Contents

- Azure Boards Documentation
- What is Azure Boards?
- Why use Azure Boards?
- Best practices for Agile project management
- Quick reference
- Get started
  - Start using Azure Boards
  - Sign up and invite some teammates
  - Plan and track work
  - Customize your boards
  - Track work as a Stakeholder
  - Manage your project
  - System fields
  - Keyboard shortcuts
- Basic concepts
  - Backlogs, boards, & plans
  - Best tool for the job
  - Agile glossary
  - Choose a process
  - About work items
  - Area & iteration paths
  - Workflow states & categories
  - Linking & traceability
  - Link types
  - Work tracking limits
  - Process guidance
    - Agile culture
    - Agile process guidance
    - Agile workflow

- Scrum process guidance
- Scrum workflow
- CMMI process
  - CMMI process guidance
  - CMMI workflow
  - CMMI background
  - Manage projects
    - Project management process
    - Project activities
    - Project inception
    - Plan a project
    - Manage change
    - Manage risk
    - Iteration activities
    - Plan an iteration
    - Manage issues
  - Engineering management
    - Process overview
    - Develop requirements
    - Arrange requirements into a product plan
    - Create a solution architecture
    - Implement development tasks
    - Build a product
    - Track bugs
- Changes made to process templates

Permissions & access

- Quick guide to default permissions & access
- Default permissions & access (Security)
- Set work tracking & plan permissions
- Grant or restrict permissions to select tasks

Configure & customize

Configure & customize guidance

- Define area paths & assign to a team
- Define iteration (sprints) & configure team iterations
- Set up your backlogs & boards
- Configure your backlog view
- Change column options
- Filter backlogs, boards, queries, and plans
- Add & manage teams
  - About teams & Agile tools
  - Add a team
  - Add a team administrator
  - Add users to a project or team
  - Configure team tools
- Set the Work Items experience (Visual Studio)
- Manage process templates
- Work across projects FAQs

Azure Boards & GitHub

- GitHub integration
- Install the Azure Boards app
- Connect Azure Boards (Cloud)
- Connect Azure Boards (On-premises)
- Link GitHub commits, pull requests, & issues
- Configure status badges
- Add or remove repositories
- Change GitHub repository access
- Troubleshoot GitHub connection
- Enable DevSecOps with Azure and GitHub

Work items

- About work items
- Quick reference
- View & add work items
- Add & update a work item
- Filter your work items

- Manage requirements
- Manage bugs
- Manage issues
- Link work items
- Copy/clone
- Collaborate with others
  - Follow work
  - Use @mentions in discussions
  - Add tags to work items
  - Share information
  - Copy list
  - Email or print work items
  - Move work items from one team to another
- Modify, delete, import, or export
  - Bulk modify (web)
  - Bulk add or modify (Excel)
  - Bulk import or update (CSV)
  - Change work item type, move work items
  - Remove, delete, or restore
  - Delete test artifacts
  - Use work item templates
- Samples
  - Sample work item templates
  - Define a work item URL
- Go mobile

Boards & Kanban

- About Boards and Kanban
- Kanban board quickstart
- Reorder cards
- Add tasks or child items as checklists
- Define features & epics
- Filter your Kanban board

- Kanban board basics
- Add, run, & update inline tests
- Customize your boards
  - Kanban board controls
  - Enable live updates
  - Add or edit columns
  - Customize cards
  - WIP limits
  - Expedite work (swimlanes)
  - Split columns
  - Definition of done
  - Show bugs on backlogs & boards
- Best practices, monitoring, & reports
  - Kanban best practices
  - Kanban concepts
  - View/configure CFD
  - Configure Lead/Cycle time widget

Backlogs

- About backlogs
- Create your backlog
- Define features & epics
- Organize your backlog (map or reparent)
- Filter your backlogs
- Forecast
- Customize your backlogs
  - Configure your backlog view
  - Filter your backlog
  - Set up your backlogs & boards
  - Backlog controls
  - Change column options
  - Display rollup progress bars or totals
  - Select backlog levels

- Show bugs on backlogs & boards
- Best practices, monitoring, & reports
  - Refine your backlog
  - Rollup of work & other fields
  - View/configure velocity
- Sprints & Scrum
  - About Sprints and Scrum
  - Change sprint dates
  - Forecast
  - Plan and work a sprint
    - 1. Assign backlog items to a sprint
    - 2. Add tasks
    - 3. Set sprint capacity
    - 4. Adjust work to fit capacity
    - 5. Share your sprint plan
    - 6. Update the Taskboard
  - Customize your sprint backlogs
    - Customize a Taskboard
    - Change column options
    - Filter your sprint views
    - Set working days
  - Best practices, monitoring, & reports
    - Scrum best practices
    - Scrum concepts
    - View/configure sprint burndown
    - Configure a Burndown or Burnup widget
- Queries & search
  - About managed queries
  - Query quick reference
  - Semantic search
  - Get started with queries
    - View, open, run a query

- Define a query
- Manage queries & query folders
- Chart a query
- Triage work items
- Filter your queries
- Best practices to define a query
- Query FAQs
- Set query permissions (Security)
- Additional query operations
  - Change column options
  - Cross-service and enhanced query operations
  - Bulk import or update work items (CSV)
  - Bulk add or modify work items (Excel)
  - Define a query as a hyperlink
- Sample queries
  - Query by title, ID, or description
  - Query by assignment or workflow changes
  - Query by area or iteration
  - Query by date or current sprint
  - Query based on tags
  - Query history and discussion
  - Query by field comparison
  - Query by a numeric field
  - Query by picklist value
  - Query by links or attachments
  - Query by build & test integration fields
- Reference
  - Query fields, operators & macros
  - Work Item Query Language (WIQL) syntax
- Plans & portfolios
  - Plans, Agile at scale
  - Portfolio management

- Configure a hierarchy of teams
- Review team Delivery Plans
- Track dependencies
- Filter your Delivery Plans
- Add or edit a Delivery Plan
- Edit plan permissions
- Delivery Plans FAQs
- Scaled Agile Framework®
  - Implement Scaled Agile Framework®
  - SAFe® concepts & Azure Boards
  - Configure Azure Boards
  - Customize Azure Boards
  - Manage SAFe® programs and portfolios
  - View roadmaps and monitor progress
- Best practices, monitoring, & reports
  - Agile culture
  - Practices that scale
  - Visibility across teams
- Track work, code, builds, and releases
- End-to-end traceability
- Use #ID to link to work items
- Configure Git repositories to integrate with work tracking
- Configure pipelines to support work tracking
- Drive Git development
- View deployment stages on work item form
- Auto complete work items with pull requests
- Request and track feedback
  - About feedback
  - Test & Feedback extension
    - Request stakeholder feedback
    - Provide stakeholder feedback
    - Get insights from tests

- Install the extension
- Test in Connected mode
- Test in Standalone mode
- Explore work items
- Add to existing bugs
- Provide feedback without a request
- Track stakeholder feedback requests
- Microsoft Feedback client
  - Request feedback
  - Provide feedback
- Track feedback requests
- Set feedback permissions (Security)
- Enable remote audio capture
- Change the audio device or annotation tool

Excel, Project, PowerPoint

- Connect to an Office client
- Bulk add or modify work items (Excel)
- Excel FAQs
- Project & Storyboarding (Deprecated)
  - Create backlog items & tasks (Project)
  - Schedule projects (Project)
  - Rollup work using Project
  - Storyboard your ideas (PowerPoint)
  - Add & share storyboard shapes (PowerPoint)
- Troubleshooting
  - Office integration issues
  - Resolve Excel data conflicts
  - Resolve Excel data validation errors
  - Resolve invalid links in an Excel tree list
- Project portal & process guidance (Deprecated)
  - Configure or add a project portal
  - Configure or redirect process guidance

Integrations & extensions
Overview
Migration and integration
Delivery Plans
Feature Timeline
Epic Roadmap
Dependency Tracker
Azure Boards with Slack
Azure Boards with Teams
Project Server Integration (Deprecated)>>
TFS-SharePoint Integration (Deprecated)>>
Troubleshooting
FAQs
Set up your backlogs & boards
Fix "Ordering backlog items is disabled"
Work item form caching
Field reference
Work item field index
Work item fields
Code review & feedback fields
Bugs, issues, & risks fields
Change request fields
Requirements fields
Review meeting fields
REST API Reference
Work Items
Fields
Boards
Backlogs
Queries
Plans
Work Item Query Language

DevOps Resource Center
- What is Agile?
- What is Agile development?
- Creating productive teams
- What is Scrum?

Resources
- Dashboards, Charts, Reports, & Widgets
- Process customization
- Azure Test Plans
- Client support
- Azure DevOps CLI

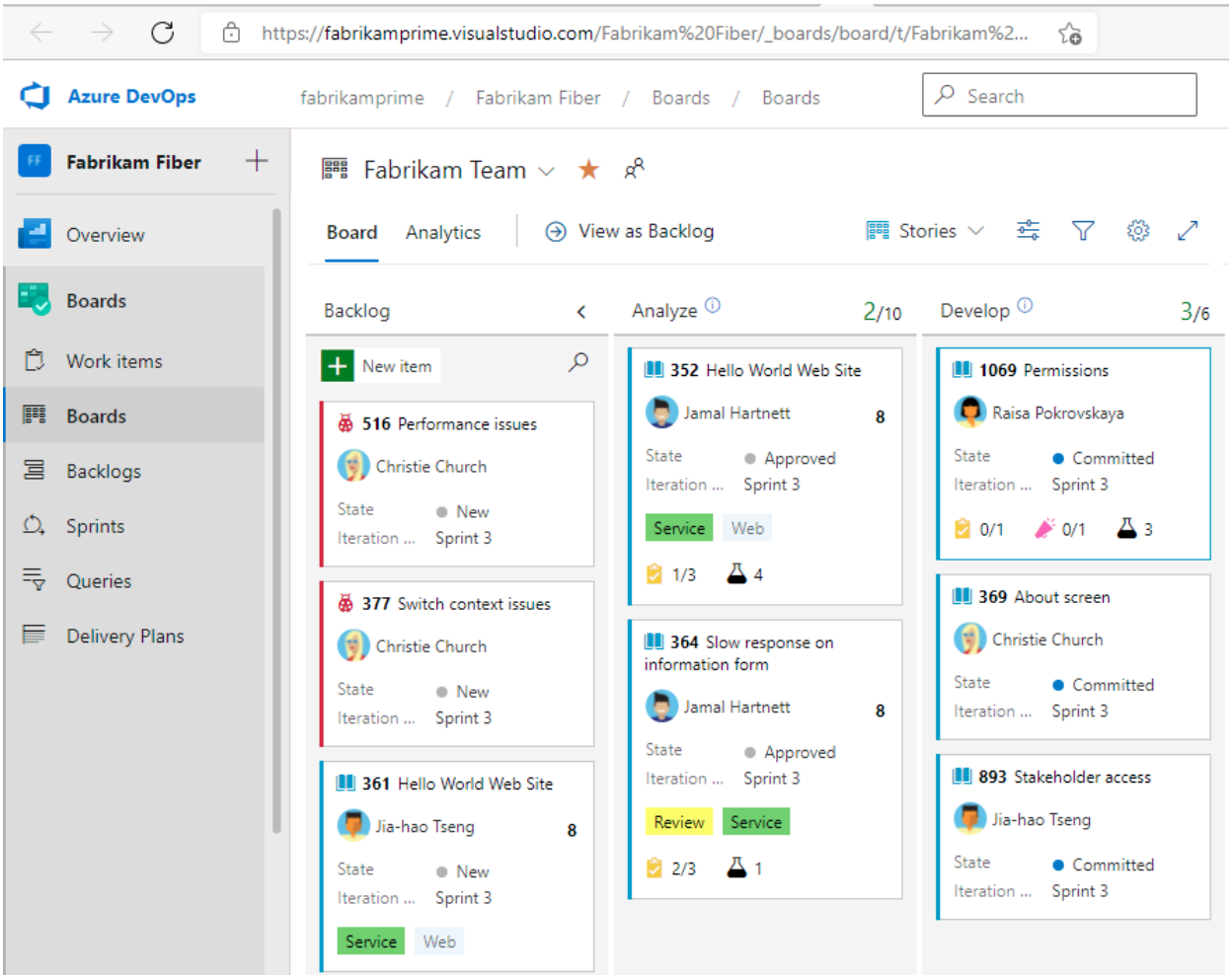
What is Azure Boards?

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure Boards provides software development teams with the interactive and customizable tools they need to manage their software projects. It provides a rich set of capabilities including native support for Agile, Scrum, and Kanban processes, calendar views, configurable dashboards, and integrated reporting. These tools scale as your business grows.

Quickly and easily track work, issues, and code defects associated with your project. The Kanban board, shown in the following image, is just one of several tools that allows you to add, update, and filter user stories, bugs, features, and epics.

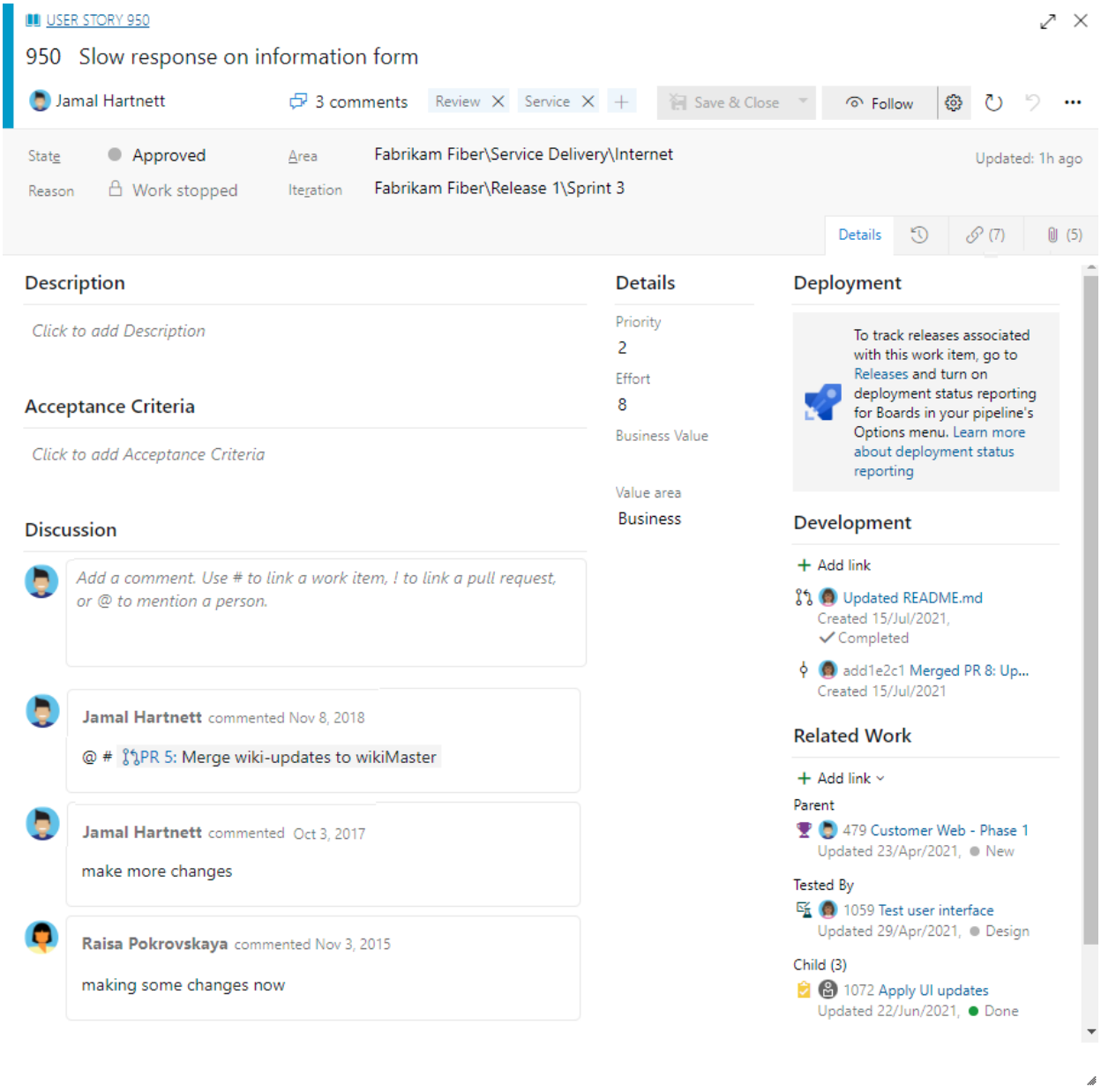


If you're ready to start using Azure Boards, see Sign up for free and invite others to collaborate in Azure Boards. Need more information? See Reasons to use Azure Boards to plan and track your work.

> **NOTE**
> This article applies to Azure DevOps Services and Azure DevOps Server 2019 and later versions. Most of the information is valid for earlier on-premises versions, however, images show only examples for the latest version.

Track user stories, bugs, features, and epics

Track various types of work using the default work item types—such as user stories, bugs, features, and epics. Or, customize these types or create your own. Each work item form provides a standard set of system fields and controls, including **Discussion** for adding and tracking comments, **History**, **Links**, and **Attachments**. The **Deployment**, **Development**, and **Related Work** controls support tracking when code is released or changed, and relationships between work items.

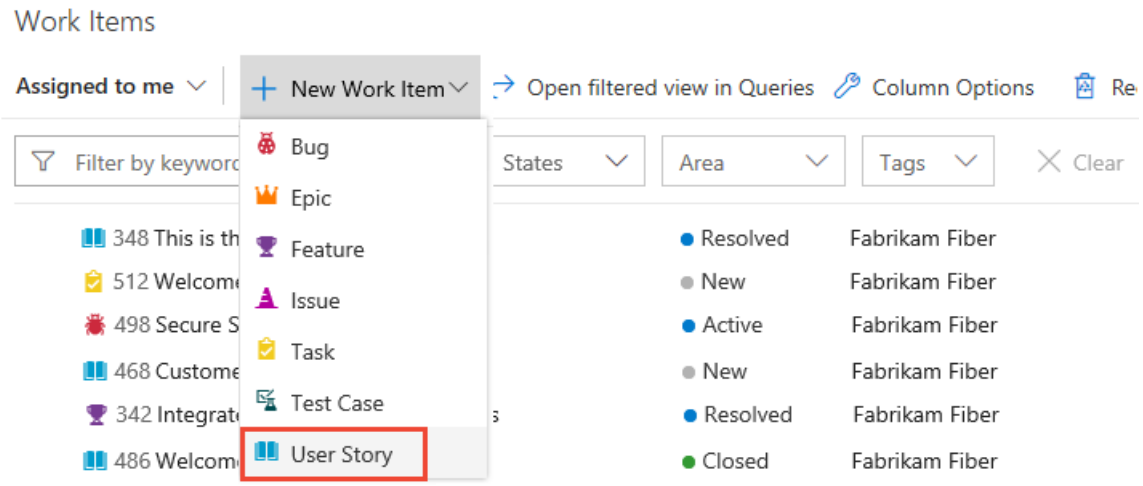


Use interactive backlogs, boards, lists, and calendar views

Azure Boards provides several hubs, each providing a set of interactive tools. Each tool provides a filtered set of work items. Most tools support adding and updating work items. To learn more about effective use of these tools, see Best tool to add, update, and link work items.

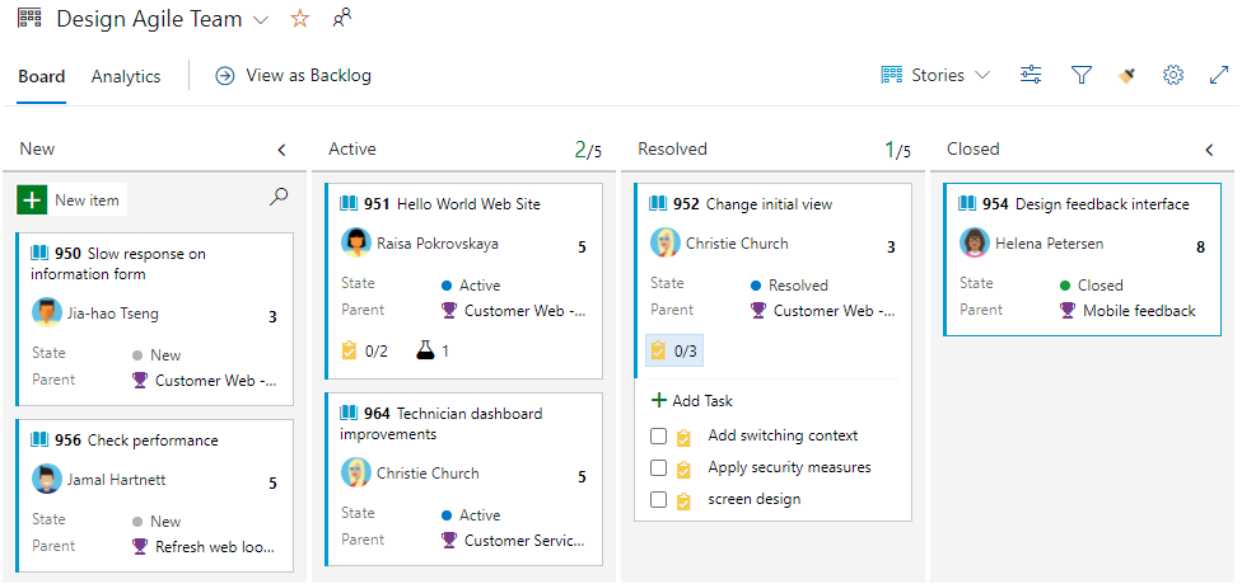
Open Work Items for personalized filtered lists

Use the **Work items** hub to quickly find work items that are assigned to you. Access various lists based on select criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.



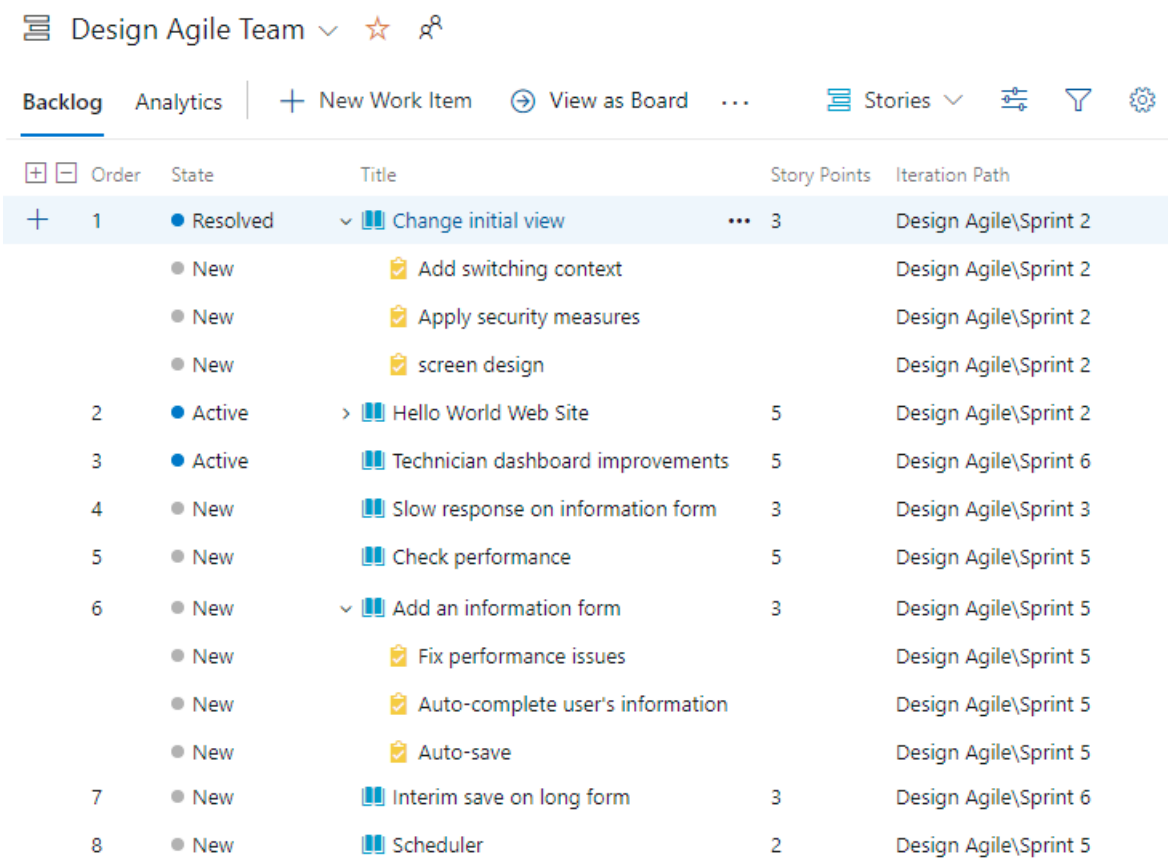
Use Boards to add, update, and track status

Use the Boards hub to view work items as cards and perform quick status updates through drag-and-drop. The feature is similar to sticky notes on a physical whiteboard. Use to implement Kanban practices and visualize the flow of work for a team.



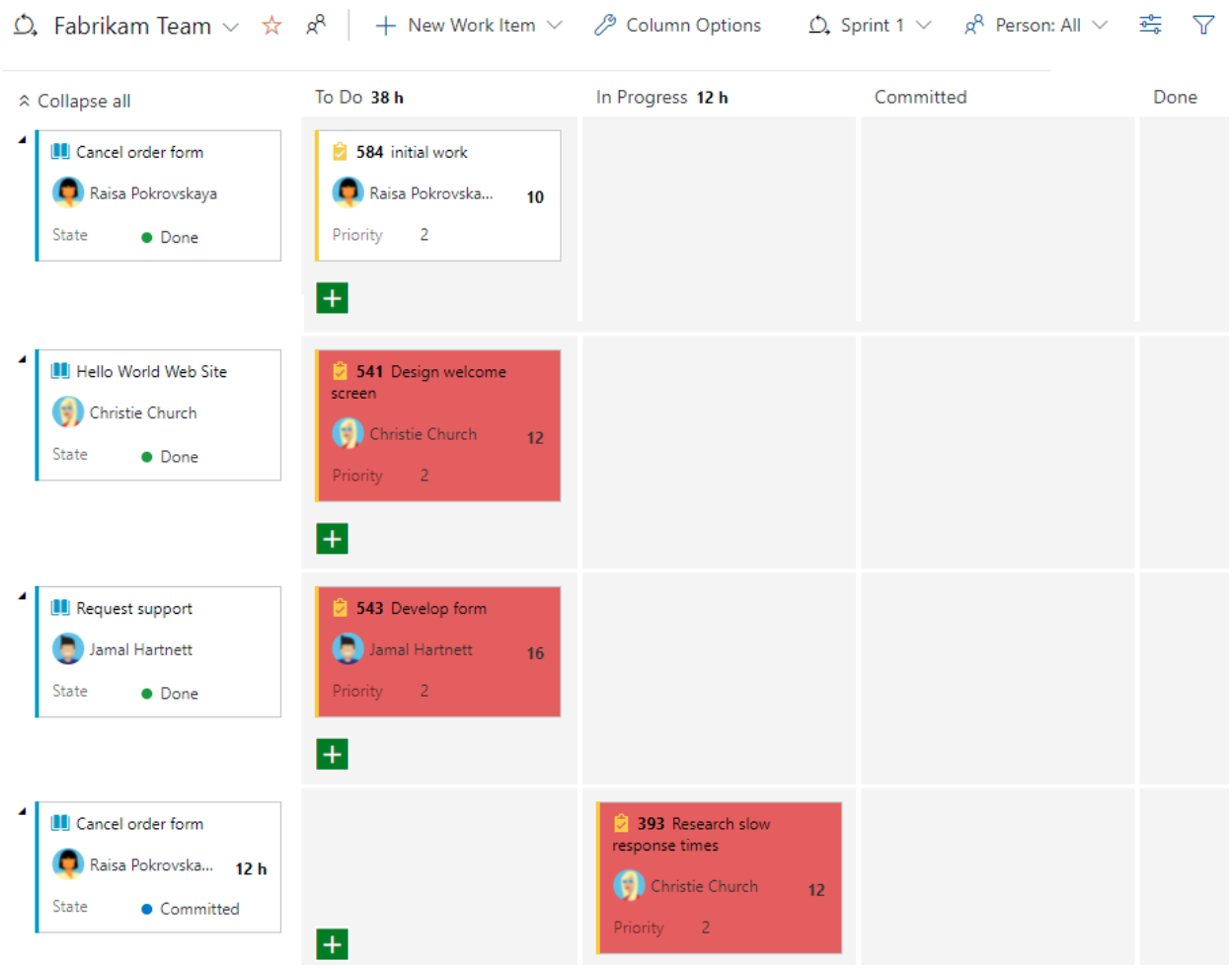
Use Backlogs to add, prioritize, organize, and assign work

Use the Backlogs hub to view, plan, order, and organize work items. A product backlog represents your project plan and a repository of all the information you need to track and share with your team. Portfolio backlogs allow you to group work under features and epics.



Use Sprints for Scrum processes

Open the Sprints hub to access a team's filtered view of work items based on a specific iteration path, or sprint. From your backlog, you can assign work to an iteration path by using drag-and-drop. Implement Scrum practices by interacting with a backlog list or card-based taskboard.

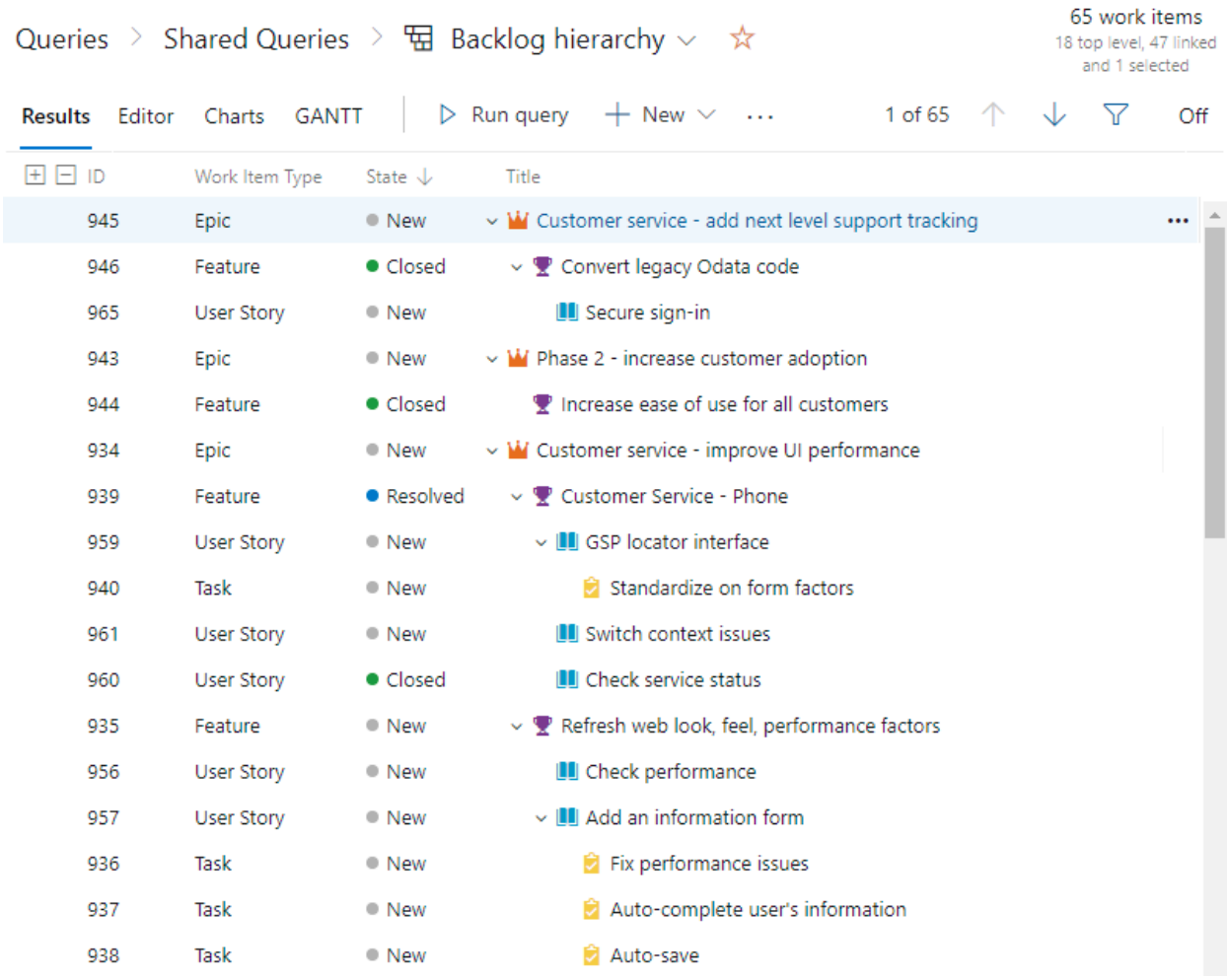


Create work item lists and charts with Queries

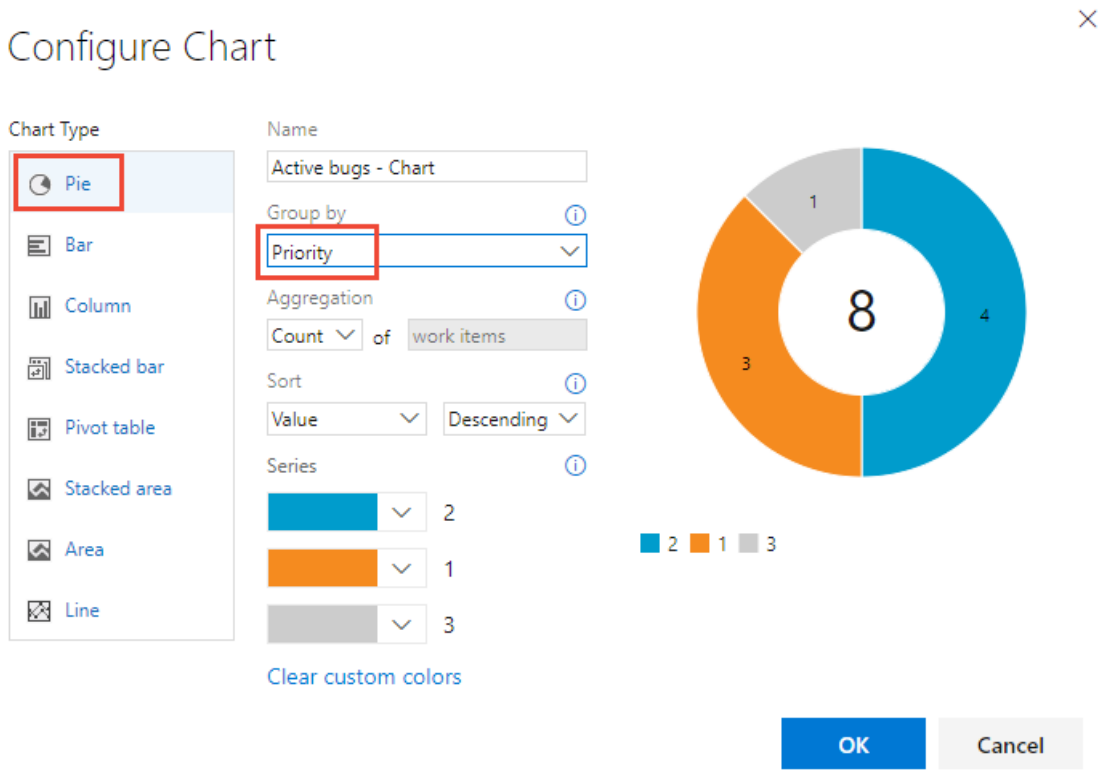
Access the Queries hub to generate custom lists of work items. Queries support the following tasks:

- Find groups of work items with something in common.
- Triage work to assign to a team member or sprint and set priorities.
- Perform bulk updates.
- View dependencies or relationships between work items.
- Create status and trend charts that you can optionally add to dashboards.

Backlog hierarchy



Query charts

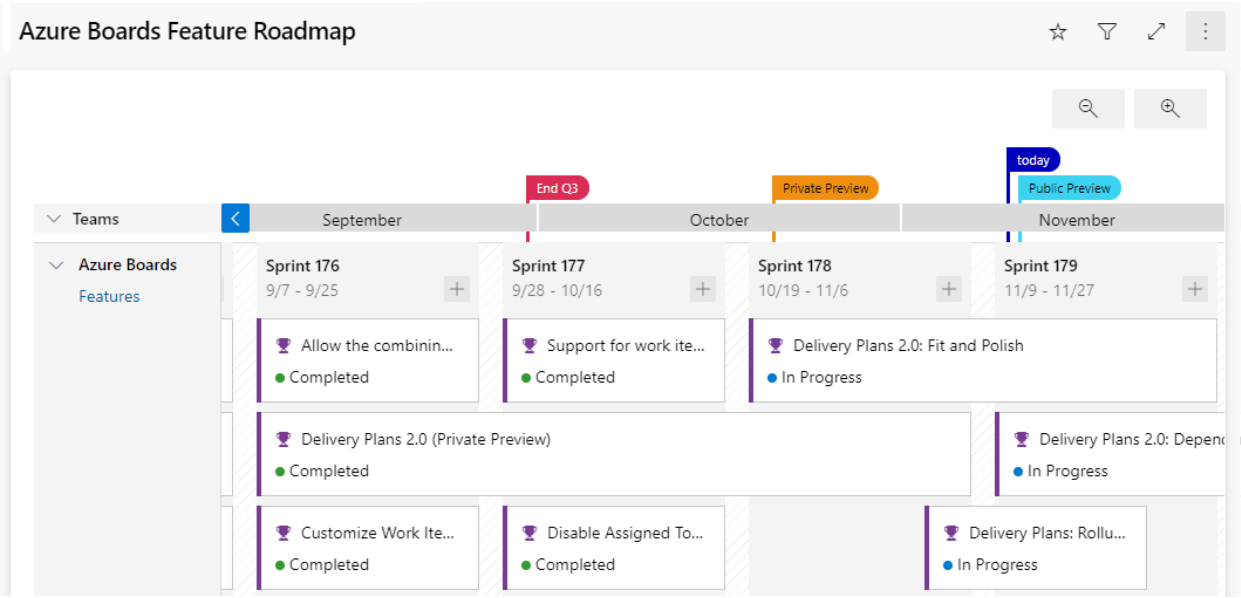


Use Delivery plans for calendar views and dependencies

With the Delivery Plans hub, management teams can view deliverables and track dependencies across several teams in a calendar view. Delivery plans are fully interactive, supporting the following tasks:

- View up to 15 team backlogs, including a mix of backlogs and teams from different projects

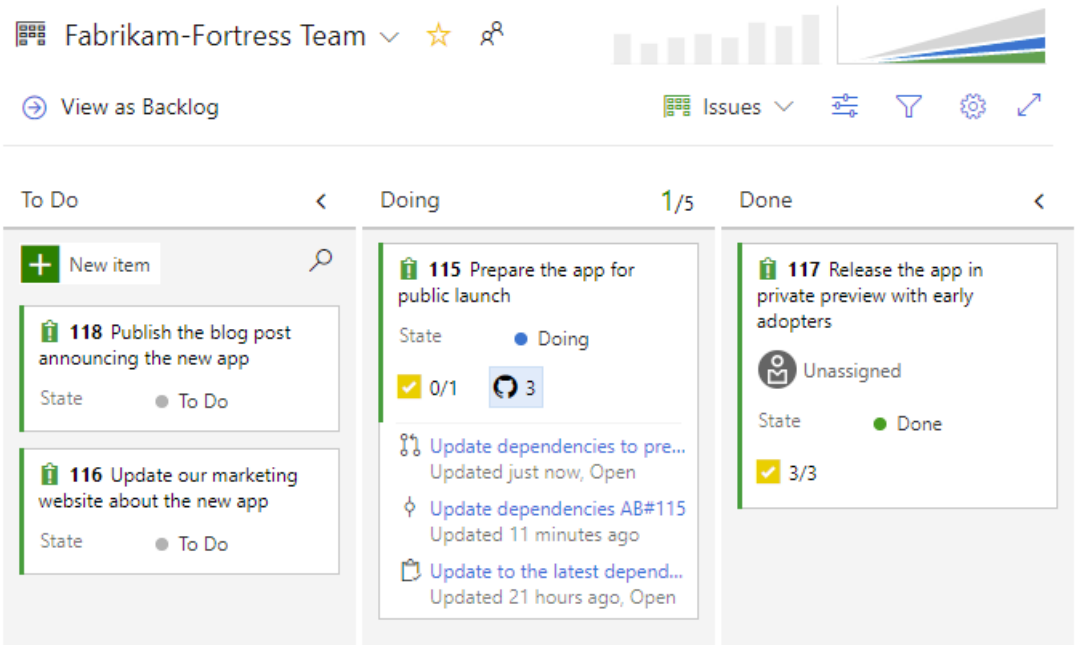
- View custom portfolio backlogs and epics
- View work that spans several iterations
- Add backlog items from a plan
- View rollup progress of features, epics, and other portfolio items
- View dependencies that exist between work items.



Use GitHub, track work in Azure Boards

By connecting Azure Boards with GitHub repositories, you enable linking between GitHub commits, pull requests, and issues to work items. You can use GitHub for software development while using Azure Boards to plan and track your work. Azure Boards provides the scalability to grow as your organization and business needs grow.

From your Kanban board, you can see and quickly open linked GitHub commits, pull requests, or issues for more detail. To learn more, see GitHub & Azure Boards.

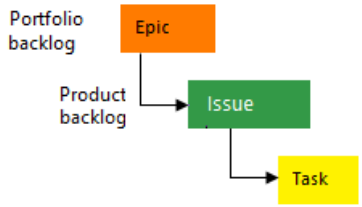


Implement Agile, Scrum, and Kanban processes

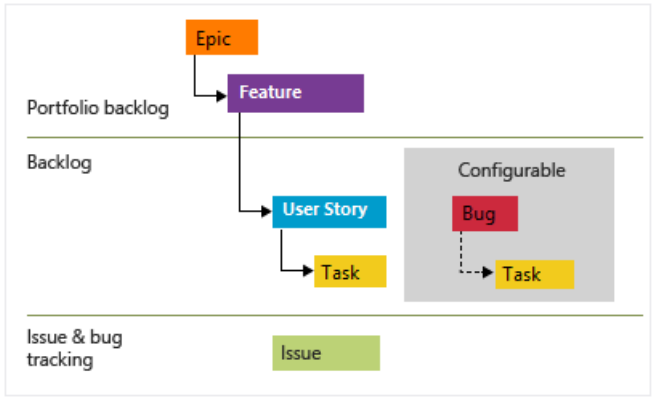
Azure Boards is designed to support software development processes through the default process models

selected for a project. Each process provides a set of work item types with a natural hierarchy as shown in the following images.

Basic provides the simplest model that tracks work through Issues, Tasks, and Epics.

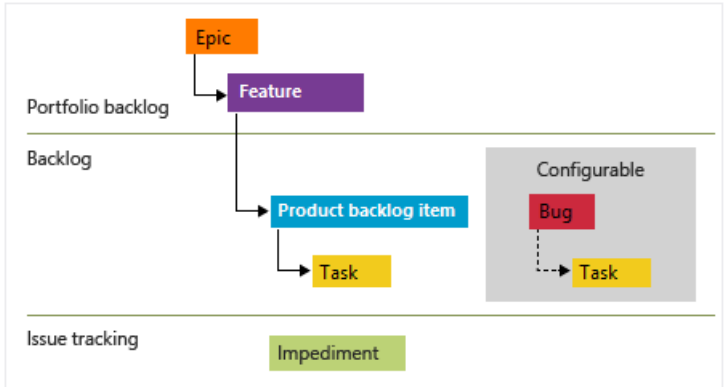


Agile supports Agile planning methods (learn more about Agile methodologies at the Agile Alliance), including Scrum, and tracks development and test activities separately. This process works great if you want to track user stories and (optionally) bugs on the Kanban board, or track bugs and tasks on the taskboard.

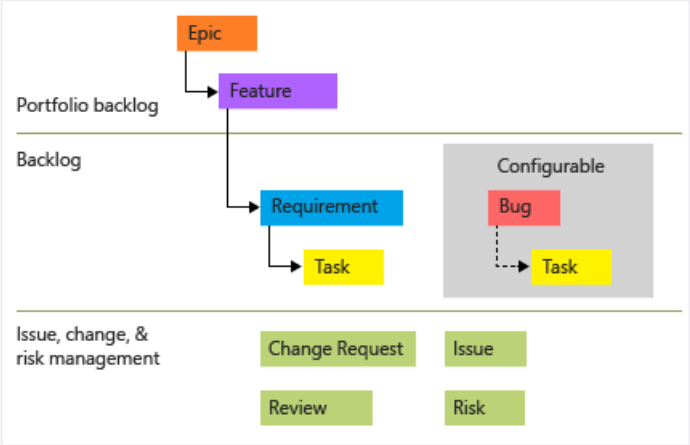


Scrum tracks work using product backlog items (PBIs) and bugs on the Kanban board or viewed on a sprint taskboard.

This process supports the Scrum methodology as defined by the Scrum organization

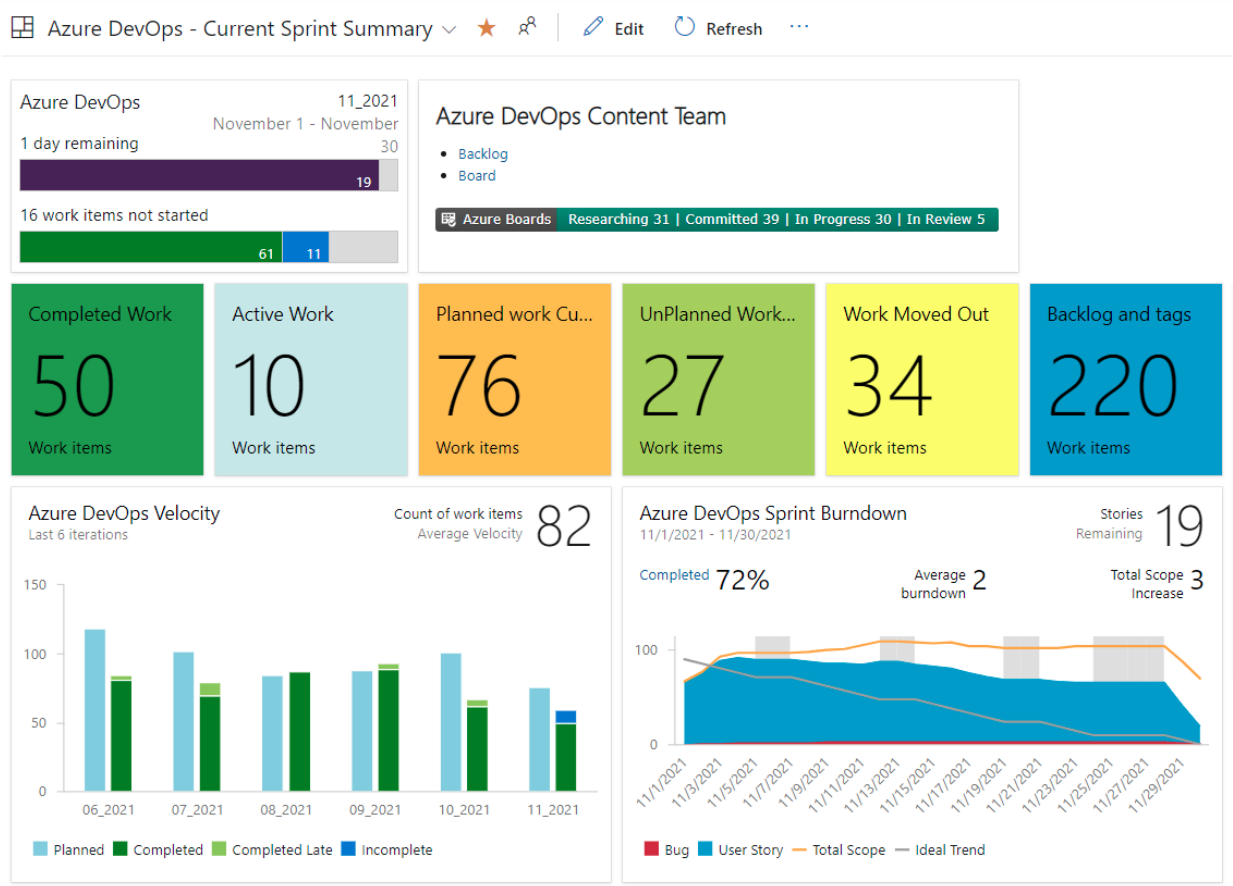


Capability Maturity Model Integration (CMMI) supports a framework for process improvement and an auditable record of decisions. With this process, you can track requirements, change requests, risks, and reviews. This process supports formal change management activities.



Configurable dashboards and Power BI reports

With dashboards, teams can create customized views to gain visibility into their status, view progress, and analyze trends. Dashboards provide flexibility to share information and improve workflow processes. Each team can tailor their dashboards to share information and monitor their progress.



Also, you can use Power BI to create custom, complex reports based on custom queries of the Analytics service. The Analytics service is the reporting platform for Azure DevOps. It is optimized for fast read-access and server-based aggregations. Use it to answer quantitative questions about the past or present state of your projects.

To learn more, see About dashboards, charts, reports, & widgets and What is the Analytics service?.

Gain visibility through end-to-end traceability

With Azure Boards, you gain the advantage of full integration with the Azure DevOps platform. Azure DevOps is designed to provide end-to-end traceability, tracking work from requirements to deployment. You gain insight at each step of decisions made and software deployed. Some of the traceability tasks supported include:

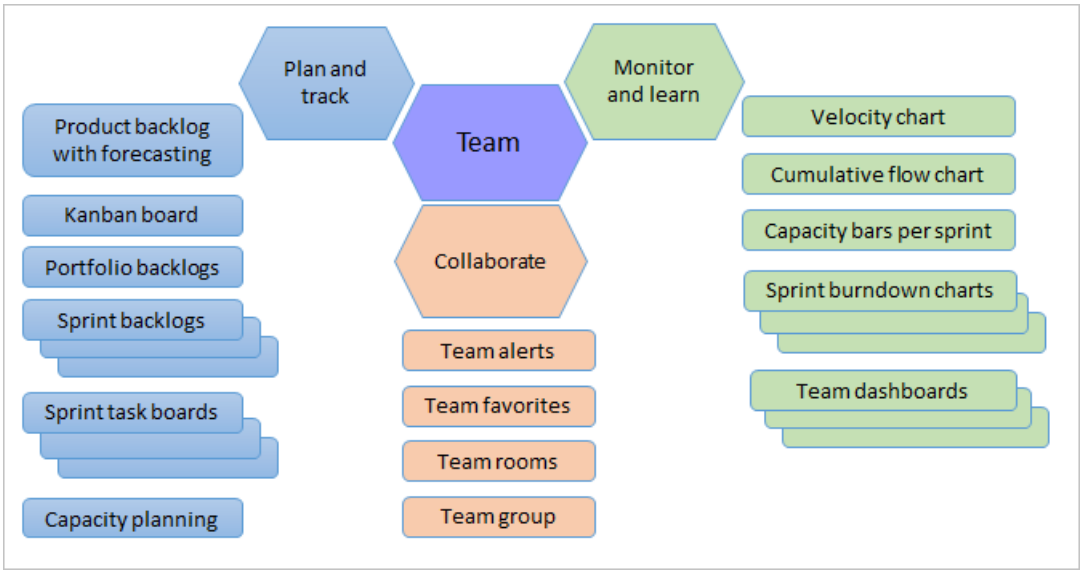
- Create a branch from a requirement
- Create a pull request of updated branch
- Validate the pull request using a build pipeline
- Create and run inline tests on requirements
- Merge the pull request into the main, default branch
- Deploy changes into production with deployment status to Azure Boards
- Monitor and report on requirements traceability

To learn more about these features, see End-to-end traceability and Cross-service integration and collaboration overview.

Support independent, autonomous teams

A team refers to a group of project members who work in a particular product area. Those areas are represented as **Area Paths**. Area paths are hierarchical paths that denote the possible areas of ownership in an organization. A team is defined by a name, its members, and its area paths.

Boards, Backlogs, Sprints, and Delivery Plans rely on team configurations. For example, if you want to add a Kanban board or product backlog, you define a team. For more information on teams, see About teams and Agile tools.



Azure Boards works with your favorite tools. Integrate with Microsoft Teams and Slack to enable efficient ChatOps.

Extensions provide support for other tools. An extension is an installable software unit that adds new capabilities to your projects. Find extensions in the Azure DevOps Marketplace. Extensions can support planning and tracking of work items, sprints, scrums, and more and collaboration among team members.

Video: Plan your work with Azure Boards

Related articles

- Reasons to use Azure Boards to plan and track your work
- Best tool to add, update, and link work items
- Configure and customize Azure Boards
- The DevOps journey at Microsoft

- Agile culture
- Practices that scale
- About teams and Agile tools

Reasons to use Azure Boards to plan and track your work

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

We know you have a choice of tracking systems. So why use Azure Boards to plan and track your work, bugs, and customer issues?

In What is Azure Boards?, we describe the main features you get with Azure Boards. Here, we provide 11 compelling reasons to take Azure Boards for a free test spin.

Start simply, scale as you grow

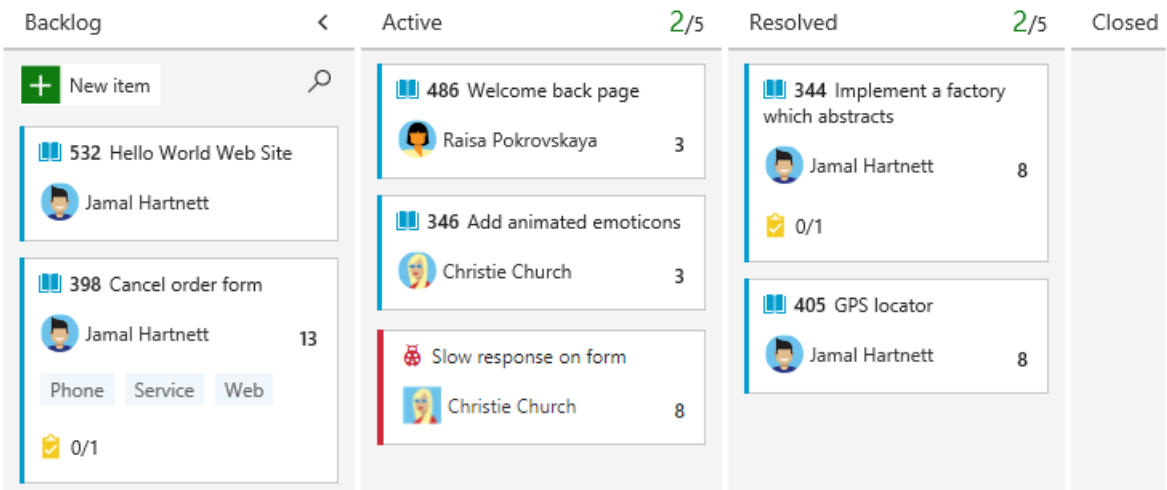
Azure Boards provides you with a set of predefined work item types to support tracking features, user stories, bugs, and tasks. You can quickly get up and running by using your product backlog or Kanban board. Whatever Agile method you use, Azure Boards supports you with the tools you need to implement that method.

As your organization grows, you can add teams to provide them the autonomy to track their work according to their needs.

> **TIP**
> Quickly add work items by using your backlog or Kanban board. Or, use work item templates to simplify defining work items by setting values for select fields.

Visual, interactive tools

Visual tools help teams quickly see and share progress. For example, with Kanban boards you can add work, update the status of work, and monitor work in progress.



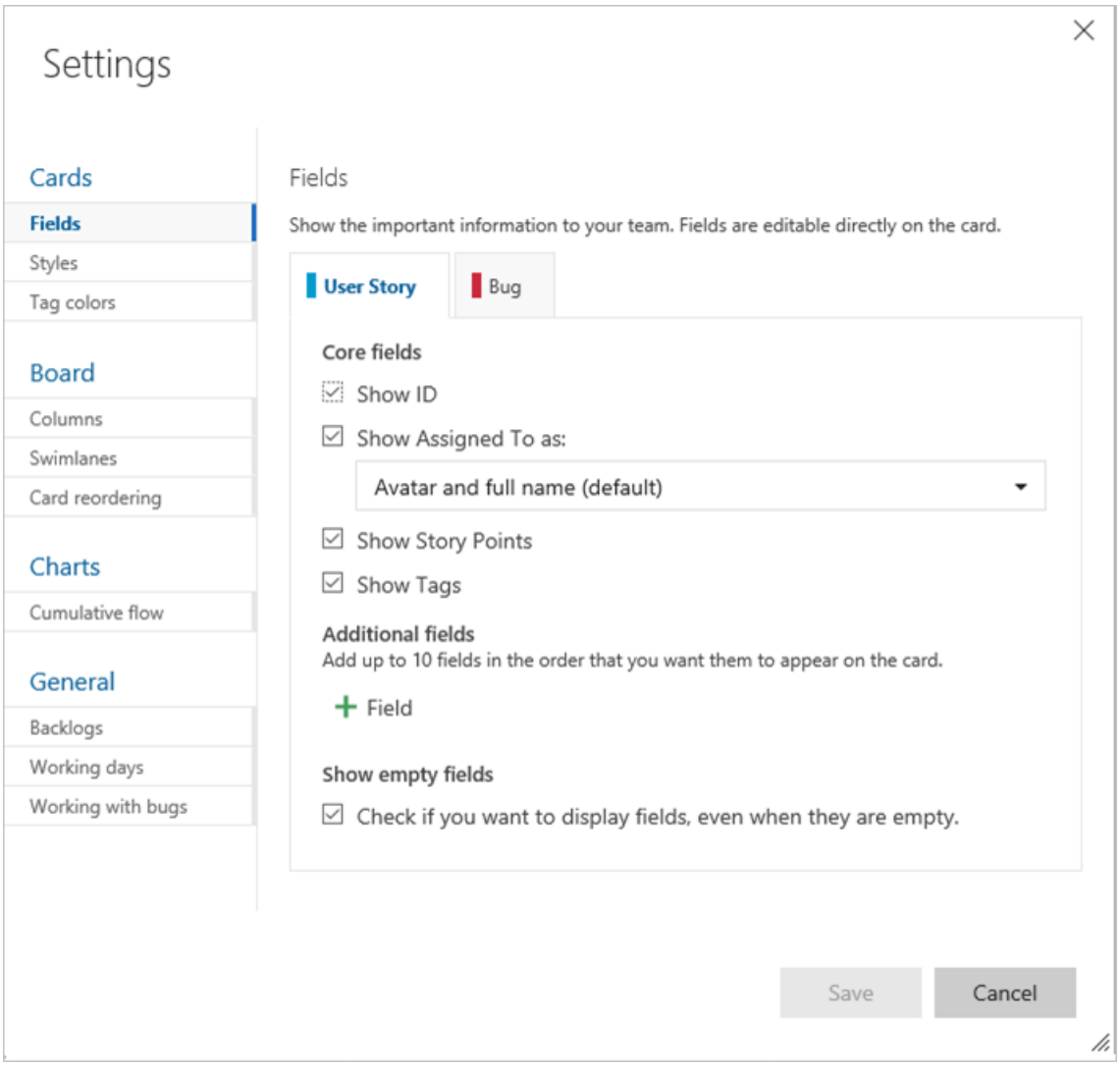
With product backlogs, you can quickly add work items and organize work to keep the most important work at the top of the stack. And, with delivery plans, teams can share their plans against a calendar view.

Use built-in scrum boards and planning tools to help your teams run effective stand-ups, planning meetings, and retrospectives.

Easy to customize

Kanban boards, taskboards, and delivery plans are easy to configure and customize through the user interface.

For example, with Kanban boards, you can configure columns, swim lanes, card styles, fields shown on cards, and more. You can configure them all through a common configuration dialog.



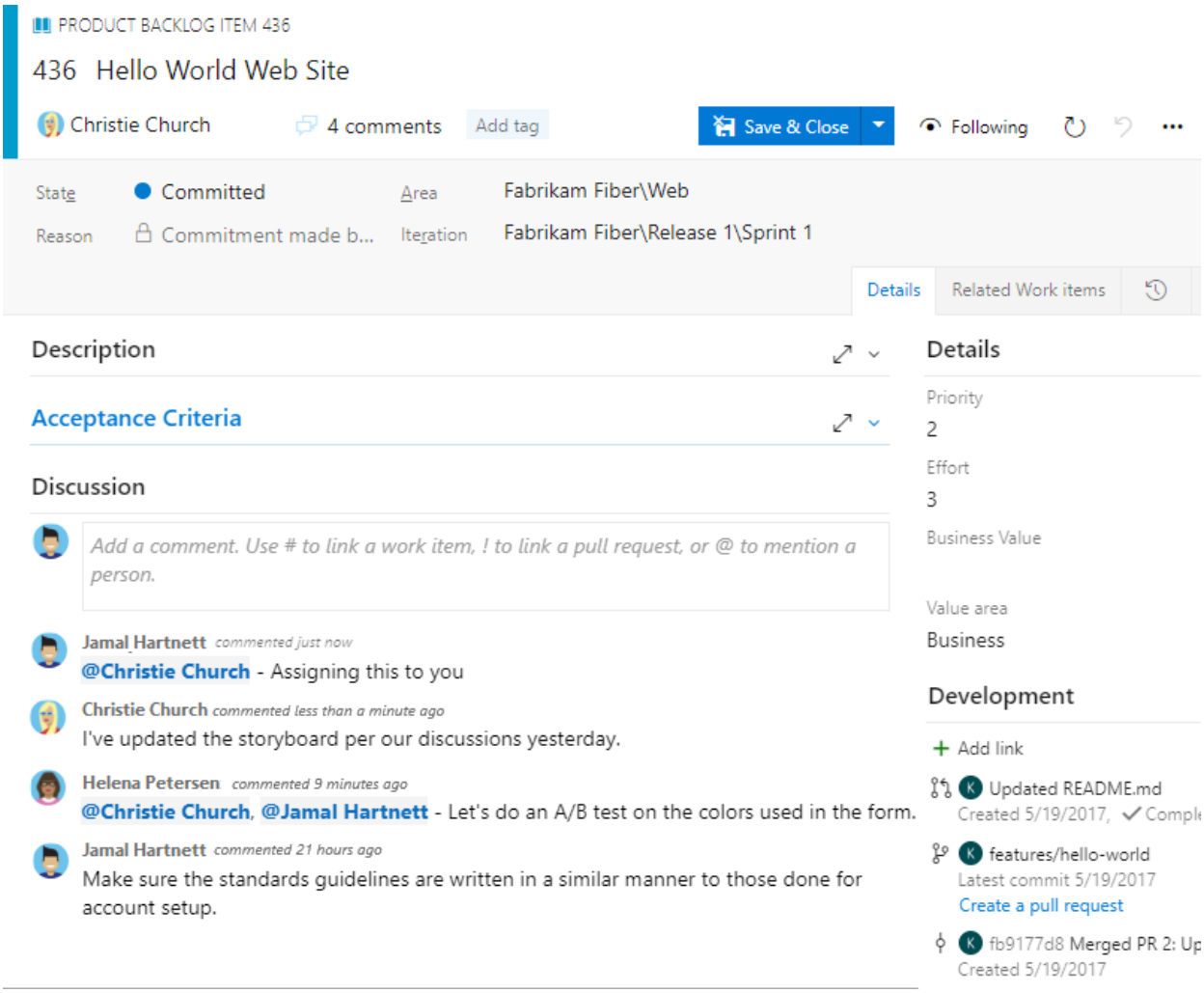
TIP

Define area paths and iteration paths to group work items by product or feature, You also can group work items into sprints, milestones, or other time-related periods.

And, you can easily add custom fields, work item types, and portfolio backlogs.

Built-in social tools and communication

Work item forms provide built-in discussion that you can use to capture questions, notes, and communication as they occur. With this feature, you can maintain a history of what a team decides on any particular work item. You can quickly bring a team member or an entire team into the conversation by using @mentions.



Capture information, generous cloud storage

Work items are designed to track all the information you need to track. You can edit in rich text, drag and drop inline images, and add larger attachments. You can add attachments up to 60 MB and as many as 100 attachments.

Also, you can link work items within a hierarchy or by simple related links. Each work item form maintains a history of changes, so you can review what changed, who made the change, and when.

Quickly find what you need and get notified of changes

As your project grows, the number of work items used to track it grows. To support your ability to quickly find a specific work item, Azure Boards provides you with easy-to-use tools:

- A follow feature so you can follow work items to monitor updates and changes
- Pivot views that show you work items assigned to you, that you elected to follow, were recently modified, and more
- A powerful query engine that filters work item lists based on any field and used to update or triage work items
- Fast, flexible ad-hoc search with quick inline filters
- Alert management settings so you can personalize the alerts you receive when work items are assigned to you or are changed, or other filter criteria

Monitor status and progress with built-in dashboards and analytics

With Azure Boards, you gain access to many tools to generate reports to support tracking status and trends.

By using configurable dashboards, you can add one or more widgets. You configure widgets to display the information and data you want, such as the following bug burndown widget.

Along with dashboards, you have access to the Analytics service. This service is optimized for fast read-access and server-based aggregations. By using Analytics views and Power BI, you can create highly sophisticated reports on the project data of interest.

Office integration

Project managers who want to use familiar tools can import and export work item queries to and from Microsoft Office Excel or import and export work items using .csv files. To learn more, see:

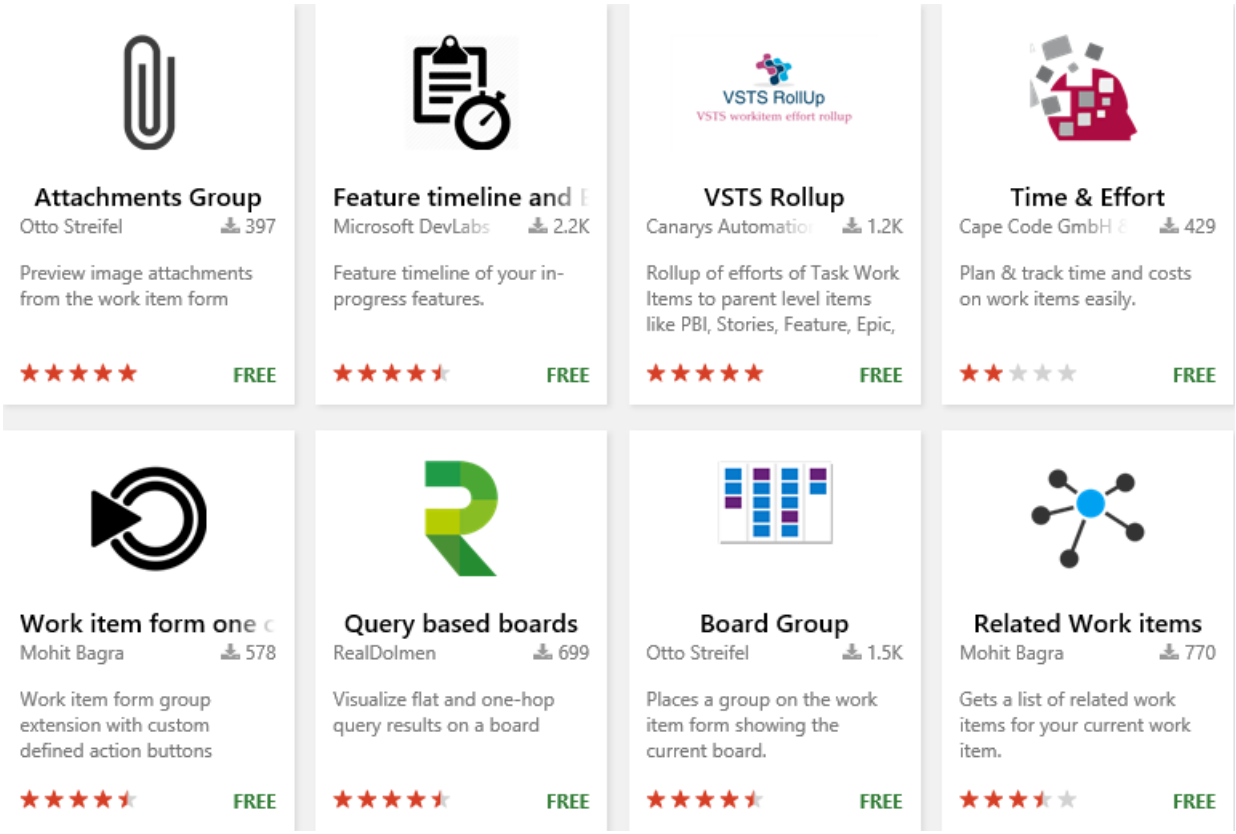
- Bulk import or update work items using CSV files
- Bulk add or modify work items with Excel

Project managers who want to use familiar tools can import and export work item queries to and from Microsoft Office Excel. To learn more, see Bulk add or modify work items with Excel.

Extensions and extensibility

You can gain even greater functionality by adding Marketplace extensions, many of which are free. An extension is an installable unit that adds capabilities to Visual Studio, Azure DevOps Services, or Visual Studio Code. You can find extensions within these products or in the Visual Studio Marketplace, Azure DevOps tab.

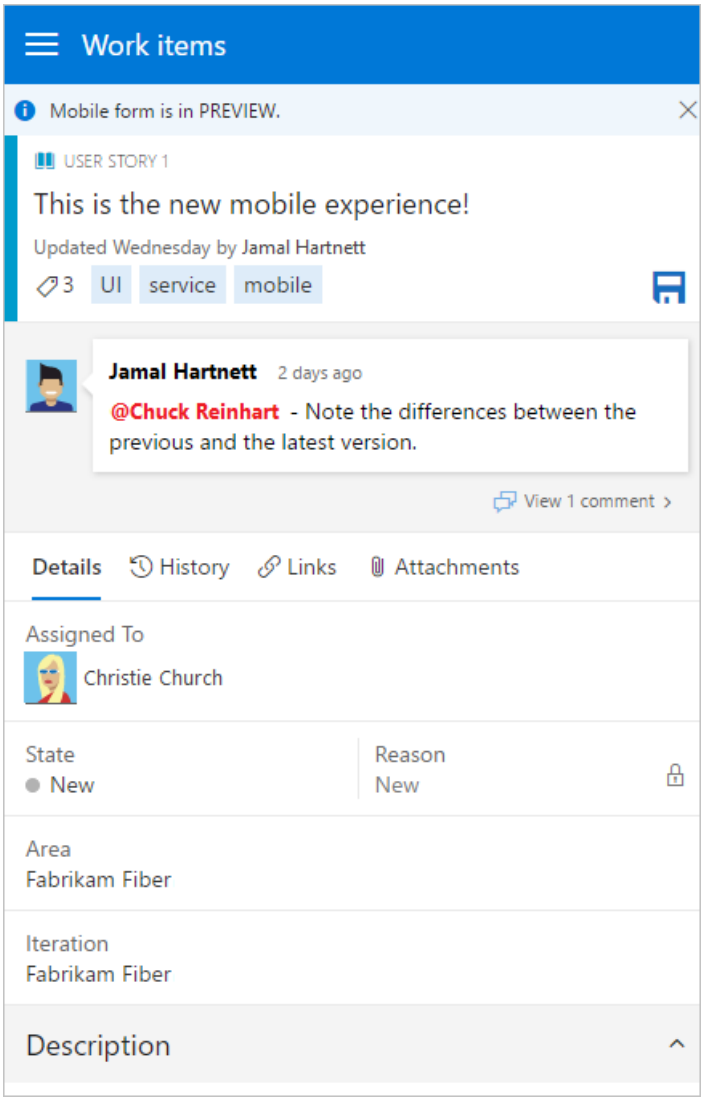
Here are a few extensions available from the Marketplace.



Also, by using the REST API, you can create your own extensions or tools to integrate with Azure DevOps Services.

Updates via a mobile browser

Azure Boards makes it easy to stay on top of changes as they occur. By using the mobile browser, you can be notified and respond to changes made to work items.



Start for free

Last but not least, you can start for free and add up to five free users and unlimited stakeholders.

Get started today. To learn how, see Sign up for free and invite others to collaborate on your project.

Related articles

- Best practices for Agile project management
- Best tool to add, update, and link work items

Best practices for "light-weight" Agile project management

10/4/2022 • 17 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure Boards provides a choice of Agile planning tools, many of which work in combination with each other. This article provides a get-started guide for project managers new to Azure Boards. If you and your teams want to take a minimal tracking approach to plan and manage your projects, start with this guide. Also, if you are moving from waterfall project management to Agile methods, start with this guide.

> **NOTE**
> If your team is committed to practicing Kanban or Scrum methods, see instead About Boards and Kanban or the tutorials for implementing Scrum.

In this article, we make the following suggestions and guidance:

- Configure your teams to support rollup of development user stories to project management features
- Define and work in a sprint cadence
- Use User Stories and Features to track deliverables
- Use your team's Feature and Product backlogs to create your product plan
- Use tags to support queries and filtering
- Forecast your product plan to gain insight into when deliverables can ship, set milestones
- Manage dependencies by linking work items
- Assign work to sprints
- Review progress and deliverables using the Features backlog, rollup, and delivery plans
- Engage with process improvement during sprint planning and retrospectives

> **NOTE**
> This article applies to Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features included in this article, such as Rollup, Analytics, and some portfolio planning tools, are only available for the cloud at this time.

Configure your teams

Azure Boards provides each team a set of Agile tools to plan and track work. Each project defines a default team, which you can start using immediately. However, if you have several development or feature teams, we recommend that you define a team in Azure DevOps for each feature team. This way, each team can work autonomously while collaborating with each other.

Best practice tips:

- Configure teams along the value streams your organization wants to deliver
- Define a team for each development group of six to twelve developers
- Configure development teams to support rollup to project management feature teams

To learn how:

- Configure a hierarchy of teams
- Add a team, move from one default team to several teams
- Agile culture
- Scale Agile to Large Teams

Configure your sprints

Sprints—specified by Iteration Paths—are defined for a project and then selected by teams. A sprint cadence can vary between one week to four weeks or longer. Also, you can define sprints within a hierarchy that includes release trains. You assign work to sprints that teams commit to deliver at the end of the sprint. These Azure Boards tools rely on sprint assignments to a team Sprint backlogs, Taskboard, and Forecast and Delivery plans.

Best practice tips:

- Define a sprint cadence that all teams within your product group will use
- Define at least six or more iterations that will support planning for the next six to twelve months
- Determine how teams will use iterations to manage backlog items
  - Unassigned sprint work is assigned to the default backlog, or
  - Unassigned sprint work is assigned to a designated future backlog sprint.

To learn how:

- About Area and Iteration Paths, Define and assign Iteration Paths
- Define Iteration Paths and configure team iterations

Choose the work item types you'll use

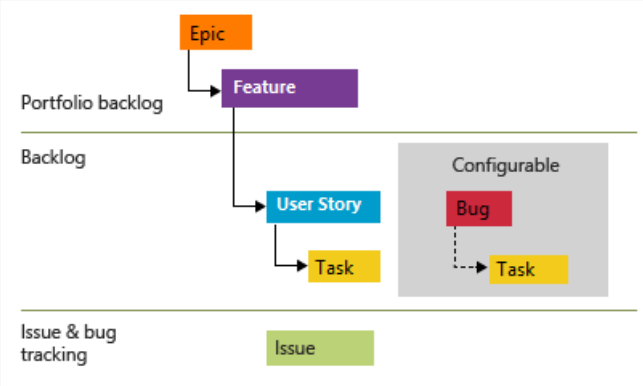
Determine which work item types your team will use to capture customer requirements and development work. If your project is based on the Agile process, we recommend using User Stories, Bugs, and Features.

> **NOTE**
> The guidance provided in this article are based on the Agile process.

If your project is based on another process, such as Basic, Scrum, or CMMI, you have a choice from those shown in the following images. Also, each team can determine how they want to track bugs.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

> **NOTE**
> *Requirements* specify expectations of users for a software product. In Azure Boards, requirements are defined by work items that appear on your product backlog. They correspond to User Stories (Agile), Product Backlog Items (Scrum), Issues (Basic), or Requirements (CMMI) based on the process selected for your project. They also belong to the Requirements Category which manages which work item types appear on the product backlog.

Best practice tips:

- Use **Features** to capture customer features you want to ship
- Quickly add features or requirements from the backlog and fill in details later
- Use **Requirements**—User Stories (Agile), Issues (Basic) Product Backlog Items (Scrum), or Requirements (CMMI)—to break-down Features into work the development team will own
- Use Bugs to capture code defects
- Map Requirements to Features to track progress at the project management level
- Size Requirements to be completed within a sprint
- Size Features to be completed within a sprint or several sprints
- Size Epics to be delivered quarterly or to some milestone objective
- Let Developers use Tasks to break-down their work as needed.

As project managers, you manage the Features and the development team manages the Requirements. By mapping them using parent-child links, you gain visibility into the progress of your features. Each work item you add to your team backlog is automatically assigned the default area path and iteration path set for your team.

If you have larger initiatives or scenarios that require shipping several Features, you can group these under Epics, again using parent-child links.

To learn how:

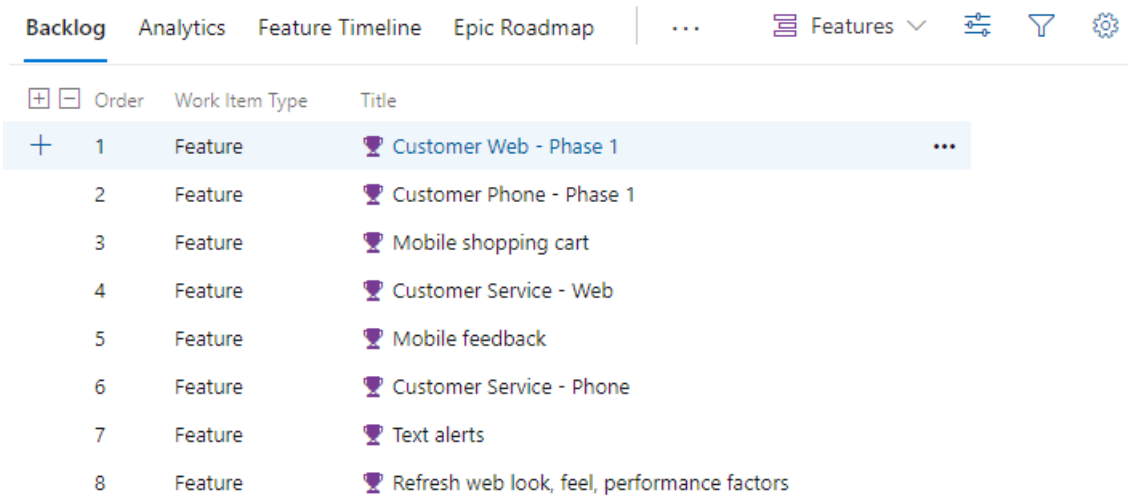
- Define features and epics
- Create your backlog
- Organize your backlog (map or reparent)

Create your product plan

Create your product plan using the Features Backlog. The development team then creates their product plan using the Product Backlog. Periodically you'll want to review and groom your product plans.

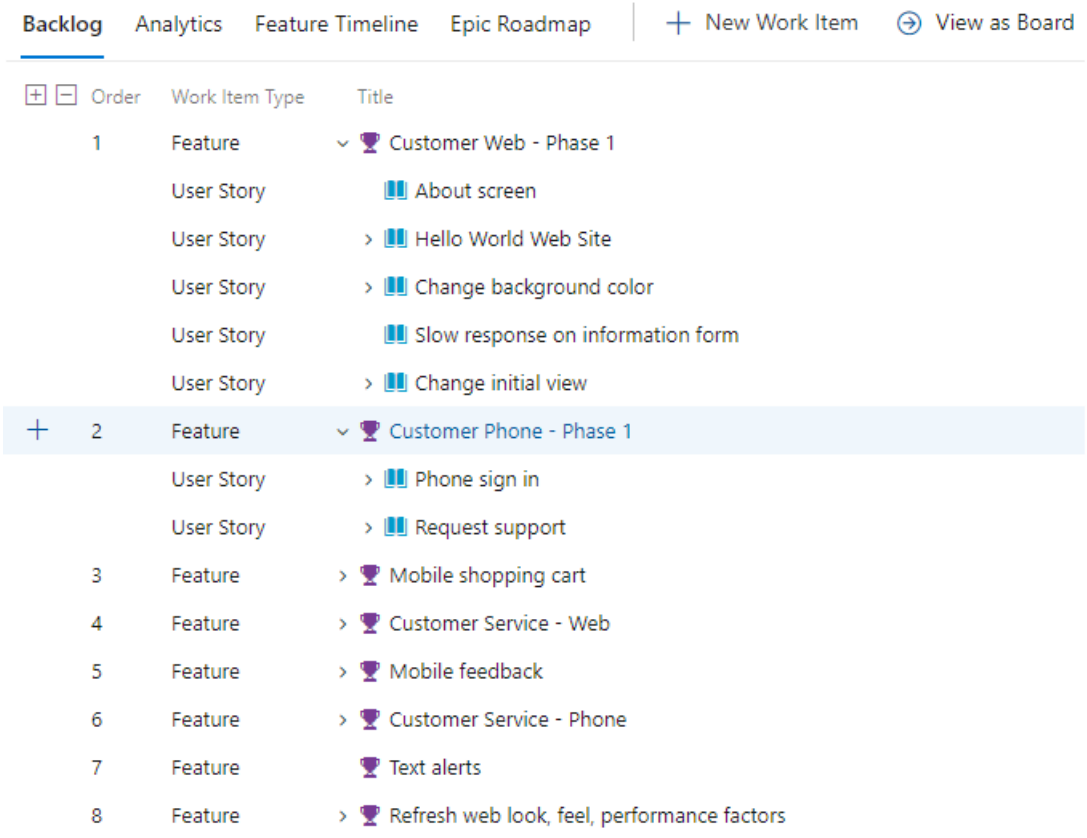
Features backlog

Project managers initiate the product plan by adding features to the Features backlog. Each feature should represent a shippable deliverable that addresses a customer need.



Product backlog

Development teams add User Stories to their product backlog so that the User Story is automatically assigned the team's default Area Path and Iteration Path. Then, they can map those stories under each Feature that represents the work they'll do to implement the Feature. Each User Story should be sized so that they can be completed within a sprint.



Groom each backlog

Periodically review each backlog to perform these tasks:

- Define work to be performed
- Reorder work items using drag-and-drop so that they appear in priority order
- Open work items and add details
- Assign work to team members or to sprints
- Capture technical debt and non-feature work required to support a healthy ecosystem of delivery

- Map unparented work to the Feature they belong to
- (Optional) Estimate size of requirements to help determine team velocity and support forecasting

> **TIP**
> You can monitor team velocity based on estimates assigned to completed work or a simple count of work items completed during sprints. However, to use the Forecast feature, you must assign a value to the Story Points, Effort, or Size field. If you don't want to estimate requirements, you can simply assign a value of 1 to requirement estimates and then use the Forecast tool based on a count of work items.

Best practice tips:

- Periodically refine your backlog
- Make sure features and requirements are sized appropriately
- Define the acceptance criteria and the definition of done for features and work
- Map unmapped work to Features
- Set your view options to support the backlog tasks you want to accomplish
- Forecast your backlog

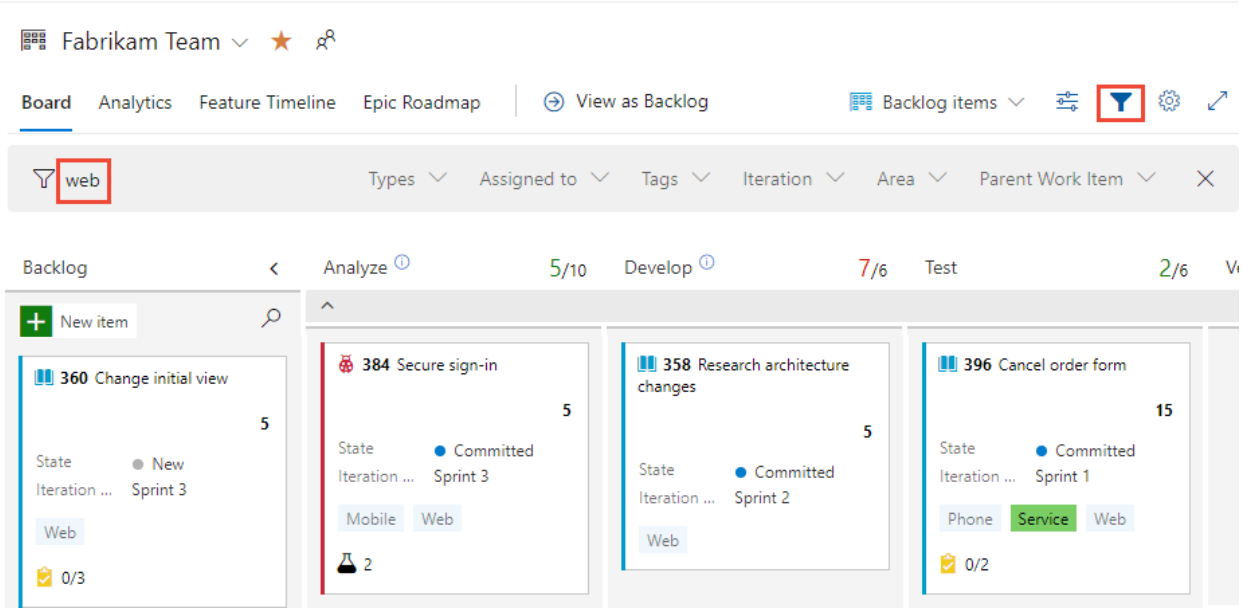
To learn how:

- Refine your backlog
- Define features and epics
- Create your backlog
- Configure your backlog view
- Forecast your product backlog

Use tags to support queries and filtering

With work item tags, team members can assign ad-hoc tags to work items. You can use these tags to filter backlogs and boards as well as query on work items. For tags to be useful to the team, provide some general guidance on how your team should use tags. Consider documenting this guidance in a central place, such as the project wiki.

The following image illustrates a Kanban board filtered on the *web* keyword which displays cards with the *Web* tag.



Best practice tips:

- Have a policy in place about how your teams will use tags

- Indicate how you'll use tags to support queries, filtering, reporting
- Consider using tags to identify cross-team or cross-project dependencies

To learn how:

- Add work item tags to categorize and filter lists and boards
- Filter your Kanban board
- Create a Wiki for your project

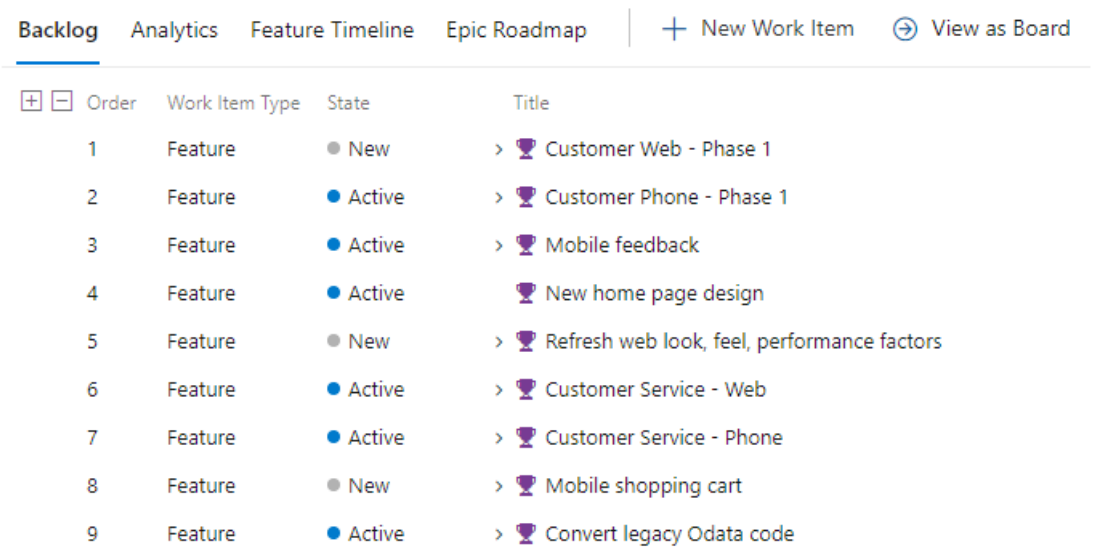
Forecast and milestone planning

To gain insight into what features can ship when, use the **Forecast** tool. This tool requires that you provide estimates to the Story Points, Effort, or Size field for each requirement. If you want to forecast on a simple count of work items, then simply assign the value of **1** to requirement estimates.

Order the features backlog in priority order

As project managers, you'll want to always have your features backlog in priority order. This conveys to the development team which features are most important to complete first.

Here the features backlog shows the sequence of features to ship.



Order the requirements backlog based on parent features

First you want to make sure you are completing the requirements needed to ship features. As shown in the following image, the requirements backlog has been ordered according to the features you want to ship. This ordering assumes that all requirements in a feature must be complete in order to ship it. Also, Story Points have been assigned to each User Story.

Backlog | Analytics | Feature Timeline | Epic Roadmap | + New Work Item | View as Board | Column Options | ...

Order	State	Parent	Story Points	Title
1	New	Customer Web - Phase 1	3	About screen
2	New	Customer Web - Phase 1	3	Change initial view
3	New	Customer Web - Phase 1	5	Hello World Web Site
4	New	Customer Web - Phase 1	3	Slow response on information form
5	New	Customer Web - Phase 1	5	Change background color
6	New	Customer Phone - Phase 1	2	Phone sign in
7	New	Customer Phone - Phase 1	3	Request support
8	New	Mobile feedback	8	Design feedback interface
9	New	Mobile feedback	8	Develop mobile interface
10	New	Refresh web look, feel, performance factors	3	Add an information form
11	New	Refresh web look, feel, performance factors	5	Check performance
12	New	Refresh web look, feel, performance factors	3	Interim save on long form
13	New	Customer Service - Web	5	Technician dashboard improvements
14	New	Customer Service - Web	8	Scheduler
15	New	Customer Service - Web	3	Check issues with permissions
16	New	Customer Service - Phone	5	Switch context issues
17	New	Customer Service - Phone	5	GSP locator interface
18	New	Customer Service - Phone	2	Check service status

Forecast the requirements backlog

With estimates assigned to each requirement, you can set a team velocity. In the example below, we specify 12 for the velocity, equivalent to stating that on average the team can complete 12 Story Points per sprint. The Forecast tool shows which requirements and features the team can complete within the next six sprints. Using the Planning tool, you can quickly assign requirements to the forecasted sprints.

To see the full image, click the image to expand. Choose the close icon to close.

Backlog | Analytics | Feature Timeline | Epic Roadmap | + New Work Item | View as Board | Column Options | ...

Forecasting based on velocity of **12**

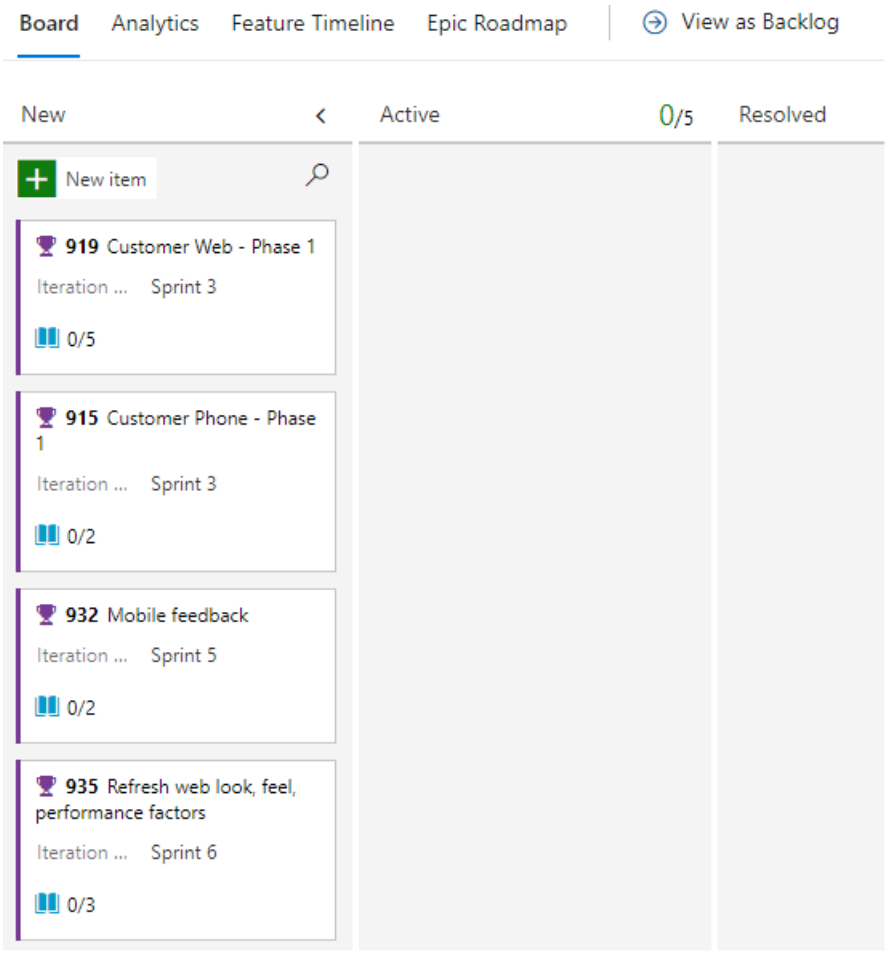
Forecast	Order	State	Parent	Story Points	Title	Iteration Path
	1	New	Customer Web - Phase 1	3	About screen	Design Agile\Sprint 2
	2	New	Customer Web - Phase 1	3	Change initial view	Design Agile\Sprint 2
Sprint 2	3	New	Customer Web - Phase 1	5	Hello World Web Site	Design Agile\Sprint 2
	4	New	Customer Web - Phase 1	3	Slow response on information form	Design Agile\Sprint 3
	5	New	Customer Web - Phase 1	5	Change background color	Design Agile\Sprint 3
	6	New	Customer Phone - Phase 1	2	Phone sign in	Design Agile\Sprint 3
Sprint 3	7	New	Customer Phone - Phase 1	3	Request support	Design Agile\Sprint 3
Sprint 4	8	New	Mobile feedback	8	Design feedback interface	Design Agile\Sprint 4
	9	New	Mobile feedback	8	Develop mobile interface	Design Agile\Sprint 5
	10	New	Refresh web look, feel, performance factors	3	Add an information form	Design Agile\Sprint 5
Sprint 5	11	New	Refresh web look, feel, performance factors	5	Check performance	Design Agile\Sprint 5
	12	New	Refresh web look, feel, performance factors	3	Interim save on long form	Design Agile\Sprint 6
Sprint 6	13	New	Customer Service - Web	5	Technician dashboard improvements	Design Agile\Sprint 6
	14	New	Customer Service - Web	8	Scheduler	Design Agile

Getting good at estimates and having predictable team velocities are useful team goals for process improvement.

Update your Features board

With a forecast of when a feature will ship, you can update each feature's iteration path. Quickly assign values to

a feature by adding those fields to the card on the Kanban board as shown in the following image.



Milestone planning

Milestone markers aren't used in Azure Boards work tracking, except for Delivery Plans. Delivery Plans provide a calendar view and allow you to define a milestone marker. However, you can use one or more of the following options to mark a work item as a milestone:

- Simply prepend or append the word **Milestone** in the title of your work item
- Add a work item tag labeled **Milestone**
- Add a custom field labeled **Milestone** and populate it with a pick list of milestones
- Link work items using the Predecessor/Successor or Related link type to a milestone work item
- Assign a milestone work item to the sprint in which it's targeted for completion.

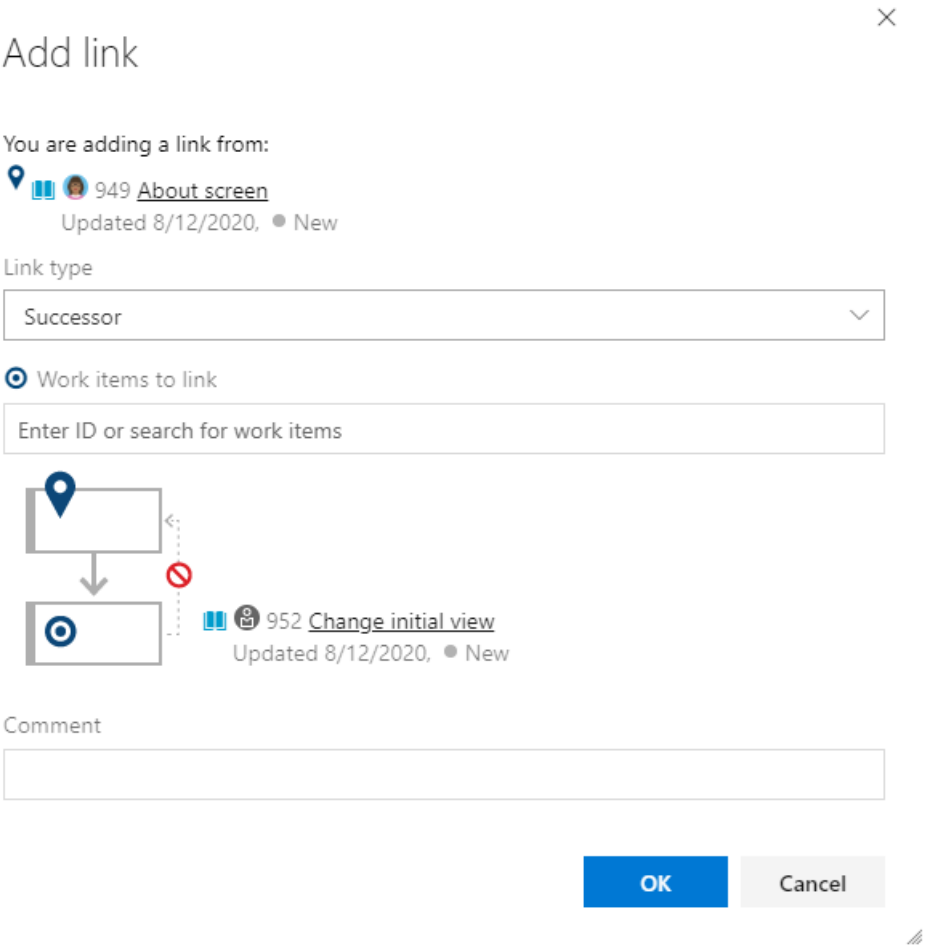
Manage dependencies

In Microsoft Project, you manage tasks that depend on the completion of other tasks by linking them. To manage dependencies in Azure Boards, you can add similar linking by adding Predecessor/Successor link types to work items. You add these links from the Add link dialog for a work item.

Add link dialog

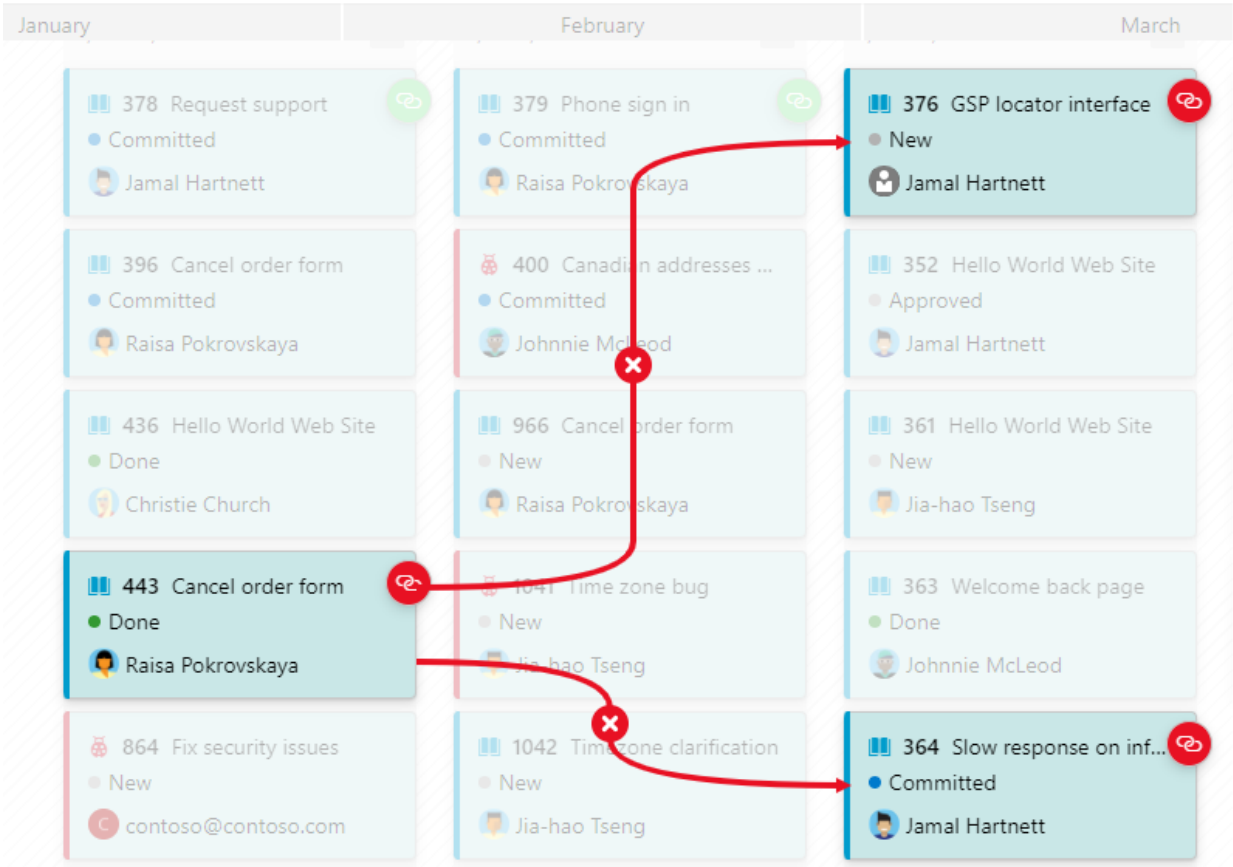
Azure Boards supports many link types to track related work. Choose the **Predecessor/Successor** link types to track work with dependencies. A quick way to link work items is to add a tag to work items that participate in producing or consuming dependencies, create a query based on this tag, and then add the required links from the triage mode of the query results.

The following Add link dialog illustrates how two work items are linked using the Successor link type.



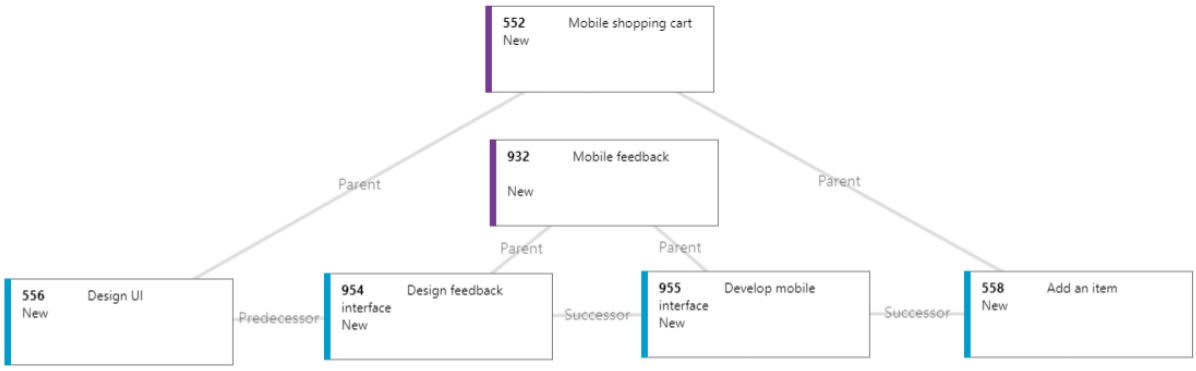
Visualize work item relationships

You can view dependencies and identify dependencies that have issues with Delivery Plans. As shown in the following image, you can toggle the display of dependency lines between linked work items. To learn more, see Track dependencies using Delivery Plans.



With the Work Item Visualization Marketplace extension, you can visualize the link relationships among several work items as shown in the following image.

To see the full image, click the image to expand. Choose the close icon to close.



Minimum Viable Product versus Critical Path Management

Azure Boards doesn't provide a native view of the critical path. In part, as Agile methodologies favor a Minimum Viable Product (MVP) over Critical Path Management (CPM). By using MVP, you identify the shortest path and dependencies by prioritizing epics, features, stories and tasks. For additional context, see The Critical Path on Agile Projects and Running a lean startup on Azure DevOps.

Best practice tips:

- Add a `dependency` tag to work items participating in dependency management
- Use **Predecessor/Successor** link types to track dependencies of work owned by other teams or within other projects
- Create queries to track, add, and triage dependencies
- Use Delivery Plans to view work that you have dependencies on from other teams
- Use the Work Item Visualization Marketplace extension to visualize dependencies for a specific work item within the work item form.

> **NOTE**
> Marketplace extensions are not supported features of Azure Boards and therefore not supported by the product team. For questions, suggestions, or issues you have when using these extensions, visit their corresponding extension page.

To learn how:

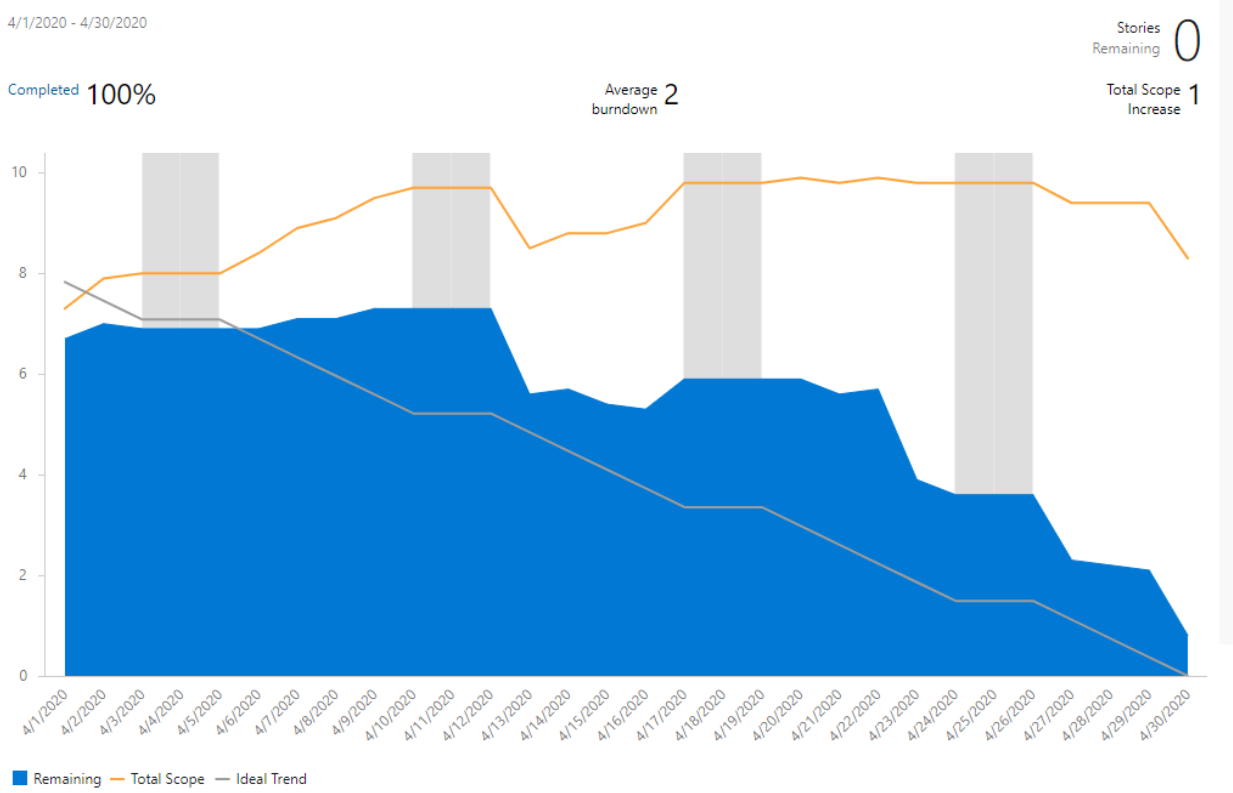
- Link user stories, issues, bugs, and other work items
- Triage work items
- Track dependencies by using Delivery Plans

Work in sprints

Sprints allow the development team to focus on completing a pre-selected set of work. Work assigned to a sprint appears on the team's sprint backlog. Sprint backlogs are defined only for product backlogs, not for portfolio backlogs.

Sprint burndown chart

By updating the status of work daily throughout a sprint, you can easily track sprint progress with the Sprint burndown chart, as shown in the following image.



Best practice tips

Each sprint, perform the following tasks:

- Plan each sprint with your team
- Use the team's Sprint backlog to review sprint deliverables
- Ensure each sprint work item is assigned to a team member
- Ensure each work item is scoped to be completed within the sprint
- Ensure the acceptance criteria for the work is well defined and understood
- Update the status of sprint work items as work moves from a *New* to *Active* to *Completed* state to track sprint burndown
- Check in with other teams on dependencies that your team's work depends on
- Monitor sprint progress using the Sprint burndown chart

To learn how:

- Assign backlog items to a sprint
- Configure and monitor sprint burndown
- Define features and epics

Review progress and feature deliverables

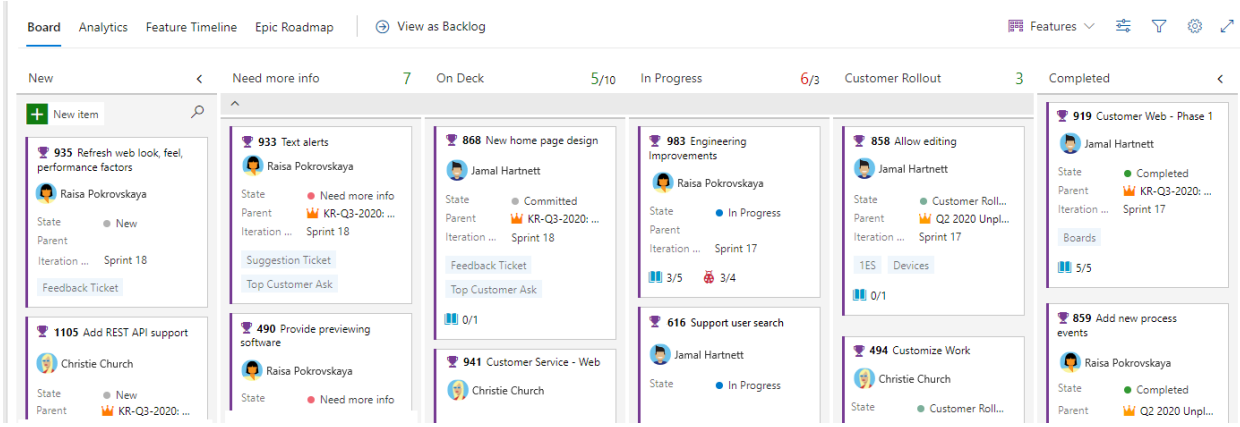
The three main tools you'll want to use to review progress and deliverables are:

- Features Kanban board
- Features backlog with rollup columns
- Delivery plans

Features Kanban board

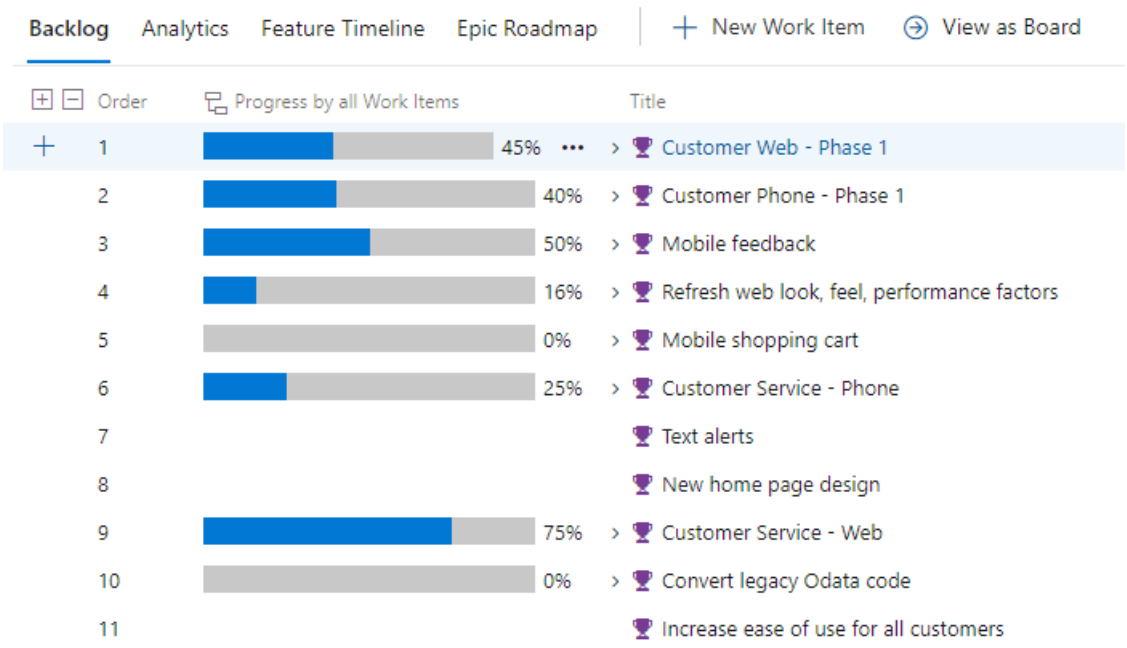
Your Features board is another place to review progress and ensure the continuous flow of deliverables. The following image illustrates a customized Features board. In progress columns have been added such as *Need more info*, *Spec Complete*, *In Progress*, and *Customer Rollout*. These provide a more natural set of states as Features get proposed, researched, designed, developed, and then deployed to production.

To see the full image, click the image to expand. Choose the close icon to close.



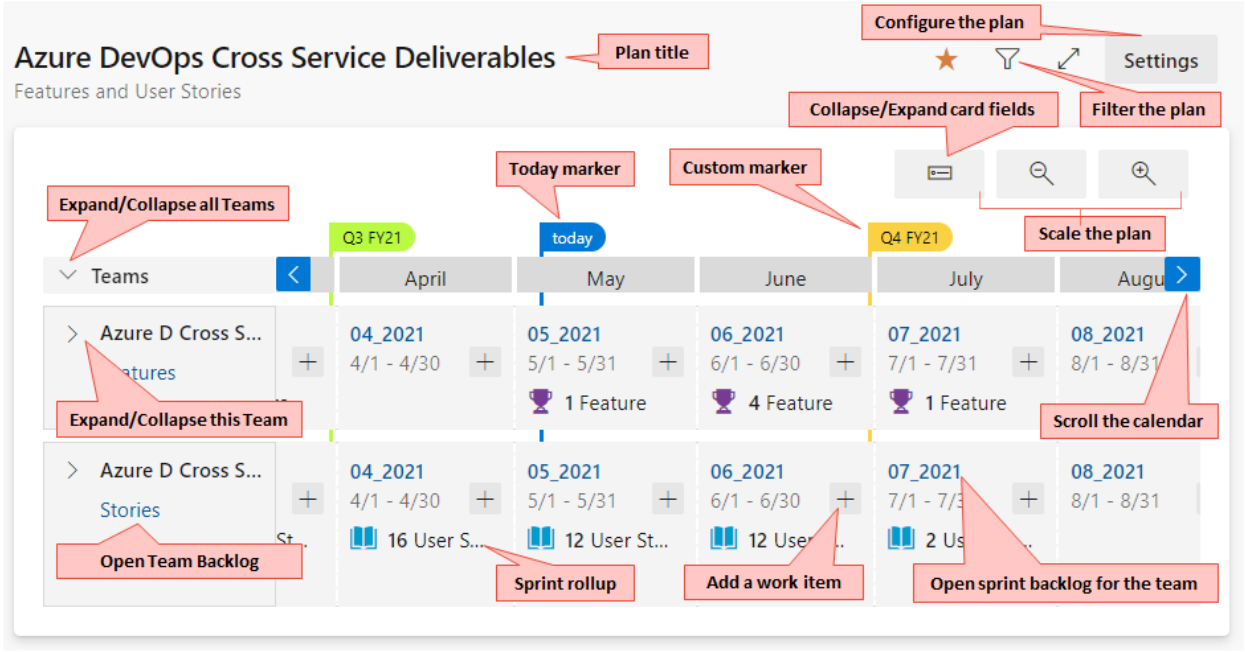
Rollup

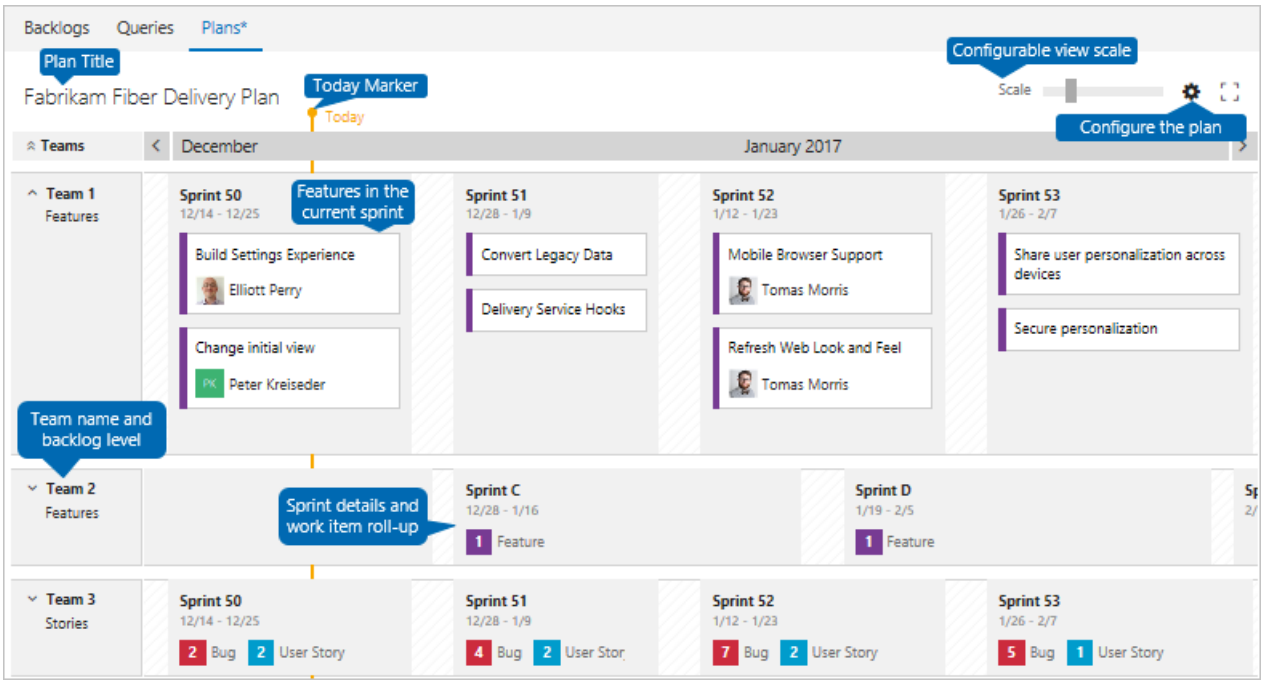
One quick and visual way to monitor progress is from the Features backlog. By adding the rollup progress bar column, you can see what percentage of work items are completed for each feature, as shown in the following image.



Delivery plans and multiple team deliverables

To review features delivered across several teams, configure a delivery plan. Delivery plans provide an interactive board to review a calendar schedule of stories or features several teams plan to deliver.





Best practice tips

- Customize your Features Kanban board to support your team's processes
- Add fields to cards so that you can update their values quickly and easily
- Update the Iteration Path (Sprint) of Features as you gain clarity as to when they will ship
- Review the Features board to talk through status, blocks/issues/risks/changes,and update status
- Use the filter feature to focus on tagged items, assigned by features, a specific sprint and more
- Add rollup columns to your Feature backlog to monitor overall progress based on work item count completion
- Use delivery plans to review features being delivered by several teams and discuss cross-team dependencies

To learn how:

- Add columns to your Kanban board
- Customize cards
- Filter your Kanban board
- Display rollup progress or totals

- Review team Delivery Plans

Process improvement

At the heart of Agile methods is continuous improvement. To improve your processes, you need to have shared goals and a shared plan. To initiate process improvement activities, consider adding them through regular practices, such as:

- Sprint planning
- Setting sprint goals
- Conducting regular retrospectives

Consider the following questions when setting goals:

- *What are you learning about your customers? What do you need to know?*
- *What data is being measured? Is it actionable? What data needs to be measured?*
- *How is the flow of deliverables? Is it as expected? Where can improvements be made?*
- *Are your team members empowered to do their best? What tools or information would help them improve?*
- *How well is information being shared? How well are teams collaborating?*
- *How well is your team managing technical debt and closing bugs?*

Some of the Agile tools you can use to support process improvement are team velocity, team dashboards, and the Retrospectives Marketplace extension.

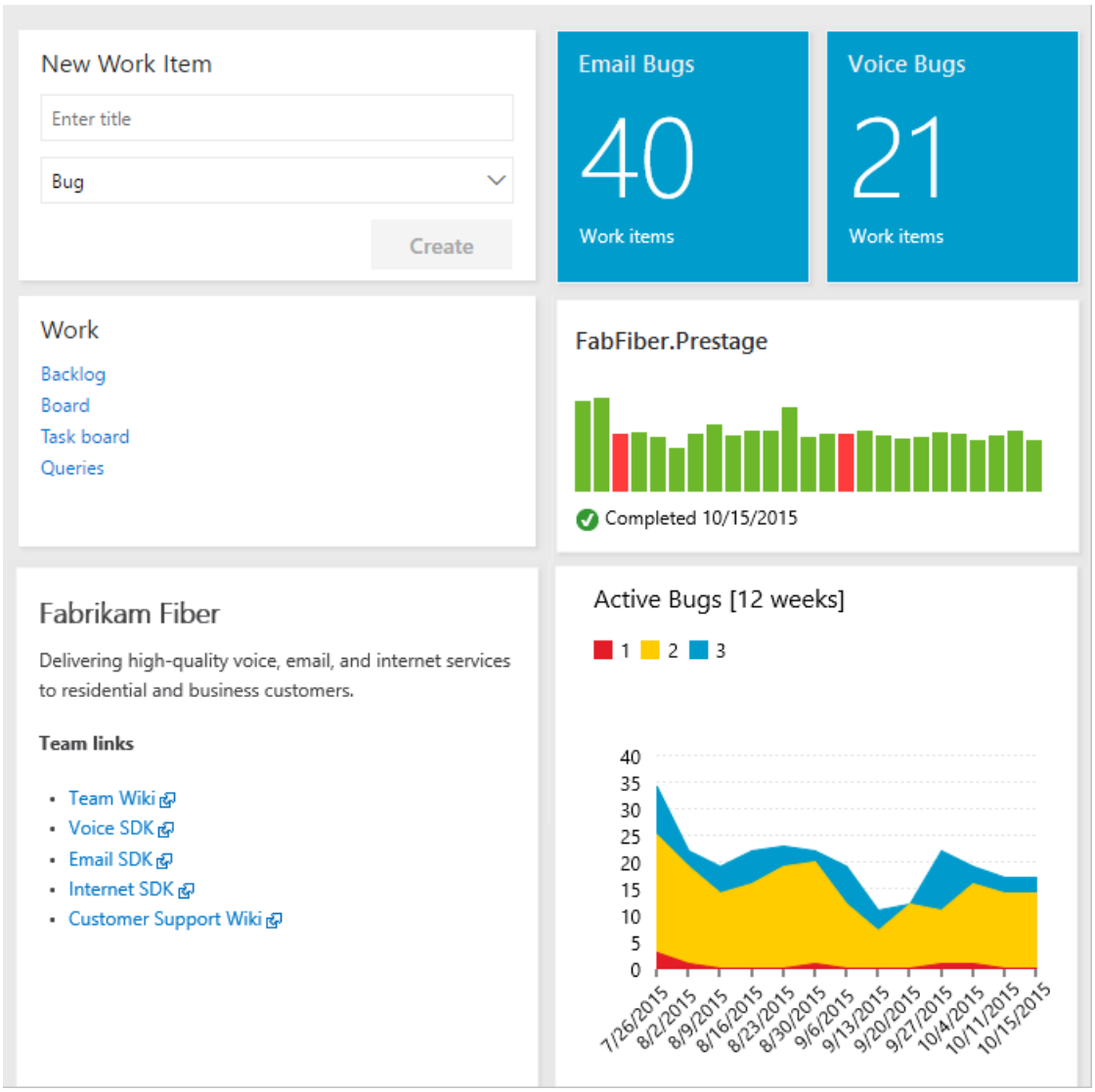
Team velocity

From the team velocity chart, you can gain an understanding as to how well the team is planning and executing a sprint. As shown in the following example, the velocity chart shows the planned, completed, completed late, and incomplete count of work items for several sprints. Teams can review this chart to help determine how well they are estimating and executing and how they might improve.



Team dashboards

Teams can define one or more dashboards to share information and monitor real-time data on work progress.



Best practice tips

- Identify process improvement goals that your team can agree to, write them down and review them periodically
- Use team dashboards to share information and work tracking charts which you and your team review periodically
- At sprint planning meetings, have your team identify at least one sprint goal related to process improvement
- Conduct regular retrospectives to capture what went well, what didn't go well, and actions to improve
- Maintain an improvement tracking board, such as that available with the Retrospectives Marketplace extension.

To learn how:

- View or configure team velocity
- Add, rename, and delete dashboards
- Scaling Agile - Practices that scale
- Retrospectives Marketplace extension

Try this next

Configure and customize Azure Boards

Related articles

- Manage requirements

- Tasks supported by Backlogs, Boards, Taskboards, and Plans
- Work with multi-team ownership of backlog items
- 11 Reasons for using Azure Boards to plan and track your work

Industry articles

- Agile and a continuous improvement mindset
- What is KAIZEN™

Key concepts and work item tasks in Azure Boards and Azure DevOps

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use this index to quickly access concepts related to work items and information on adding and updating work items—such as users stories, features, tasks, and bugs.

Key concepts

- Agile glossary
- Agile process
- Area Paths
- Autocomplete work items
- Assigned to
- Basic process

- Chart for work items widget
- Charts
- Client tools
- CMMI process
- Customization process models
- Dependencies
- Delivery plans

- Filtering
- Following
- Inheritance process model
- Iteration Paths
- Keyboard shortcuts

- Link types
- Linking and traceability
- Mobile browser
- New Boards Hub
- New work item widget
- On-premises XML process model
- Permissions and access
- Process guidance
- Process models

- Queries
- Recycle bin

- Remote linking
- Rich text fields
- Rollup
- Scrum process
- State categories

- Queries
- Recycle bin
- Rich text fields
- Rollup
- Scrum process
- State categories

- Queries
- Recycle bin
- Requirements
- Rich text fields
- Rollup
- Scrum process
- State categories

- Tags
- Track bugs as requirements or tasks
- Track dependencies

- Visual Studio work item experience

- Work item fields
- Work item form
- Work item templates
- Work item types
- Work tracking limits
- Workflow

Work item user tasks

Tasks listed below are available to users with Contributor permissions and Basic access.

- Add a work item
- Add Epics
- Add Features
- Add items to a backlog
- Add items to a Kanban board
- Add tags
- Add tasks
- Add to discussion
- Apply a template to a work item
- Assign work to a team member

- Attach files

- Bulk add or remove tags
- Bulk modify work items (Excel)
- Bulk modify work items (Web)

- Capture work item as a template
- Change work item type
- Copy or clone a work item
- Copy work item URL
- Copy list of work items
- Create a branch
- Create a work tracking chart

- Define a work item template
- Define work item URL
- Delete work item tags
- Delete work items
- Display rollup

- Define a work item template
- Define work item URL
- Delete work item tags
- Delete work items

- Email work item list
- Export a work item list
- Filter a backlog, board, or plan
- Filter the History tab
- Follow a work item
- Forecast work items
- Get notified of work item changes
- Group work items

- Link to cross-organization work items
- Link to development objects
- Link to GitHub commits and pull requests
- Link to work items from a wiki
- Link work items
- List work items
- List work items in a wiki

- Manage bugs
- Manage issues or impediments
- Manage work item tags
- Map work items
- Move work items to a sprint
- Move work items to another project

- Open work items
- Print work items
- Prioritize backlog items
- Query work item history
- Query for work items

- Reassign work items
- Remove work items
- Request feedback
- Restore deleted work items

- Start storyboarding
- Track dependencies
- Triage work items

- Update status of tasks (Taskboard)
- Update status of work items (Kanban board)
- Use #ID to link
- Use @mentions

- View history
- View work items (mobile)
- View work items (web)
- View work assigned to me
- View work I'm following
- View work I've recently viewed or updated
- View work recently completed
- View work recently created
- View work where I'm mentioned

- View history
- View work items (mobile)
- View work items (web)

Administrative customization tasks

Tasks listed below must be performed by an administrator who has the necessary permissions, as they affect all users and teams within a project.

You customize work item types using the Inheritance process model.

You customize work item types using either the Inheritance process model or On-premises XML process model, depending on the selection made when you created the project collection on which the project is defined.

Inherited process model

- Add a checkbox (Boolean) field
- Add a custom field
- Add a custom work item type

- Add/remove custom fields
- Add/remove custom groups
- Add/remove custom pages
- Add/remove a custom control
- Add/remove custom rules to a field
- Add a person-name/Identity
- Add a picklist (drop-down menu)
- Add a rich-text (HTML) field
- Add, edit, or remove a WIT workflow state

- Change a field label
- Change the WIT color or description
- Change the reference process from Agile to Scrum
- Change the reference process from Basic to Agile
- Change the reference process from Scrum to Agile
- Create a project

- Define Area Paths
- Define Iteration Paths
- Delete field
- Delete a WIT
- Enable/disable a WIT

- Modify a default pick list
- Move the field within the layout
- Remove a field from form
- Restrict modification
- Set Required/Default options

- Add a custom field
- Add a custom work item type (WIT)
- Add/remove custom fields from a WIT
- Add/remove custom groups on a WIT form
- Add/remove custom pages on a WIT form
- Add/remove a custom control on a WIT form
- Add/remove custom rules to a field
- Add a person-name/Identity
- Add a picklist (drop-down menu)
- Add a rich-text (HTML) field
- Add, edit, or remove a WIT workflow state

- Change a field label
- Change the WIT color or description
- Create a project

- Define Area Paths
- Define Iteration Paths

- Delete field
- Delete a WIT

- Enable/disable a WIT
- Modify a default pick list
- Move the field within the layout

- Remove a field from a form
- Restrict modification
- Set Required/Default options

On-premises XML process model

You customize work item types using the On-premises XML process model. For more customization options, see On-premises XML process customization.

- Add a checkbox (Boolean) field (TFS 2017.2)
- Add a custom field
- Add a custom pick list
- Add or modify a field
- Add rules to a field
- Add a custom control field
- Add fields that integrate with test, build, and version control
- Add/remove State or Reason fields (customize workflow)

- Change a field attribute
- Change a field label
- Define Area Paths
- Define Iteration Paths
- Define global lists
- Delete a field
- Modify a predefined pick list
- Remove a field from a form

Related articles

- Query quick reference
- Work item field index
- Quick guide to default permissions and access
- Best tools for adding, updating, and linking work items

How to get started with Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

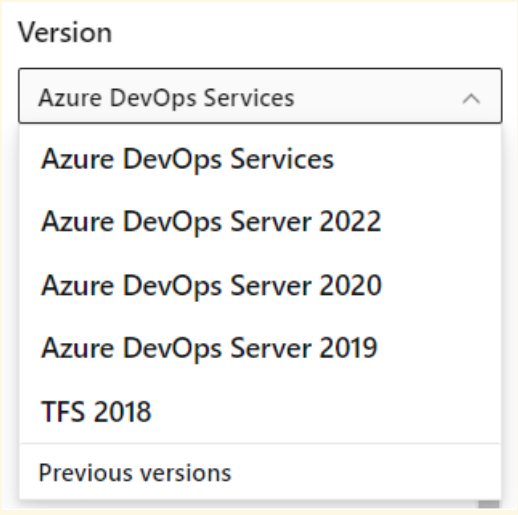
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use this guide to sign up and start using Azure Boards. Start with Sign up and invite some teammates.

Then, read Plan and track work to start adding and tracking work items on the Kanban board. To add columns, swimlanes, or fields to your board, see Customize your boards.

> **NOTE**
> This quickstart guide illustrates how to sign up, create a project, and start tracking work using the Basic or Agile processes. If you want more information on working with other processes which offer other work item types, such as user stories and bugs, then see Choose a process. You can create additional projects using different processes.
>
> If you use GitHub and want to track your issues in Azure Boards, see GitHub & Azure Boards.

If you're tasked with managing Azure Boards settings, review Manage your Azure Boards project for other configurations and resources that you may want to make.

> **IMPORTANT**
> 
>
>
> To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
> To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

Related articles

- Why use Azure Boards?
- Default permissions & access (Security)
- Agile glossary

More resources

- Best tool for the job
- Work items
- Sprints (Scrum)
- Web portal navigation

Sign up for free and invite others to collaborate in Azure Boards

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services

Sign up for an Azure DevOps and Azure Boards to start planning and tracking work. We walk you through the sign-up process for Azure Boards. Follow these steps when you want to accomplish the following tasks:

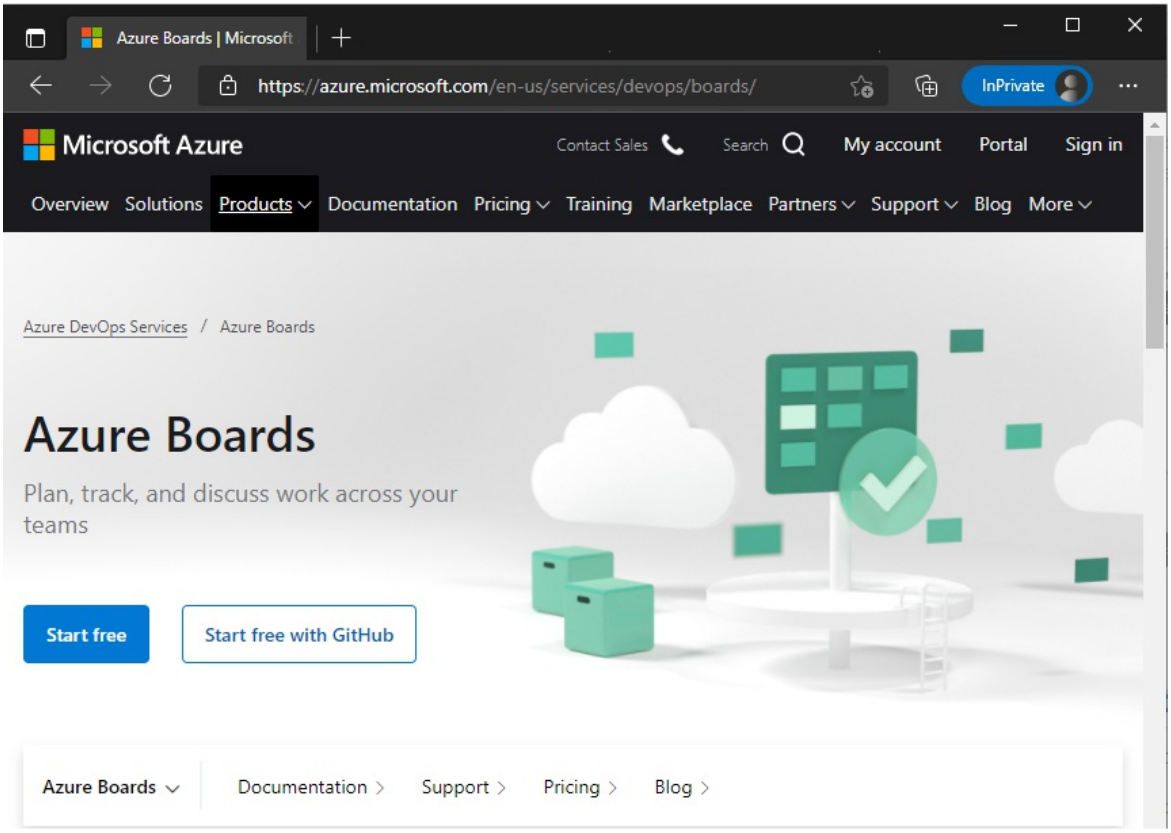
- Create a private or public project with Azure Boards enabled services
- Track work using the Basic, Agile, Scrum, or CMMI process for tracking work
- Create a Git or Team Foundation Version control (TFVC) repository for managing your code
- Invite others to contribute to your project

To sign up and enable all Azure DevOps Services, see Sign up, sign in to Azure DevOps. To learn more about the choices available to you, see these articles:

- What is Azure Boards?
- Choose a process work tracking work
- Choose the right version control for your project.

Choose your sign-up option

1. From your web browser, open the Azure Boards sign-up page. Supported web browsers include the latest versions of Edge, Internet Explorer, Safari (Mac), Firefox, and Chrome.



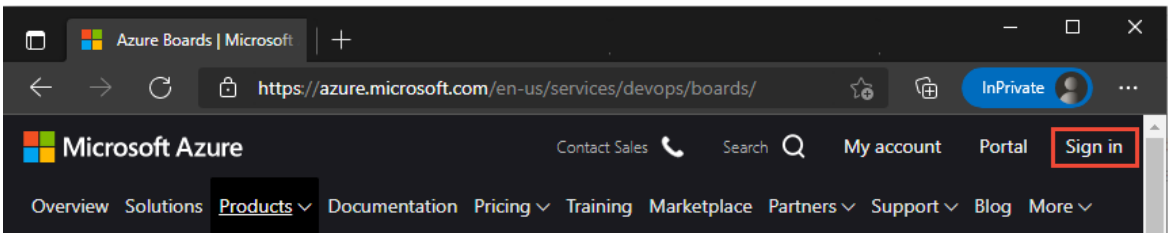
2. Choose one of the following buttons based on the account you want to use.

- **Start free**: Choose this option when:
  - You have a Microsoft account and will sign in using your account email address, phone number, or Skype ID. If you're a Visual Studio subscriber and you get Azure DevOps as a benefit, use the Microsoft account associated with your subscription. Go to Sign up with a personal Microsoft account
  - You want to sign up using a general email address you want to use. Continue to Sign up by creating an account using your email address.

  > **TIP**
  > You can sign up with any valid email address. Signing up for Azure Boards enables your email address as a Microsoft account.

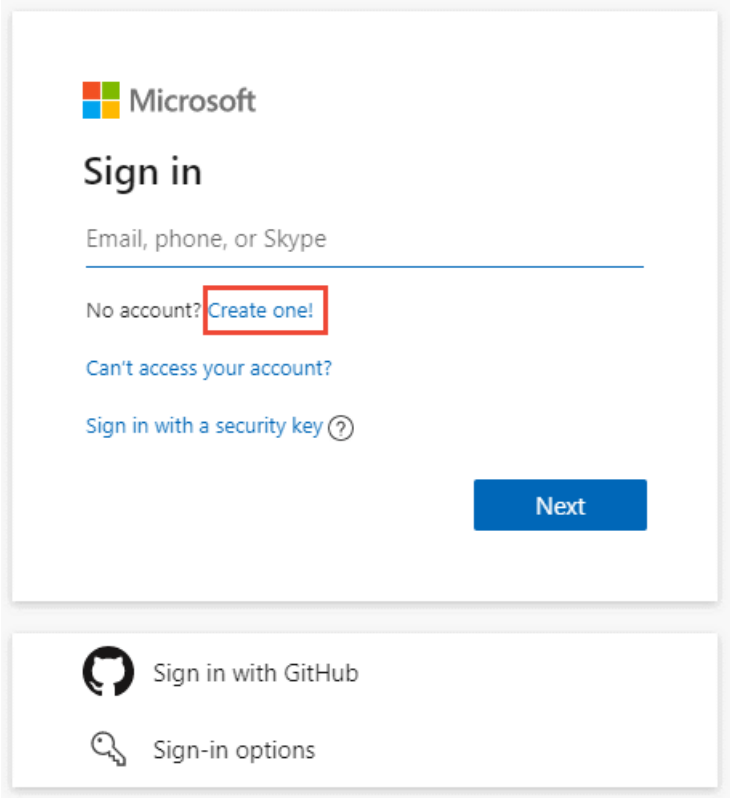
- **Start free with GitHub**: Choose this option if you have an existing GitHub account. Then go to Sign up with a GitHub account.

3. If you've already signed up or have an organization set up to use Azure Boards, choose the **Sign in** link.

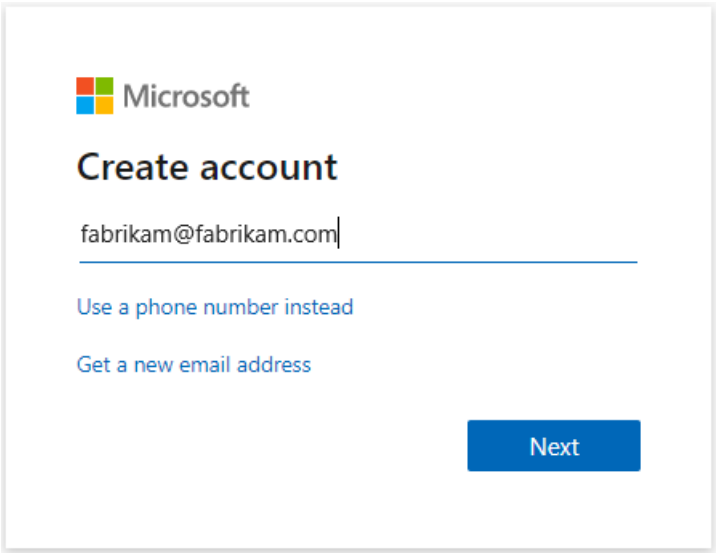


Sign up by creating an account using your email address

1. To sign up with a valid email address, choose **Create one!**. Enter the email address you want to use.

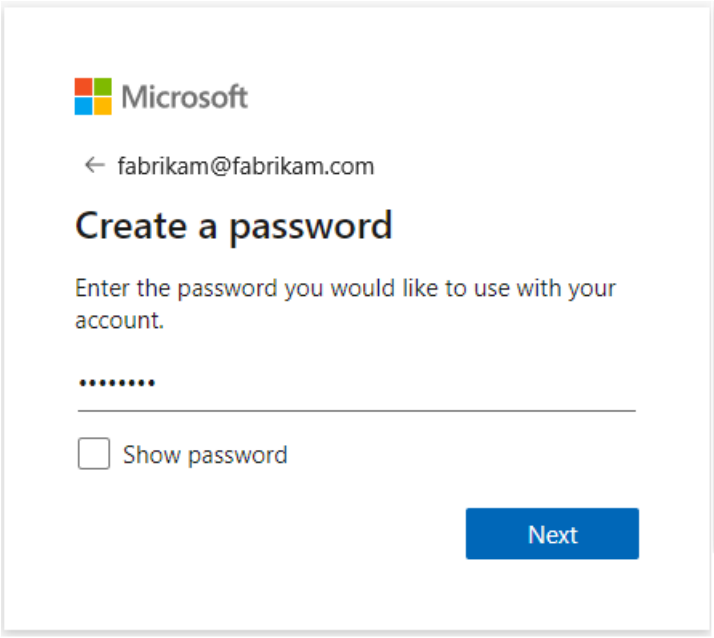


2. Enter the email address you want to use and then choose **Next**.

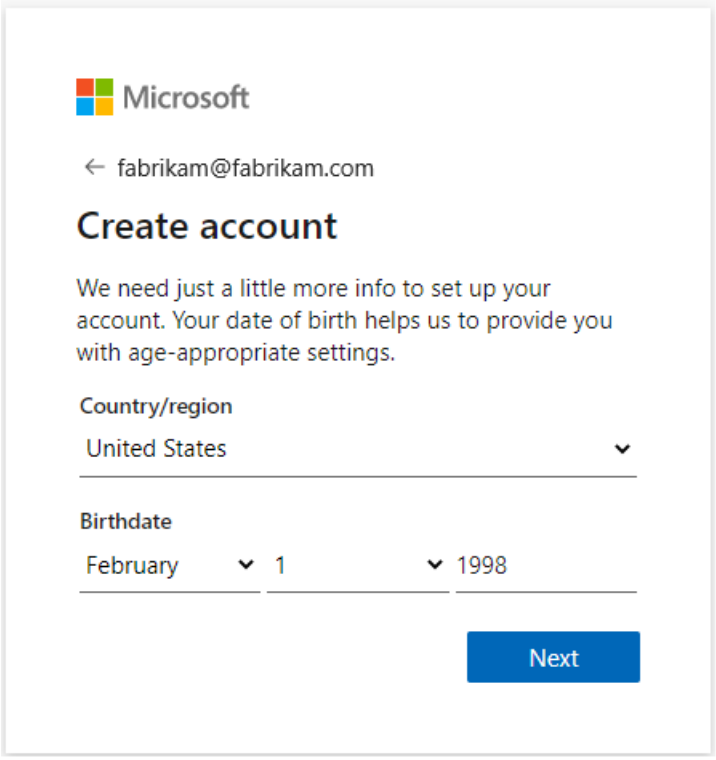


Or, you choose **Get a new email address** to create an Outlook or Hotmail account at this time. To learn more, see create a Microsoft account.

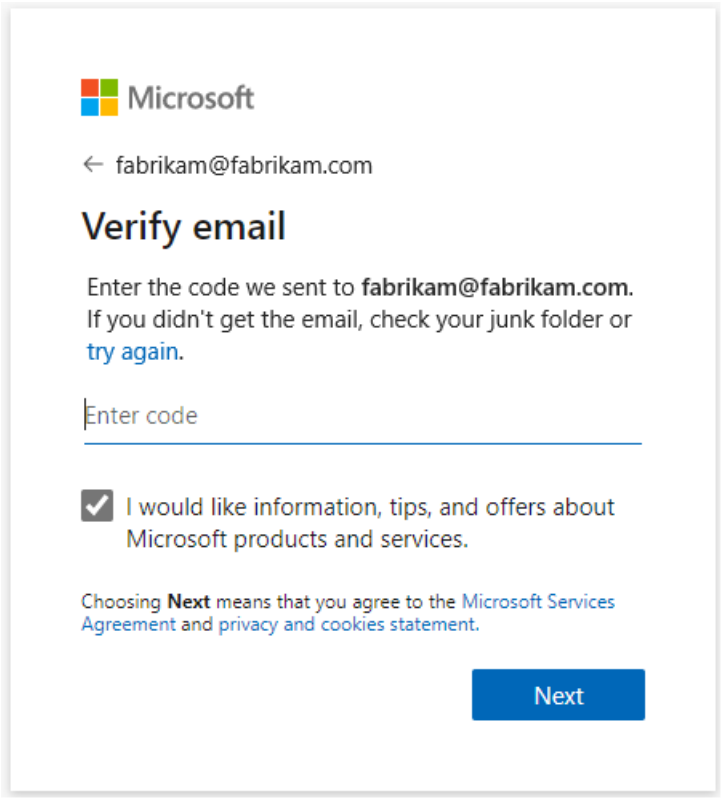
3. Enter the password you want to use with your Azure Boards Microsoft account and then choose **Next**.



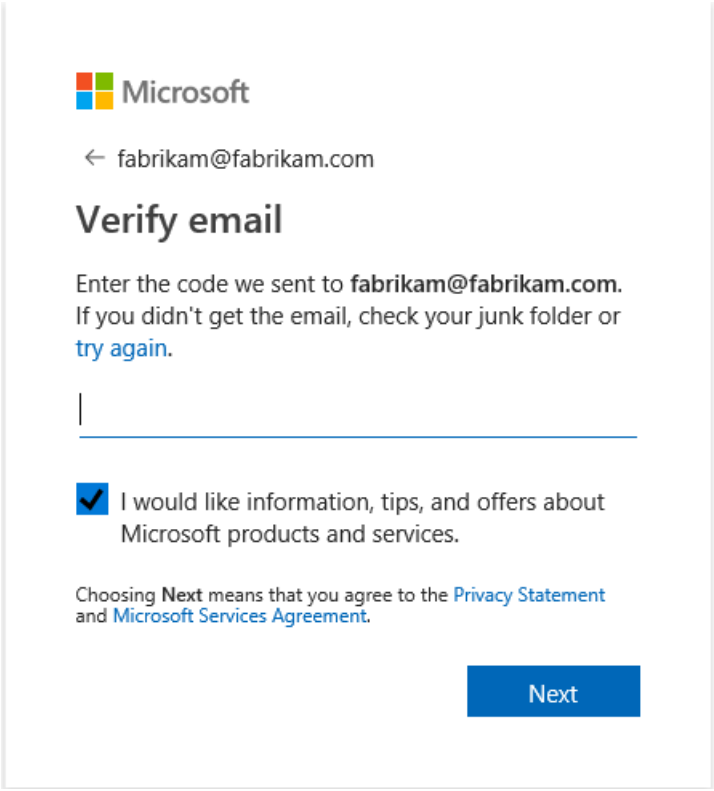
4. Choose the region and specify your birthday to complete your account registration and then choose **Next**.



5. Enter the code sent to your email address to verify your account and then choose **Next**.

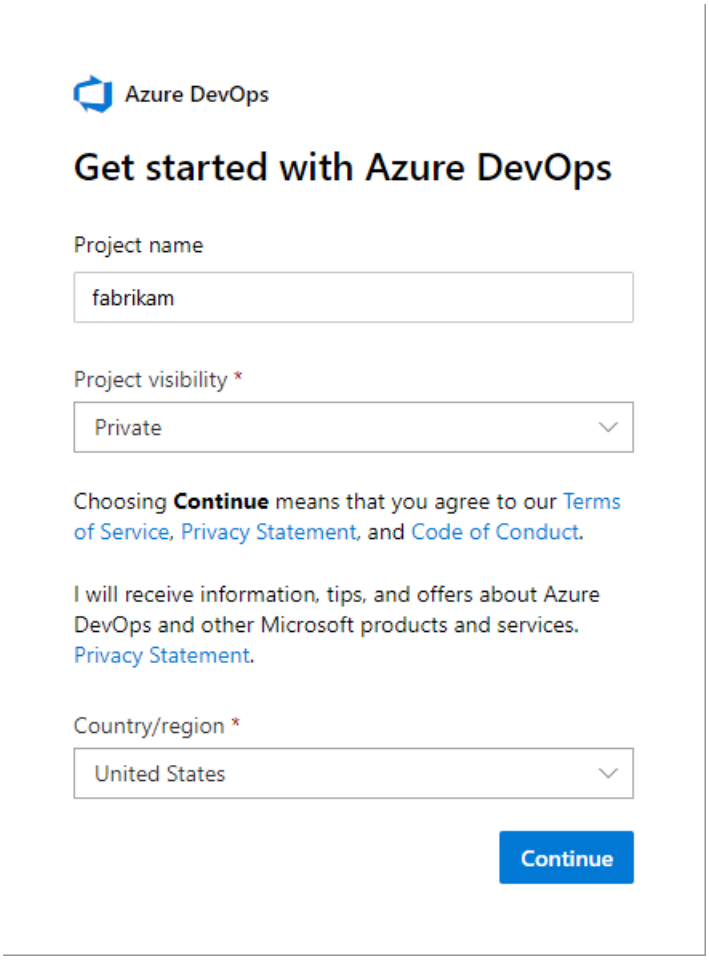


6. Check your email account and enter the code provided. Choose **Next**.



An organization is created based on your account name. Sign in to your organization at any time by entering `https://dev.azure.com/{yourorganization}` in your web portal. You can change the organization name as indicated in Change organization or project settings later in this article.

7. Also, a project is created based on your account name. You can change the project name later. To get started with Azure DevOps, choose **Continue**.



- **Project name**: Can't contain special characters (such as /: \ ~ & % ; @ ' " ? < > | # $ * } { , + = [ ]), can't

begin with an underscore, can't begin or end with a period, and must be 64 characters or less.

- **Visibility**: Choose **Public** if you want to create an open-source project. Otherwise, choose **Private**, so only people who you give access to can view your project.

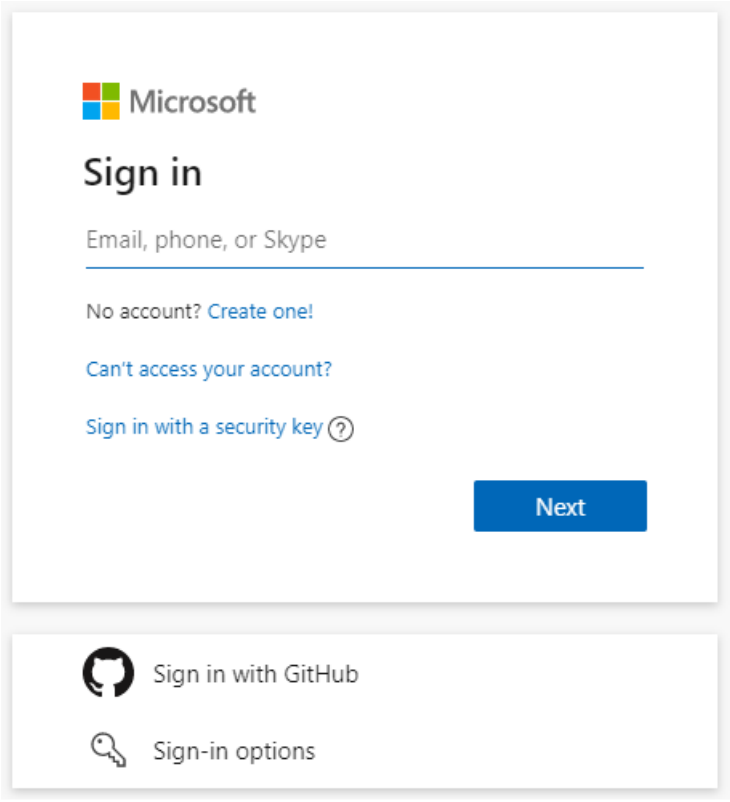
8. Your next step is to start using your Kanban board to track issues and tasks, or invite other users to collaborate with your project.

> **NOTE**
> Your project was created using the Basic process which uses Epics, Issues, and Tasks to track work. If you want a project that uses the Agile, Scrum, or CMMI process, then you can add another project and specify the process through advanced setting options as described in the next section. See Create a project using Advanced settings.

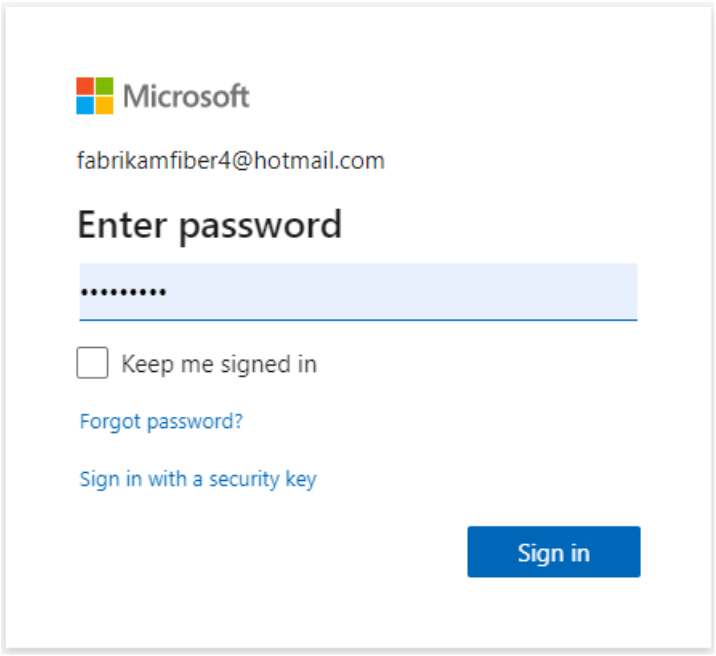
Sign up with a personal Microsoft account

If you have a Microsoft account or Azure Active Directory organizational account, follow these steps to sign up for Azure Boards.

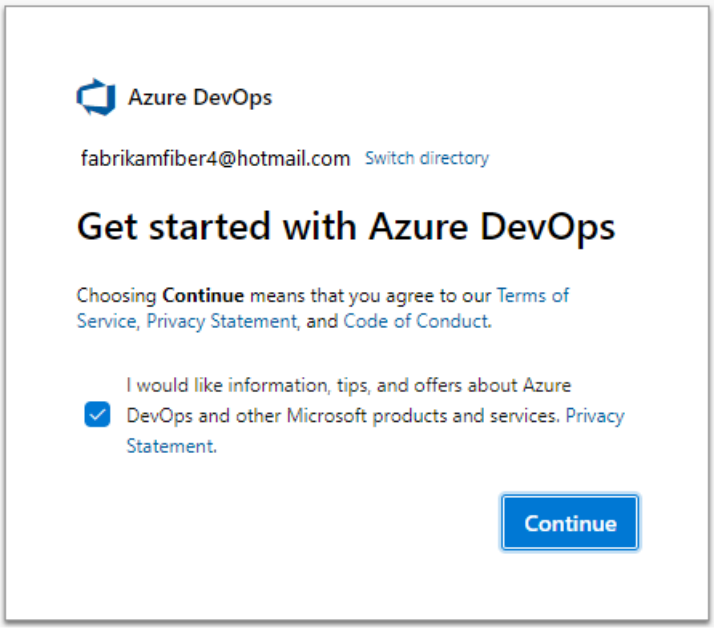
1. Enter your email address, phone number, or Skype ID for your Microsoft account. If you're a Visual Studio subscriber and you get Azure DevOps as a benefit, use the Microsoft account associated with your subscription. Select **Next**.



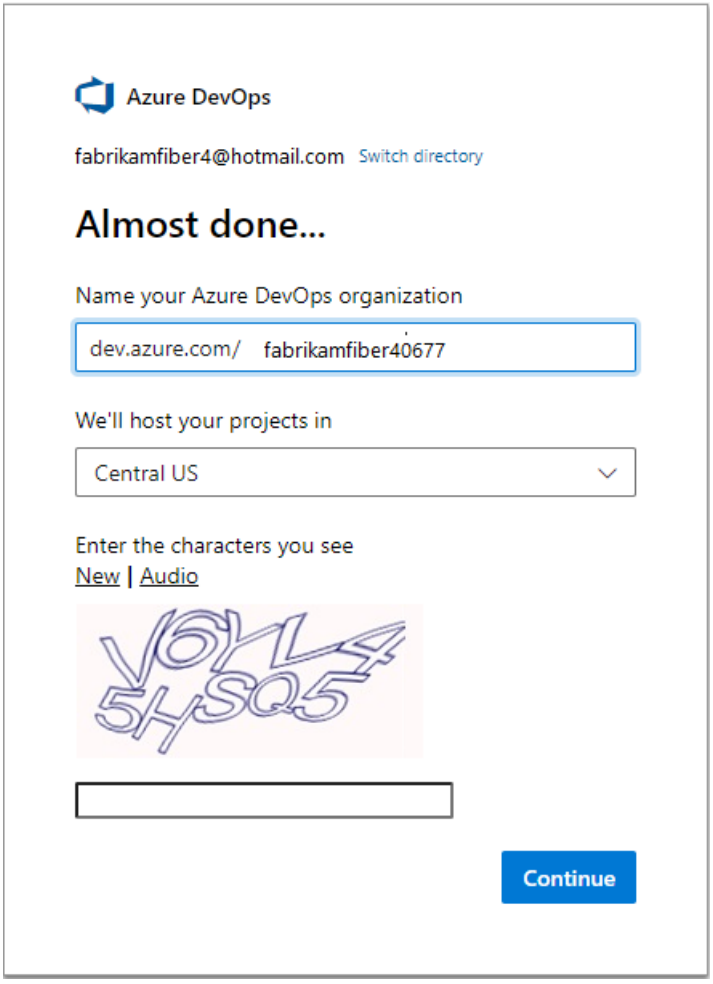
2. Enter your password and select **Sign in**.



3. To get started with Azure Boards, select **Continue**.



4. An organization is suggested based on the account you used to sign in. You can modify the account name. Choose the region where you want your projects hosted. Then enter the characters you see into the text box, and then choose **Continue**.



An organization is created based on the name entered in the **Name your Azure DevOps organization** box.

Use the following URL to sign in to your organization at any time:

```
https://dev.azure.com/{yourorganization}
```

You can change the organization name as indicated in Change organization or project settings later in this article.

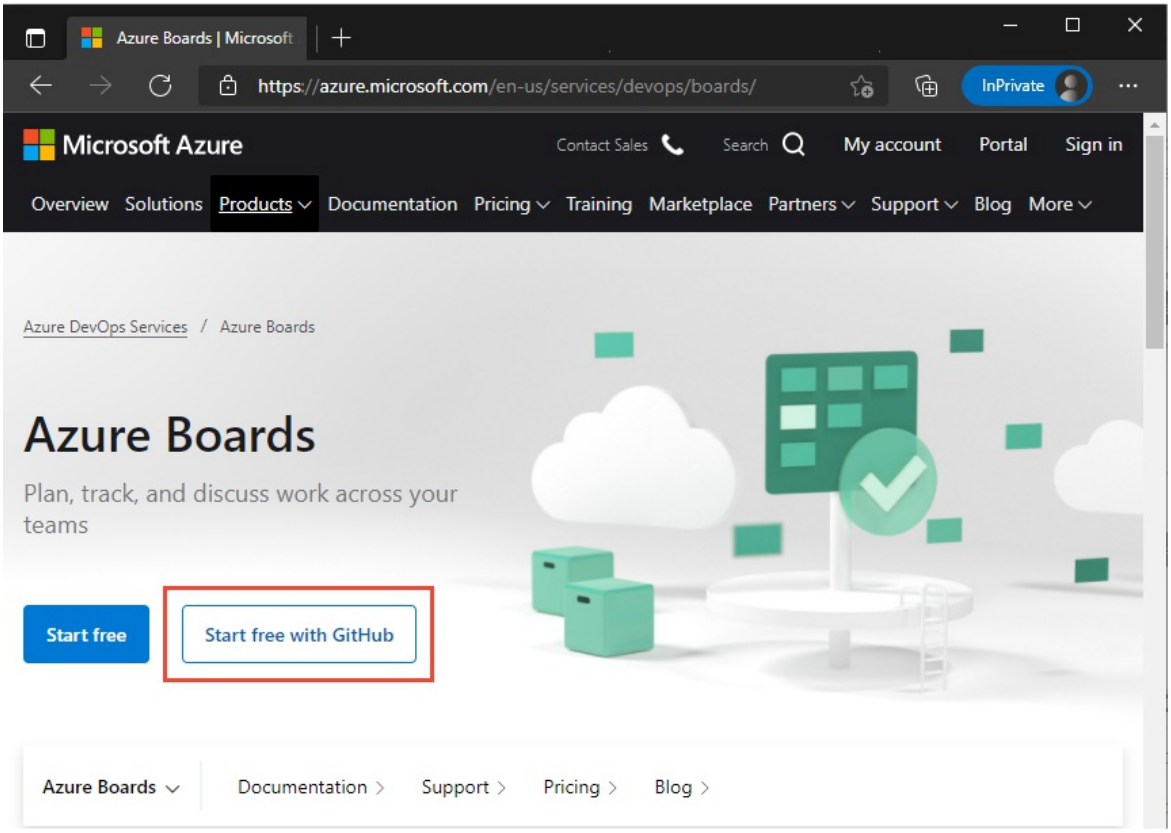
5. To complete your sign-up process, go to create a project.

Sign up with a GitHub account

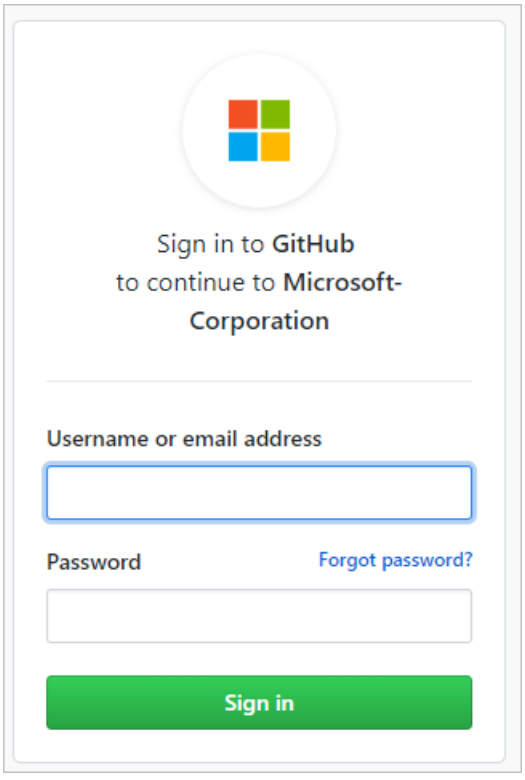
If you have a GitHub account, follow these steps to sign up for Azure Boards.

> **IMPORTANT**
> If your GitHub email address is associated with an Azure AD-backed organization in Azure DevOps, you can't sign in with your GitHub account, rather you must sign in with your Azure AD account.

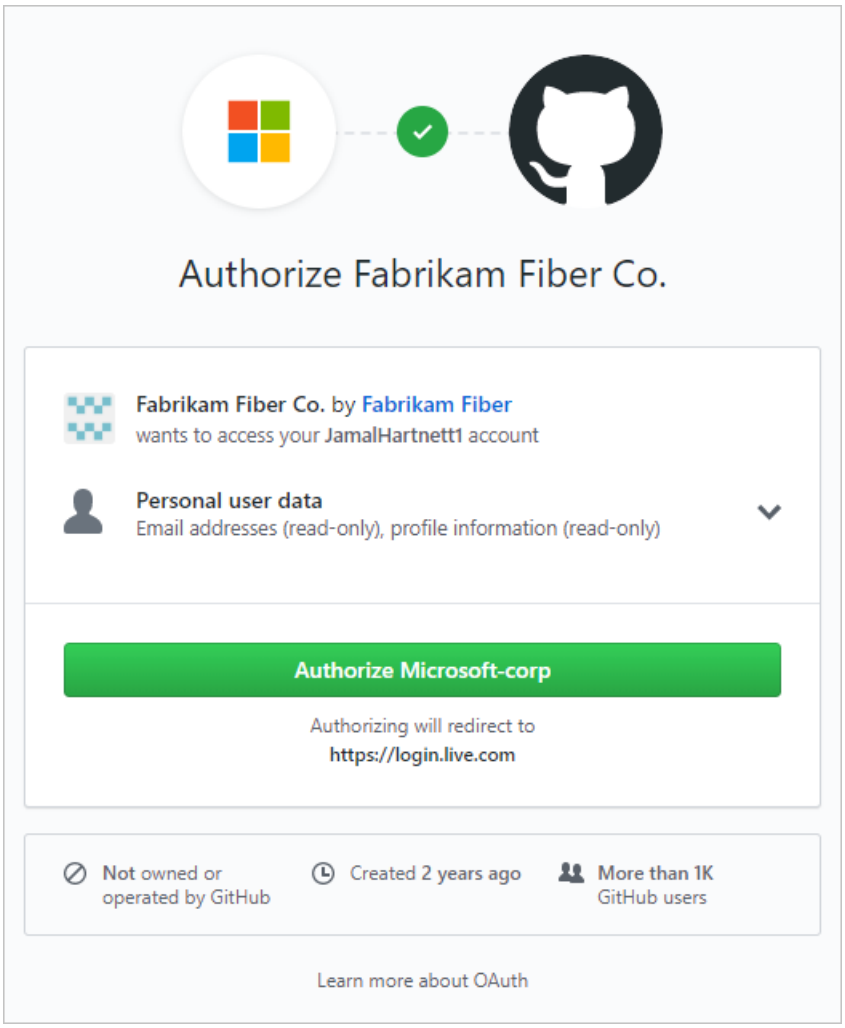
1. From the Azure Boards sign-up page, choose **Start Boards free with GitHub**.



2. Enter your GitHub account credentials, and then select **Sign in**.

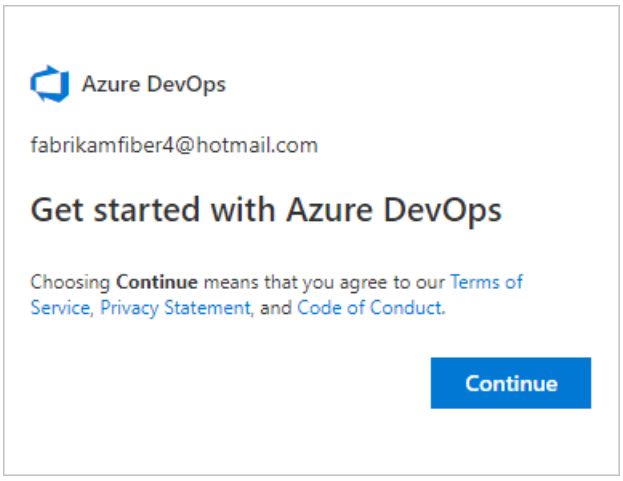


3. Select **Authorize Microsoft corporation**.



For more information about GitHub authentication, see FAQs.

4. Choose **Continue**.



An organization is created based on your GitHub account.

Use the following URL to sign in to your organization at any time:

```
https://dev.azure.com/{yourorganization}
```

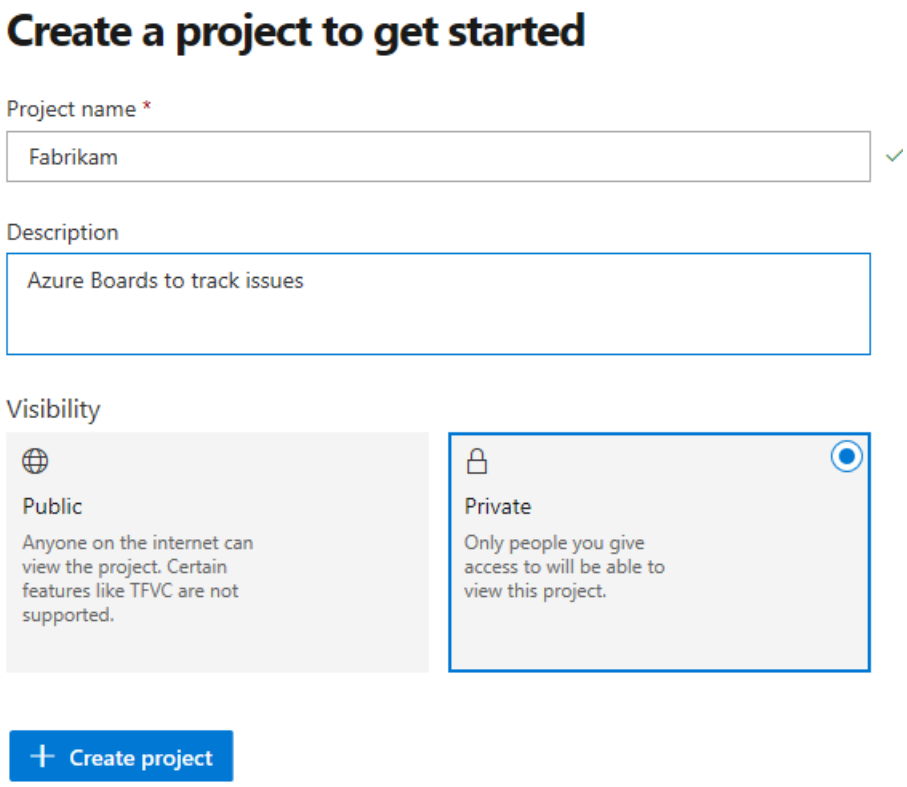
You can change the organization name as indicated in Change organization or project settings later in this article.

5. To complete your sign-up process, go to create a project.

Create a project

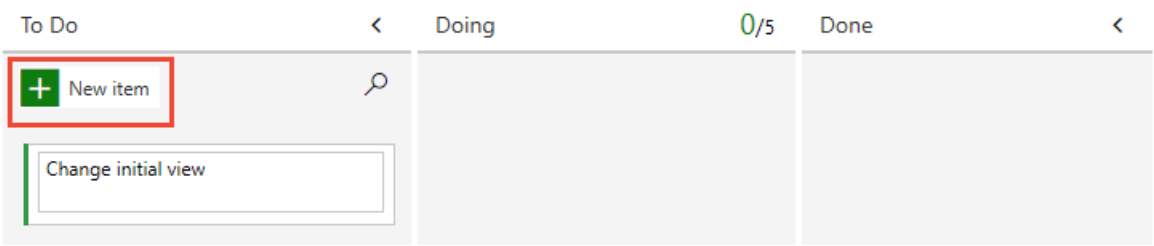
If you signed up for Azure DevOps with an existing Microsoft account or GitHub identity, you're automatically prompted to create a project. You can create either a public or private project. To learn more about public projects, see What is a public project?.

1. Enter a name for your project, select the visibility, and optionally provide a description. Then choose **Create project**.



- **Project name**: Can't contain special characters (such as / : \ ~ & % ; @ ' " ? < > | # $ * } { , + = [ ]), can't begin with an underscore, can't begin or end with a period, and must be 64 characters or less.
- **Visibility**: Choose **Public** if you want to create an open-source project. Otherwise, choose **Private**, so only people who you give access to can view your project.

2. Your Kanban board automatically appears. You're now set to start tracking issues, tasks, and features, or invite other users to collaborate with your project.



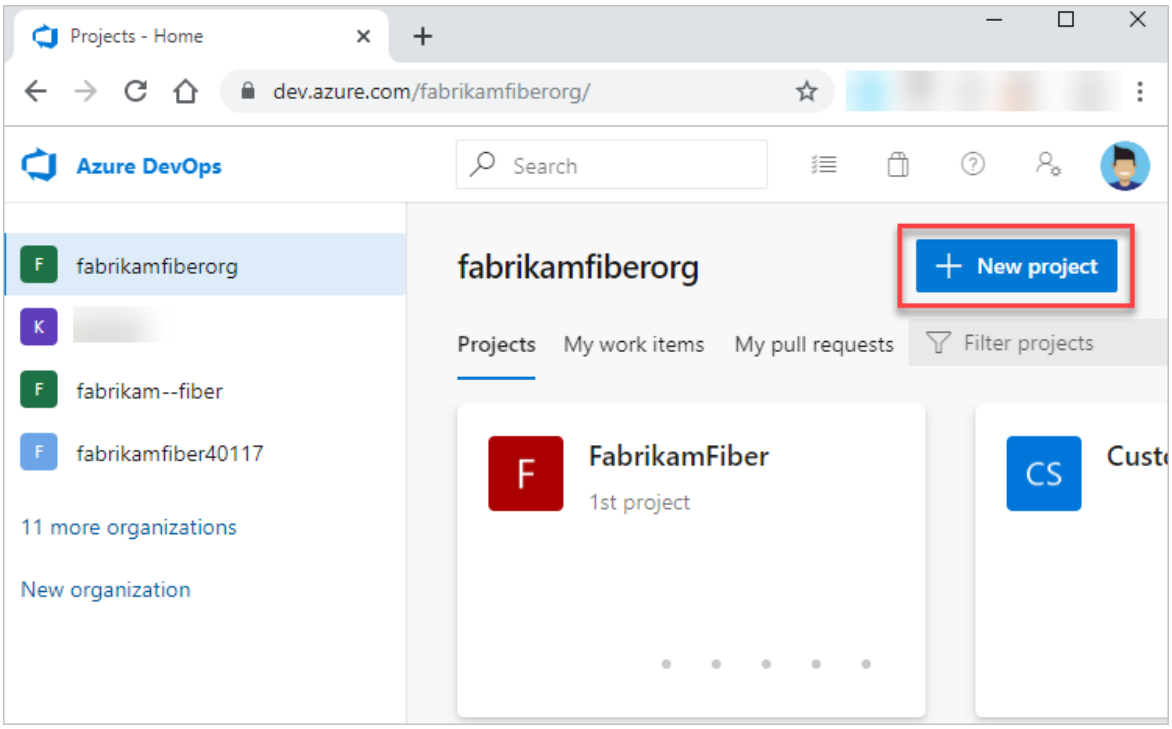
NOTE

Your first project was created using the Basic process which uses Epics, Issues, and Tasks to track work. If you want a project that uses the Agile, Scrum, or CMMI process, then you can add another project and specify the process through advanced setting options as described in the next section. See Choose a process for a comparison of processes.

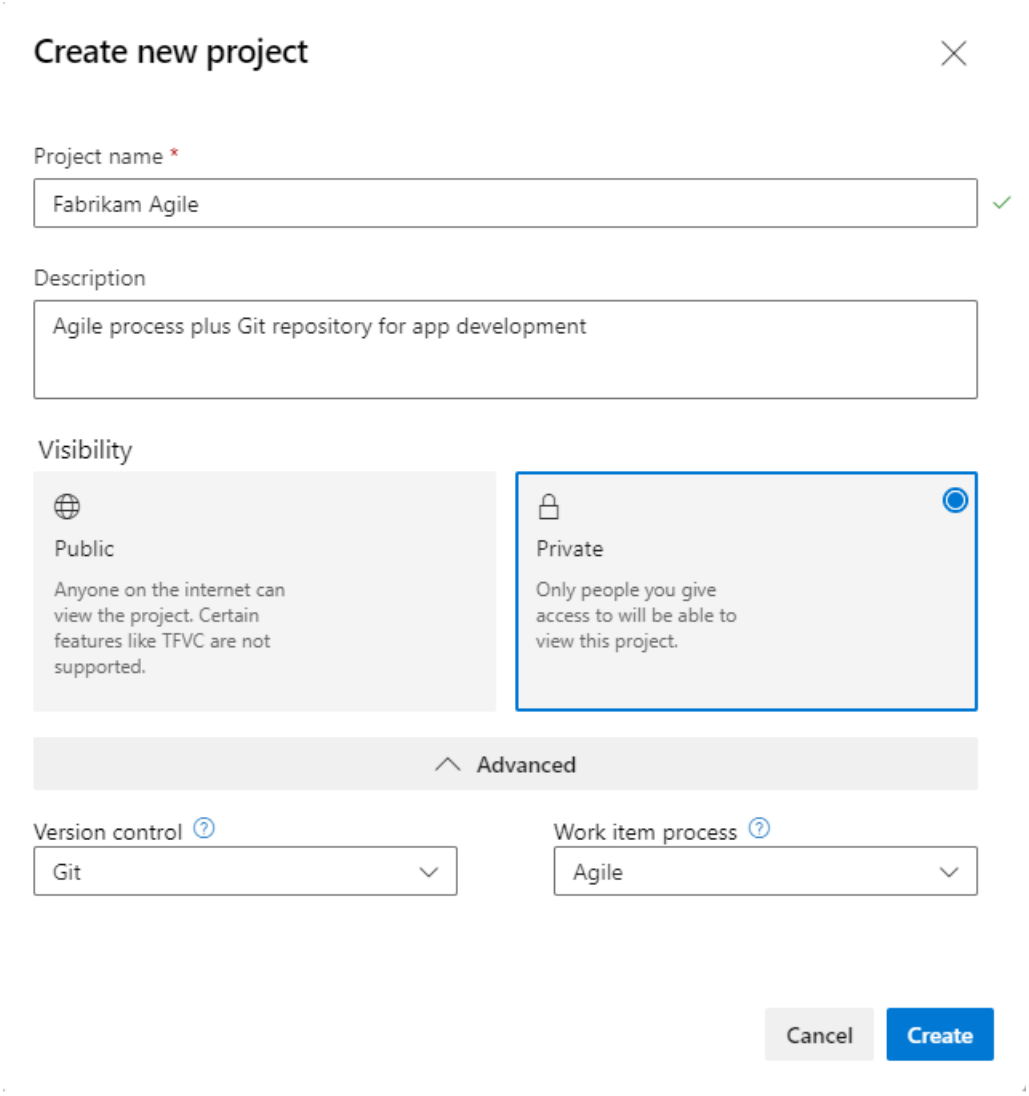
Create a project with Advanced options

Your first project is automatically created using the Basic process and a Git repository. If you want to use the Agile, Scrum, or CMMI process and a different repository, you can create another project and choose the process by expanding the **Advanced** settings. You can then delete the project with the process you don't want to use.

1. Select **Azure DevOps** to open the **Projects** page, and then select **New project**.



2. Fill out the form, expand **Advanced** to choose the options available for Version control and Work item process.

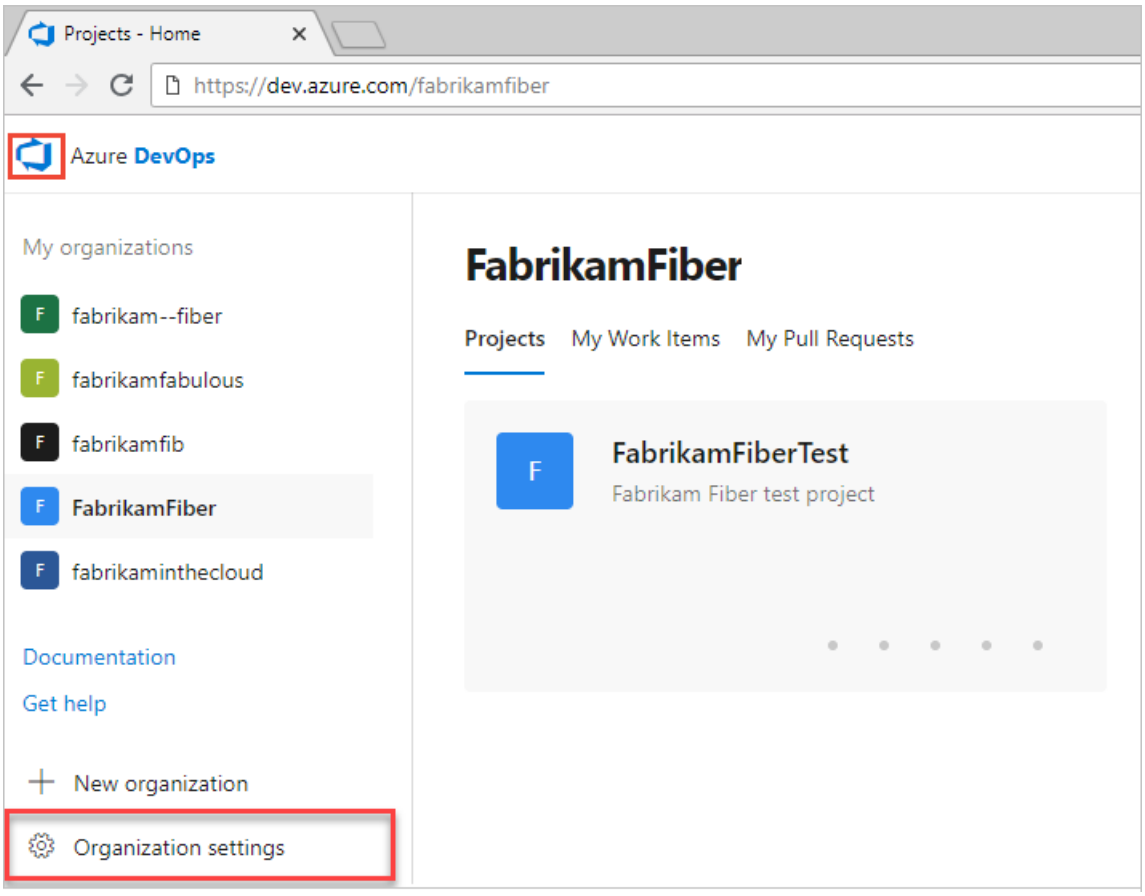


3. Choose **Create** to complete the action.

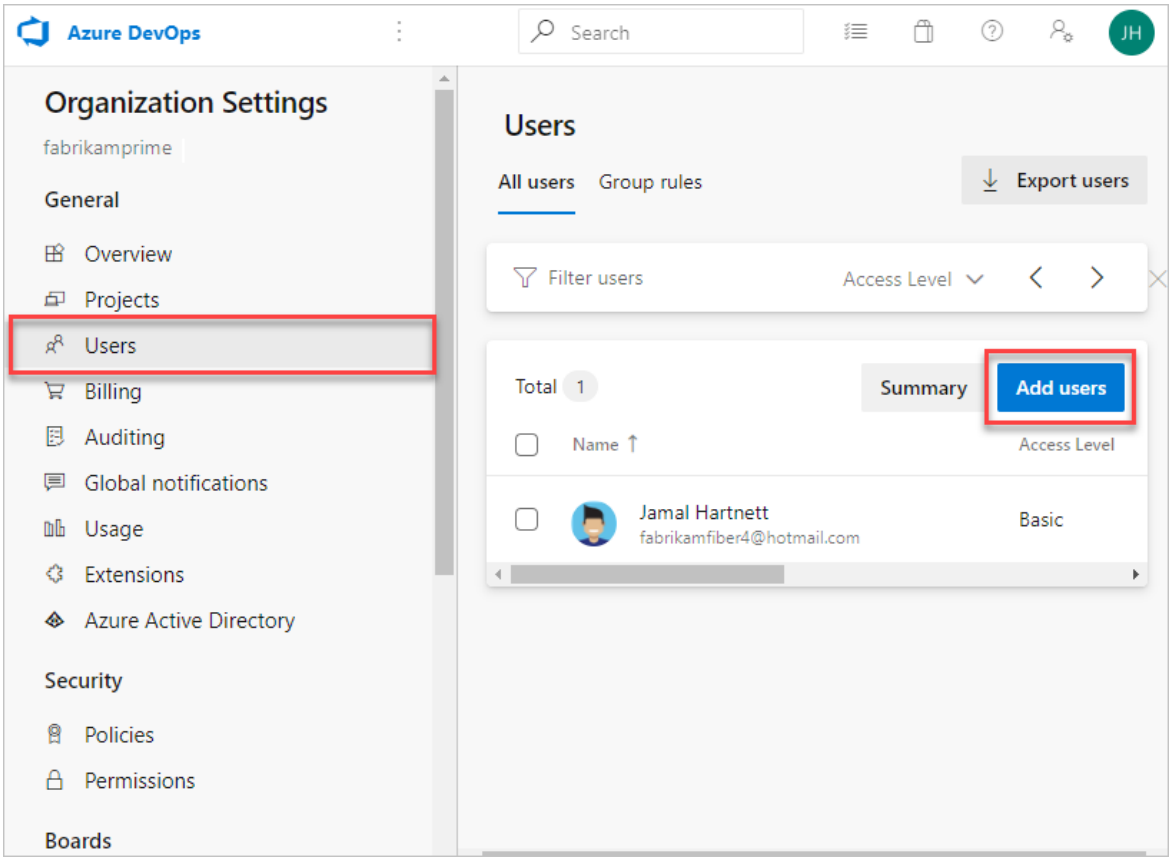
Invite team members to your project

You can add and invite others to work on your project by adding their email address to your organization and project.

1. From your project web portal, choose the Azure DevOps icon, and then select **Organization settings**.



2. Select **Users** > **Add new users**.



3. Complete the form by entering or selecting the following information:

- **Users:** Enter the email addresses (Microsoft accounts) or GitHub IDs for the users. You can add several email addresses by separating them with a semicolon (;). An email address appears in red when it's accepted.
- **Access level:** Assign one of the following access levels:

    - **Basic**: Assign to users who must have access to all Azure Boards features. You can grant up to five users **Basic** access for free.
    - **Stakeholder**: Assign to users who will have limited access to features to view, add, and modify work items. You can assign an unlimited amount of users Stakeholder access for free.
- **Add to project**: Select the project you named in the preceding procedure.
- **Azure DevOps Groups**: Select one of the following security groups, which will determine the permissions the users have to perform select tasks (To learn more, see Default permissions and access for Azure Boards.):
    - **Project Readers**: Assign to users who only require read-only access.
    - **Project Contributors**: Assign to users who will contribute fully to the project.
    - **Project Administrators**: Assign to users who will configure project resources.

> **NOTE**
> Add email addresses for personal Microsoft accounts and IDs for GitHub accounts unless you plan to use Azure Active Directory (Azure AD) to authenticate users and control organization access. If a user doesn't have a Microsoft or GitHub account, ask the user to sign up for a Microsoft account or a GitHub account.

4. When you're done, select **Add** to complete your invitation.

For more information, see Add users or groups to a team or project and Add organization users for Azure DevOps Services.

Change your organization or project settings

You can rename and delete your organization, or change the organization location. To learn more, see the following articles:

- Manage organizations
- Rename an organization
- Change the location of your organization

You can rename your project or change its visibility. To learn more about managing projects, see the following articles:

- Manage projects
- Rename a project
- Delete a project
- Change the project visibility, public or private

Try this next

Track issues and tasks

Related articles

- Manage projects
- Manage organizations
- About access levels
- Define organizations and projects

Plan and track work in Azure Boards

10/4/2022 • 21 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You track your work by creating work items. This article walks you through creating issues and tasks using a Kanban board. You can learn the Basic process or the Agile process for creating these items.

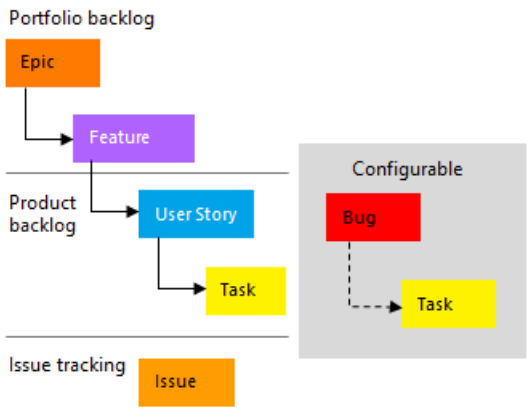
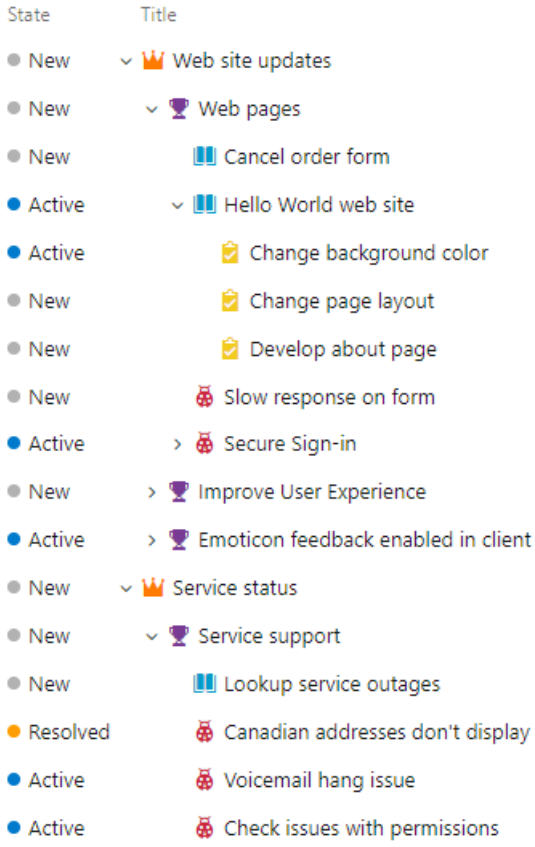
Choose one of the following four system processes—**Agile**, **Basic**, **Scrum**, or **Capability Maturity Model Integration (CMMI)**—for guidance depending on what process was selected for your project. For an overview of each of these processes, see Choose a process.

> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

- Agile process
- Basic process
- Scrum process
- CMMI process

The Agile process provides several work item types—for example, user stories, tasks, bugs, features, and epics among others—to plan and track work. We recommend you start by adding user stories. If you need to group them into a hierarchy, you can define features. To track other details of work, you can add tasks to a user story.

WORK ITEM TYPES	BACKLOG HIERARCHY

WORK ITEM TYPES	BACKLOG HIERARCHY
	

Within each work item form, you can describe the work to be done, assign work to project contributors, track status, and collaborate with others through the Discussion section.

Here we show how to add user stories and child tasks from the web portal and add details to those work items.

Prerequisites

- After you connect to a project, you can add work items. If you don't have a project yet, create one in Azure DevOps.
- To add work items to a board, and use all other board features, you must be granted **Basic** access and have been added as a member of the Contributors or Project Administrators group.
- If you have been granted **Stakeholder** access for a private project and have been added as a member of the Contributors or Project Administrators group, you can view boards, open and modify work items, and add child tasks to a checklist. However, you can't reorder or reparent a backlog item using drag-and-drop, nor update a field on a card.
- If you have been granted **Stakeholder** access for a public project, and have been added as a member of the Contributors or Project Administrators group, you have full access to all Boards features.

- After you connect to a project, you can add work items. If you don't have a project yet, create one in Azure DevOps.
- To add work items to a board, and use all other board features, you must be granted **Basic** access and have been added as a member of the Contributors or Project Administrators group.

- If you have been granted **Stakeholder** access and have been added as a member of the Contributors or Project Administrators group, you can view boards, open and modify work items, and add child tasks to a checklist. However, you can't reorder or reparent a backlog item using drag-and-drop, nor update a field on a card.

> **NOTE**
> The ability for Stakeholders to drag-and-drop cards to different columns requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

- After you connect to a project, you can add work items. If you don't have a project yet, create one in Azure DevOps.
- To add work items to a board, and use all other board features, you must be granted **Basic** access and have been added as a member of the Contributors or Project Administrators group.
- If you have been granted **Stakeholder** access for a private project and have been added as a member of the Contributors or Project Administrators group, you can view boards, open and modify work items, and add child tasks to a checklist. However, you can't update the status of a backlog item or reorder or reparent a backlog item using drag-and-drop, nor update a field on a card.
- If you have been granted **Stakeholder** access for a public project, and have been added as a member of the Contributors or Project Administrators group, you have full access to all Boards features.

For details, see Default permissions and access for Azure Boards

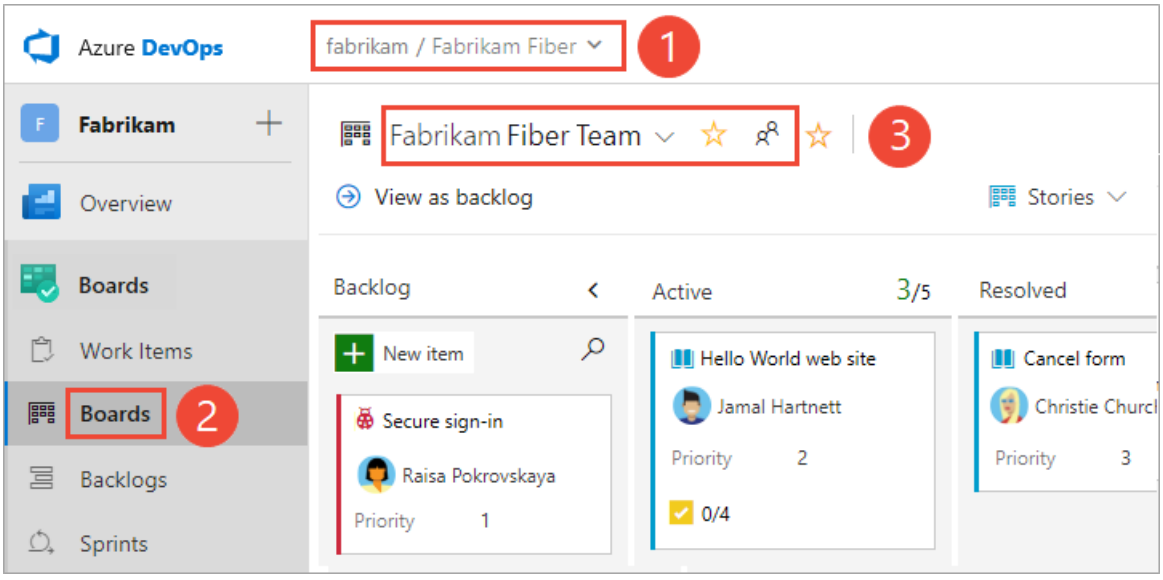
> **NOTE**
> The images shown in this article correspond to the latest version of Azure Boards. While they may differ from those shown in earlier, on-premises versions of Azure DevOps, they are similar in the functions described unless otherwise noted.

Open your Kanban board

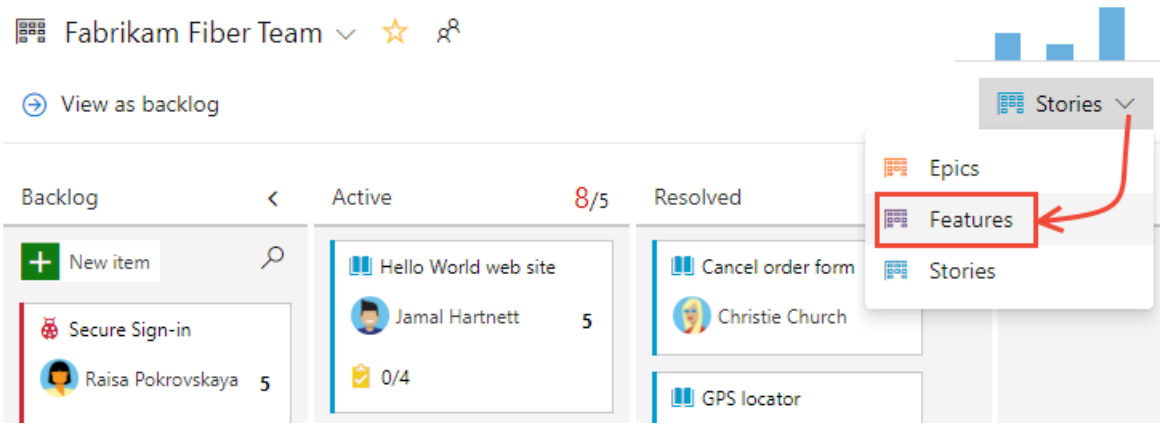
A Kanban board is provisioned with the addition of each project and each team. You can only create or add Kanban boards to a project by adding another team. To learn more, see About teams and Agile tools.

- Agile process
- Basic process
- Scrum process
- CMMI process

The User Stories Kanban board is the best tool for quickly adding user stories and child tasks. To open, choose **Boards > Boards**.



The Features Kanban board is the best tool for quickly adding features and user stories that are children of those features. To open the Features board from the Stories board, choose **Features** from the board selector.

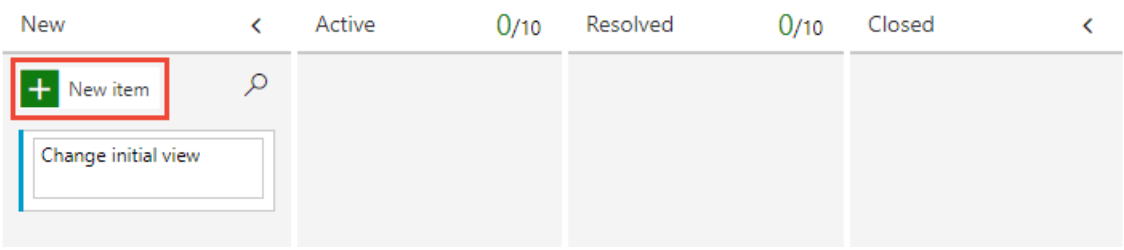


Add work items to your board

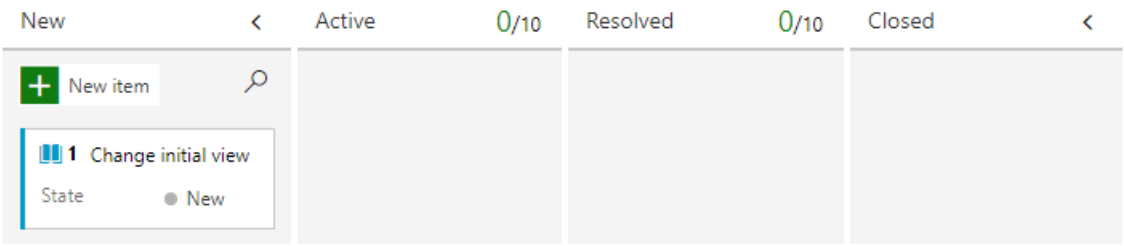
Work items you add to your board are automatically assigned the default **Area Path** and **Iteration Path** assigned to the team. To learn more, see Configure team settings.

- Agile process
- Basic process
- Scrum process
- CMMI process

1. From the Stories board, choose **New item** and start adding those stories you want to track.



2. Enter return and the system assigns a work item ID to the user story.



3. To track the work you want to manage, add as many user stories that you need.

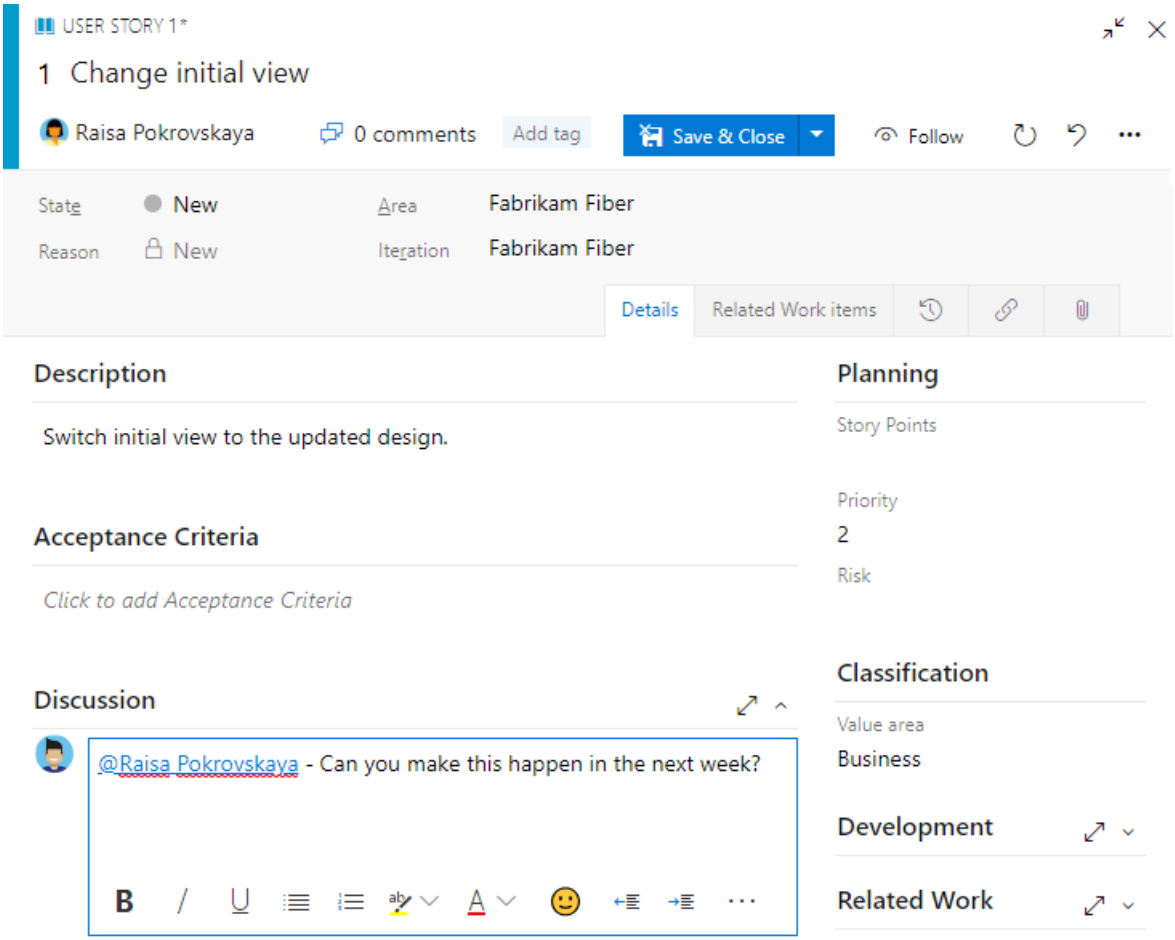
Add details to a board item

Choose the issue or user story title to open it. Change one or more field values, add a description, or make a note in the **Discussion** section. You can also choose the **Attachments** tab and drag-and-drop a file to share the file with others.

- Agile process
- Basic process

- Scrum process
- CMMI process

For example, here we assign the story to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa.



Choose **Save & Close** when done.

Field descriptions

Field	Usage
Title	Enter a description of 255 characters or less. You can always modify the title later.
Assigned To	Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu lists only team members or contributors to the project.

> **NOTE**
> You can only assign work to a single user. If you need to assign work to more than one user, add a work item for each user and distinguish the work to be done by title and description. The Assigned To field only accepts user accounts that have been added to a project or team.

State	

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current status.

Reason

Use the default first. Update it when you change state as need. Each State is associated with a default reason.

Area (Path)

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting. To change the dropdown list of areas, see Define area paths and assign to a team.

Iteration (Path)

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later during a planning meeting. To change the drop-down list of iterations, see Define iteration paths and configure team iterations.

Description

Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

Acceptance Criteria

Provide the criteria to be met before the work item can be closed. Define what "Done" means by describing the criteria for the team to use to verify whether the backlog item or bug fix is fully implemented. Before work begins, describe the criteria for customer acceptance as clearly as possible. Have conversations between the team and customers to determine the acceptance criteria. These criteria help ensure a common understanding within the team to meet customers' expectations. Also, this information provides the basis for acceptance testing.

Priority

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **1**: Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible.
- **2**: Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately.
- **3**: Resolution of the work item is optional based on resources, time, and risk.
- **4**: Resolution of the work item is not required.

Value Area

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **Architectural**: Technical services to implement business features that deliver solution .
- **Business**: Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default).

Effort, Story Points, Size

Provide a relative estimate of the amount of work required to complete an issue. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include

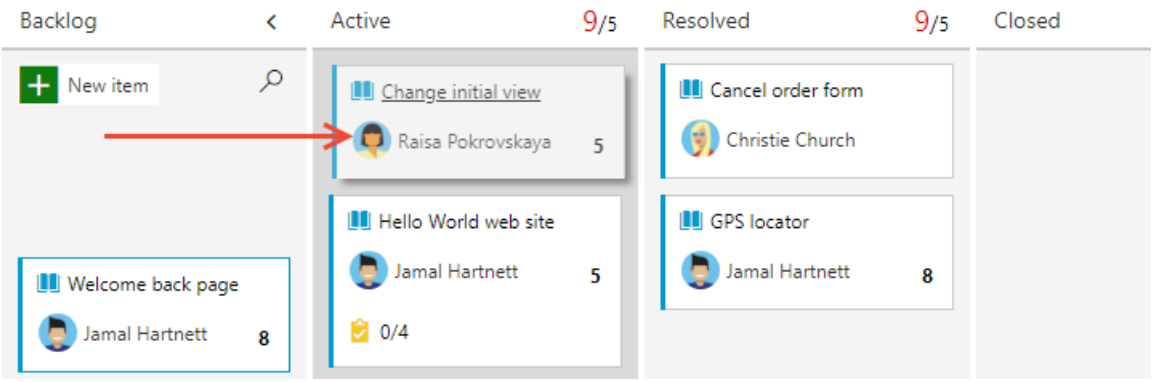
powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, etc.). Use any numeric unit of measurement your team prefers.
The estimates you set are used to calculate team velocity and forecast sprints.

Update work status

The State field tracks the status of a work item. With the Kanban board, you can quickly update the status of backlog items by dragging and dropping them to a different column. This feature requires that you have Basic access or higher.

- Agile process
- Basic process
- Scrum process
- CMMI process

As work starts, drag the user story card from the **Backlog** column to the **Active** column. Once work is ready for review, move to the **Resolved** column. After it's reviewed and accepted, move to the **Closed** column.



You can add or rename columns as needed, see Customize your board.

> **TIP**
> You can add or rename columns as needed, see Customize your board.

Add tasks

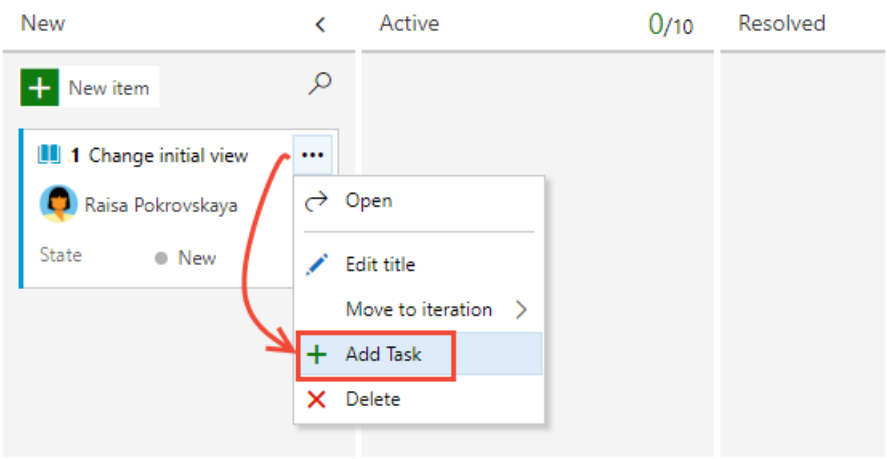
Task checklists provide a quick and easy way to track elements of work that are important to support completing a backlog item. Also, you can assign individual tasks to different team members.

> **TIP**
> Tasks that you create from the Kanban board are automatically assigned the **Area Path** and **Iteration Path** of their parent work item.

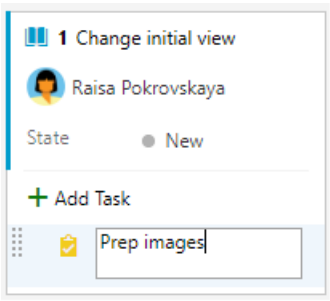
Tasks that you create from the Kanban board show up on your sprint taskboard. Also, tasks that you create from the sprint backlog or taskboard show up within tasks checklists on the Kanban board.

- Agile process
- Basic process
- Scrum process
- CMMI process

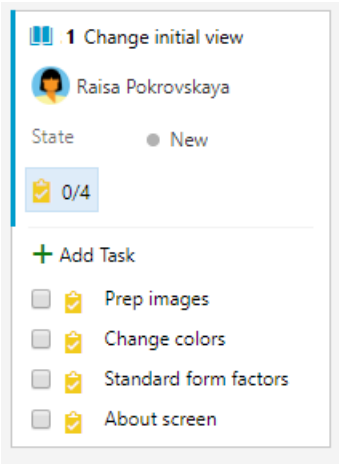
1. To start adding tasks, choose the ··· actions icon for the story and select the + **Add Task** option.



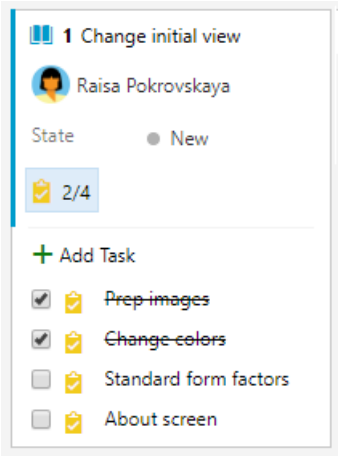
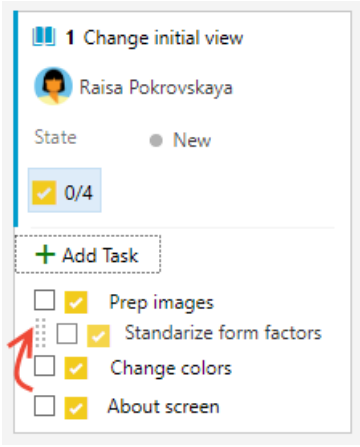
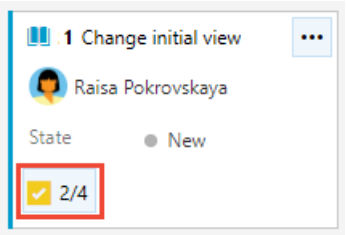
Enter a title for the task and type Enter when done.



2. If you have many tasks to add, keep typing your task titles and type Enter.



3. You can mark a task as done, expand or collapse the task checklist, or reorder and reparent tasks.

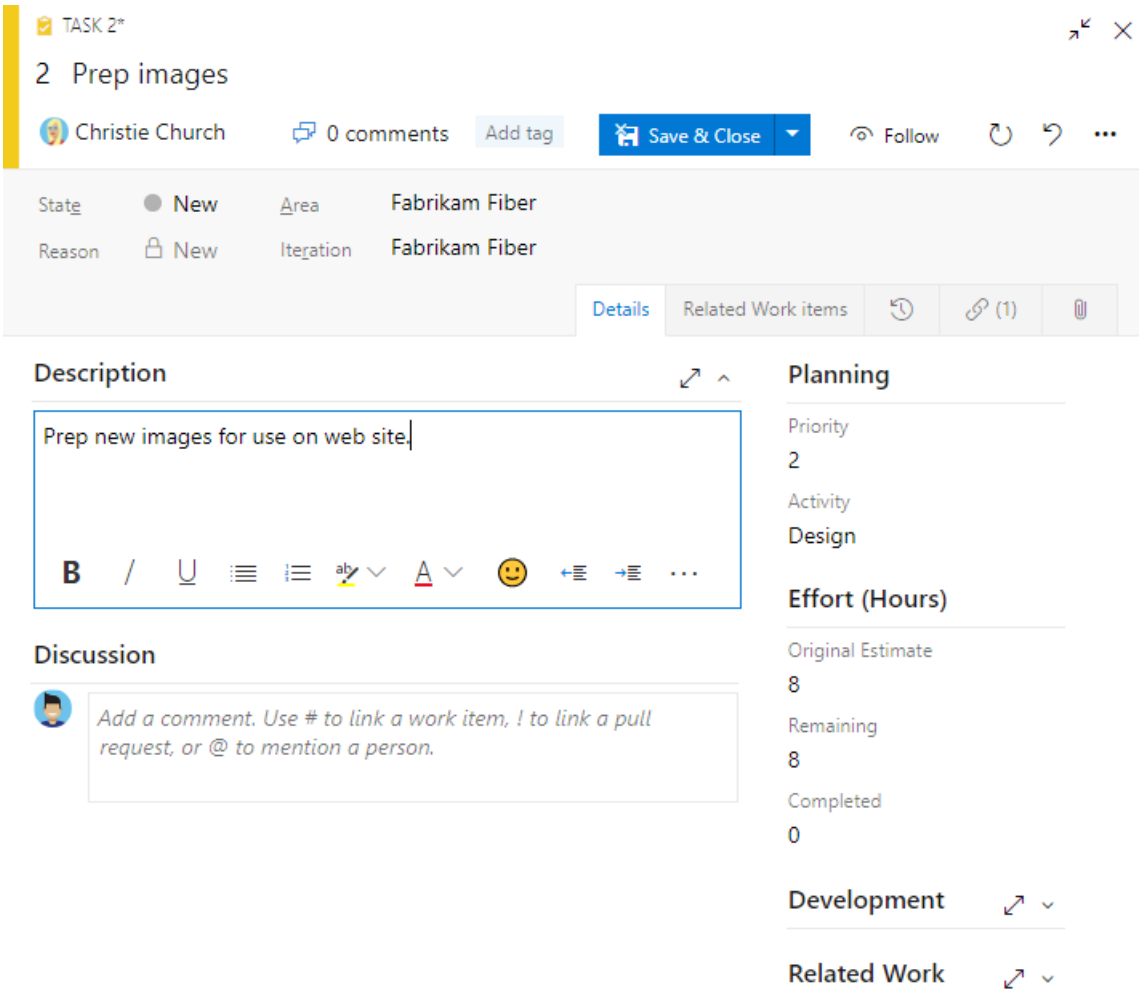
MARK A TASK AS DONE	REORDER AND REPARENT TASKS	EXPAND OR COLLAPSE THE CHECKLIST
To mark a task as complete, check the task checkbox. The task State changes to **Done**. 	To reorder a task, drag it within the checklist. To reparent a the task, drag it to another issue on the board. 	To expand or collapse a task checklist, simply choose the task annotation. 

Add details to a task

If you have details you want to add about a task, choose the title, to open it. Change one or more field values, add a description, or make a note in the **Discussion** section. Choose **Save & Close** when done.

- Agile process
- Basic process
- Scrum process
- CMMI process

Here we assign the task to Christie Church.



Field descriptions

In addition to the fields you can define for a backlog item—user story, issue, product backlog item, or requirement—you can specify the following fields for a task to support capacity and time tracking.

> **NOTE**
> There are no inherent time units associated with this field even though the taskboard always shows "h" for hours in relationship to Remaining Work. You can specify work in any unit of measurement your team chooses.

Field	Usage
Activity	The type of activity that's required to do a task.To learn more about how this field is used, see Capacity planning. Allowed values are: • Deployment • Design • Development • Documentation • Requirements • Testing
Discipline (CMMI process)	

The type of activity that's required to do a task.To learn more about how this field is used, see Capacity planning. Allowed values are:

- **Analysis**
- **Development**
- **Test**
- **User Education**
- **User Experience**

Original Estimate

The amount of estimated work required to complete a task. Typically, this field doesn't change after it is assigned.

Remaining Work

The amount of work that remains to finish a task. You can specify work in hours or in days. As work progresses, update this field. It's used to calculate capacity charts and the sprint burndown chart.
If you divide a task into subtasks, specify Remaining Work for the subtasks only.

Completed Work

The amount of work spent implementing a task. Enter a value for this field when you complete the task.

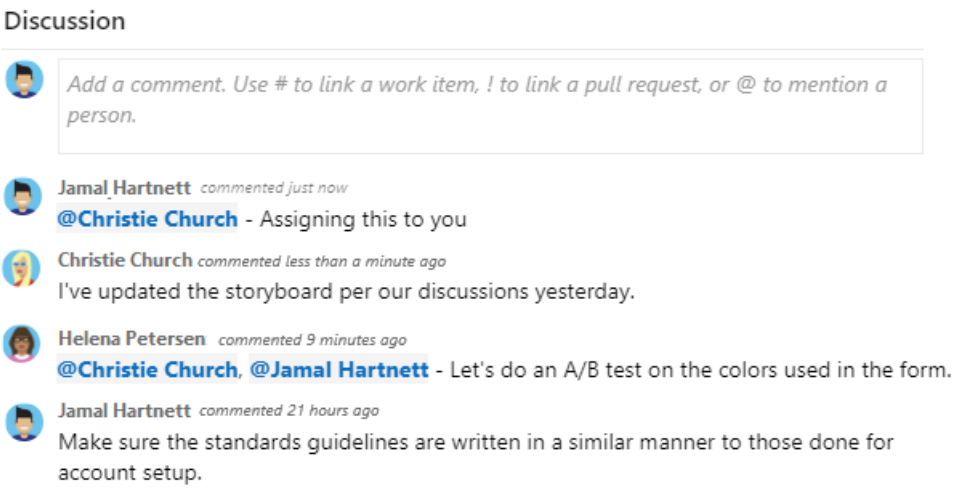
Task Type (CMMI only)

Select the kind of task to implement from the allowed values:

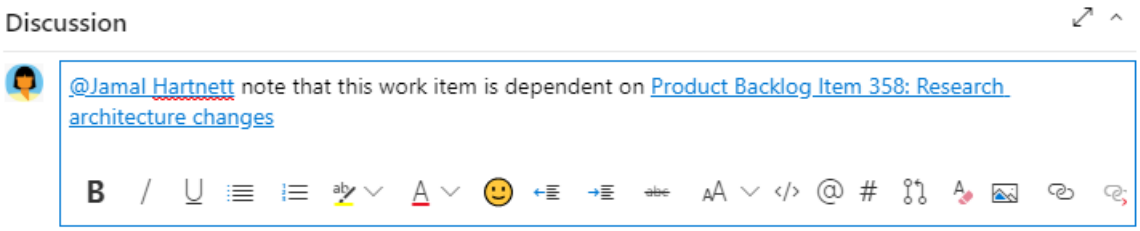
- **Corrective Action**
- **Mitigation Action**
- **Planned**

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



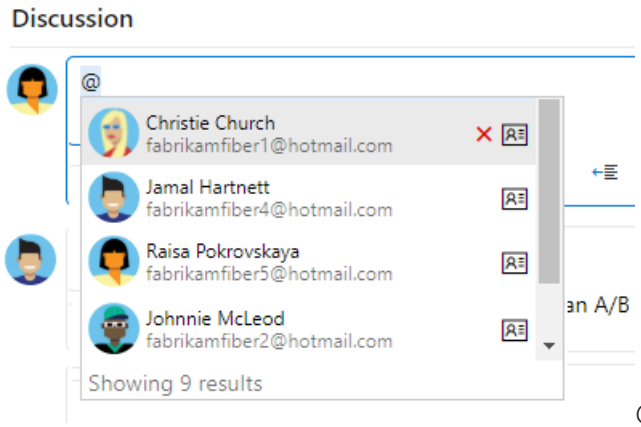
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the **History** field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons —@ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.

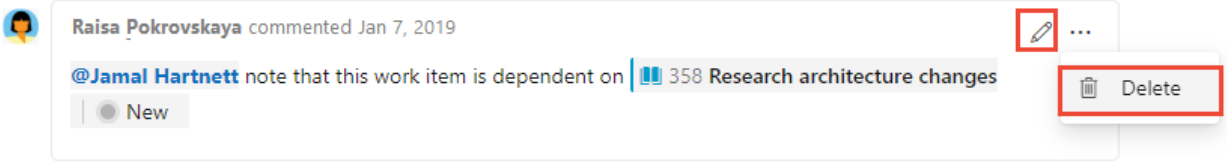


@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon and then choose **Delete**.



> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently

referenced will appear from which you can select.

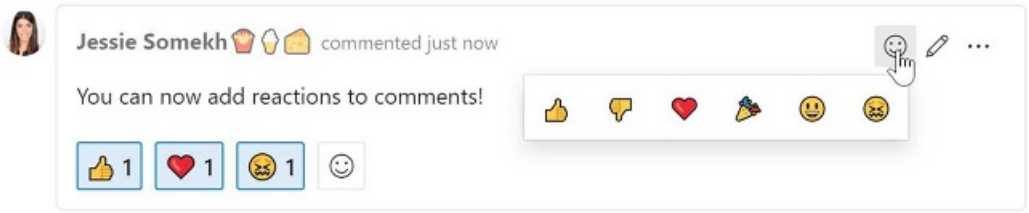
To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

You can't edit or delete comments once you've entered them.

> **IMPORTANT**
> For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



Next step

Customize your board

Related articles

- Azure Boards FAQs
- Index to field descriptions
- Add tags to issues or tasks

Customize your Kanban boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This article shows how to customize a Kanban board. You have one Kanban board for each active product or portfolio backlog.

You can configure your Kanban board in several ways to support specific tracking needs. For example:

- Update fields directly from the card.
- Highlight cards based on field assignments.
- Add columns to track other workflow states.
- Add swimlanes to speed up work or differentiate work assigned to different service classes.

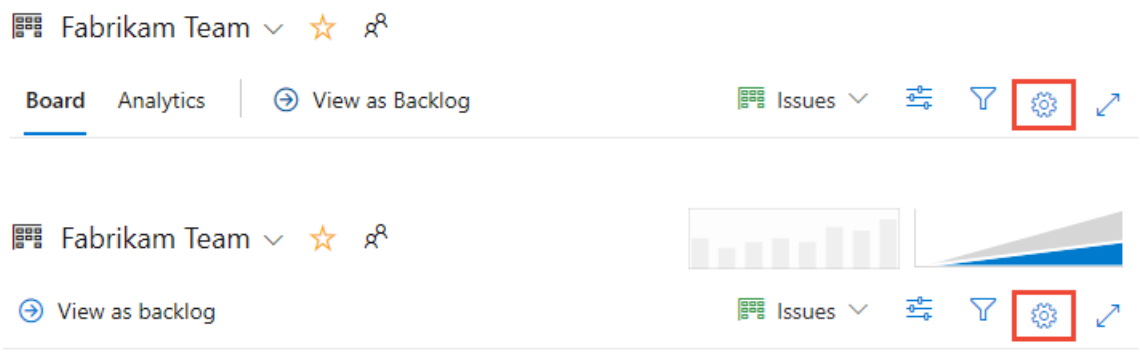
Open board settings

- Basic process
- Agile process

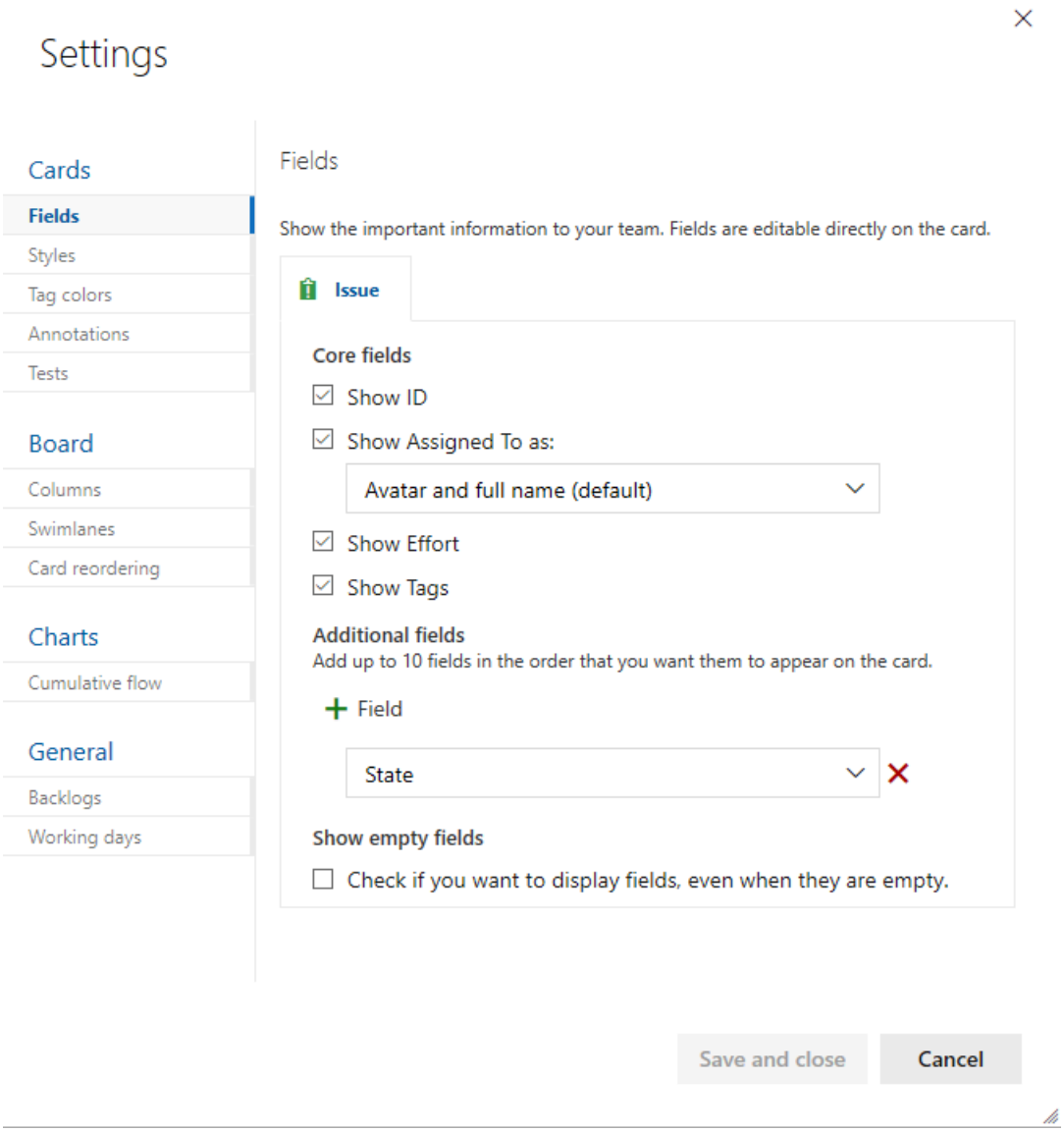
Each team can customize their Issues and Epics boards and sprint Taskboards.

> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

1. Open your Kanban board.
2. Choose the gear icon to configure the board setting settings.



The Settings dialog opens.



3. Choose one of the following tabs based on what you want to customize:

   Common customizations:

   - **Fields**: Set what fields appear on cards.
   - **Columns**: Add, rename, and configure columns for the board.
   - **Swimlanes**: Add swimlanes to a board to easily track high priority levels, distinguish different service classes, or track blocked items.
   - **Backlogs**: Enable or disable issues or epics from being tracked on backlogs and boards.

   Less common customizations:

   - **Styles**: Specify the card color to display based on field criteria you specify.
   - **Tag colors**: Specify the tag color to display based on tag criteria you specify.
   - **Annotations**: Enable or disable task or test annotations.
   - **Tests**: Configure how you want tests to appear and behave on the cards.
   - **Card reordering**: Choose to enable/disable changing of the backlog priority when dragging and dropping cards on the board.
   - **Cumulative flow**: Set how you want the cumulative flow diagram to display.
   - **Working days**: Set the active weekdays to use when tracking capacity and burndown.

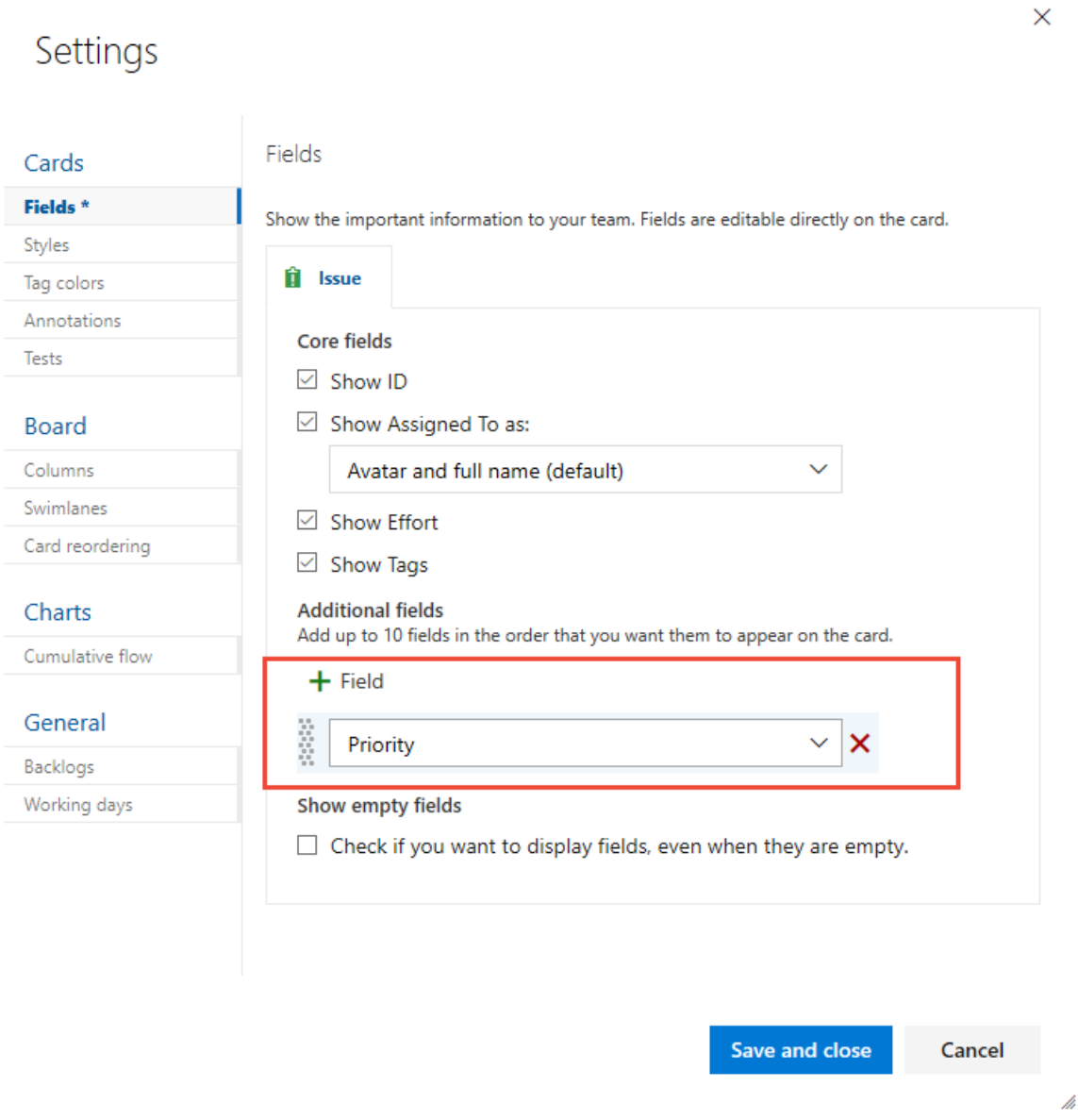
Customize fields on a card

To quickly assign fields from the card without having to open the work item, add them to display on the card.

- Basic process
- Agile process

1. To configure the fields displayed on cards, choose the **Fields** tab.
2. Check or uncheck those fields you want to display or not display on the board.
3. To add a field, choose + **Field** to add a field.

   For example, here we uncheck **Show ID** and add the **Priority** field.



4. To remove a field, choose the ✕ delete icon next to the field.
5. Choose **Save and close** when done. To learn more, see Customize cards.

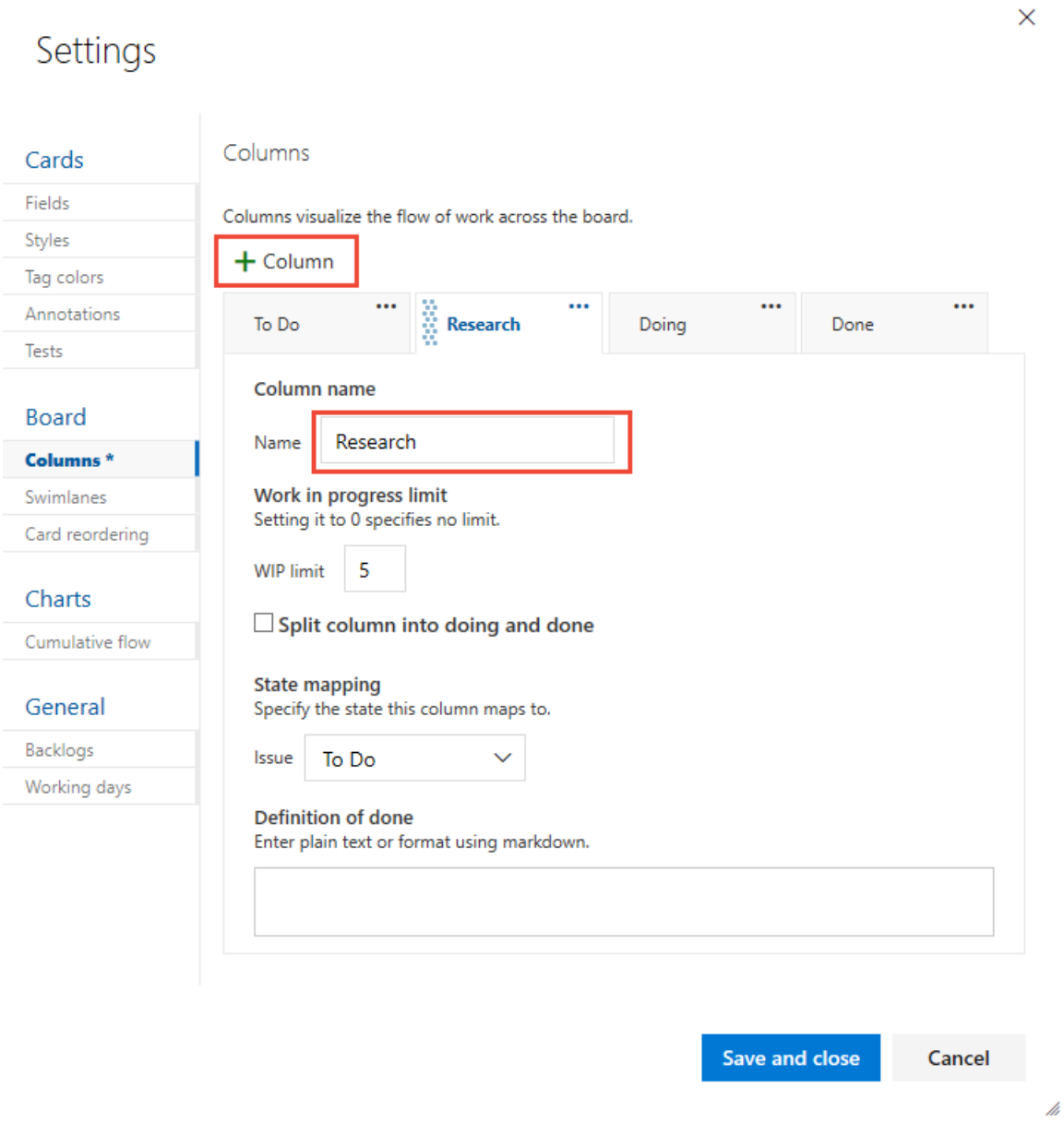
Customize your columns

You can add columns or change column names.

- Basic process
- Agile process

1. To add a column or change column settings, choose **Columns** tab.

   Here we choose + **Column**, and then specify the name as **Research**.



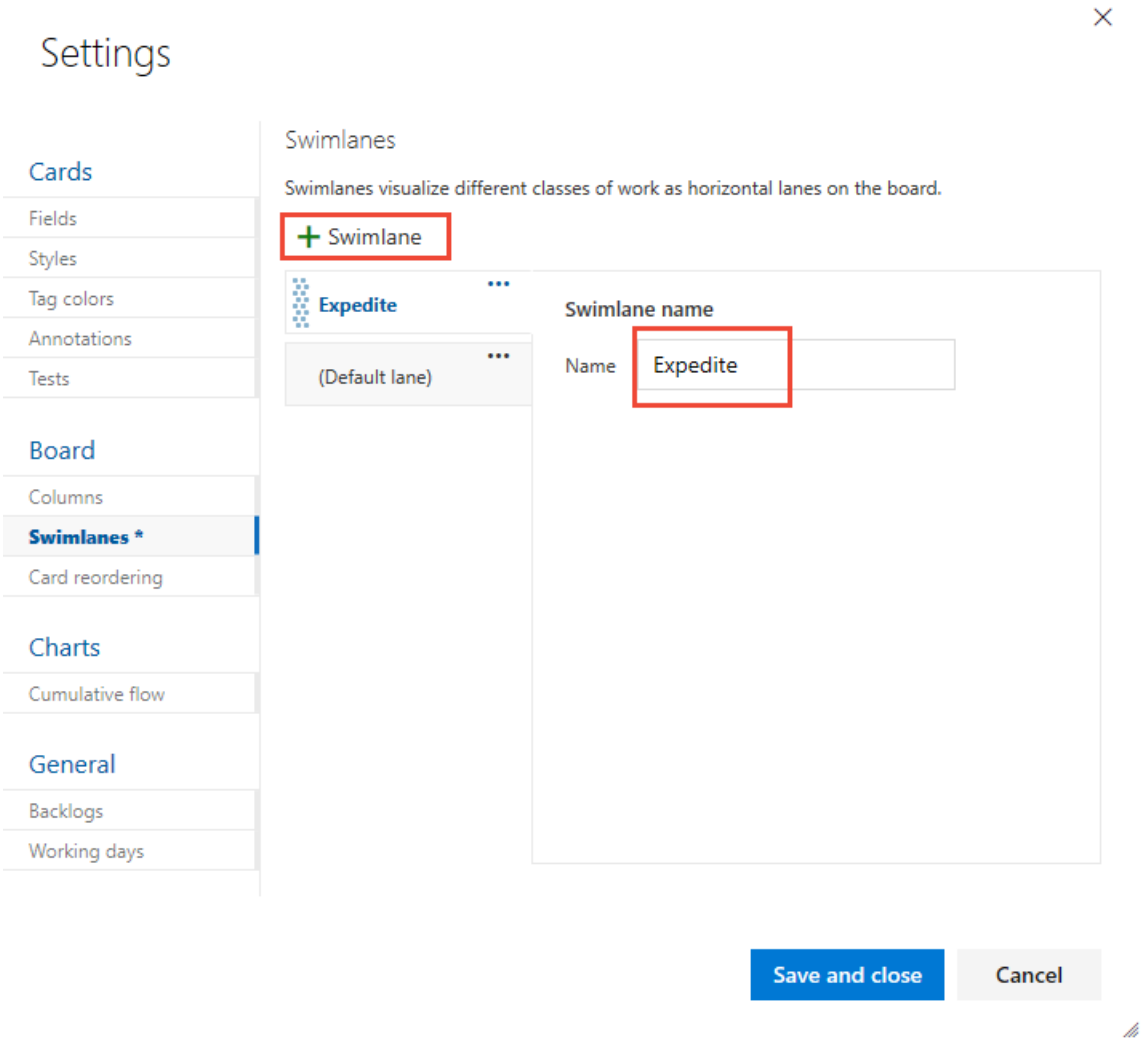
2. Choose **Save and close** when done. To learn more about column settings, see Add columns to your Kanban board.

Customize your swimlanes

Swimlanes appear as rows within the Kanban board and provide yet another way to categorize work you want to track.

1. To add a swimlane, choose **Swimlanes** tab and choose + **Swimlane**, and then specify the name.

   Here we label the swimlane **Expedite**.



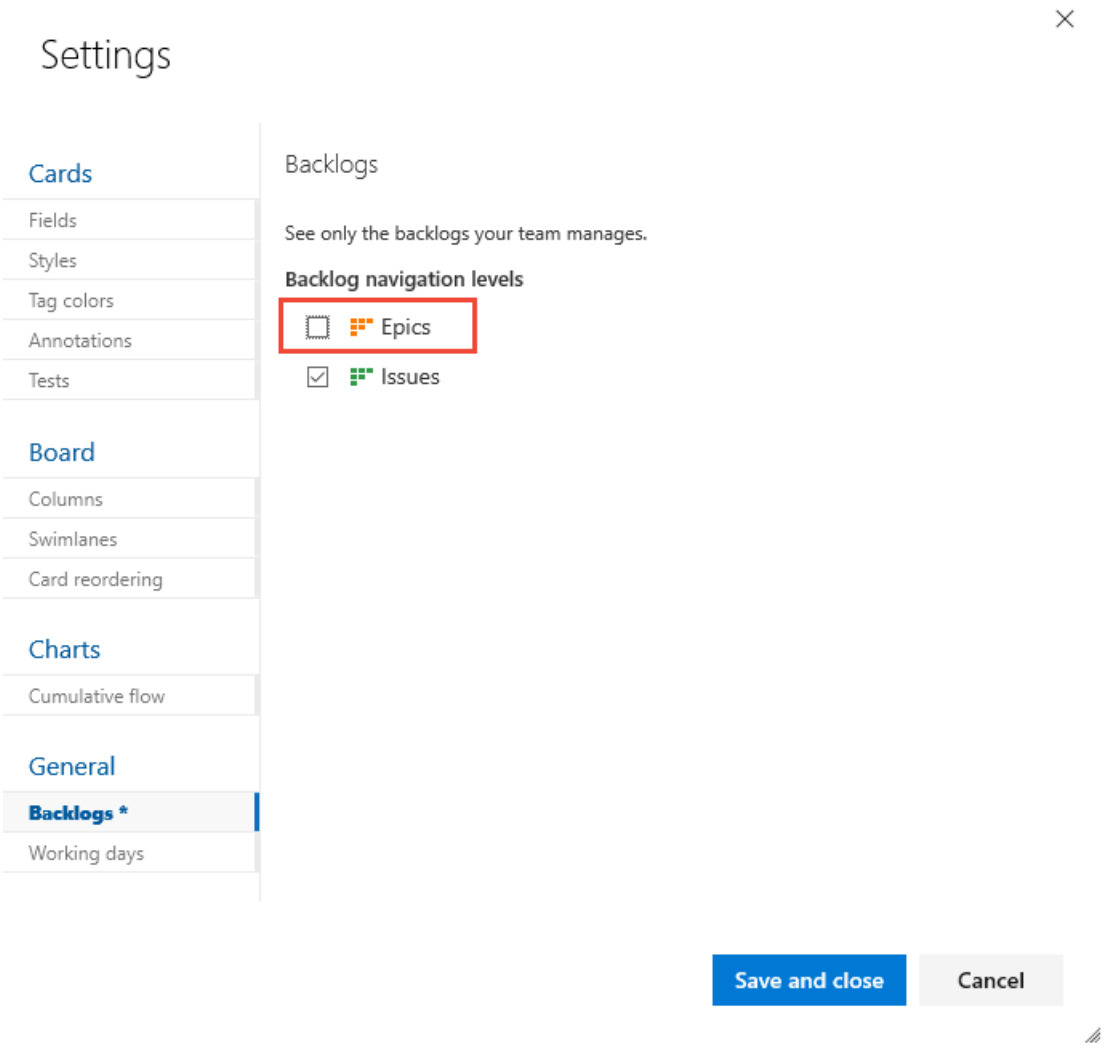
2. Choose **Save and close** when done. To learn more about working with swimlanes, see Expedite work with swimlanes.

Add or remove backlog levels

- Basic process
- Agile process

If you decide you don't want to use Epics to track work, you can turn it off and it won't show up as a board or backlog. By default, it's enabled for new projects.

1. Choose **Backlogs** tab and uncheck the work item type you no longer want to track on backlogs and boards.



2. Choose **Save and close** when done.

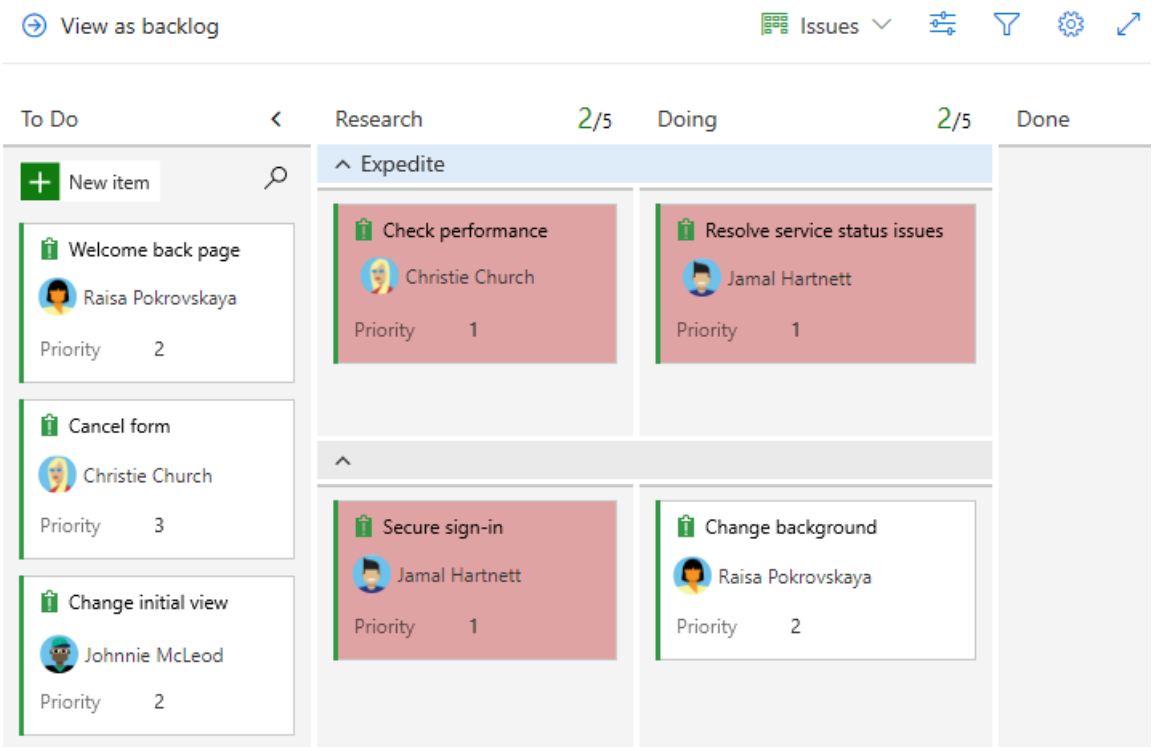
> **NOTE**
> Contributors will still be able to create Epics from other views, they just won't be able to view Epics within a backlog or board. To completely disable the Epic work item type, see Add and manage work item types, Enable or disable a WIT.

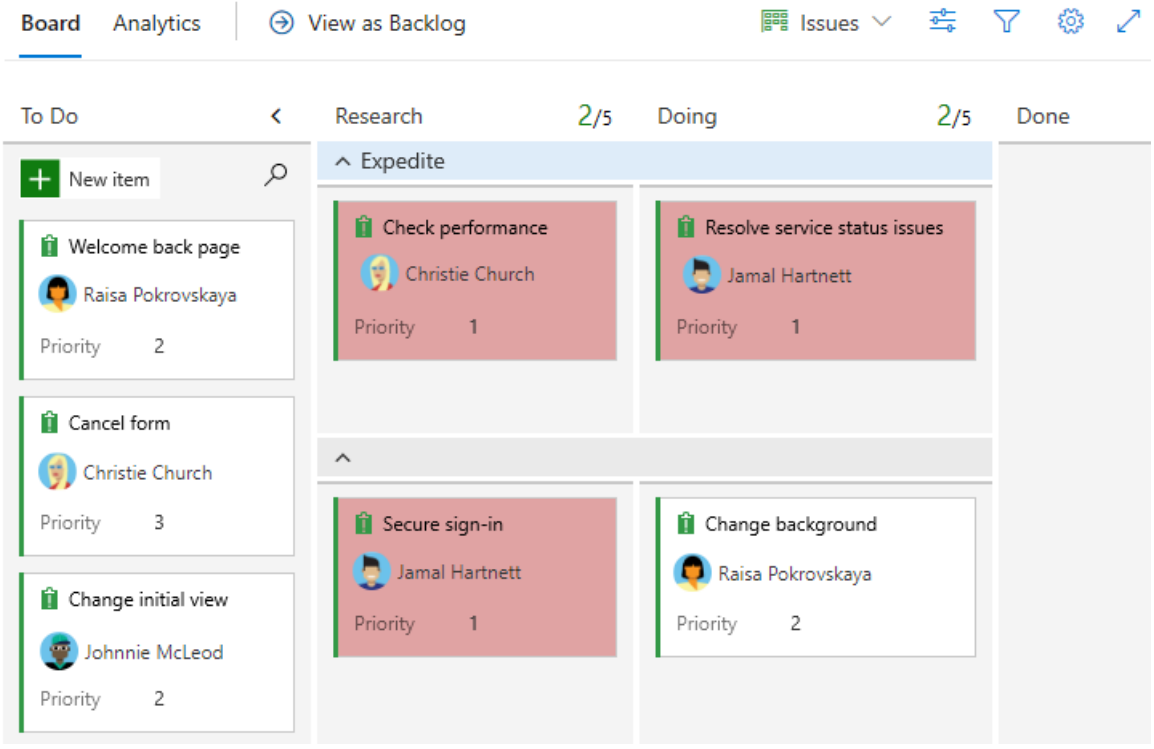
Review your changes

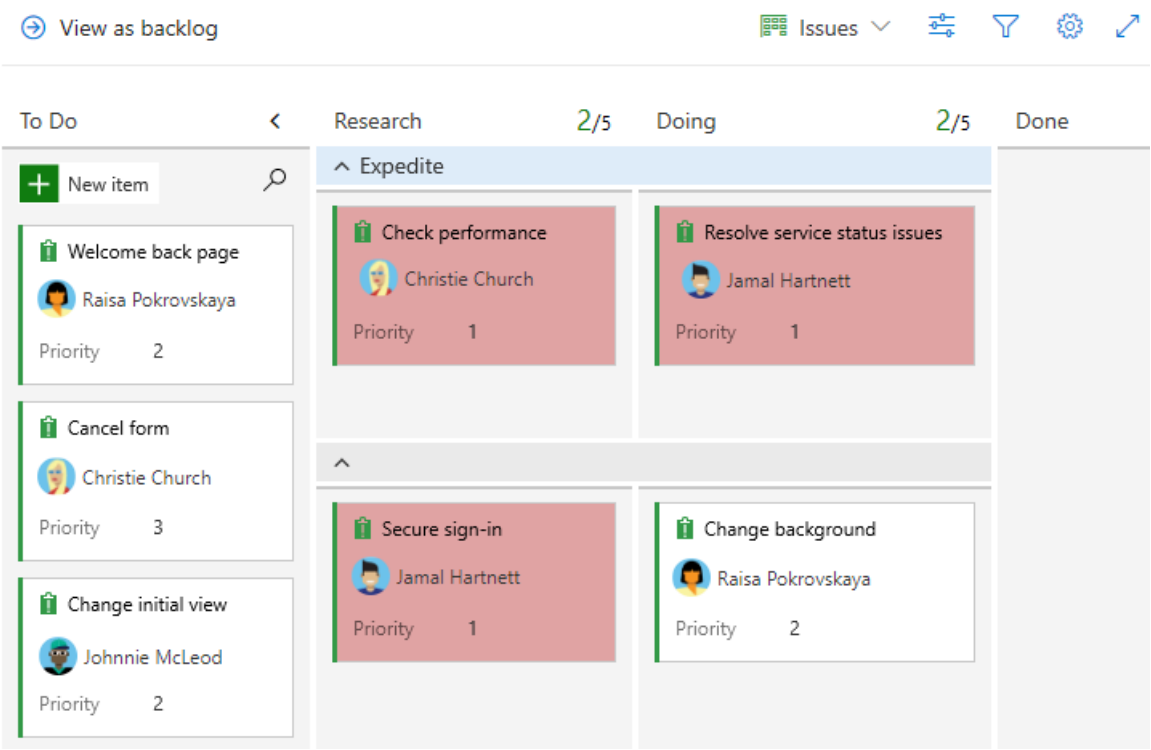
After you close the settings dialog, refresh (F5) your board to view your changes. Verify that all changes appear as expected, or revisit the Settings dialog to make a change.

- Basic process
- Agile process

Here we show the customizations made in this article. The following image also shows a style applied to the color when the Priority=1.


View as backlog
Issues
To Do
Research 2/5
Doing 2/5
Done
New item
Expedite
Welcome back page
Raisa Pokrovskaya
Priority 2
Cancel form
Christie Church
Priority 3
Change initial view
Johnnie McLeod
Priority 2
Check performance
Christie Church
Priority 1
Resolve service status issues
Jamal Hartnett
Priority 1
Secure sign-in
Jamal Hartnett
Priority 1
Change background
Raisa Pokrovskaya
Priority 2


Board
Analytics
View as Backlog
Issues
To Do
Research 2/5
Doing 2/5
Done
New item
Expedite
Welcome back page
Raisa Pokrovskaya
Priority 2
Cancel form
Christie Church
Priority 3
Change initial view
Johnnie McLeod
Priority 2
Check performance
Christie Church
Priority 1
Resolve service status issues
Jamal Hartnett
Priority 1
Secure sign-in
Jamal Hartnett
Priority 1
Change background
Raisa Pokrovskaya
Priority 2



Next step

Manage your project

Related articles

- Customize cards (addresses **Styles**, **Tag colors**, **Annotations**, and **Tests**)
- Card reordering
- Work in Progress limits
- Split columns
- Definition of Done
- Set working days
- Cumulative flow

Get started as a Stakeholder

10/4/2022 • 16 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items, manage build and release pipelines, and view dashboards. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team. For a quick overview of the features available to Stakeholders, see Stakeholder access quick reference.

> **NOTE**
> For public projects, Stakeholder access gives users greater access to features. To learn more, see Default roles and access for public projects. For information about working with pipelines, see these articles: Build your GitHub repository and Build OSS repositories.

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items, view and approve pipelines, and view dashboards. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team.

Stakeholder access is one of several supported access levels as described in Stakeholder access quick reference. To get access as a Stakeholder, ask your organization owner or Project Collection Administrator to add you to a project with Stakeholder access.

Stakeholder access is one of several supported access levels as described in Stakeholder access quick reference. To get access as a Stakeholder, ask your server administrator to add you to a security group that has Stakeholder access.

> **NOTE**
> Azure Boards supports several Agile methods such as Kanban and Scrum. Depending on what methods your team uses, you'll want to become familiar with other tools that Azure Boards supports. This article focuses on getting familiar with work items and the Kanban board. For additional information, see Related articles at the end of this article.

Use this tutorial to learn how to do the following tasks:

- Sign in to a project
- Understand which work item types are available to your project
- Open the Kanban board and open a work item
- Add details, tags, or comments to a work item
- View the product backlog
- Find work assigned to you, or query for other work items
- Understand what features are and aren't available to users with Stakeholder access

Connect to the web portal of a project

You must have been added to the Azure DevOps project and been granted Stakeholder or higher access level.

1. Choose the link provided in the email invitation you should have received. Or, open a browser window and enter the URL for the web portal.

`https://dev.azure.com/OrganizationName/ProjectName`

`http://ServerName:8080/tfs/DefaultCollection/ProjectName` For example, to connect to the server named *FabrikamPrime* and project named *Contoso*, enter `http://FabrikamPrime:8080/tfs/DefaultCollection/Contoso`.

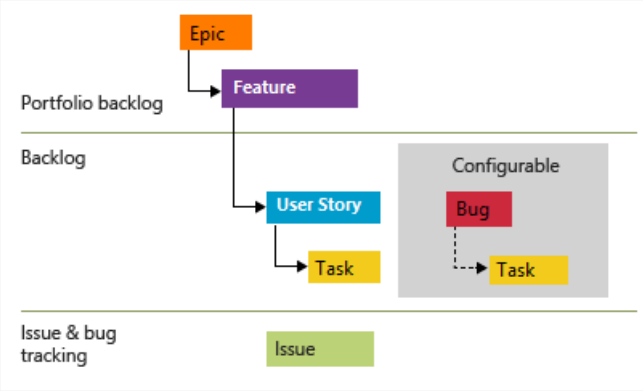
2. Enter your credentials. If you can't sign in, ask the organization owner or Project Administrator to add you as a member of the project with Stakeholder access.

Understand work items and work item types

Work items support planning and tracking work. Each work item represents an object stored in the work item data store. Each work item is based on a work item type and is assigned an identifier which is unique within an organization or project collection. Different work items are used to track different types of work as described in About work items. The work item types available to you are based on the process used when your project was created—Agile, Basic, Scrum, or CMMI—as illustrated in the following images.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

Open your Kanban board from the web portal

You can start viewing work items once you connect to a project.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.

   To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

> **TIP**
> Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level.



1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



> **TIP**
> Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level. Here we have selected **Backlog Items** for the Scrum process.



1. To view your Kanban board, open your project from a web browser. Select **Work** > **Backlogs** > **Stories**, and then select **Board**.



If you don't see **Work**, your screen size might be reduced. Select the three dots ( ) icon. Then select **Work** > **Backlogs** > **Board**.



2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.

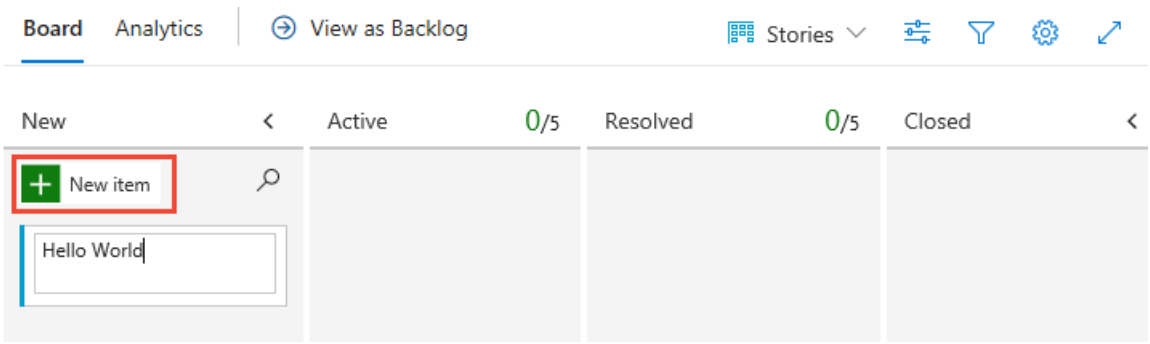


Your Kanban board appears.



Add work items

From the Kanban board, you can add work items, open them, and modify them. To add work items, open the backlog by choosing the **Backlog** link. To add a work item, select the + plus sign, enter a title, and then press Enter.

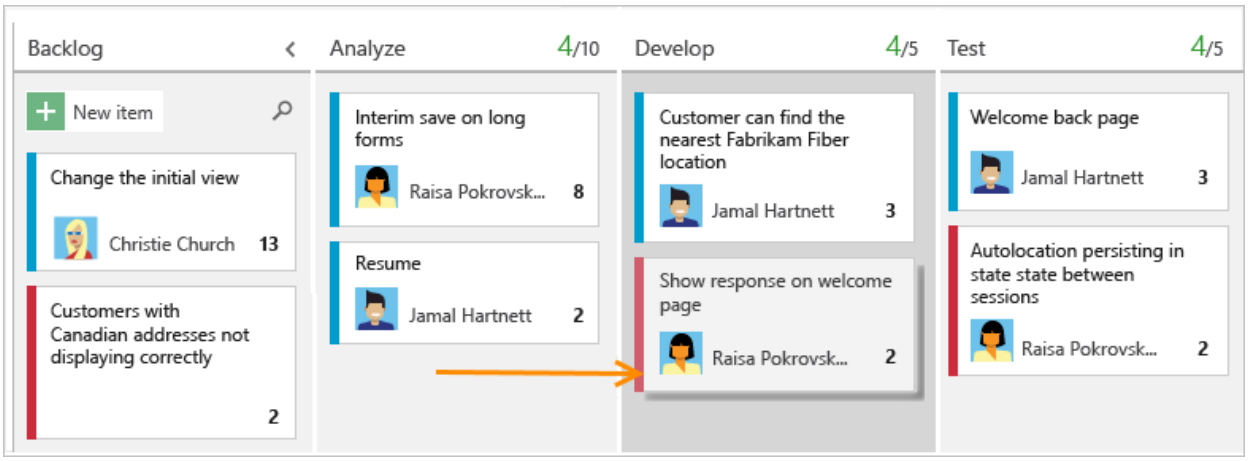


Or, you can add work items to the bottom of the product backlog. Open the backlog by choosing the **Backlog** link.

From the Kanban board, you can't add work items, but you can open them and annotate them. To add work items, open the backlog by choosing the **Backlog** link. Also, you can't update the status of a work item by drag-and-drop to a different column or reorder cards within a column.

Update status of work items

As work completes in one stage, update the status of an item by dragging it to a downstream stage.



> **NOTE**
> The ability for Stakeholders to drag-and-drop cards to different columns requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

Add details to a work item

To add information to a work item, open it by double-clicking the title or by selecting it and then typing Enter. Change one or more field values, add a description, add a tag, or add a comment in the **Discussion** section. You can also choose the **Attachments** tab and drag-and-drop or upload a file to share with others.

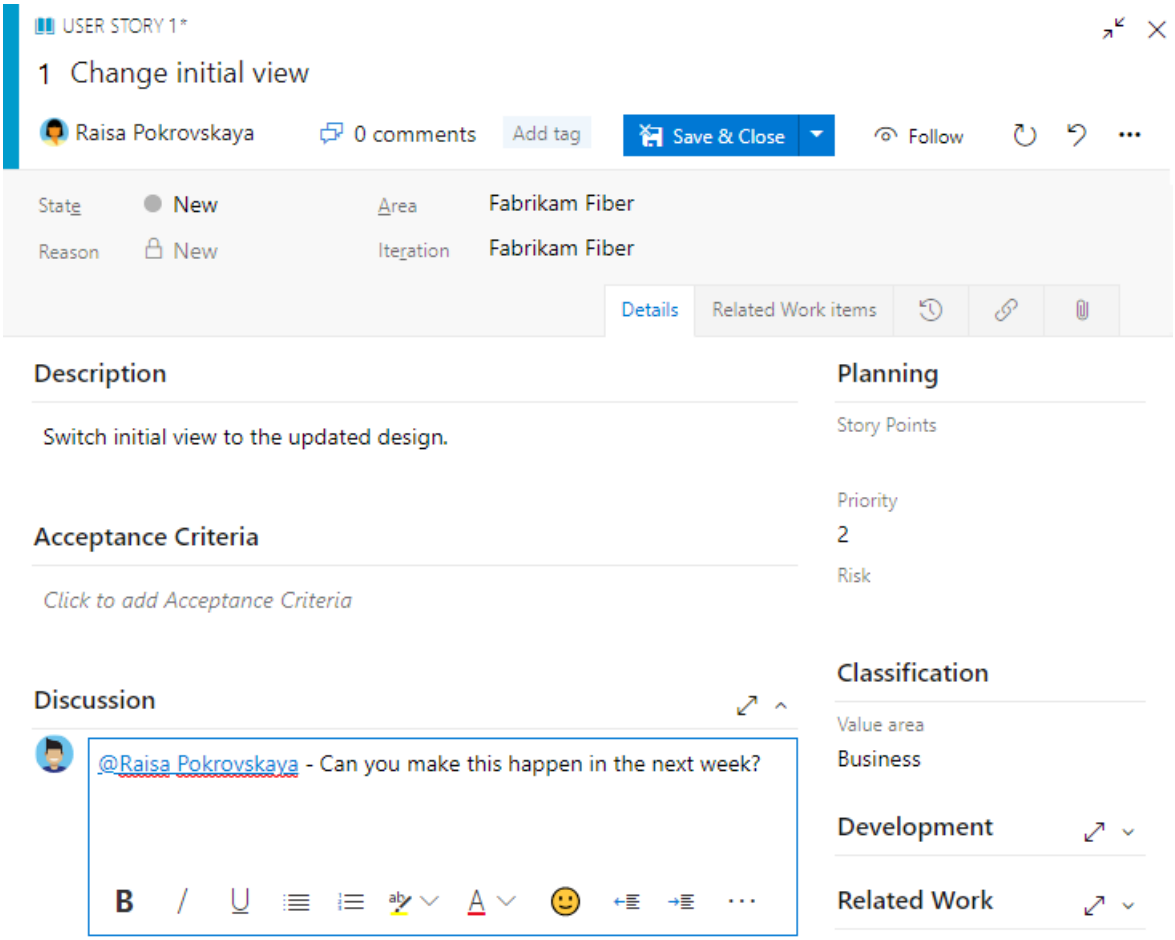
You can only assign work to a user who has been added to the project.

> **NOTE**
> The work item form you see may differ from those shown in the following images. The basic functionality is the same, however, changes have been made with different versions of Azure DevOps.

- Agile process

- Basic process
- Scrum process
- CMMI process

For example, here we assign the story to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa. Choose **Save & Close** when done.



Field descriptions

Field	Usage
Title	Enter a description of 255 characters or less. You can always modify the title later.
Assigned To	Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu lists only team members or contributors to the project.

> **NOTE**
> You can only assign work to a single user. If you need to assign work to more than one user, add a work item for each user and distinguish the work to be done by title and description. The Assigned To field only accepts user accounts that have been added to a project or team.

State	

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current status.

Reason

Use the default first. Update it when you change state as need. Each State is associated with a default reason.

Area (Path)

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting. To change the dropdown list of areas, see Define area paths and assign to a team.

Iteration (Path)

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later during a planning meeting. To change the drop-down list of iterations, see Define iteration paths and configure team iterations.

Description

Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

Acceptance Criteria

Provide the criteria to be met before the work item can be closed. Define what "Done" means by describing the criteria for the team to use to verify whether the backlog item or bug fix is fully implemented. Before work begins, describe the criteria for customer acceptance as clearly as possible. Have conversations between the team and customers to determine the acceptance criteria. These criteria help ensure a common understanding within the team to meet customers' expectations. Also, this information provides the basis for acceptance testing.

Priority

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **1**: Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible.
- **2**: Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately.
- **3**: Resolution of the work item is optional based on resources, time, and risk.
- **4**: Resolution of the work item is not required.

Value Area

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **Architectural**: Technical services to implement business features that deliver solution .
- **Business**: Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default).

Effort, Story Points, Size

Provide a relative estimate of the amount of work required to complete an issue. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include

powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, etc.). Use any numeric unit of measurement your team prefers.
The estimates you set are used to calculate team velocity and forecast sprints.

Add tags to a work item

Tags are useful for filtering backlogs, boards, and queries. As a Stakeholder, you can add existing tags to a work item, however, you can't add new tags.

From the web portal, open a work item and choose **Add tag** and type a keyword of an existing tag. Or, select from the list of previously assigned tags.

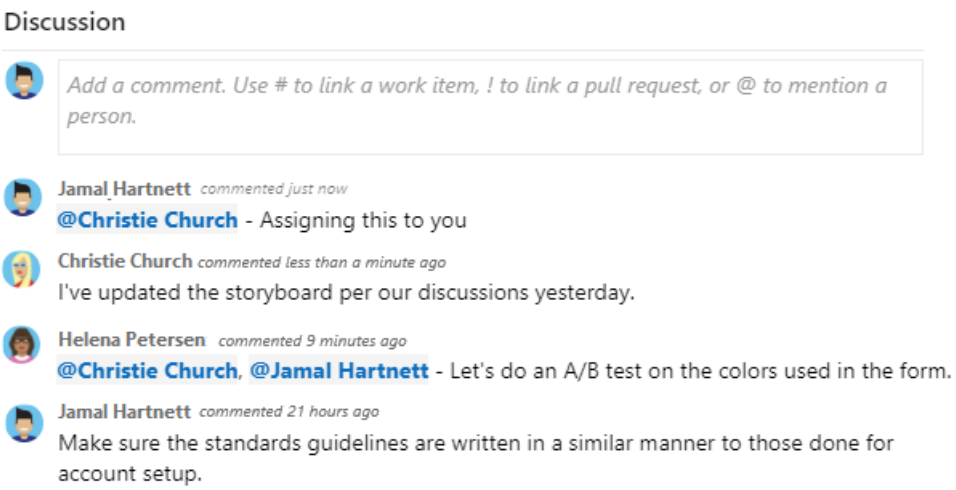


Tags that appear in the tag bar are already assigned to the work item. To unassign a tag, choose the x on the tag, Web ☒.

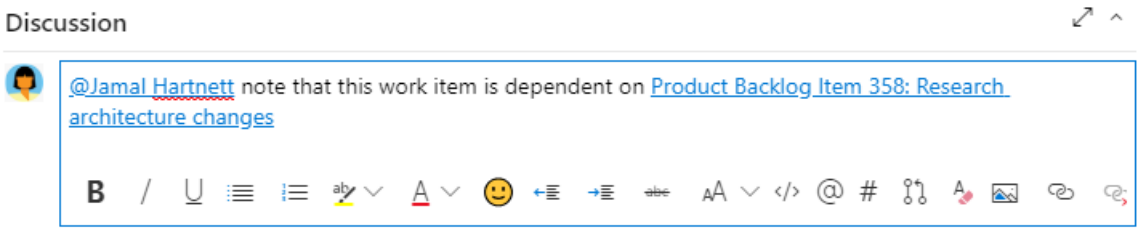
> **NOTE**
> By default, all Contributors and Stakeholders of public projects are granted permissions to add new and existing tags. Stakeholders in private projects can add tags that are already defined, but not add new tags. To grant or restrict permissions to create new tags, you set the project-level **Create tag definition** permission. To learn more, see Change project-level permissions.

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



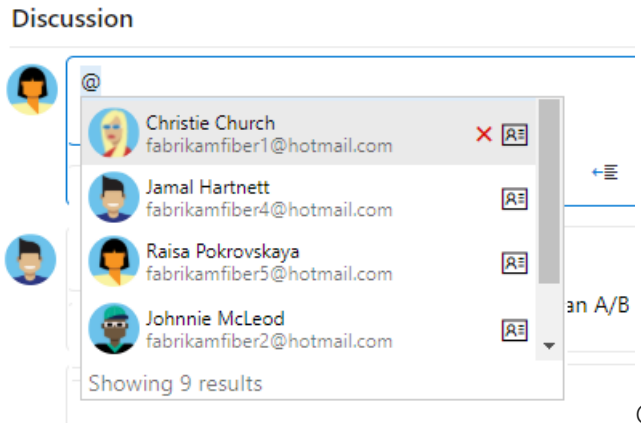
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the **History** field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons —@ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.

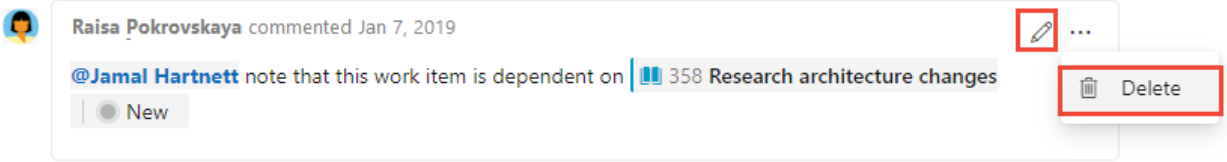


@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon and then choose **Delete**.



> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently

referenced will appear from which you can select.

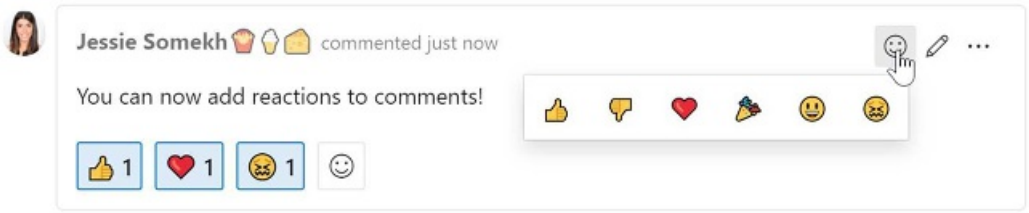
To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

You can't edit or delete comments once you've entered them.

> **IMPORTANT**
> For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

Add a reaction to a comment

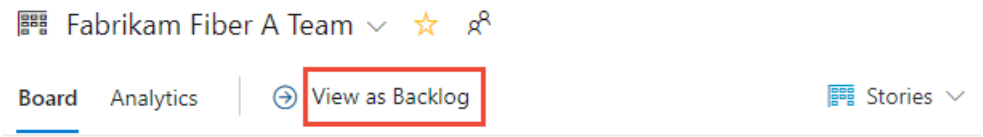
Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



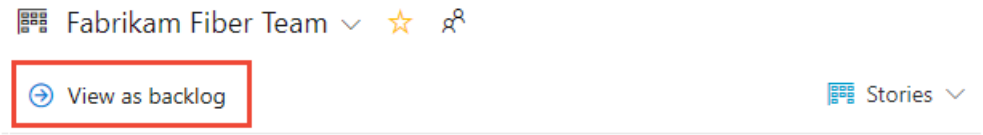
Check the backlog and prioritized work

You can check the product backlog to see how the team has prioritized work. Backlog items appear in priority order. Work item types may include bugs depending on the settings made for the team.

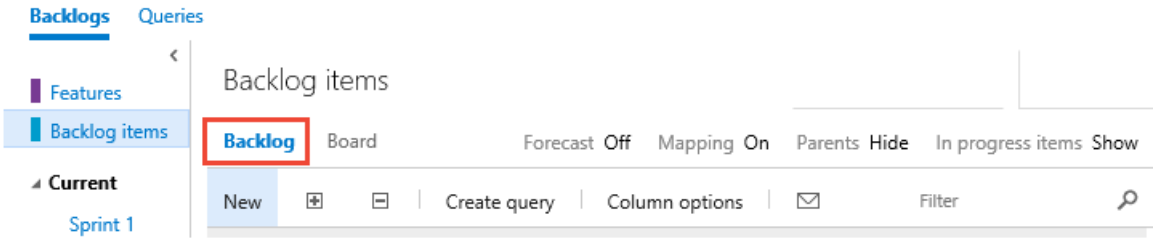
From the Kanban board, choose **View as backlog**.



From the Kanban board, choose **View as backlog**.



From the Kanban board, choose **Backlog**.

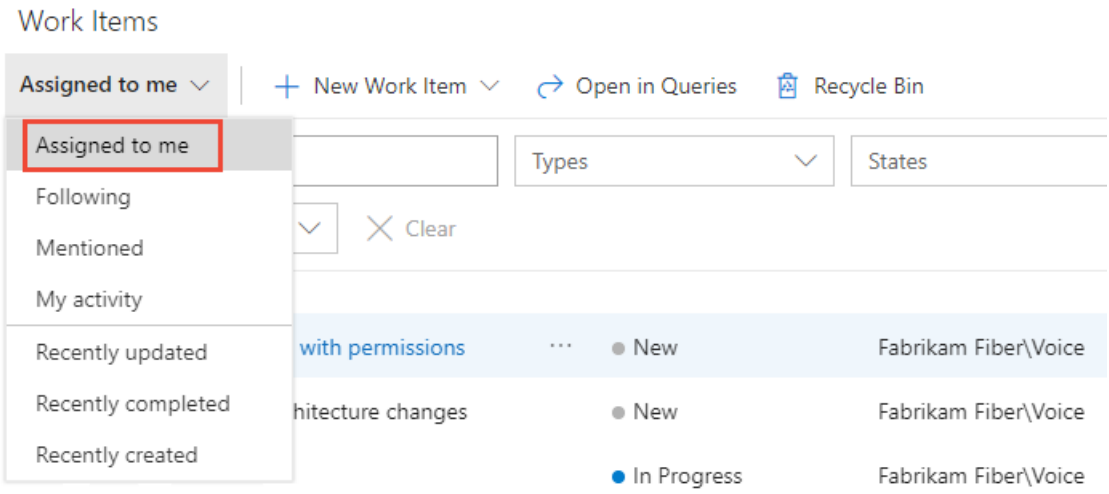


You should see the list of backlog items listed in priority order. You can add a backlog item which will be placed at the bottom of the list. With Stakeholder access, you can't re-prioritize work.

To view or edit a work item, select it and choose **Enter**.

Find work assigned to you, or query for other work items

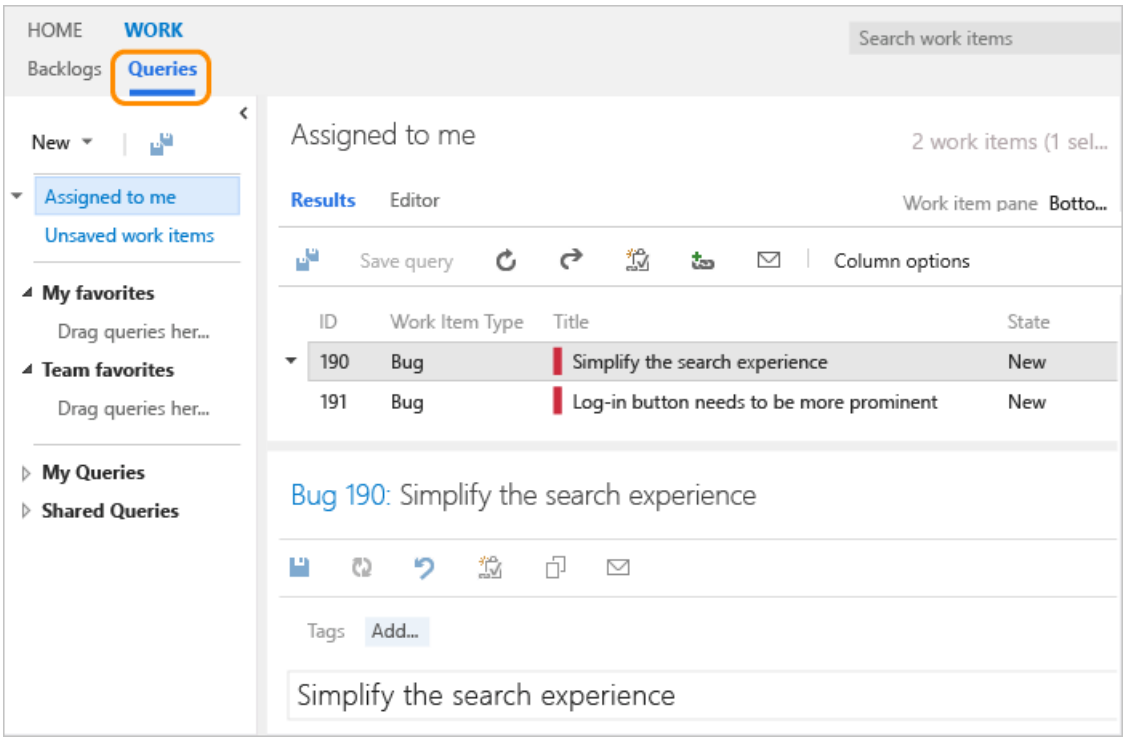
1. Choose **Boards>Work Items**, and then select **Assigned to me**.



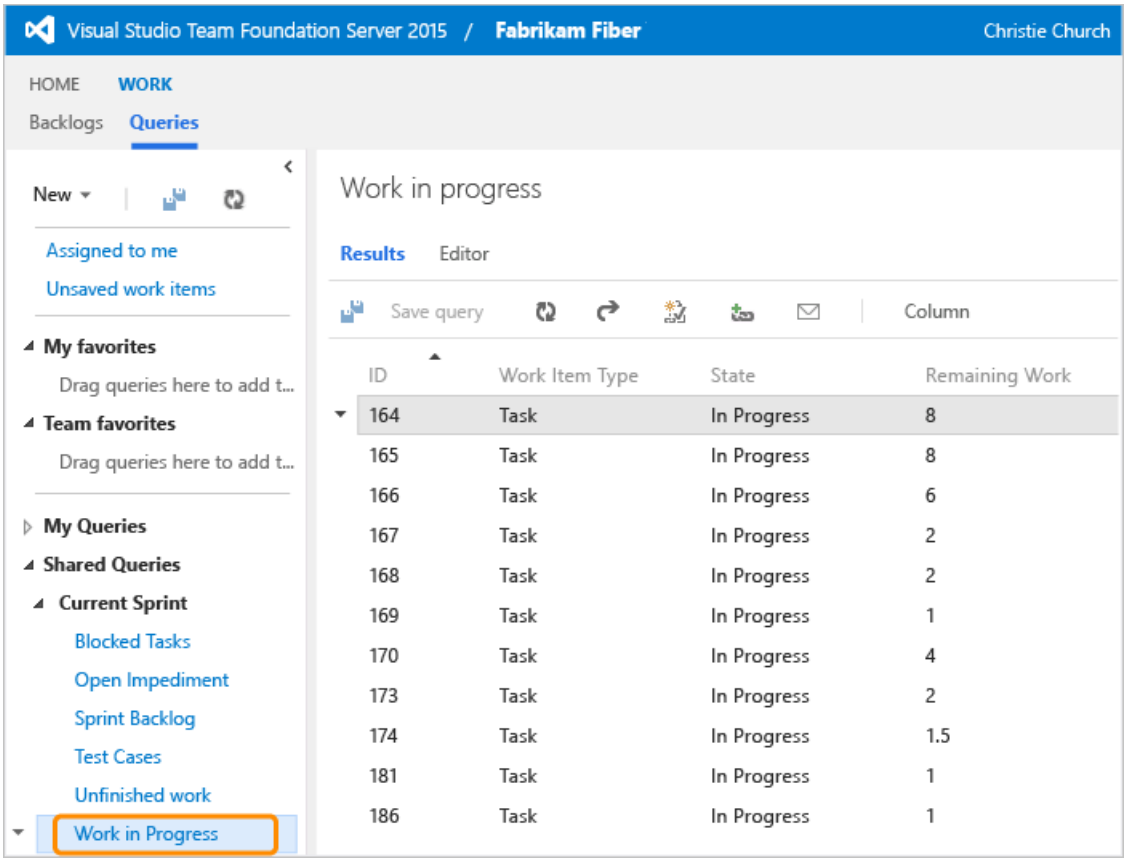
   You can focus on relevant items inside a project using one of the seven pivots as described next. Additionally, you can filter and sort each pivot view. For details, see View and add work items using the Work Items page.

2. To query for work items, see View, run, or email a work item query.

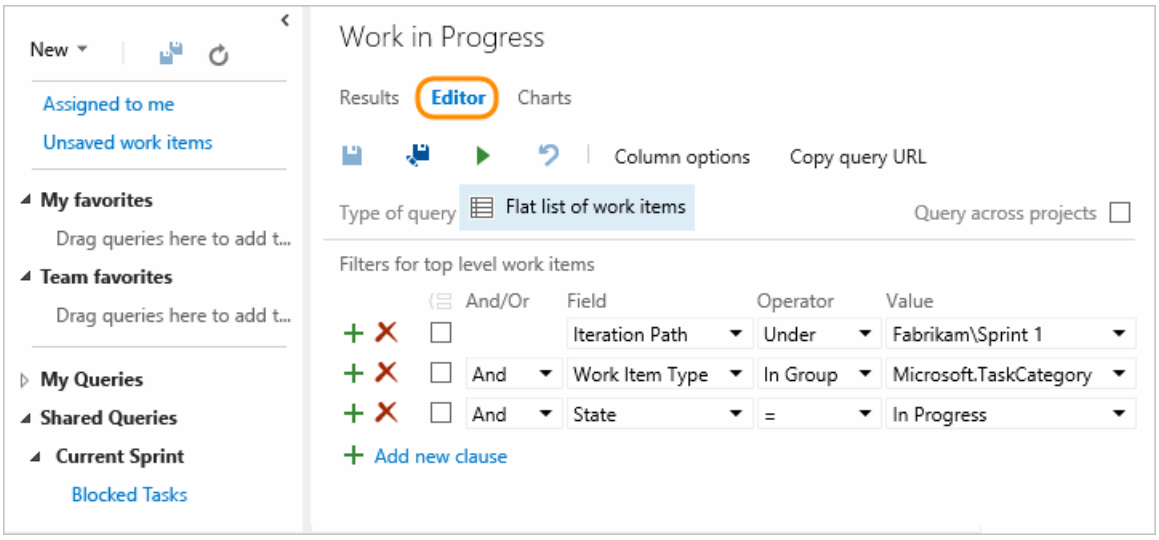
1. Open **Work>Queries** and select **Assigned to me** to see the list of work items assigned to you.



2. Or, open any of the queries defined in the Shared Queries folder.



3. And, you can create new queries or edit existing queries and save them under My Queries folder.



Related articles

For a comparison chart of Stakeholder versus Basic access, see this feature matrix. See also these quickstart guides:

- Add work items
- Create your backlog
- Kanban quickstart
- Access levels
- Change access levels

Configure and manage your Azure Boards project

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can start using Azure Boards project and configure resources as you go. No up-front work is required. Most settings define defaults.

As an organization owner or a project administrator, there are a few items you might want to attend to at the start. If you own a large organization, you'll want to consider other tasks to structure your projects. More tasks can be structured to help support multiple teams or software development apps.

Specifically, consider doing one or more of the following tasks:

- Add users to your project. To assign users to issues or tasks, you need to add them to your project.
- Share your project vision. To support people who will contribute to your project, provide them some directions via the project summary page, or through your project wiki.
- Define area and iteration paths. Define Iteration Paths if you work with Scrum methods or want to time-box your issues and tasks.
- Customize your issues or tasks. If you need more fields to track data, or other type of work item, you can customize your process.

Add users to your project

You add users to a team or project so they can contribute to the team and project. Users can be added to multiple teams and projects.

Users that have been added to an organization, can easily be added to a project by adding them to a team or inviting them to contribute to a project.

Team administrators can add users to their team which automatically adds them to the project. By adding users to a team, you make team-specific tools aware of them, such as the team security group, Team Members widget, and sprint capacity planning tools. To learn more about teams, see About teams and Agile tools.

Members of the **Project Administrators** group can add users to a project. Adding users to a team or project automatically adds them to the project's **Contributors** group. Members of this group have permissions to most features needed to contribute to work items, code, builds, and releases. For an overview of default permissions, see Default permissions quick reference.

Once users have been added to a project or organization, you can browse for their display name or user name (email alias) from any people-picker tool. Users can connect to a project and access features available through a supported client or the web portal.

To learn more, see the following articles:

- Add users or groups to a team or project
- Manage your organization or project collection, Add users to your organization
- Connect to a project

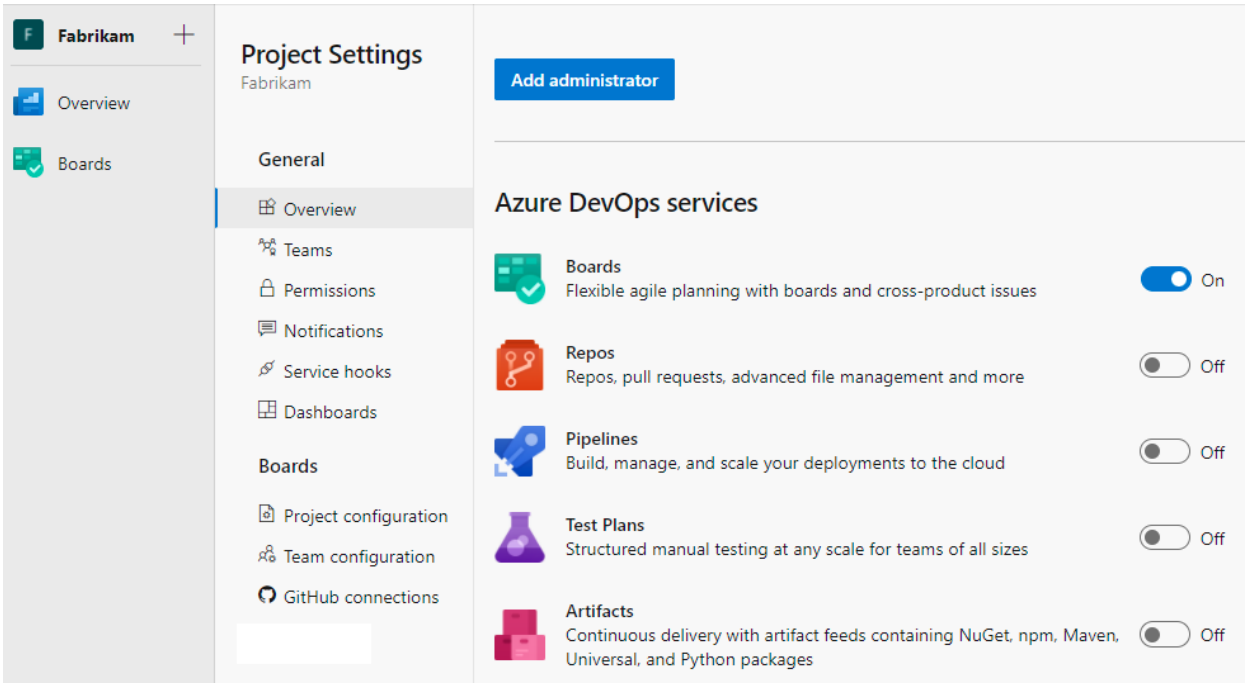
Share your project vision, set up a project wiki

Each project has a summary page that's useful for sharing information through **README** files. Or, redirect users to a project Wiki. For users who are new to your project, we recommend that you set up your project summary page. Or, you can provision a Wiki. Use these features to share established processes and procedures for your project.

Each project has a summary page that's useful for sharing information through **README files**. For users who are new to your project, we recommend that you set up your project summary page. Or, you can provision a Wiki. Use these features to share established processes and procedures for your project.

Remove unused services

To simplify the web portal user interface, you can disable select services. For example, if you use a project only to plan and track work, then disable all services except for **Boards** as shown in the following image. Once you disable the services, refresh your browser to see the changes take effect. To learn more, see Turn a service on or off.



Manage security and permissions

Access to select tasks is controlled by permissions and security groups. To quickly understand the defaults configured for your project, see Default permissions and access.

The following table lists the permissions assigned at the project-level. All of these permissions are granted to members of the **Project Administrators** group, except for the **Delete shared Analytics views** and **Edit shared Analytics views** permissions which are not set. For a description of each permission, see Permissions and groups reference, Groups.

The following table lists the permissions assigned at the project-level. All of these permissions are granted to members of the **Project Collection Administrators** group. For a description of each permission, see Permissions and groups reference, Groups.

> **NOTE**
> Permissions associated with Analytics requires that the Inherited process model is selected for an on-premises project collection.

General

- Delete team project
- Edit project-level information
- Manage project properties
- Rename team project
- Suppress notifications for work item updates
- Update project visibility
- View project-level information

- Delete team project
- Edit project-level information
- Manage project properties
- Rename team project
- Suppress notifications for work item updates
- View project-level information

- Delete team project
- Edit project-level information
- Manage project properties
- Rename team project
- View project-level information

Boards

- Bypass rules on work item updates
- Change process of team project
- Create tag definition
- Delete and restore work items
- Move work items out of this project
- Permanently delete work items

- Bypass rules on work item updates
- Change process of team project
- Create tag definition
- Delete and restore work items
- Move work items out of this project
- Permanently delete work items

- Create tag definition
- Delete and restore work items
- Permanently delete work items

Analytics

- Delete shared Analytics views
- Edit shared Analytics views
- View analytics

Test Plans

- Create test runs
- Delete test runs
- Manage test configurations

- Manage test environments
- View test runs

To learn more about security and setting permissions at the project-level, review the following articles:

- Get started with permissions, access, and security groups
- Change permissions at the project-level

Add members to the Project Administrators group

The person who creates a project is automatically added as a member to the **Project Administrators** group. Members of this group have permissions to manage project configuration, repositories, pipeline resources, teams, and all project-level permissions.

It's always a good idea to have more than one person who has administrative privileges. To add a user to this group, see Change permissions at the project level, Add members to the Project Administrators group.

Grant or restrict permissions

Permissions are managed at the following three levels and through role-based assignments.

- object
- project
- organization or collection

As a member of the **Project Administrators** group, you can grant or restrict permissions for all objects and at the project-level. To delegate specific tasks to others, we recommend that you add them to a built-in or custom security group or add them to a specific role. To learn more, see the following articles.

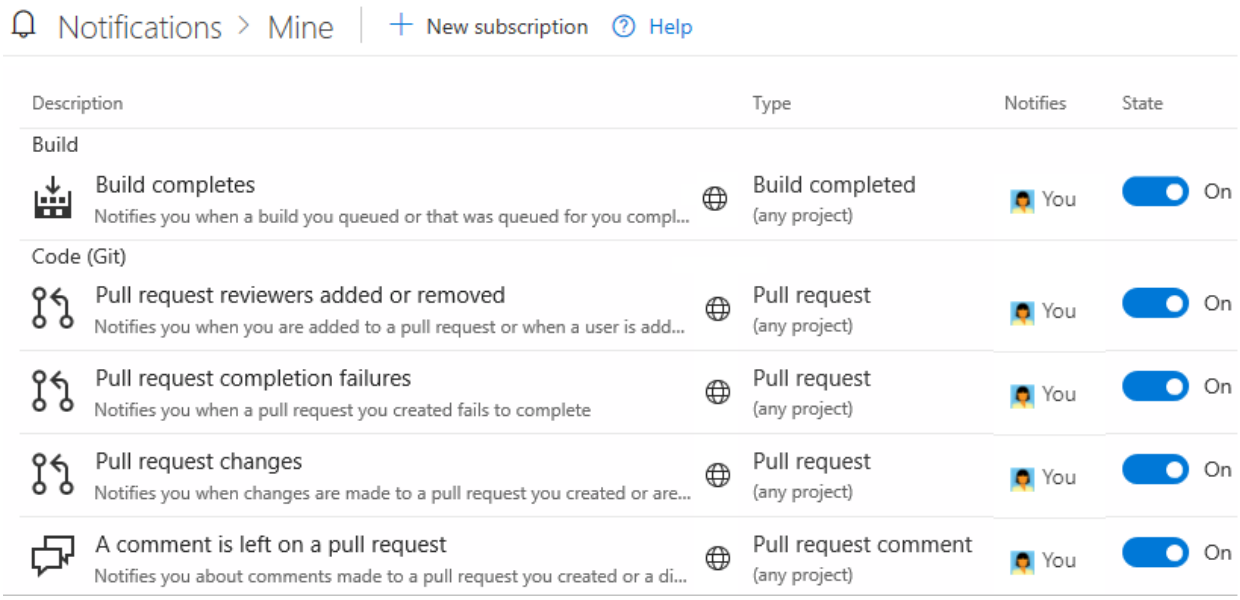
- Role-based permissions
- Add or remove users or groups, manage security groups
- Grant or restrict access to select features and functions
- Set object-level permissions

Review and update notifications

A number of notifications are predefined for each project you add. Notifications are based on subscription rules. Subscriptions arise from the following areas:

- Out-of-the-box or default subscriptions.
- Team, project, and organization or collection subscriptions defined by a team administrator or member of the **Project Administrators** or **Project Collection Administrators** groups.

If users believe they're getting too many notifications, you can direct them to opt out of a subscription.



Determine traceability requirements

If you're using most of Azure DevOps Services—Boards, Repos, Pipelines, and Test Plans— you'll want to alert your teams to those features that support end-to-end traceability. To get started, we recommend that you review the following articles:

- Cross-service integration and collaboration overview
- End-to-end traceability

Configure and customize Azure Boards

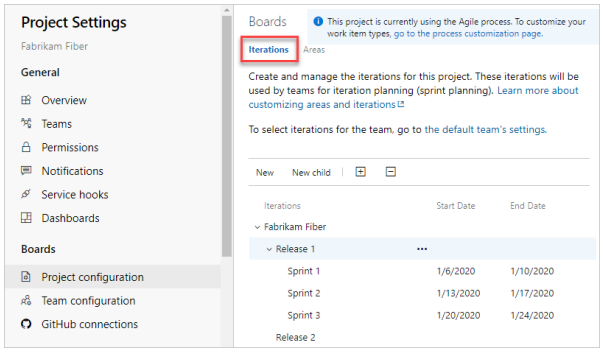
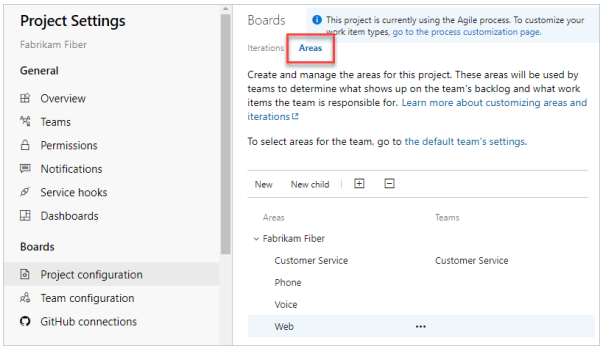
You can configure and customize Azure Boards to support a number of business requirements for planning and tracking work. At a minimum, you'll want to configure the following elements:

- Area paths to group work items by team, product, or feature area
- Iteration paths to group work into sprints, milestones, or other event-specific or time-related periods.

If you're new to Azure Boards and want an indepth overview of what you can configure and customize, see Configure and customize Azure Boards.

Define area and iteration paths to track work

If you support several products, you can assign work items by feature area by defining area paths. To assign work items to specific time intervals, also known as sprints, you configure iteration paths. To use the Scrum tools —sprint backlogs, taskboards, and team capacity—you need to configure several sprints. For an overview, see About areas and iteration paths.

ITERATIONS	AREAS
	

Customize work-tracking processes

You and your team can start using all work-tracking tools immediately after you create a project. But often, one or more users want to customize the experience to meet one or more business needs. You can customize the process easily through the user interface. As such, you'll want to establish a methodology for who will manage the updates and evaluate requests.

> **NOTE**
> By default, organization owners and users added to the **Project Collection Administrators** security group are granted permission to create, edit, and manage processes used to customize the work-tracking experience. If you want to lock down who is able to perform these tasks, you can set permissions at the organization-level to **Deny**.

To learn more, see these articles:

- About process customization and inherited processes
- Customize a project
- Add and manage processes

- On-premises XML process customization
- Add or modify a field to track work
- Add or modify a work item type

Install and manage extensions

To add new features and capabilities to Azure Boards, install extensions from the Azure DevOps Marketplace. You can install free, preview, or paid

To learn more, see Install free extensions for Azure DevOps. To learn about building your own Azure DevOps extensions, see developing and publishing extensions.

Integrate with other services

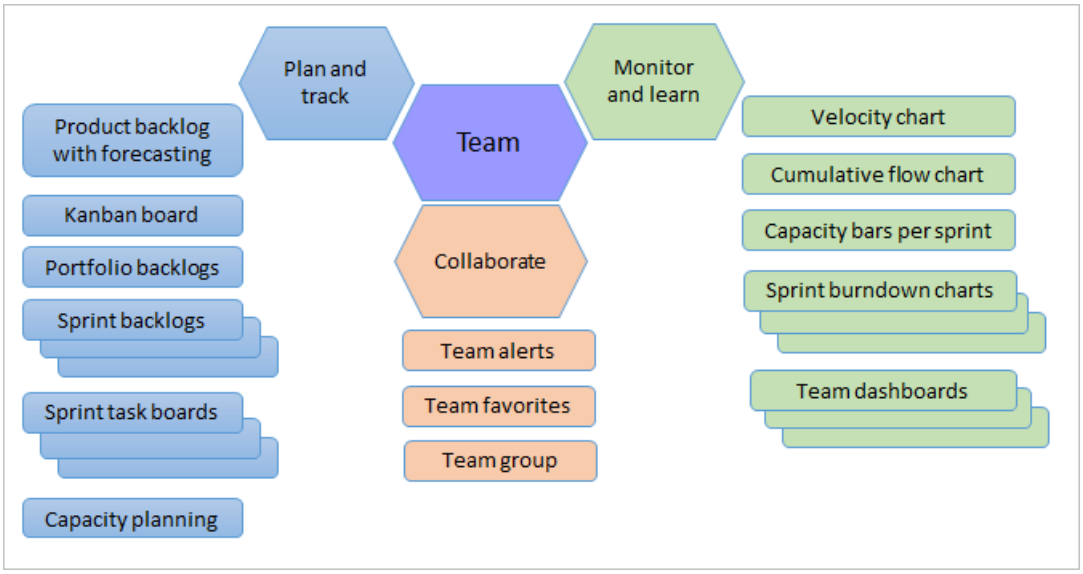
Azure DevOps supports integration with Azure, GitHub, and many other services. As a member of the **Project Administrators** group, you can configure integration with many of these services. To learn more, see the following articles.

- Azure DevOps and GitHub integration overview
- Azure Boards and GitHub integration
- Azure Boards with Microsoft Teams
- Azure Boards with Slack

- Integrate with service hooks

Add teams to scale your project

As your organization grows, we recommend that you add teams to scale your project. Each team gets access to their own set of customizable Agile tools.



To learn more, see the following articles:

- About projects and scaling your organization
- Add a team, move from one default team to several teams
- Add a team administrator

Related articles

- Web portal navigation
- Set user preferences
- Enable a preview feature
- Get started managing your organization or project collection

Index of system or basic work item fields in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use this index to look up a description of a field used to track an issue, task, or epic. This reference includes all fields defined for the Basic process. If you use another process—such as Agile, CMMI, or Scrum—see Work item field index for more field definitions.

> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

To support other tracking needs, you can define your own custom work item fields.

Alphabetical index

A

- Activated By
- Activated Date
- Activity
- Area ID
- Area Path
- Assigned To
- Attached File Count

B

- Board Column
- Board Column Done
- Board Lane

C

- Changed By
- Changed Date
- Closed By
- Closed Date
- Completed Work
- Created By
- Created Date

D-E

- Description
- Effort
- External Link Count

H-I

- History
- Hyperlink Count
- ID
- Iteration ID (System)
- Iteration Path (System)

L-N-P

- Link Comment
- Link Description
- Node Name
- Parent
- Priority

R

- Reason
- Related Link Count
- Remaining Work
- Remote Link Count
- Resolved By
- Resolved Date
- Resolved Reason

S

- Stack Rank
- Start Date
- State
- State Change Date

T

- Tags
- Team Project
- Title

W

- Watermark
- Work Item Type

Add and modify fields

To add fields to a process, modify select field attributes, and review fields added to a process, see Customize a process.

Also, you can view the list of fields defined for an organization and the WITs that reference them.

Related articles

- About managed queries

Keyboard shortcuts for Azure Boards and Team Explorer

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can use the keyboard shortcuts listed in this article when you work within Azure DevOps or Team Explorer. Along with these shortcuts, you can assign your own shortcuts in Visual Studio from the **Tools/Options/Environment/Keyboard** page.

For specific guidance on navigating within the web portal, see Web portal navigation.

Web portal

You can use these keyboard shortcuts when working in the web portal for Azure DevOps.

Navigate within lists

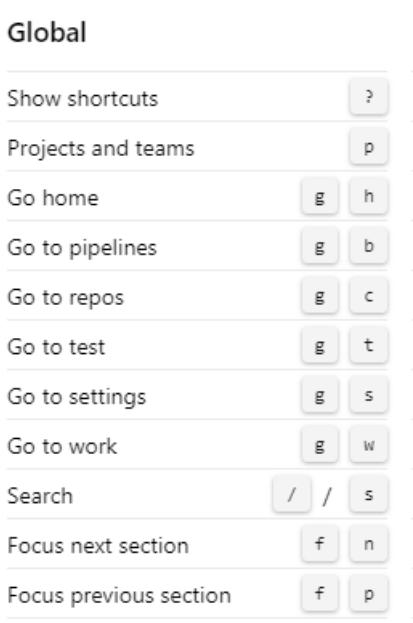
SHORTCUT	ACTION
Tab	Move focus
← →	Move focus left/right
↑ ↓	Move focus up/down
Ctrl+Home	Move focus to top of list
Ctrl+End	Move focus to bottom of list
Ctrl+↑↓	Move item up/down within list
Shift+↑↓	Highlight consecutive items
Menu	Open context menu
Esc	Dismiss context menu
Enter	Choose selected menu item

Web portal, global shortcuts

Type **?** to access the Global and page-specific shortcuts.

> **NOTE**
> The following shortcuts are available for TFS 2015.2 and later versions. Type **?** to access the Global and page-specific shortcuts.

You can use the following keyboard shortcuts from the web portal.

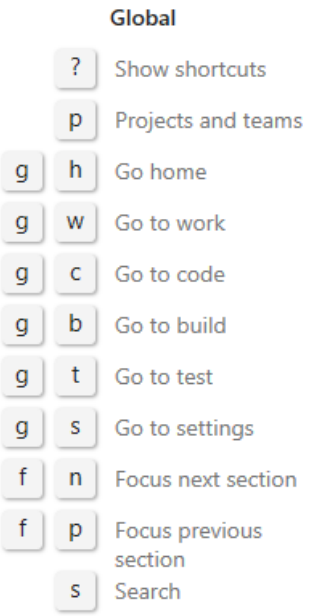


? Show keyboard shortcuts
p Go to **Projects and teams**

g,h Go to **Projects home**
g,b Go to **Pipelines**
g,c Go to **Repos**
g,t Go to **Test Plans**
g,s Go to **Project Settings**
g,w Go to **Boards**

/ or **s** Move focus to search
f,n Focus next section
f,p Focus previous section

You can use the following keyboard shortcuts from the web portal.



? Show keyboard shortcuts
p Go to Projects and teams

g,h Go to Projects home
g,w Go to Boards or Work
g,c Go to Repos or Code
g,b Go to Pipelines or Build and release
g,t Go to Test Plans or Test
g,s Go to Project Settings

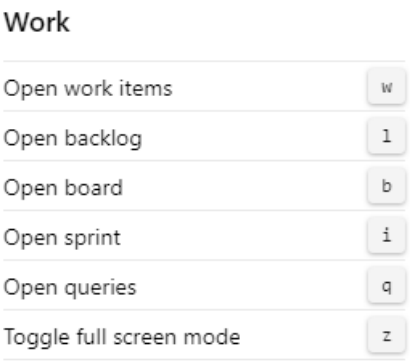
f,n Focus next section
f,p Focus previous section
/ Move focus to search

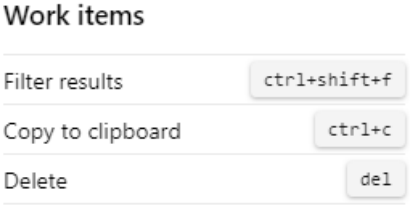
Page-specific shortcuts only work when in a specific page. For example, type **g c** to open the **Code** page, and then type **c p** to create a pull request. These navigation shortcuts work as long as the focus isn't on an input control.

Work items

You can use the following keyboard shortcuts when working from the **Boards > Work Items** or **Work > Work Items** page.

> **NOTE**
> The following shortcuts are available from the web portal for Azure DevOps Server 2019 and later versions.

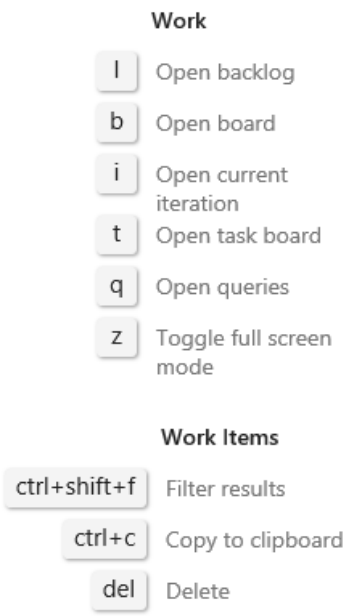




w Open work items
l Open backlog

b Open board
i Open sprint
q Open queries
z Toggle full screen

Ctrl+Shift+f Filter results
Ctrl+c Copy to clipboard
Delete Delete



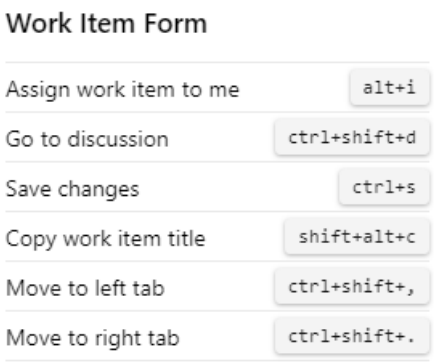
l Open backlog
b Open board
i Open current iteration
t Open task board
q Open queries
z Toggle full screen

Ctrl+Shift+f Filter results
Ctrl+c Copy to clipboard
Delete Delete

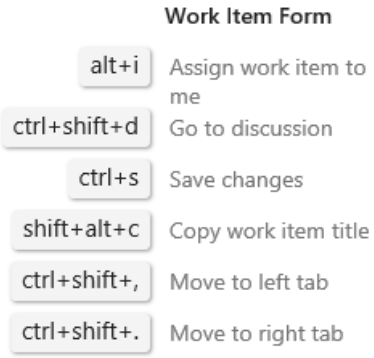
Work item form shortcuts

You can use the following keyboard shortcuts when interacting with a work item form. To view the valid shortcuts, enter ? from within the form.

> **NOTE**
> The following shortcuts are available from the web portal for Azure DevOps Server 2019 and later versions.



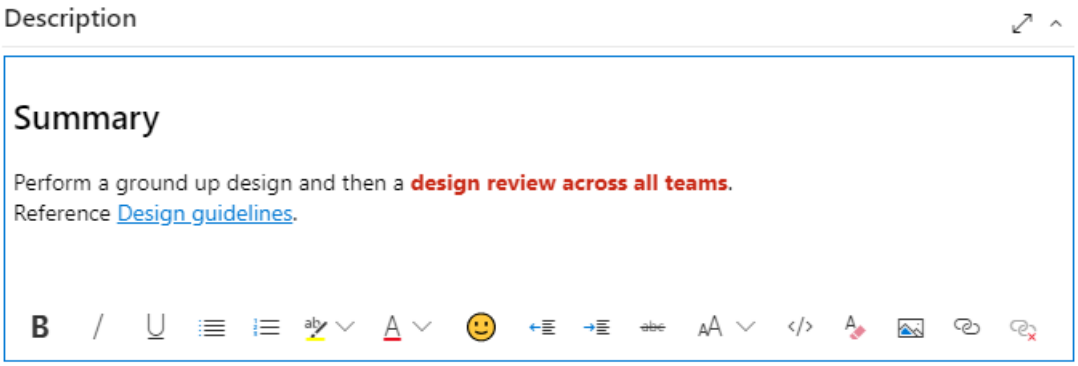
Alt+i	Assign work item to me
Ctrl+Shift+d	Go to discussion
Ctrl+s	Save changes
Shift+Alt+c	Copy work item title
Ctrl+Shift+,	Move to left tab (page)
Ctrl+Shift+.	Move to right tab (page)



Alt+i	Assign work item to me
Ctrl+Shift+d	Go to discussion
Ctrl+s	Save changes
Shift+Alt+c	Copy work item title
Ctrl+Shift+,	Move to left tab (page)
Ctrl+Shift+.	Move to right tab (page)

Rich text field shortcuts

The rich text editor tool bar displays below the text entry area when you click your cursor within the each text box that can be formatted.



You can use the following keyboard shortcuts when working in a web browser on one of the following operating systems. (**Command** = ⌘)

WINDOWS OS SHORTCUTS	MAC OS SHORTCUTS
Ctrl + **b** = Bold **Ctrl** + **c** = Copy text **Ctrl** + **i** = Italics **Ctrl** + **k** = Insert hyperlink **Ctrl** + **s** = Save **Ctrl** + **u** = Underline **Ctrl** + **v** = Paste text **Ctrl** + **y** = Redo **Ctrl** + **z** = Undo **Ctrl** + **.** = Bullet list **Ctrl** + **/** = Numbered list **Shift** + **:** = Emoji library	**Command** + **b** = Bold **Command** + **c** = Copy text **Command** + **i** = Italics **Command** + **k** = Insert hyperlink **Command** + **s** = Save **Command** + **u** = Underline **Command** + **v** = Paste text **Command** + **Z** = Redo **Command** + **z** = Undo **Command** + **.** = Bullet list **Command** + **/** = Numbered list **Shift** + **:** = Emoji library

The rich text formatting toolbar appears above each text box that can be formatted. It only becomes active when you click within the text box.



You can use the following keyboard shortcuts when working in the editor from a web browser running on a Windows operating system.

Ctrl + **b** = Bold
Ctrl + **c** = Copy text
Ctrl + **i** = Italics
Ctrl + **k** = Insert hyperlink
Ctrl + **s** = Save
Ctrl + **u** = Underline
Ctrl + **v** = Paste text
Ctrl + **y** = Redo
Ctrl + **z** = Undo
Ctrl + **.** = Bullet list
Ctrl + **/** = Numbered list
CTRL+**Spacebar** = Remove formatting

Boards

You can use the following keyboard shortcuts from any Kanban board, that is, when working from **Boards**>**Boards** or **Work**>**Board** page.

Kanban board

Add new item	n
Add new child item	c
Select first item	home
Open item	enter
Filter results	ctrl+shift+f
Move item up	ctrl+up
Move item down	ctrl+down
Move item left	ctrl+left
Move item right	ctrl+right
Move item to top of column	ctrl+home
Move item to bottom of column	ctrl+end
Move item to swimlane above	ctrl+shift+up
Move item to swimlane below	ctrl+shift+down
Rename item	f2
Show/hide empty fields	e
Expand all swimlanes	o
Collapse all swimlanes	u
Select first/next swimlane above	shift+pageup
Select last/next swimlane below	shift+pagedown

n Add new item
c Add new child item

Home Select first item
Enter Open item

Ctrl+Shift+f Filter results

Ctrl+↑ Move item up
Ctrl+↓ Move item down
Ctrl+← Move item left
Ctrl+→ Move item right

Ctrl+Home Move item to top of column
Ctrl+End Move item to bottom of column
Ctrl+Shift+↑ Move item to swimlane above
Ctrl+Shift+↓ Move item to swimlane below

F2 Rename item
e Show/hide empty fields
o Expand all swimlanes
u Collapse all swimlanes

Shift+Pageup Select first/next swimlane above
Shift+Pagedown Select last/next swimlane below



n Add new item
c Add new child item

Home Select first item
Enter Open item

Ctrl+Shift+f Filter results

Ctrl+↑ Move item up
Ctrl+↓ Move item down
Ctrl+← Move item left
Ctrl+→ Move item right

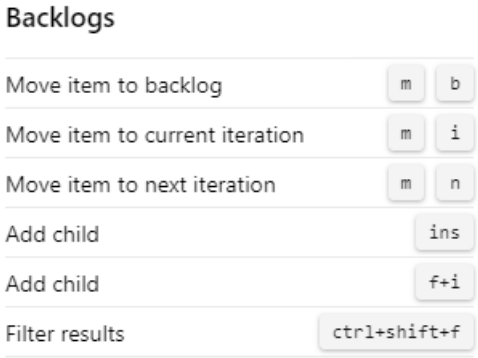
Ctrl+Home Move item to top of column
Ctrl+End Move item to bottom of column
Ctrl+Shift+↑ Move item to swimlane above
Ctrl+Shift+↓ Move item to swimlane below

F2 Rename item
e Show/hide empty fields
o Expand all swimlanes
u Collapse all swimlanes

Shift+Pageup Select first/next swimlane above
Shift+Pagedown Select last/next swimlane below

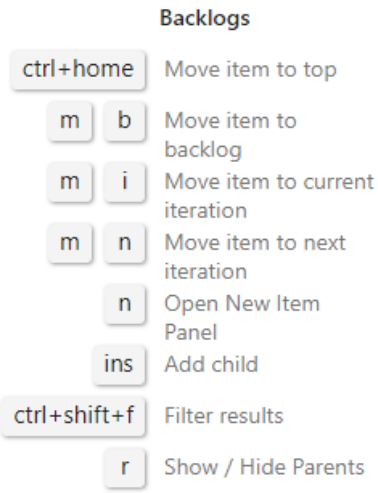
Backlogs

You can use the following keyboard shortcuts when working from a **Boards > Backlogs** page. These shortcuts work when you are on a product backlog, portfolio backlog, or sprint backlog page.



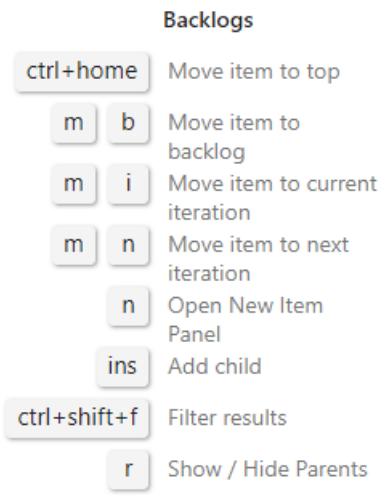
m,b Move item to backlog
m,i Move item to current iteration
m,n Move item to next iteration

Ins	Add child
f+i	Add child
Ctrl+Shift+f	Filter results



Ctrl+Home	Move item to top
m,b	Move item to backlog
m,i	Move item to current iteration
m,n	Move item to next iteration
Ins	Add child
Ctrl+Shift+f	Filter results<b

You can use the following keyboard shortcuts when working from a **Work>Backlogs** page. These shortcuts work when you are on a product backlog, portfolio backlog, or sprint backlog page.

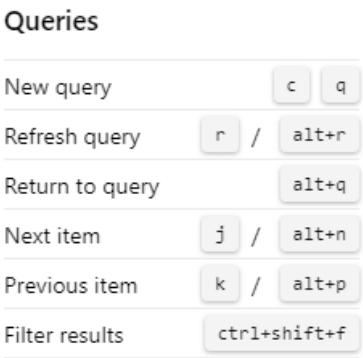


Backlogs

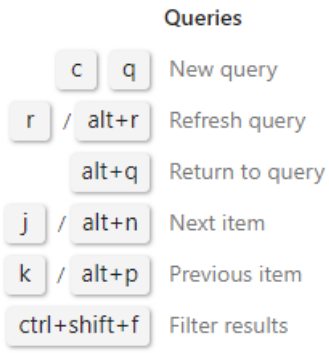
Ctrl+Home	Move item to top
m,b	Move item to backlog
m,i	Move item to current iteration
m,n	Move item to next iteration
n	Open new item panel
Ins	Add child
Ctrl+Shift+f	Filter results
r	Show/Hide Parents

Queries

You can use the following keyboard shortcuts when working with queries in the web portal. To view the valid shortcuts, enter **?** from **Boards>Queries** or **Work>Queries**.



c q	New query
r or **Alt+r**	Refresh query
Alt+q	Return to query
j or **Alt+n**	Next item
k or **Alt+p**	Previous item
Ctrl+Shift+f	Filter results



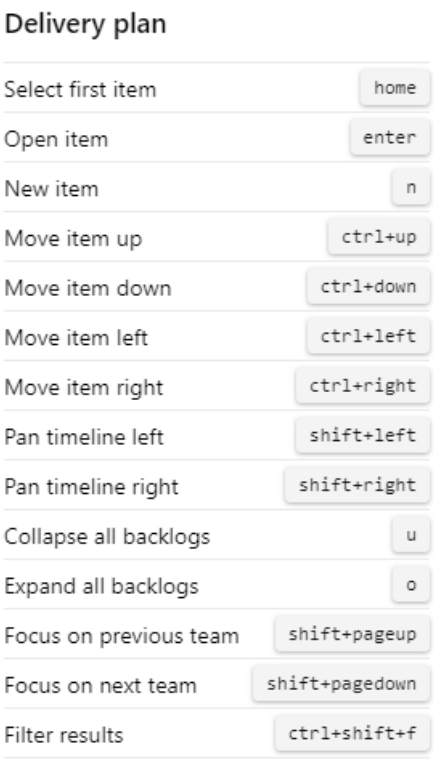
c q	Add new query
r or **Alt+r**	Refresh query
Alt+q	Return to query
j or **Alt+n**	Select next item
k or **Alt+p**	Select previous item
Ctrl+Shift+f	Filter results

Plans

You can use the following keyboard shortcuts when interacting with a delivery plan. To view the valid shortcuts, enter **?** when viewing a plan from the **Boards > Plans** or **Work > Plans** page.

To toggle between card titles only and card details, enter **t**.

> **NOTE**
>
> Type **?** to access Global and service-specific shortcuts.



Home Select first item
Enter Open item
n New item

Ctrl+↑ Move item up
Ctrl+↓ Move item down
Ctrl+← Move item left
Ctrl+→ Move item right
Shift+← Pan timeline left
Shift+→ Pan timeline right

u Collapse all backlogs
o Expand all backlogs

Shift+pageup Focus on previous team
Shift+pagedown Focus on next team
Ctrl+Shift+f Filter results

Team Explorer navigational shortcuts

Use these shortcuts when working in Team Explorer.

Navigate

Ctrl+0,a Open web portal
Ctrl+0,b Open Build
Ctrl+0,c Open Connect
Ctrl+0,d Open Documents
Ctrl+0,e Open Branches (Git)
Ctrl+0,g Open Changes (Git)

Ctrl+0,h Open Home
Ctrl+0,m Open My Work (TFVC)
Ctrl+0,p Open Pending changes (TFVC)
Ctrl+0,r Open Reports
Ctrl+0,s Open Settings
Ctrl+0,w Open Work items
Ctrl+0,y Open Synchronization (Git)

Ctrl+' Move focus to search box
Alt+0 Move focus to top of page
Alt+1...9 Move focus to visible section [1 through 9]
Alt+↑↓ Move focus to next/previous section

Context menu

<↓ Open a context menu
Esc Dismiss a context menu
← → Move focus left/right
↑↓ Move focus up/down
Enter Choose Context menu

Work item commands

Alt+m,g Open work item
Alt+m,i Add a work item
Alt+m,q Add a query
Shift+Alt,c Copy selected work item
Shift+Alt,l Link to new work item
Enter Open selected work item

You can use query results shortcuts whenever you have a list of work items, such as the query results view or a list of linked work items within a work item form.

QUERY EDITOR	ACTION	QUERY RESULTS	ACTION
← →	Move focus left/right	← →	Scroll left/right
↑↓	Move focus up/down	**PgUp/PgDn**	Scroll up/down
Shift+↑↓	Highlight consecutive clauses	**Shift+↑↓**	Highlight consecutive rows
Shift+←	Move focus left one field at a time	**Shift+Alt,n**	Move focus to next item
Shift+→	Move focus right one field at a time	**Shift+Alt,p**	Move focus to previous item
End	Move focus to end of current clause	**End**	Move focus to bottom of list
Enter	Move focus down	**Enter**	Open selected work item
Tab	Move focus right, one field at a time	**Home**	Move focus to top of list

QUERY EDITOR	ACTION	QUERY RESULTS	ACTION
Ctrl+c	Copy selected clause	+/-	Expand/collapse current row
Ctrl+s	Save changes (editor)	Ctrl+s	Save changes (results)
Ctrl+v	Paste copied clause	F5	Refresh
Del	Delete contents of current field or clause		

Related articles

- Keyboard shortcuts for Microsoft Test Manager
- Customize Visual Studio keyboard shortcuts
- Default keyboard shortcuts for Visual Studio
- Accessibility Features of Visual Studio
- Web portal navigation

Install Team Explorer

Team Explorer is a plug-in to Visual Studio. By installing the free Visual Studio Community, other Visual Studio version, or Visual Studio Team Explorer 2017 you gain access to Team Explorer.

Learn more about working in Team Explorer.

Tasks supported by Backlogs, Boards, Taskboards, and Plans

10/4/2022 • 12 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

What can you do from a backlog view versus a board view? How do these tasks differ from delivery plans? How do changes you make in one show up on the other? What customizations can you make for each?

Which view should you use to work with Agile methods?

In a nutshell...

- Backlogs display work items as a list and boards display them as cards
- You use your product backlog to quickly plan and prioritize your work
- You use your sprint backlogs and taskboards when you work in Scrum
- You use your Kanban board to update work status and when you employ Kanban methods
- Each backlog is associated with a board, changes to priority order you make in one are reflected in its corresponding board
- Plans allow you to review the deliverables for several teams across sprints and a calendar schedule
- Backlogs, boards, and plans are configurable for each team.

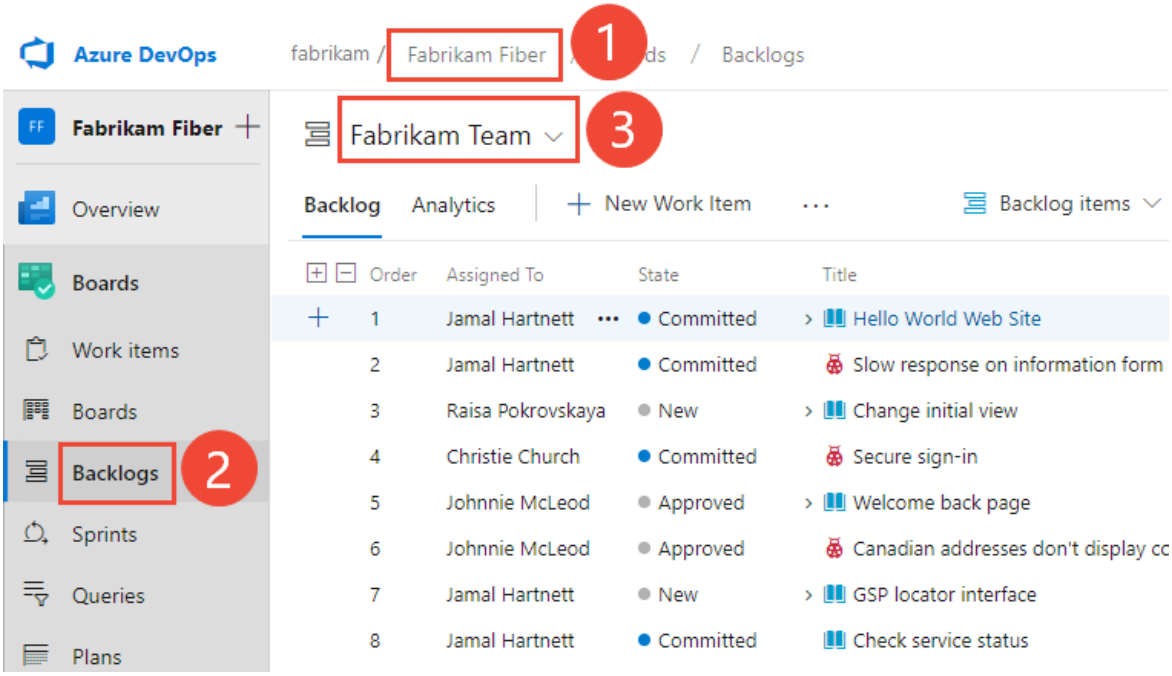
With list backlogs, you can quickly develop your project plan, group and prioritize work, and make bulk updates on selected work items. With boards, you can quickly update status and fields displayed for each work item.

And with plans, you can monitor progress, deliverables, and dependencies across several teams.

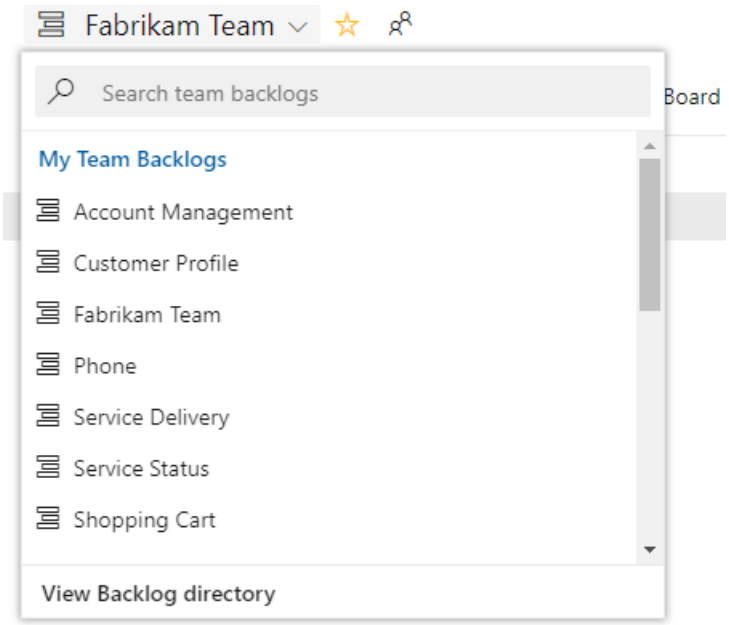
Access a backlog or board

You access your backlogs and boards from a web portal. If you work from the Stories (Agile) or Backlog items (Scrum) pages, you have access to the product backlog and Kanban board. When you work from a sprint page, you have access to the sprint backlog and taskboard. For an overview of working in Scrum or Kanban, see What is Azure Boards?.

1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



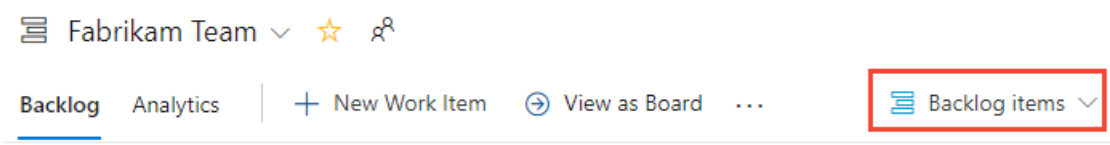
To select another backlog, open the selector and then choose a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.



TIP

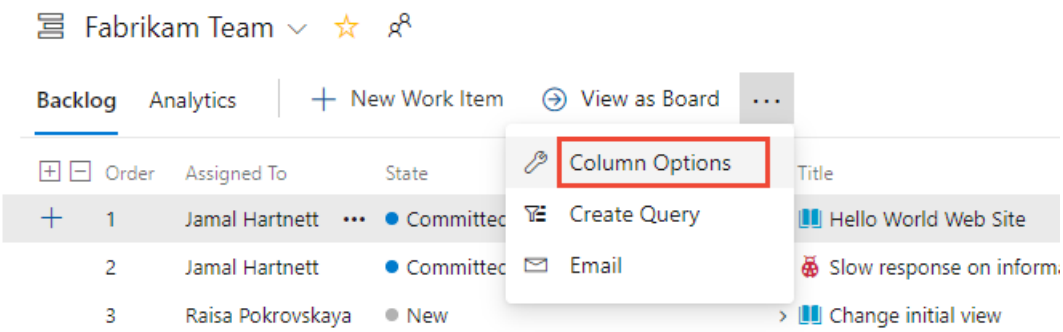
Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



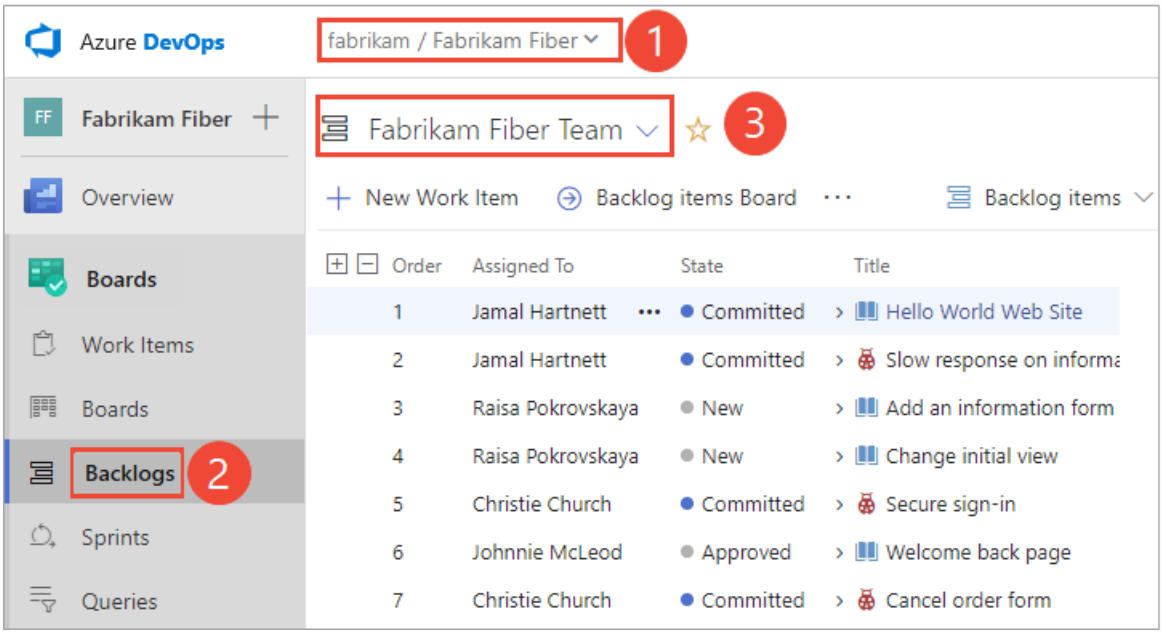
> **NOTE**
> The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.

3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.

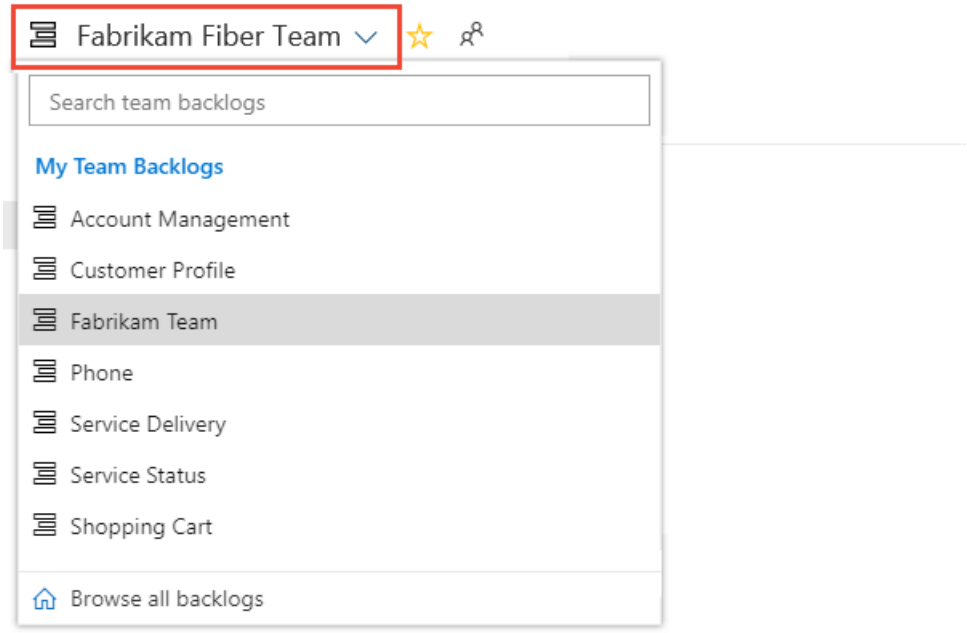


> **TIP**
> Each team member has several tools to configure their backlog view: **Expand/Collapse one level**, **Column Options**, **Backlog level selector**, **View options**, and **Filter** toolbar. Options set for each backlog level are distinct and persist until changed. To learn more, see Configure your backlog view.

1. (1) Check that you've selected the right project, (2) choose **Boards** > **Backlogs**, and then (3) select the correct team from the team selector menu.



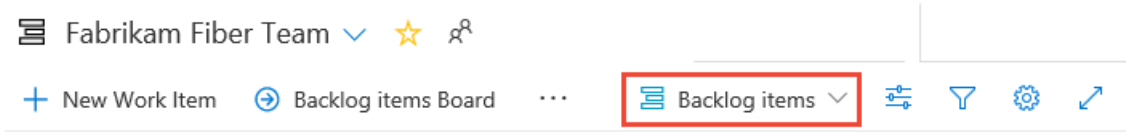
To choose another team, open the selector and select a different team or choose the **Browse all backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



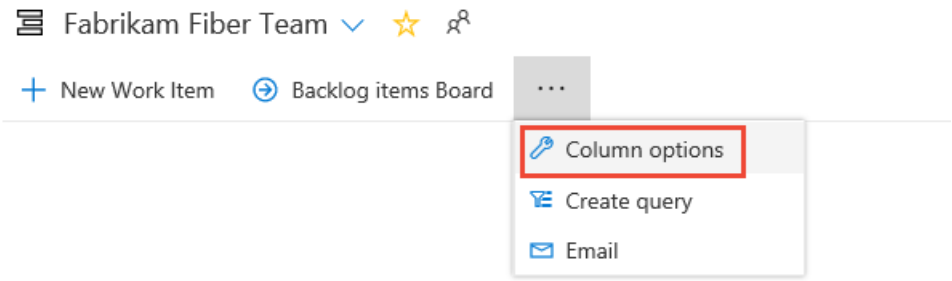
TIP

Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

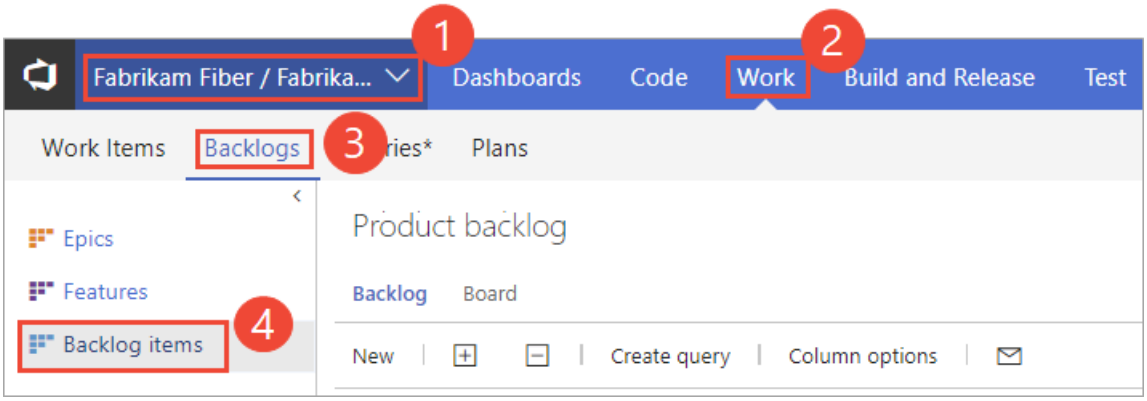
2. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level.



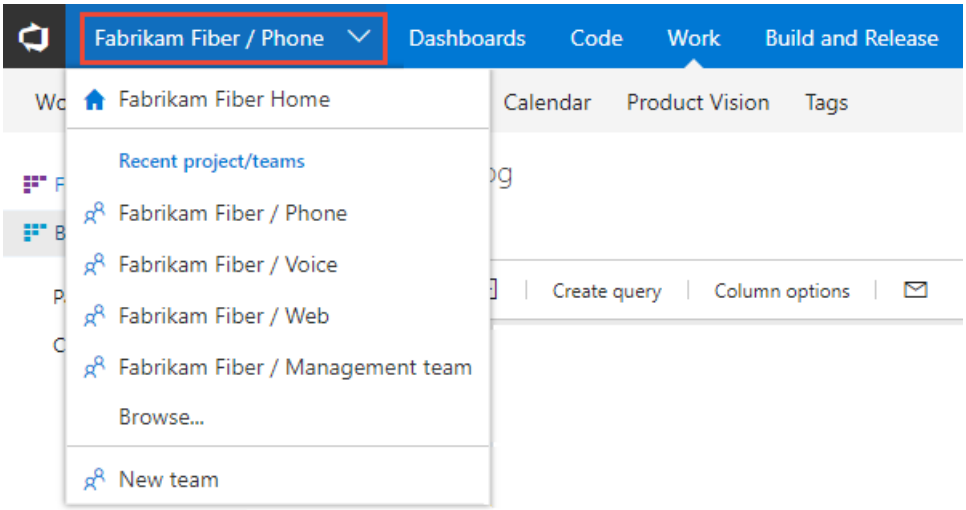
3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.



From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).



To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



Three classes of backlogs, two types of boards

To manage work, you have access to three classes of backlogs (portfolio, product, and sprint) and two types of boards (Kanban and Taskboards). Backlogs list work items, boards display work items as cards. Backlog and board views provide similar and distinct features to support planning and tracking.

You use work items to share information, assign work to team members, track dependencies, organize work, and more. You can apply different filters to your backlogs and boards to just show those items of interest.

> **NOTE**
> For guidance on configuring and customizing your project and teams to support your business needs, review Configuration and customization of Azure Boards.

Portfolio, product, and sprint backlogs

Portfolio backlogs typically track high-level features, scenarios, or epics. Your product backlog contains a prioritized list of user stories, deliverables, or works you plan to build or fix. Portfolio backlogs help you organize your product backlog into a hierarchy of elements. Sprint backlogs contain just those items that each team is working on during a scheduled sprint or iteration period.

For details about working in each type of backlog, see Create your backlog, Define features and epics, and Sprint planning (sprint backlogs).

TIP

You can't sort a backlog by column. However, you can use the Create Query option on each backlog to create a query that you can sort on any field column you choose. To learn more about queries, see Use the query editor to list and manage queries.

Kanban and Taskboards

Kanban and Taskboards support visualizing the flow of work and monitoring metrics to optimize that flow. Kanban boards track requirements, are sprint-independent, and you monitor the flow through the cumulative flow chart. Taskboards track tasks defined for a sprint and you monitor the flow via the sprint burndown chart.

For details about working in each type of board, see Kanban basics and taskboard.

Feature support across backlogs and boards

The following table indicates those elements or tasks associated with each type of backlog and board.

Associated element or task

Backlog type: Portfolio

Backlog type: Product

Board type: Kanban

Backlog type: Sprint

Board type: Task

Corresponding backlog or board type

Kanban

Kanban

Portfolio or product

Task

Sprint

Add items and child items (see notes 1, 2)

Yes

Yes

Yes

Yes

Yes

Reorder items

Yes

Yes

Yes

Yes

Yes

Map items

Yes (except the top-level portfolio backlog)

Yes

No

No

No

Filter by keyword and select fields

Yes

Yes

Yes

Yes

Yes

Show/hide parents

Yes (except the top-level portfolio backlog)

Yes

No

No

No

Show/hide in progress items (see note 3)

Yes

Yes

No

No

No

Forecast

No

Yes

No

No

No

Customize: show bugs (see note 1)

No

Yes

Yes

Yes

Yes

Configure: Columns

Yes, see Column options

Yes, see Column options

Yes, see Add columns

Yes, see Column options

Yes, see Customize Taskboard

Display rollup progress, counts, or totals

Yes, see Display rollup

Yes, see Display rollup

No

No

No

Customize: Add more backlog or board views

Yes, see Select backlog navigation levels

Yes, when you add another team (see note 4)

Yes, see Select backlog navigation levels

Yes, see Schedule sprints

Yes, see Schedule sprints

Customize cards

n/a

n/a

Customize: cards

n/a

Customize Taskboard

In-context reports

Cumulative flow Velocity

Cumulative flow Velocity

Cumulative flow Velocity

Sprint burndown

Sprint burndown

Duration (see note 5)

Project or release

Project

Project

Sprint

Sprint

Associated element or task

Backlog type: Portfolio

Backlog type: Product

Board type: Kanban

Backlog type: Sprint

Board type: Task

Corresponding backlog or board type

Kanban

Kanban

Portfolio or product

Task

Sprint

Add items and child items (see notes 1, 2)

Yes

Yes

Yes

Yes

Yes

Reorder items

Yes

Yes

Yes

Yes

Yes

Map items

Yes (except the top-level portfolio backlog)

Yes

No

No

No

Filter

Text or tags

Text or tags

Text or select fields

Text

Backlog items or people

Show/hide parents

Yes (except the top-level portfolio backlog)

Yes

No

No

No

Show/hide in progress items (see note 3)

Yes

Yes

No

No

No

Forecast

No

Yes

No

No

No

Customize: show bugs (see note 1)

No

Yes

Yes

Yes

Yes

Customize: Columns

Yes, see Column options

Yes, see Column options

Yes, see Add columns

Yes, see Column options

Yes, see Customize Taskboard

Customize: Add more backlog or board views

Yes, see Select backlog navigation levels

Yes, when you add another team (see note 4)

Yes, see Select backlog navigation levels

Yes, see Schedule sprints

Yes, see Schedule sprints

Customize: cards

n/a

n/a

Yes

n/a

Customize Taskboard

Charts

Cumulative flow Velocity

Cumulative flow Velocity

Cumulative flow Velocity

Sprint burndown

Sprint burndown

Duration (see note 5)

Project or release

Project

Project

Sprint

Sprint

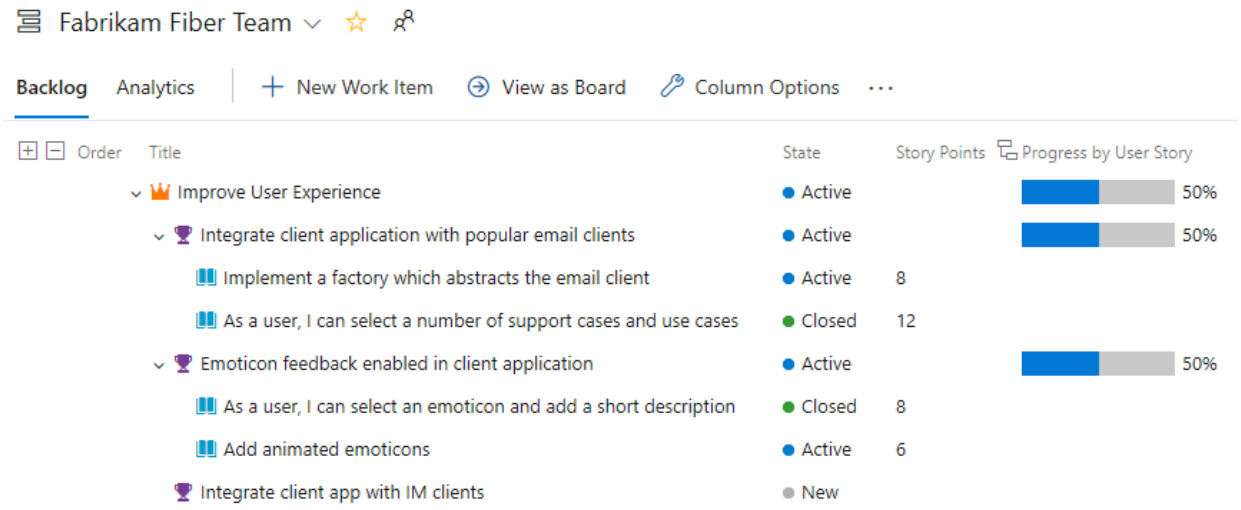
NOTE

1. Each team can determine how they want to track bugs: as requirements, as tasks, or not at all. When tracked as requirements, they appear in your product backlog, sprint backlogs, and Kanban board. When tracked as tasks, they appear in your sprint backlogs and taskboards. For details, see Show bugs on backlogs and boards.
2. Work items that appear on each team backlog and board meet the criteria defined for the team selected area and iteration paths.
3. The **In progress items Show/Hide** control is another filter you can apply to your product and portfolio backlogs. This control essentially shows or hides those work items where work has begun. It's useful to show/hide In Progress items when forecasting sprint work.
4. When you add a team, you essentially add another product backlog associated with that team. Each team can then manage their own set of sprint backlogs and portfolio backlogs. See Manage teams and configure team tools for details.
5. Duration refers to how you use your backlog or board to plan and track work over time. Once you change the State of a work item to done or completed, it no longer appears on a portfolio or project backlog. As you complete each sprint, the system maintains a history of your activity. You can review past sprints and sprint burndown charts by choosing the sprint listed under the Past section. For more information, see Sprint burndown.

Display rollup progress, counts, or totals

Product and portfolio backlogs support the display of rollup columns. You can add one or more rollup columns to display progress bars, counts of descendant work items, or totals of select numeric fields. Your settings persist for each page you customize and are only valid for your views.

Progress bars indicate the percentage of descendant items within a hierarchy that are closed or completed. *Counts* display the total number of descendant items. And, *Totals* provide a sum of numeric fields, such as Effort, Story Points, Completed Work, or Remaining Work of descendant items.
For example, progress bars are shown here for a portfolio backlog.



"In Progress" filter for the backlog

The In progress items **Show/Hide** filter causes some backlog items to display or not display. Bugs and other backlog items aren't listed when **In progress items=Hide** and their assigned State corresponds to an In Progress state category. Bugs in a New state will display, however, bugs in an Assigned state won't. To learn more about state categories, see Workflow states and state categories.

On your backlog, set **In progress items** to **Show** to see all active bugs and other items on your backlog.

Delivery Plans display team deliverables

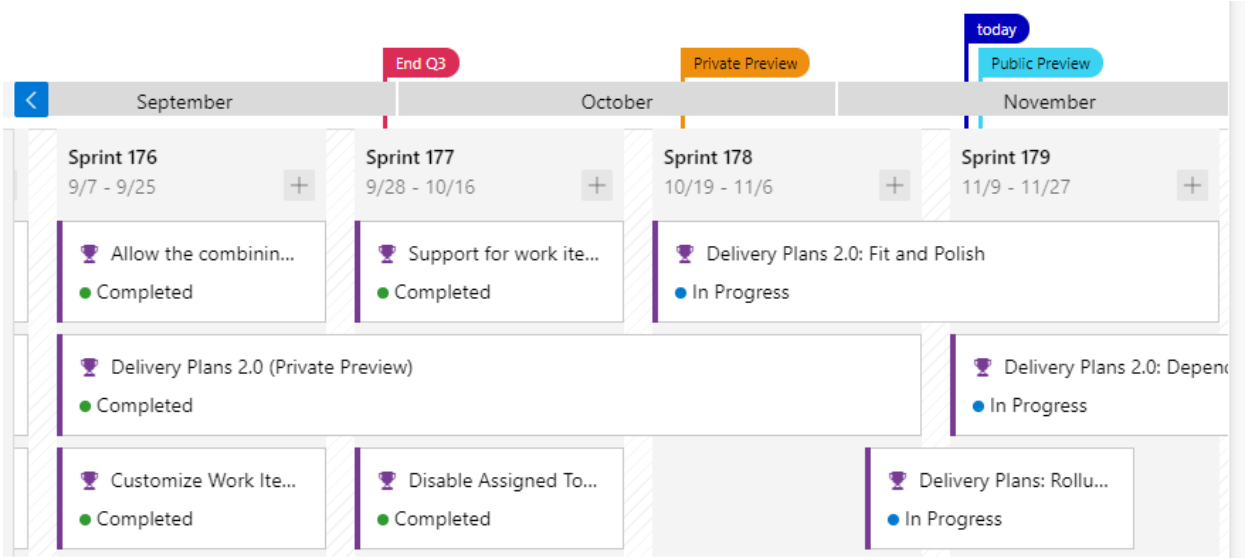
With Delivery Plans, you gain tailor-made views across several teams and their development backlogs—stories, features, or epics. You can use these views to drive alignment across teams by overlaying several backlogs onto your delivery schedule.

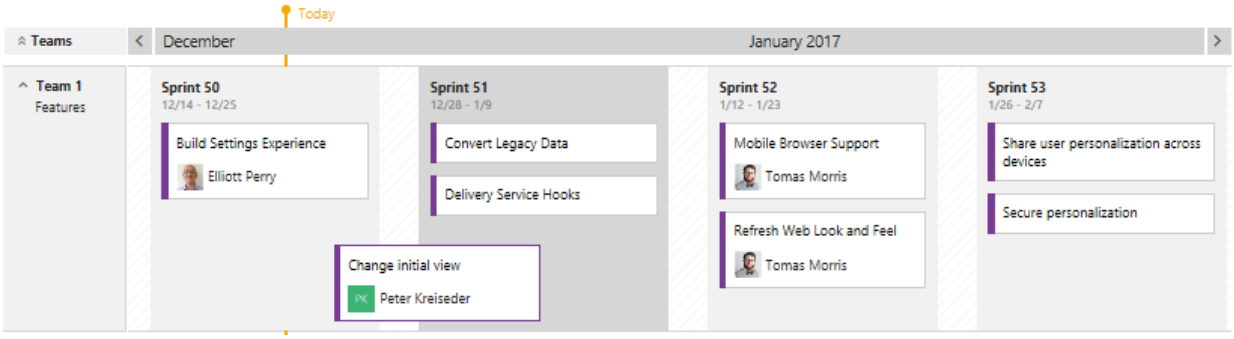
> **NOTE**
>
> A new version of Delivery Plans is available in public preview for Azure Boards. This feature is now part of Azure Boards and not an extension. To enable it, see Manage or enable features and turn on **New Delivery Plans**. This new version of Delivery Plans provides support for the following tasks:
>
> - Epics can be added to a delivery plan
> - Work item cards can span iteration boundaries
> - Drag and drop borders show when a work item starts and ends
> - You can add backlog items to a team from a plan
> - You can view work item dependencies
> - Stakeholders can view plans

You install Delivery Plans from the Visual Studio Marketplace, in the Azure DevOps tab.

All users with basic access can view, add, and configure Delivery Plans. Stakeholders, however, don't have access to Delivery Plans.

When you configure a plan, you select the team or teams and backlog levels of interest. To learn more about Delivery Plans, see Review team plans.





Taskboard items versus query list items

You may wonder why the items that are shown on the taskboard differ from those items that are listed in a query that was created from its corresponding sprint backlog.

It's possible to assign tasks to an iteration but not have them linked to a parent backlog item. These items will show up in the created query, but might not show up on the taskboard itself. The system runs the query and then applies a few background processes before displaying the taskboard items.

These reasons can cause work items that belong to the Task Category to not appear on a sprint backlog or taskboard:

- The task hasn't been linked to a parent backlog item. Only those bugs and tasks that you've linked to a parent product backlog item (Scrum), user story (Agile), or requirement (CMMI) whose iteration path is set to the sprint will appear on the sprint backlog page.
- The task is a parent of another task, or the user story is a parent of another user story. If you've created a hierarchy of tasks or user stories, only the child-level tasks or the child-level stories at the bottom of the hierarchy appear.
- The task's linked parent corresponds to a backlog item defined for another team. Or, the area path of the task's parent backlog item differs from the task's area path.

Customize backlog and board levels

If you need more than three backlog and board levels, you can add more. To learn how, see Customize your backlogs or boards for a process.

If you need more than three backlog levels and board levels, you can add more. You can also add or modify the fields that are defined for a work item type (WIT) or add a custom WIT. To learn how, see the following articles based on the process model used to update your project:

Inheritance process model:

- Customize your backlogs or boards for a process.
- Customize an inheritance process

On-premises XML process model:

- Add portfolio backlogs
- Customize the On-premises XML process model

If you need more than three backlog and board levels, you can add more. To learn how, see Add portfolio backlogs.

Related articles

Now that you understand how backlogs, boards, and plans work, get started using them to plan and track your work.

A few things to keep in mind...

- Every team owns their own backlog. To add a new set of backlogs and boards, you add a new team
- To have work completed by several teams roll up to a portfolio backlog, you'll want to setup the team hierarchy
- Every backlog has a corresponding Kanban board you can use to track progress and update status
- Each team can control how bugs show up on their backlogs
- When you add child items, they're linked to their parent using parent-child links, which support hierarchical views and tree queries

More articles of interest:

- About teams and Agile tools
- Add work items
- Dashboards

More tools from the Marketplace

You may find other tools to help plan and track your work from the Visual Studio Marketplace, Azure DevOps tab.

Best tool to add, update, and link work items in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure Boards provides you several tools—many designed to support a single task and others that support several tasks. This article provides a guide to the best tool for specific tasks that will help you work most efficiently.

Work item form

If you want to make a single update to one work item, you can make an update from within the work item form. When you want to add or update several work items at a time, then you'll want to use a backlog or query.

Work item form controls | Work item field index

Best tool for:

- Updating a work item field for a single work item
- Adding to the discussion, mentioning others in the discussion
- Choosing to follow or unfollow a work item
- Driving Git development, creating a branch
- Adding a link to another work item or external object
- Copying or cloning a work item
- Deleting the work item

Additional supported tasks:

- Share information
- Copy the work item URL
- Capturing the work item to use as a template
- Updating fields of the work item by applying an existing template

Work Items

Use the Work Items page to quickly focus on work items of interest to you.

Best tool for:

- Listing and filtering work items of interest to you, specifically work items that meet the following criteria:
  - That are assigned to you
  - That you chose to follow
  - Where you were mentioned
  - That you've recently viewed or updated
  - That has been recently updated, completed, or created for the project.

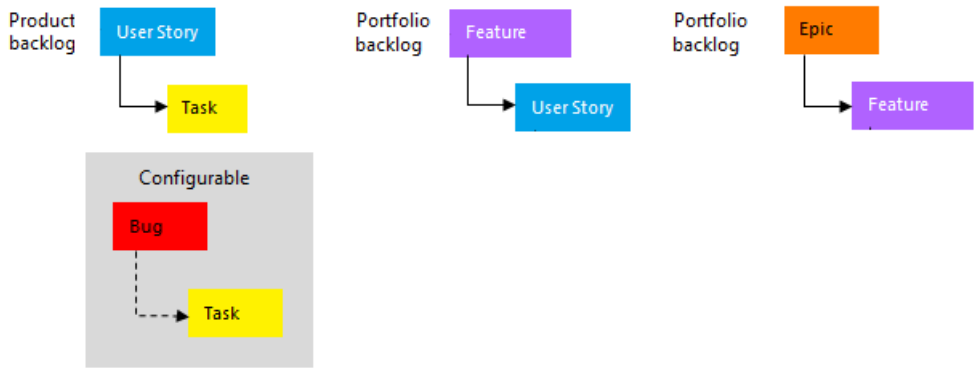
Other supported tasks:

- Add a work item
- Restore work items from the recycle bin

- View work items through a mobile browser

Boards

The two types of Kanban boards, product backlog and portfolio backlogs, provide the quickest means for adding user stories and portfolio work item types. You can also quickly add and update the status of child items within a hierarchy. As shown in the following image for the Agile process, when you add tasks to user stories, users stories to features, or features to epics, you automatically create parent-child links between the work items.



Product backlog board | Features or Epics board | Customize cards

Best tool for:

- Implementing Kanban methods
- Viewing the flow of work from start to completion
- Quickly adding product backlog and portfolio backlog items
- Updating the status of backlog items
- Adding linked child items (checklists, user stories or features)
- Focusing the view based on assignment, tags, or other filter criteria
- Adding, running, and updating inline test cases)

Additional supported tasks:

- Assigning a work item, updating a field displayed on a card
- Monitoring cumulative flow
- Monitoring lead time and cycle time control charts

- Assigning a work item, updating a field displayed on a card
- Monitoring cumulative flow

> **NOTE**
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans.
> If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

Backlogs

You can quickly add and prioritize your product and portfolio backlogs, which list work items either as a flat or hierarchical list. You can also quickly add and reparent child items within a hierarchy.

Product backlog | Portfolio backlogs

Best tool for:

- Managing your product backlog, developing your project plan
- Quickly adding product and portfolio backlog items
- Moving backlog items in priority order
- Creating, viewing, and modifying a hierarchy of backlog items
- Organizing your backlog, linking or mapping backlog items to portfolio backlog items
- Planning a sprint
- Forecasting work
- Emailing a list of backlog items
- Focusing the list based on assignment, tags, or other filter criteria

Additional supported tasks:

- Bulk modifying work items
  - Change work item type
  - Move work item to a different project
  - Assign work items, change the iteration
  - Add or remove tags
  - Delete work items
- Creating a Git branch from one or more work items
- Monitoring team velocity

- Bulk modifying work items
  - Assign work items, change the iteration
  - Delete work items
  - Add or remove tags
- Creating a Git branch from one or more work items
- Restoring work items from the recycle bin
- Monitoring team velocity

Sprint tools

Sprint tools provide teams a focused view of work items they've assigned to a specific sprint. You can add tasks to work items and prioritize your sprint backlog.

Sprint backlog | Taskboard | Capacity

Best set of tools for:

- Implementing Scrum methods
- Adding tasks to backlog items
- Configuring team capacity
- Monitoring and adjusting team capacity
- Updating remaining work, and task status
- Emailing or sharing a sprint plan

Additional supported tasks:

- Monitoring sprint burndown
- Bulk modifying work items

Queries

Queries enable you to filter work items within or across projects for the purposes of listing, updating, or sharing work items.

- Queries | Query operators

Best tool for:

- Listing items to run bulk updates, assign, or reassign
- Listing a tree of parent-child related work item or dependent work items
- Triaging work items (review, set priority, update)
- Creating simple progress and trend charts
- Emailing a list work items

Additional supported tasks:

- Create a chart and add it to a dashboard
- Create a chart to get a count of items or sum a field
- Create a chart that shows a burndown or burnup over time

Plans

When you want to review the schedule of stories or features your teams plan to deliver, use Delivery Plans. Plans show scheduled work items that are assigned to sprints of selected teams against a calendar view.

Best tool for:

- Viewing product or portfolio work items assigned to several teams against a calendar schedule

Additional supported tasks:

- Moving a work item to a different iteration

Office integration tools

> **NOTE**
> Starting with Azure DevOps Server 2019 and Visual Studio 2019, the Team Foundation plug-in for Office is deprecating support for Office Project and Storyboarding with PowerPoint.

Use **Excel** to complete the following tasks:

- Add or update many work items and their fields
- Add or change hierarchical links between work items.

Test tools, test artifacts, and bugs

Testing tools used several work item types—such as test plans, test suites, test cases, and more. You create and manage them from **Test Plans/Test** or using one or more test tools. Several of these tools also support creating bugs.

Test tool

Best tool for:

Test Plans

- Test Plans and Test Suites

- Test Cases
- Test Cases, grid view
- Shared Steps, Shared Parameters
- Delete test artifacts

Additional supported tasks:

- Track test status
- Order manual tests within suites
- Export test plans and test suites
- Assign testers to test cases

Test Runner

- Running tests, adding bugs

Test & Feedback extension

- Exploratory testing, capture feedback in connected mode
- Adding bugs, tasks, and test cases linked to a work item
- Requesting feedback
- Providing feedback

Additional supported tasks:

- Updating existing bugs
- Capture feedback in standalone mode
- Capture screen recordings of your app during testing

Other tools

Tool

Best tool for:

Adhoc search

- Find a specific work item using its ID or a keyword
- Find one or more work items across all projects in a fast, flexible manner
- Run full text search across all work item fields
- Review work items assigned to a specific team member
- Search against specific work item fields to quickly narrow down a list of work items
- Determine what key words will support a managed search.

Work item templates

- Capture templates
- Apply templates to update work items
- Use templates to create work items
- Manage work item templates.

Request and capture feedback

- Request feedback
- Give feedback using Microsoft Feedback Client

Notifications

- Manage personal notifications
- Manage team and project notifications
- Manage organization notifications

Favorites

- Set personal and team favorites

Marketplace extensions

Other tools become available when you install one of the Extensions for Azure DevOps, Boards category.

See also Azure Boards extensions developed by Microsoft.

Related articles

- Set up your Backlogs and Boards
- Navigate in the web portal
- Navigate in Team Explorer
- Why use Azure Boards?

Terms and concepts used when tracking work items in Azure Boards

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Microsoft Agile glossary is a short dictionary of terms used in tracking work using Azure Boards. More terms are defined in the following articles:

- Kanban key concepts
- Sprints and Scrum key concepts
- Work item field index
- Project management and navigation glossary

Agile methods

A family of engineering best processes with a goal of enabling rapid delivery of high-quality software and a business approach that aligns development with customer needs and company goals. In this paradigm, frequent inspection and adaptation are necessary, with team work, self-organization, and accountability all critical to project success.

Agile tools

A suite of web-based tools used to track work and support Agile methodologies. Agile tools support the core Agile methods—Scrum and Kanban—used by software development teams today. Learn more: About Agile tools and Agile project management.

Area path

Area paths are used to group work items by team, product, or feature area. Iteration paths are used to group work into sprints, milestones, or other event-specific or time-related periods. You can use area paths to define a hierarchy of paths. To learn more, see About area and iteration paths.

Bugs

A type of work item that records a potential source of dissatisfaction with the product. The common name of a work item type for tracking code defects. Each team can choose how they want to manage bugs. Some teams like to track bugs along with requirements on the backlog. Other teams like to track bugs as tasks performed in support of a requirement. The bugs then appear on their Taskboard. Learn more: Manage bugs.

Categories

Groups one or more work item types to support flexible reporting, queries, and other functions made available through Agile tools. Categories support the process configuration used by the web portal backlog and taskboard pages. For example, you can add custom work item types to the Requirements category and manage them using the product backlog and Kanban boards. To learn more, see Use categories to group work item types.

Collections

A collection is a container for a number of projects in Azure DevOps. A default collection is created when you sign up with Azure DevOps Services or install Team Foundation Server. Within Azure DevOps Services, a collection corresponds to an organization. For on-premises TFS deployments, you can add and manage collections to specify the logical and physical resources available to the projects within the collection.

Learn more: About projects and scaling your organization, Manage organizations or Manage project collections in Team Foundation Server.

Dashboards

Dashboards are user-configurable interactive signboards that provide real-time information. Dashboards are associated with a team and display configurable widgets to show information. To learn more, see Add and manage dashboards.

Discussion

Area within a work item form that supports adding and reviewing comments made about the work being performed. This way, you capture all comments within the work item rather than maintaining a long email thread. Within the discussion section, you can use the **@mention** control to notify another team member about the discussion. Simply type @ and their name.

Learn more: Work item form controls.

Favorites

Tagging an object as a favorite is a method used to support quick navigation by yourself or other team members. You can tag work item queries and build definitions as personal and team favorites. Other objects you can tag as a favorite for yourself only include code branches, delivery plans, test plans, and teams or projects. To learn more, see Set personal or team favorites.

Fields

Fields support tracking a piece of information about the work to perform. Values you assign to a field are stored in the work tracking data store that you can query and generate charts to view status and trends. Your project contains 100 or more data fields. You update data by modifying the data field within a work item. Each work item is associated with a work item type (WIT), and the data you can track corresponds to the fields assigned to the WIT. For a definition of each predefined field, see Work item field index.

Follow

Tagging specific work items or pull requests to follow them is a method used to receive email updates about changes that are made to them. To learn more, see Follow a work item or pull request.

Global lists

Defines a list of menu items or picklist items that are shared across WITs and projects within a project collection. Global lists help to minimize the work that is required to update lists. You can define global lists within WITs that you upload with your process template. Learn more: Manage global lists for work item types. (Only supported for Hosted XML and On-premises XML process models)

Global workflow

Specifies both work item fields and global lists that multiple projects and types of work items can share. Learn more: Manage global workflow (Only supported for On-premises XML process model).

Hidden types categories

Specifies the set of work item types that you don't want users to create manually. By default this set includes:

- Code Review Request and Code Review Response
- Feedback Request and Feedback Response
- Shared Steps and Shared Parameter
- Test Plan and Test Suite

You can use TFS Team Project Manager, an open-source client available from GitHub to quickly determine which WITs belong to the Hidden Types Category.

Hosted XML process model

The Hosted XML process model provides support for customizing work tracking objects and Agile tools for a project by modifying and importing a process template. This process model is only available for select accounts hosted on the Azure Boards cloud platform. To learn more, see Hosted process model.

Issues or impediments

A type of work item that helps track unplanned activities. Resolving an issue or impediment requires more work beyond what was scheduled based on actual requirements. Using the issue (Agile or CMMI process) or impediment (Scrum process) work item type helps you track and manage these issues until you can resolve and close them. Learn more: Manage issues and impediments.

Inheritance process model

The Inheritance process model provides support for customizing work tracking objects and Agile tools for a project through the user interface. This process model is only available for accounts hosted on the Azure Boards cloud platform. Projects inherit the customizations made to a process. To learn more, see Inheritance process model.

Issue

Agile process: An issue is a type of work item that defines an item that you want to track as it may impact the completion of other work. It's defined for the Agile process and doesn't appear on any backlog or board. See Manage issues and impediments.

Basic process: An issue is a type of work item that defines some work or code defect that needs to be tracked. It's defined for the Basic process and appears on the product backlog and Issues Kanban board.

> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

Iteration paths (aka sprints)

A time period, usually two to three weeks, used to group work items to be completed during that time period. Sprints are used in Scrum methods to support sprint planning, sprint burndown, and other Scrum processes. Iteration paths allow you to group work into sprints, milestones, or other event-specific or time-related period. Learn more: About area and iteration paths.

Kanban board

An interactive, electronic sign board that supports visualization of the flow of work from concept to completion and lean methods. Azure DevOps provides a Kanban board for each product and portfolio backlog. Learn more: Kanban basics and Kanban board features and epics.

Links and link types

Links support defining relationships between work items and other objects—such as commits, branches, pull requests, and more—using different link types. Learn more: Add link to work items, Link work items to support traceability and manage dependencies and Link types reference.

On-premises XML process model

The On-premises XML process model provides support for customizing work tracking objects and Agile tools for a project. With this model, you can update the XML definition of work item types, the process configuration, categories, and more. You can also update the attributes of fields. This process model is only available for on-premises Azure DevOps. To learn more, see On-premises process model.

Pick lists

A picklist specifies an enumerated set of values that appear within a drop-down menu in a work item form. Values also appear in the **Value** column within the query editor. The method you use to customize a picklist varies. It depends on the field and the process model. To learn more, see Customize work.

Plans (also known as delivery plans)

A plan is a configurable view that displays work from multiple teams and projects laid out within a calendar based on each team's iterations. Each row in the view represents the work from a team's product or portfolio backlog. Each card corresponds to a work item, such as user story, feature, or epic. To learn more, see Review team delivery plans.

Portfolio backlog

An interactive list of work items, similar to the product backlog, that supports organizing or grouping work under features, epics, or scenarios. Portfolio backlogs work similarly to product backlogs in that you can prioritize work and view the tree hierarchy of work. Learn more: Define features and epics.

Process

A process defines the building blocks of a work-tracking system. To customize a process, you first create an inherited process from one of the default system processes, Agile, Scrum, or CMMI. All projects that use the process see the changes you make. To learn more, see About process customization and inherited processes.

Process configuration

Specifies the default configuration and functional capabilities that your teams can access using the Agile tools. These web portal tools include the product backlog, sprint backlogs, Kanban board, and taskboard. (Only supported for Hosted XML and On-premises XML process models)

Process model

The work tracking customization method supported by your organization or collection. One of three process models are supported, Inheritance and Hosted XML for Azure Boards and On-premises XML for on-premises

Azure DevOps. Learn more: Customize your work tracking experience

Process template

Specifies an inter-related set of files that contain the XML definitions for tracking work and defining the initial configuration of other functional areas. The system provides three default process templates—Agile, Scrum, or CMMI. You can create a project and then customize it, or customize a process template that you then use to create a project. (Only supported for Hosted XML and On-premises XML process models)

Product backlog

An interactive list of work items that corresponds to a team's project plan or roadmap for what the team plans to deliver. The product backlog supports prioritizing work, forecasting work by sprints, and quickly linking work to portfolio backlog items. You can define your backlog items and then manage their status using the Kanban board.

Each product backlog can be customized by a team. Learn more: Create your backlog.

Product backlog item

A type of work item that defines the applications, requirements, and elements that teams plan to create. Product owners typically define and stack rank product backlog items which are defined with the Scrum process. Learn more: Scrum process work item types and workflow.

Projects

A project, which was previously known as a *team project*, provides a repository for source code. A project provides a place where a group of people can plan, track progress, and collaborate on building software solutions. A project is defined for an Azure DevOps Services organization or within a TFS project collection. You can use it to focus on those objects defined within the project. To learn more, see About projects and scaling your organization.

Queries

Queries are used to find and list work items. Queries support managed searches, which are used to triage work, versus ad-hoc searches, which are used to find a specific work item. Flat-list queries also support status and trend charts. To learn more, see About managed queries.

Remote linking

With remote linking, you can create link relationships between work items in one organization to work items or other objects defined in another organization. Organizations must be managed by the same Azure Active Directory. Learn more: Link work items, Link to a remote work item.

Rollup

Rollup refers to the sum of Remaining Work, Story Points, or other numeric field of child and descendent work items within a hierarchy. To add rollup columns to a product or portfolio backlog, see Display rollup progress or totals.

Rollup refers to the sum of Remaining Work, Story Points, or other numeric field of child and descendent work items within a hierarchy. Azure Boards supports some native rollup features. To learn more, see Rollup of work and other fields.

Sprints (also known as iterations)

A sprint is a time period of usually two to three weeks that's used to group work items to be completed during that time period. Sprints are used in Scrum methods to support sprint planning, sprint burndown, and other Scrum processes. Sprints are defined via iteration paths. To learn more, see About area and iteration paths (aka sprints).

Sprint backlog

An interactive list of work items that have been assigned to the same sprint or iteration path for a team. The sprint backlog supports teams that use Scrum methodologies. Learn more: Sprint planning.

Taskboard

A taskboard is an interactive board of work items that you can use to review and update tasks defined for the sprint backlog. The taskboard supports teams that use Scrum methodologies. To learn more, see Update and monitor your taskboard.

Teams

A team corresponds to a selected set of project members. With teams, organizations can subcategorize work to better focus on all the work they track within a project. Each team gets access to a suite of Agile tools. Teams can use these tools to work autonomously and collaborate with other teams across the enterprise. Each team can configure and customize each tool to meet their work requirements. To learn more, see About teams and Agile tools.

User story

A type of work item that defines the applications, requirements, and elements that teams plan to create. Product owners typically define and stack rank user stories. User story is defined with the Agile process. Learn more: Agile process work item types and workflow.

Widgets

Widgets display information and charts on dashboards. Many of them can be configured. Many widgets display information available from one or more data stores or charts created by the system. To learn more, see Widget catalog.

Work item types (WITs)

A WIT specifies the fields, workflow, and form used to track an item of work. Each WIT is associated with more than 30 system fields and several more type-specific fields. You use work items to plan and track the work required to develop your project. For an overview of predefined WITs provided with the default processes, see Choose a process.

Workflow

A workflow is an integral aspect of a work item. It's defined by its corresponding work item type. The workflow determines the logical progression and regression of work items. For the Agile process, it tracks the status of work as the work progresses from a New or Active state to a Closed or Completed state. For the Basic process, all work item types use the **To Do**, **Doing**, and **Done** states to track workflow status.

The workflow also specifies the values that appear in the State and Reason drop-down menus. To learn more, see Workflow states and state categories.

Related articles

- Refine your backlog
- Kanban best practices
- Scrum best practices.

Choose a process flow or process template to work in Azure Boards

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Anytime you create a project, you must choose a process or process template based on the process model selected for your organization or collection. When choosing a process for your project, it's important to understand the following terms:

- A *process model* refers to the model used to support projects created for an organization (Azure DevOps Services) or project collection (Azure DevOps Server). Only one process model is supported for a project at a time. A comparison of the three process models—Inheritance, On-premises XML, and Hosted XML—is provided in Customize work tracking.
- A *process* defines the building blocks of the work item tracking system and supports the Inheritance process model for Azure Boards. This model supports customization of projects through a WYSIWYG user interface.
- A *process template* defines the building blocks of the work item tracking system and other subsystems you access through Azure DevOps. Process templates are only used with the Hosted XML and On-premises XML process models. You customize projects by modifying and importing process template XML definition files.

> **NOTE**
> For guidance on configuring and customizing your project and teams to support your business needs, review Configuration and customization of Azure Boards.

For details on creating a project using the process of your choice, see Create a project. To learn more about process models, see Customize your work tracking experience.

> **TIP**
> With Azure DevOps Server, you can choose between using the Inherited process model or the On-premises XML process model. For details, see Customize your work tracking experience, Choose the process model for your project collection. To access the latest versions of the default processes/process templates:
>
> - **For Inherited process model**: Open the Process page from organizations settings. To learn more, see Manage processes.
> - **For the On-premises XML process model**:
>   - Install or upgrade to the latest version of Azure DevOps Server
>   - Download the zipped template file using the Process Template Manager. You'll need to use a version of Visual Studio that is at the same version level as TFS. You can install the latest version of Visual Studio Community for free.
> - You can access the latest versions of the default process templates installed on Azure DevOps Server, for example: `%programfiles%/Azure DevOps Server 2020/Tools/Deploy/ProcessTemplateManagerFiles/1033`. For descriptions of each file and folder, see Overview of process template files.

TIP

To access the latest versions of the default process templates:

- Install or upgrade to the latest version of TFS.
- Download the zipped template file using the Process Template Manager. You'll need to use a version of Visual Studio that is at the same version level as TFS. You can install the latest version of Visual Studio Community for free.
- You can access the latest versions of the default process templates installed on TFS 2017 here: `%programfiles%/TFS 15.0/Tools/Deploy/ProcessTemplateManagerFiles/1033` (for TFS 2015, the main folder is TFS 14.0). For descriptions of each file and folder, see Overview of process template files.

The work tracking objects contained within the default processes and process templates—Basic, Agile, CMMI, and Scrum—are the same and summarized below. The Basic process is available from Azure DevOps Server 2019.1 and later versions. For simplicity, they're referred to as a "process."

TIP

To view and manage Inherited process models, see Manage processes.

Choose a Basic, Agile, Scrum, and CMMI process

The default processes differ mainly in the work item types (WITs) they provide for planning and tracking work.

Basic is the most lightweight and is in a selective Preview. Scrum is the next most lightweight. Agile supports many Agile method terms, and CMMI, which stands for Capability Maturity Model Integration, provides the most support for formal processes and change management.

NOTE

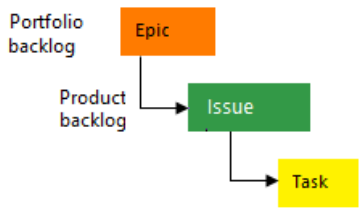
The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.

Choose the process that provides the best fit for your team.

Basic

Choose Basic when your team wants the simplest model that uses Issues, Tasks, and Epics to track work.

Tasks support tracking Remaining Work.

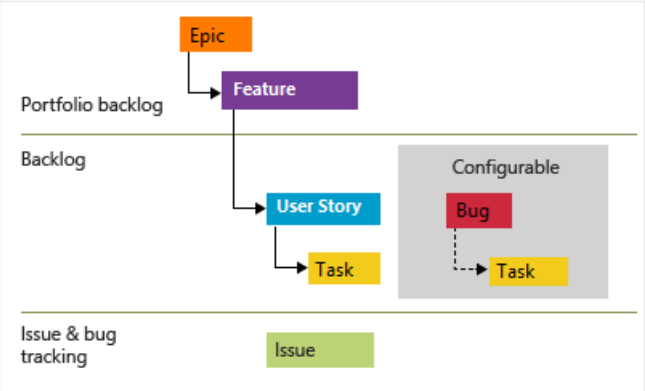


Agile

Choose Agile when your team uses Agile planning methods, including Scrum, and tracks development and test activities separately. This process works great if you want to track user stories and (optionally) bugs on the Kanban board, or track bugs and tasks on the taskboard.

You can learn more about Agile methodologies at the Agile Alliance.

Tasks support tracking Original Estimate, Remaining Work, and Completed Work.

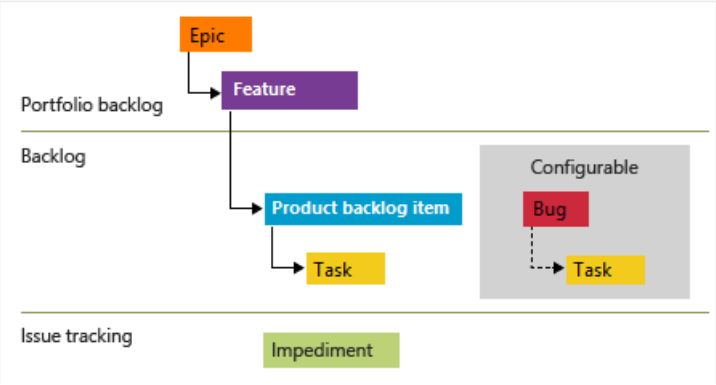


Scrum

Choose Scrum when your team practices Scrum. This process works great if you want to track product backlog items (PBIs) and bugs on the Kanban board, or break down PBIs and bugs into tasks on the taskboard.

This process supports the Scrum methodology as defined by the Scrum organization.

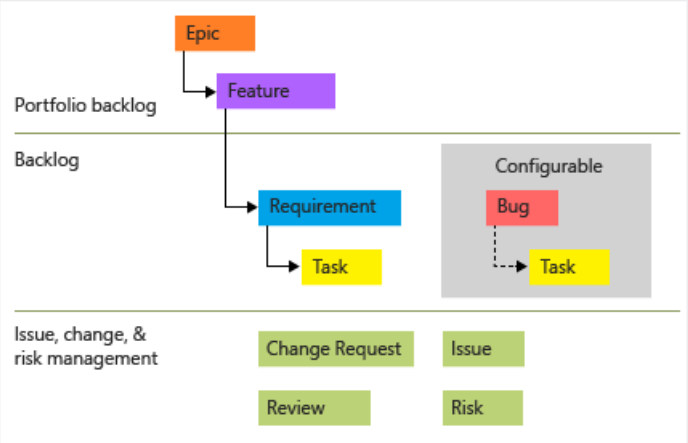
Tasks support tracking remaining work only.



CMMI

Choose CMMI when your team follows more formal project methods that require a framework for process improvement and an auditable record of decisions. With this process, you can track requirements, change requests, risks, and reviews.

This process supports formal change management activities. Tasks support tracking Original Estimate, Remaining Work, and Completed Work.



If you need more than two or three backlog levels, you can add more based on the process model you use:

- **Inheritance**: Customize your backlogs or boards for a process
- **Hosted XML or On-premises XML**: Add portfolio backlogs.

Main distinctions among the default processes

The default processes are designed to meet the needs of most teams. If your team has unusual needs and connects to an on-premises server, you can customize a process and then create the project. Or, you can create a project from a process and then customize the project.

The following table summarizes the main distinctions between the WITs and states used by the four default processes.

Tracking area

Basic

Agile

Scrum

CMMI

Workflow states

- To Do
- Doing
- Done

- New
- Active
- Resolved
- Closed
- Removed

- New
- Approved
- Committed
- Done
- Removed

- Proposed
- Active
- Resolved
- Closed

Product planning (see note 1)

- Issue

- User story
- Bug (optional)

- Product backlog item
- Bug (optional)

- Requirement

- Bug (optional)

Portfolio backlogs (2)

- Epic

- Epic
- Feature

- Epic
- Feature

- Epic
- Feature

Task and sprint planning (3)

- Task

- Task
- Bug (optional)

- Task
- Bug (optional)

- Task
- Bug (optional)

Bug backlog management (1)

- Issue

- Bug

- Bug

- Bug

Issue and risk management

- Issue

- Issue

- Impediment

- Issue
- Risk
- Review

NOTE

1. You can add these WITs from the product backlog or Kanban board. The product backlog shows a single view of the current backlog of work that can be dynamically re-ordered and grouped. Product owners can quickly prioritize work and outline dependencies and relationships.
   Also, each team can configure how they want bugs to show up on their backlogs and boards.
2. With portfolio backlogs you can define a hierarchy of backlogs to understand the scope of work across several teams and see how that work rolls up into broader initiatives. Each team can configure which portfolio backlogs appear for their use.
3. You can define tasks from the sprint backlog and taskboard. With capacity planning, teams can quickly determine if they are over or under capacity for a sprint.

Workflow states, transitions, and reasons

Workflow states support tracking the status of work as it moves from a new state to a closed or a done state. Each workflow consists of a set of states, the valid transitions between the states, and the reasons for transitioning the work item to the selected state.

IMPORTANT

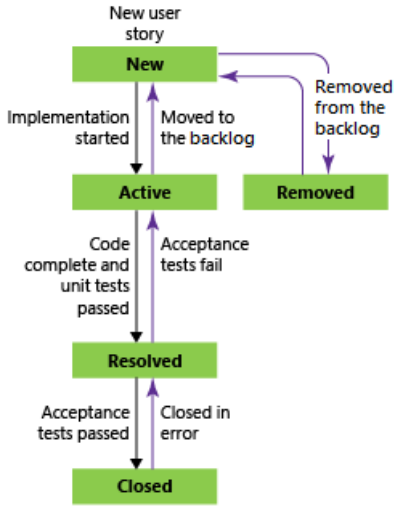
For Azure DevOps Services and Azure DevOps Server 2019, the default workflow transitions support any state to any state transition. You can customize these workflows to restrict some transitions .See Customize work tracking objects to support your team's processes.

Also, you can view the supported workflow transitions for each work item type by installing the State Model Visualization Markeplace extension. This extension adds a new hub under Boards labeled **State Visualizer**. On that page you can choose a work item type and view the workflow state model.

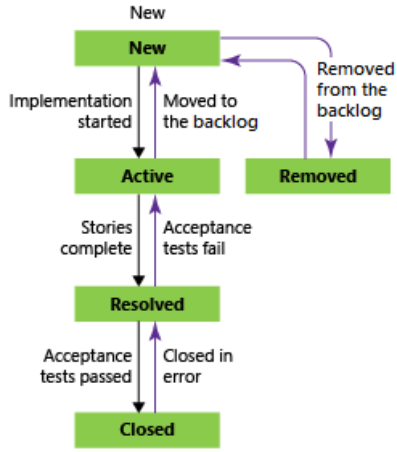
The following diagrams show the typical forward progression of those WITs used to track work and code defects for the three default processes. They also show some of the regressions to former states and transitions to removed states. Each image shows only the default reason associated with the transition.

- Agile process
- Basic process
- Scrum process
- CMMI process

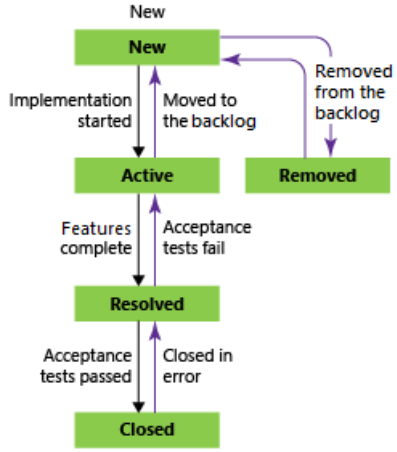
User story



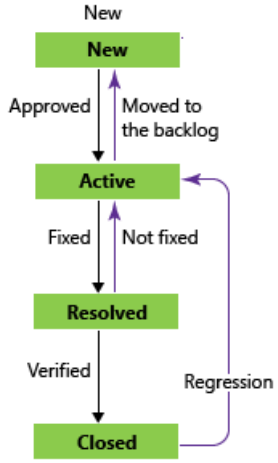
Feature



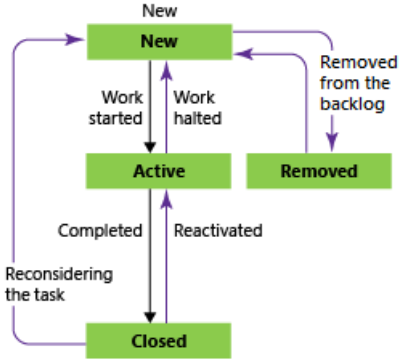
Epic



Bug



Task



Most WITs used by Agile tools, ones that appear on backlogs and boards, support any-to-any transitions. You can update the status of a work item using the Kanban board or the taskboard by dragging it to its corresponding state column.

You can change the workflow to support other states, transitions, and reasons. To learn more, see Customize your work tracking experience.

Removed, Closed, and Done states

When you change the state of a work item to Removed, Closed, or Done, the system responds like this:

- **Closed** or **Done**: Work items in this state don't appear on the portfolio backlog and backlog pages. However, they do appear on the sprint backlog pages, Kanban board, and taskboard. Also, when you change the portfolio backlog view to show backlog items, for example, to view Features to Product Backlog Items, work items in the closed and done state appear.
- **Removed**: Work items in this state don't appear on any backlog or board.

Work items are maintained in a project as long as the project is active. Even if you set them to Closed, Done, or Removed, a record is kept in the data store. You can use a record to create queries or reports.

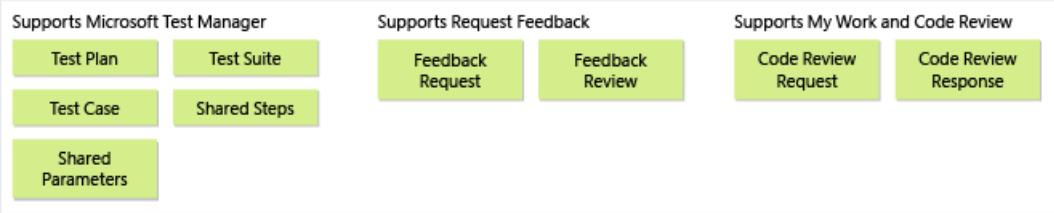
> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than 183 days (about a half a year). You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than a year old. You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

If you need to permanently delete work items, see Remove or delete work items.

Work item types added to all processes

The following WITs are added to all processes except the Basic process.



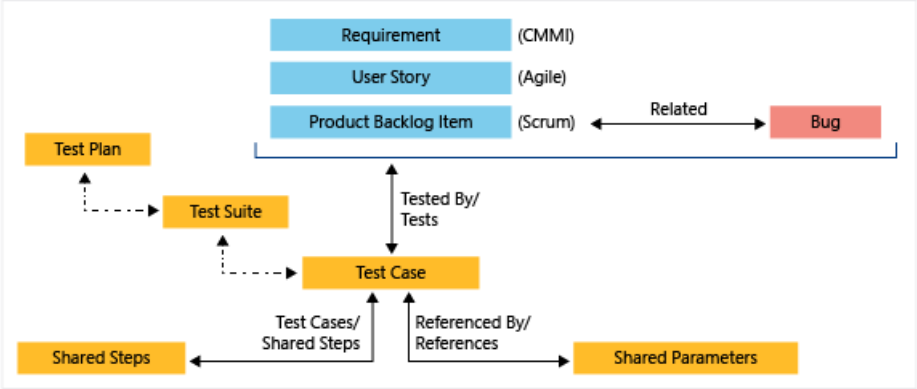
Teams create and work with these types using the corresponding tool:

- Test Plan, Test Suite, Test Case Shared Steps, and Shared Parameters: Microsoft Test Manager.
- Feedback Request and Feedback Response: Request feedback.
- Code Review Request and Code Review Response: My Work (from Team Explorer) and Code Review Request.

Work items from these type definitions aren't meant to be created manually and are then added to the Hidden Types category. Work item types added to the Hidden Types category don't appear in the menus that create new work items.

WITs that support the test experience

WITs that support the test experience and work with Test Manager and the web portal are linked together using the link types shown in the following picture.



From the web portal or Microsoft Test Manager, you can view which test cases are defined for a test suite. And you can view which test suites are defined for a test plan. However, these objects aren't connected to each other through link types. Customize these WITs as you would any other WIT. See Customize work tracking objects to support your team's processes.

If you change the workflow for the test plan and test suite, you might need to update the process configuration as described here. For definitions of each test field, see Query based on build and test integration fields.

Related articles

You can customize a process before or after you create a project that uses the process. The methods you use depend on the process model you use. To learn more, see Customize your work tracking experience.

- Upload/download process templates
- Changes made to process templates
- Configure features after an Azure DevOps Server upgrade

If you have more questions, see Azure DevOps support page.

Track user stories, issues, bugs, and other work items in Azure Boards

10/4/2022 • 15 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Track the features and requirements you're developing, code defects or bugs, and other details using work items. Work items are similar to GitHub issues, but offer different types to track various types of information.

If you're just getting started, read the information provided in this article. To jump right in and start tracking work on a Kanban board, see Plan and track work. For a quick reference to various work item tasks and key concepts, see Work item quick reference.

Plan and track work

Use work items to track anything you need to track. Each work item represents an object stored in the work item data store. Each work item is based on a work item type. And work item types have a unique identifier within an organization or project collection. The available work item types depend on the process you used when creating your project (Agile, Basic, Scrum, or CMMI).

Use work items to track anything you need to track. Each work item represents an object stored in the work item data store. Each work item is based on a work item type. And work item types have a unique identifier within an organization or project collection. The available work item types depend on the process you used when your project was created (Agile, Scrum, or CMMI).

Tasks you can do using work items

- Use different work item types (WITs) to track different types of information. Specific tools include:
  - Add backlog items, such as Issues (Basic process), User Stories (Agile), Product Backlog Items (Scrum), Requirements (CMMI)
  - Define Features and Epics
  - Define, triage, and manage Bugs
  - Add Tasks to backlog items
- Update the work item form to add information, update status, reassign another project member or sprint, link work items, attach files, and add comments
- Specify how bugs should be tracked, either as requirements or as tasks
- Add and modify work items using the web portal and other supported clients
- Assign a work item to one and only one project member
- Assign work items to a sprint via the iteration path
- Link work items to other work items or Azure DevOps objects
- Run impromptu search or queries to find or list work items
- Capture and apply work item templates to quickly fill in work item field
- Add and customize work item types
- Modify work items

Required permissions and access

As a member added to the Contributors group of a project, you can use most features provided under **Boards** or **Work**. Users with Basic access have full access to all features. Users with Stakeholder access are limited to

certain features. For details, see Stakeholder access quick reference.

To learn more about permissions and access, see Permissions and access for work tracking and Stakeholder access quick reference.

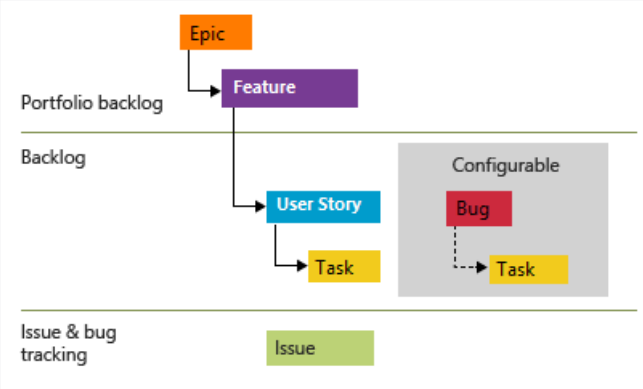
To add users to a project, see Add users to a project or team.

Work item types (WITs)

To track different types of work, different WITs are defined. The work item types available are based on the process used when your project was created—Agile, Basic, Scrum, or CMMI—as illustrated in the following images. The items in your backlog might be called user stories (agile), issues (Basic), product backlog items (Scrum), or requirements (CMMI). All four types are similar: they describe the customer value of the work and the work to do.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

Each work item type belongs to a category. Categories are used to group work item types and determine which types appear on backlogs and boards.

CATEGORY	WORK ITEM TYPE	CONTROLS BACKLOGS/BOARDS
Epic	Epic	Epic portfolio backlogs and boards
Feature	Feature	Feature portfolio backlogs and boards
Requirement	User Story (Agile) Issue (Basic) Product Backlog Item (Scrum) Requirement (CMMI)	Product backlogs and boards and Sprints backlog
Task	Task	Sprints Taskboards
Bug	Bug	Dependent on how bugs are tracked

Along with the work items types that appear on backlogs and boards, there are work item types that track testing, reviews, and feedback. These types, listed in the following table by category, are available for all processes.

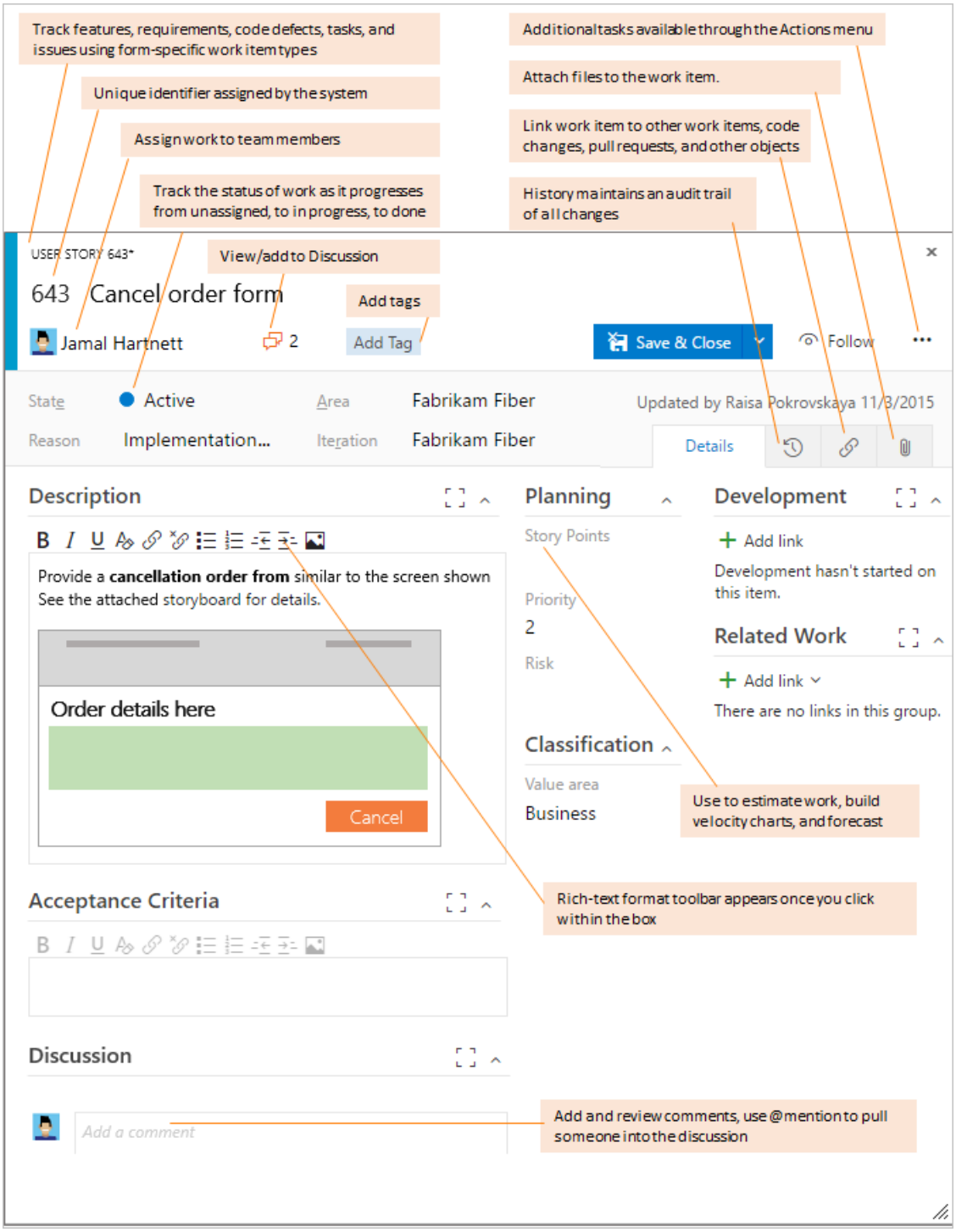
Category	Work item type	Used to track specified types of work
Code Review Request	Code Review Request	Tracks a code review request against code maintained in a Team Foundation version control (TFVC) repository. To learn more, see Day in the life of a Developer: Suspend work, fix a bug, and conduct a code review.
Code Review Response	Code Review Response	A code review response is created for each person who's requested to provide review comments.
Feedback Request		Feedback requests track requests for feedback generated through the feedback request form. See Get feedback.
Feedback Response	Feedback Response	A feedback response is created for each person and for each item for which feedback is provided through the Microsoft Feedback Client. See Get feedback.
Shared Step	Shared Step	Shared steps are used to repeat tests with different data.
Shared Parameter		Shared Parameters specify different data and parameters for running manual test cases. See Repeat a test with different data.
Test Case	Test Case	Each test case defines a manual test.
Test Plan	Test Plan	Test plan group test suites and individual test cases together. Test plans include static test suites, requirement-based suites, and query-based suites.To learn more, see Create test plans and test suites.
Test Suite	Test Suite	Test suites group test cases into separate testing scenarios within a single test plan. Grouping test cases makes it easier to see which scenarios are complete. See Create test plans and test suites.

Ideally, work items are only created from the specific tool designed to support their usage. So, to prevent users from creating work items manually, there's a Hidden Types category. All of the categories listed in the previous table are added to the Hidden Types category except for the Test Case category.

Work item form

Each work item supports tracking data contained in work item fields. Also, it captures changes as updates are made within the **History** field and comments made in the **Discussion** section. To learn more about each field, see Work item field index.

Each form contains multiple controls, as shown below and described in Work item form controls.



Track work in the web portal

You can add and update work items from the web portal. To track work using other clients, see Best tools for

adding, updating, and linking work items.

Web portal and clients that support tracking work items

You can add and update work items from the web portal and various clients. For an overview of all clients that connect to your project, see Tools and clients that connect to Azure DevOps.

Web portal

Use the web portal to accomplish the following tasks.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.
- **Backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
- **Sprints**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others, performing bulk updates, or import/export operations.
- **Delivery Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.
- **Backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
- **Sprints**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others, performing bulk updates, or import/export operations.
- **Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view. Requires installation of the Delivery Plans extension.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.
- **Backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
- **Sprints**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others, performing bulk updates, or export operations.
- **Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view. Requires installation of the Delivery Plans extension.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Backlogs**: Which provide access to:
  - **Product** and **Portfolio backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
  - **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.

    - **Sprint backlogs**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others or performing bulk updates.
- **Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view. Requires installation of the Delivery Plans extension.

Track bugs as requirements or tasks

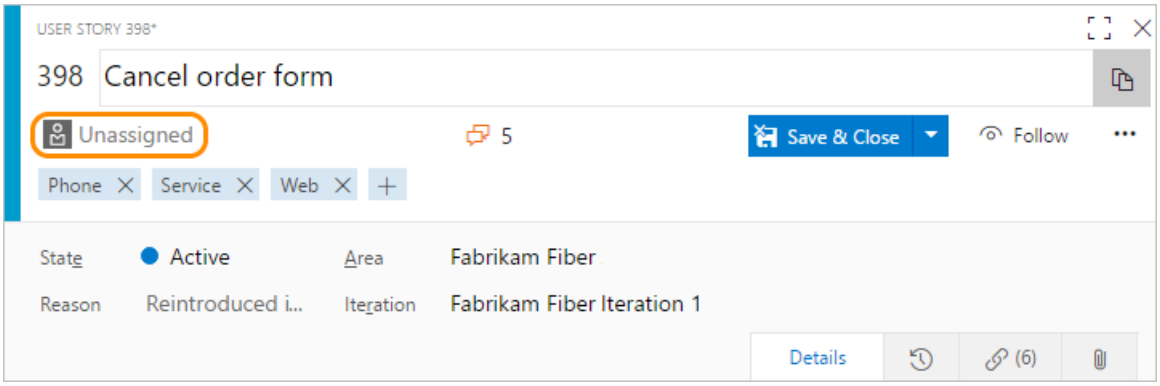
Many Scrum teams treat bugs the same as any backlog item or user story. Others see bugs as work that belongs to implementing a story, and then treat them as a task. Bugs, like product backlog items (PBIs) and user stories, represent work that needs doing. So, should you track your bugs along with other items in the product backlog items? Or, should you track your bugs as tasks linked to those backlog items? How does your team estimate work?

Based on how your team answers these questions, they can choose how they want to track bugs from one of these three choices. To change the team setting, see Show bugs on backlogs and boards.

For an overview of all team settings, see Manage teams and configure team tools.

Assign work items to a project member

You can only assign a work item to one person at a time. The **Assigned To** field is a person-name field designed to hold a user identity recognizable by the system. Within the work item form, choose the **Assigned To** field to select a project member. Or, you can begin typing the name of a project member to quickly focus your search to a select few.



Anyone who has write access to a project can assign work items, including users with Basic and Stakeholder access.

Note the following:

- You can assign a work item only to users that have been added to a project or team
- You can assign a work item to one and only one user at a time. If work is split across two or more users, consider creating separate work items that you'll assign to each user responsible for the work to complete
- Over time, the drop-down menu of person-name fields will display the names you have most recently selected
- Some drop-down menus that support assignment from a team backlog or board are automatically limited to users assigned to the team
- The system shows the display name and adds the user name when required to disambiguate identical display names

- You can assign several work items at once from the backlog or query results, see Bulk modify work items for details.

Integration with Azure Active Directory

When your system is configured with Azure Active Directory (Azure AD), then the system will synchronize person-name fields with these directories. Person-name fields include Activated By, Assigned To, Closed By, Created By, and Resolved By.

You can grant access to a project by adding security groups that you created in Azure AD or by adding accounts to existing or custom groups defined from the collection setting **Security** pages. For more information, see Add or delete users using Azure Active Directory.

Integration with Active Directory

When Azure DevOps Server is configured with Active Directory (AD), Azure DevOps synchronizes person-name fields with these directories. Person-name fields include Activated By, Assigned To, Closed By, Created By, and Resolved By.

You can grant access to a project by adding security groups you create in AD or by adding accounts to existing or custom groups defined in the collection setting **Security** pages. To learn more, see Set up groups for use in Azure DevOps Server deployments.

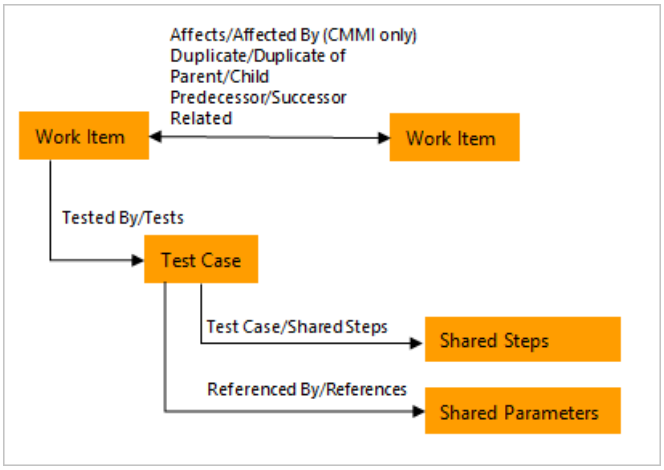
> **NOTE**
> To minimize the list of names that appear in the drop-down menus of person-name fields, you can scope the field to only those groups that you want to appear in the menu. You do this by adding one or more of the following child elements to the **FIELD** definition in the work item type definition: **ALLOWEDVALUES**, **PROHIBITEDVALUES**, and **VALIDUSER**. For details, see Define pick lists.

Assign work items to a sprint

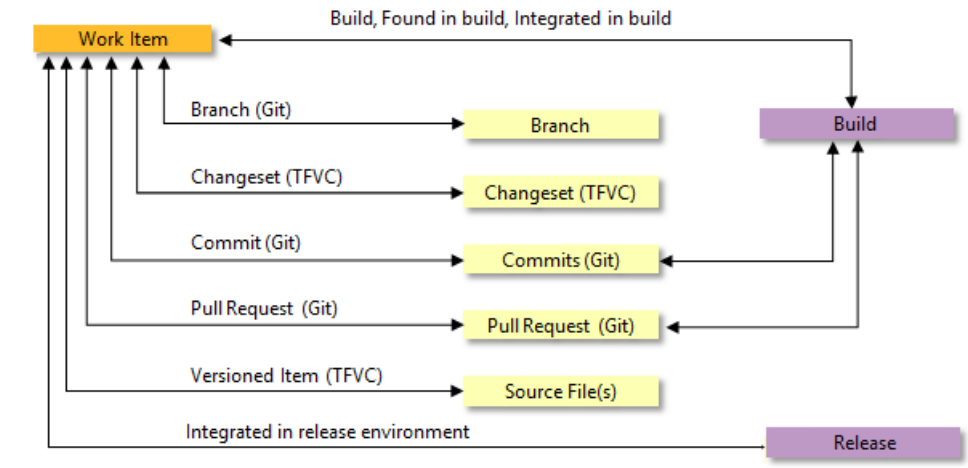
To schedule work items to be worked on during a specific time period, you assign the **Iteration Path**. First, you define the Iteration Paths for use in the project, and then each team selects the Iteration Paths that they'll use. To learn more, see Assign work to sprints.

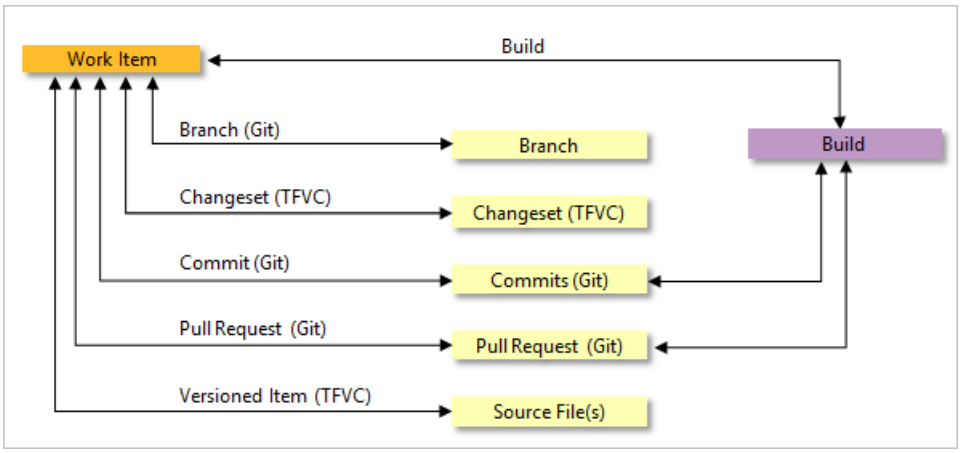
Link work items to other work or objects

As shown in the following image, you can link work items to other work items. Depending on the work item type, the link type differs. Also, special link types support a hierarchy of work items using parent-child links and predecessor-successor links to track dependencies.



Also, several tools support linking to other objects, such as builds, releases, commits, pull requests, and more. The following image demonstrates.





For a complete list of link types and supported features, see Linking, traceability, and managing dependencies and Link type reference.

Find or list work items

Use the search box to do an impromptu search for finding specific work items based on select field criteria. Or, create a query to do a managed search, which lists work items based on your query criteria. With managed searches, you can do other tasks, such as a work items triage, creating a trend or status chart, adding to the dashboard, and more.

To learn more, see these articles:

- About managed queries
- View, run, or email a query
- About managed queries
- Work item query charts

Use work item templates to quickly fill in forms

With work item templates, you can quickly create work items that have pre-populated values for your team's commonly used fields. For example, create a task template that sets the area path, iteration path, and discipline or activity whenever you use it to create a task.

To learn more, see Use templates to add and update work items.

Once you have a template defined, you can share it via email or a dashboard.

Customize a work item type

You can add or modify the fields contained within a work item type or add a custom work item type. To learn more, see Customize an inheritance process.

You can add or modify the fields contained within a work item type or add a custom work item type. To learn more, see:

- For project collections that use the Inheritance process model: Customize an inheritance process.
- For project collections that use the On-premises XML process model: Customize the On-premises XML process model.

You can add or modify the fields contained within a work item type or add a custom WIT. To learn more, see Customize the On-premises XML process model.

Try this next

Add a work item

Related articles

- Web portal navigation
- Backlogs, portfolios, and Agile project management
- About Kanban and Agile project management
- Keyboard shortcuts
- Agile, Scrum, and CMMI processes
- Work item field index
- Use categories to group work item types

About area and iteration (sprint) paths

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

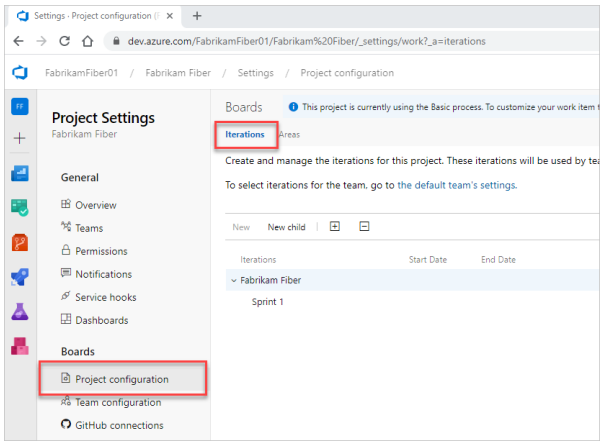
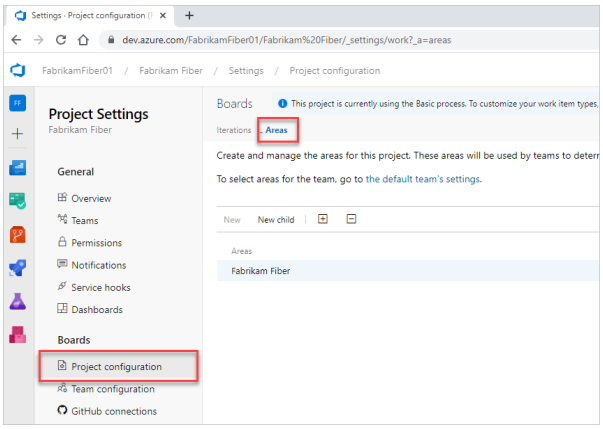
Area paths allow you to group work items by team, product, or feature area. Iteration paths allow you to group work into sprints, milestones, or other event-specific or time-related period. Both these fields allow you to define a hierarchy of paths.

You define area and iteration paths for a project. Teams can then choose which paths are used to support their backlog and other Agile tools. To understand how Agile tools use area and iteration paths, see Agile tools that rely on areas and iterations.

> **NOTE**
> Area paths and iteration paths are also referred to as *Classification Nodes*. You can manage them programmatically via the Classification Nodes (REST API) or the Azure DevOps CLI command az boards iteration.

> **NOTE**
> Area paths and iteration paths are also referred to as *Classification Nodes*. You can manage them programmatically via the Classification Nodes (REST API).

The areas and iterations you see depend on the process you used to create your project. Here we show the defaults defined for the Scrum process. No dates are set. You set dates to correspond to your sprint or release schedules.

ITERATIONS	AREAS
	

Define and assign area paths

If you're new to managing projects and teams, the most straight forward sequence for configuring your project and teams is as follows.

1. Determine the number and names of **Area Paths** that you want to support to categorize your work. At a minimum, add one area path for each team that you define.

2. Determine the number and names of teams you want to support. For guidance, review About teams and Agile tools.
3. Open **Project settings** > **Project configuration** and define the area paths to support steps 1 and 2 at the project level. Follow the steps provided later in this article: Open Project Settings, Project configuration and Add area paths.
4. Define the teams you need to support step 2. For guidance, see Add a team, move from one default team to several teams.
5. Open the team configuration and assign the default and additional area path(s) to each team. Follow the steps provided later in this article: Open team settings and Set team default area path(s).
6. Assign the area path of work items to an area path you defined. Use bulk modify to modify several work items at once.

> **NOTE**
> Organizations are limited to defining a maximum of 10,000 **Area Paths**, and assigning a maximum of 300 **Area Paths** to a single team. To learn more, see Work tracking, process, and project limits.

> **NOTE**
> While you can assign the same **Area Path** to more than one team, this can cause problems if two teams claim ownership over the same set of work items. To learn more, see About boards and Kanban, Limitations of multi-team Kanban board views.

As needed, you can do the following actions at any time:

- Add additional child nodes
- Rename an area path (except the root area path)
- Move a child node under another node
- Delete a child node
- Rename a team
- Change the area path assignments made to a team

How many areas should a team define?

You add areas to support your team's trace-ability and security requirements. Use areas to represent logical or physical components, and then create child areas to represent specific features.

Add areas when you have these requirements:

- Filter queries based on a product or feature area
- Organize or group work items by team or subteams
- Restrict access to work items based on their area.

Each team can create a hierarchy of areas under which the team can organize their backlog items, user stories, requirements, tasks, and bugs.

Avoid creating an area structure that is too complex. You can create areas to partition permissions on work items, but complex trees require significant overhead for permission management. You might find that it's too much work to duplicate the structure and permissions in other projects.

Define and assign iteration paths

Use the following guidance to configure **Iteration Paths** for your project and teams:

1. First, define the area paths and teams following the guidance provided in Define area paths and assign to a team.
2. Determine the length of the iteration you want to support. Recommended practice is to have all teams use the same sprint cadence.
3. Determine if you want a flat structure or hierarchy of sprints and releases.
4. Open **Project settings > Project configuration** and define the iteration paths to support steps 2 and 3 at the project level. Follow the steps provided later in this article: Open Project Settings, Project configuration and Add iterations and set iteration dates.
5. Open the team configuration and assign the default, backlog, and additional iteration path(s) to each team. Follow the steps provided later in this article: Open team settings and Set team default iteration path(s).
6. Each team should assign an iteration path to their work items that falls under the **Backlog iteration** path. Those work items will then show up on their product backlogs and boards. Use bulk modify to modify several work items at once. See also Assign backlog items to a sprint.

> **NOTE**
> Organizations are limited to defining a maximum of 10,000 **Iteration Paths**, and assigning a maximum of 300 **Iteration Paths** to a single team. To learn more, see Work tracking, process, and project limits.

As needed, you can do the following actions at any time:

- Add additional child iteration nodes
- Rename an iteration path (except the root path)
- Move a child iteration path under another node
- Delete a child iteration path
- Change the default and selected iteration paths assigned to a team

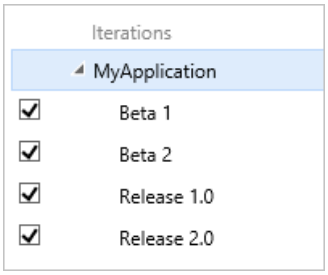
How many iterations should a team define?

You define as many child iterations as you need to reflect your project lifecycle. These paths represent a series of events, such as sprints, pre-beta and beta results, and other release milestones. A team usually leaves work items assigned to the team's default iteration if they aren't yet scheduled for work or for a release.

Add iterations to support these requirements:

- Define sprints your Scrum teams use to plan and execute their sprints
- Set up more complex multi-release and sprint cycles
- Filter queries based on sprints, milestones, or cycle time for your project
- Support future work that you're not ready to assign to a target release cycle.

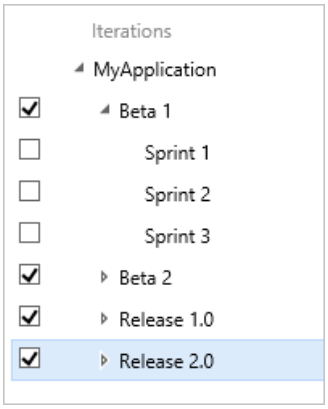
In the following example, Beta 1, Beta 2, Release 1.0, and Release 2.0 are defined for the MyApplication project.



As you create the backlog of product features and tasks, assign them to milestones. Assign the features and task by which you expect the team to finish. As your needs change, you can add events under each major milestone

that reflect how your team schedules and manages its work.

As the following example shows, the Beta 1 iteration now contains three child nodes, one for each sprint in the Beta 1 time period.



Iterations don't enforce any rules. For example, you can assign a task to an iteration but not close or complete it during that iteration. At the end of an iteration, you should find all work items that remain active or open for that iteration and take appropriate action. You can, for example, move them to a different iteration or return them to the backlog.

Naming restrictions

The **Area Path** and **Iteration Path** fields, data type=TreePath, consist of multiple node items separated by the backslash (\) character. Minimize the names of nodes and make sure you conform to the following restrictions when you're adding child nodes.

Restriction type

Restriction

Node length

- Must not contain more than 255 characters

Reserved names

- Must not consist only of a period (.) or two periods (..)
- Must not be a system-reserved name such as PRN, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, COM10, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, NUL, CON, or AUX For more information about reserved names, see File Names, Paths, and Namespaces.

Special characters for nodes

- Must not contain Unicode control characters
- Must not contain any one of the following characters:
  `\ / $ ? * : " & > < # % | +`
- Must not contain characters prohibited by the local file system. For more information about Windows character restrictions, see Naming Files, Paths, and Namespaces.

Path length

- Must not contain more than 4,000 Unicode characters

Path hierarchy depth

- Must be fewer than 14 levels deep

Supported field rules

You can specify only a small subset of rules, such as `HELPTEXT` and `READONLY` to System.XXX fields.

Related articles

As you can see, areas and iterations play a major role in supporting Agile tools and managing work items. You can learn more about working with these fields from the following articles.

- Define area paths and assign to a team
- Define iteration (sprint) paths and configure team iterations
- Agile tools and sprint definitions
- Query by date or current iteration

How workflow category states are used in Azure Boards backlogs and boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

All workflows consist of states, transitions, and reasons. Workflows are defined for a work item type. A transition supports forward and backward movement among two states. When you add a custom state, the system automatically adds transitions from the custom state to all other inherited states (except for Removed).

Each state belongs to a state category (previously referred to as a metastate). State categories support the Agile tool backlog and board views.

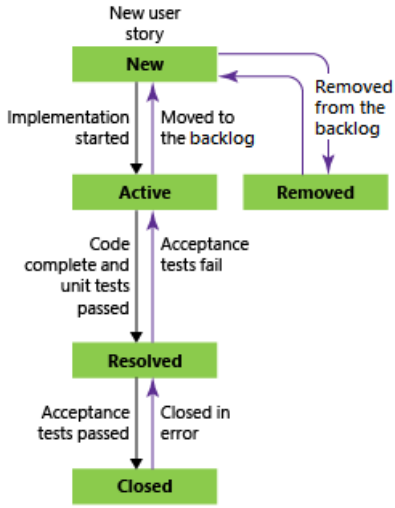
Workflow states

Workflow states define how a work item progresses from its creation to closure. The four main states that are defined for the User Story (Agile process) describe a user story's progression. The workflow states are New, Active, Resolved, and Closed. (The Removed state supports removing a work item from appearing on the backlog; to learn more, see Move, change, or delete work items.)

The natural progressions and regressions for the work item types—user story (Agile), issue (Basic) product backlog item (SCrum), and requirement (CMMI)—are as shown.

- Agile process
- Basic process
- Scrum process
- CMMI process

Workflow states: User Story, Agile process



Category states

Category states determine how Agile planning tools and select dashboard widgets treat each workflow state. The state categories used by the backlogs, boards and widgets are *Proposed*, *In Progress*, *Resolved*, and *Complete*.

Here's how the default, inherited states map to the category states for the four system processes, including Test Plan work item types. The workflow states for Test Case, Test Design, and Test Suite are the same across all four system processes.

- Agile process
- Basic process
- Scrum process
- CMMI process

Categories

Work tracking

Test tracking

Proposed: Assigned to states associated with newly added work items so that they appear on the backlog. The first column on the Kanban boards and Taskboards map to a Proposed state category.

New

Design (Test Case)

In Progress: Assigned to states that represent active work. Work items assigned to states mapped to this category appear in the backlog (unless you choose to hide them) and make up the middle columns on Kanban boards.

Active (Bug, Epic, Feature, User Story)

Active (Test Plan) In Planning (Test Suite) In Progress (Test Suite) Ready (Test Case)

Resolved: Assigned to states that represent a solution has been implemented, but aren't yet verified. Generally these states apply to bugs. Work items in a *Resolved* category state appear on the backlog by default. The Agile tools treat the *Resolved* category state exactly the same as the *In Progress* category state.

Resolved (Bug)

n/a

Completed: Assigned to states that represent work that has finished. Work items whose state is in this category don't appear on the backlog and do appear in the last column of the Kanban board. You can't modify states in this category nor can you add states to this category.

Closed (Bug, Epic, Feature, User Story)

Closed (Test Case) Completed (Test Suite) Inactive (Test Plan)

Removed: Assigned to the Removed state. Work items in a state mapped to the Removed category are hidden from the backlog and board experiences.

Removed (Epic, Feature, User Story)

n/a

> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than 183 days (about a half a year). You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than a year old. You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

Activated By/Date and Resolved By/Date fields

The system updates these fields—**Activated By**, **Activated Date**, **Resolved By**, and **Resolved Date**—when a change occurs based on corresponding workflow category states. When the workflow state changes to a *Proposed* state category, **Activated By** and **Activated Date** are updated. When the workflow state changes to a *Resolved* state category, **Resolved By** and **Resolved Date** are updated.

To learn more how workflow states map to state categories, see How workflow states and state categories are used in Backlogs and Boards.

> **NOTE**
> The logic governing the fields described here applies to Azure DevOps Services, Azure DevOps Server 2020.1 update, and later versions.

Because these fields reference the workflow state categories, custom workflow states that you add are referenced when updating the fields. To learn more about customization, see Customize the workflow for a process.

Additional notes:

- The fields get updated anytime a work item moves from any category state other than that being set. For example, if you update a work item from *New* to *Fixed*, the **Resolved By/Resolved Date** fields are updated. However, if you update from *Fixed* and *Ready for Testing*—which are in the same category state—the **Resolved By/Resolved Date** fields aren't updated.
- When you transition backwards, such as going from a *Resolved* to an *Active* state, the system clears the values for **Resolved By/Resolved Date** fields. If you got from *Active* to *New*, the system clears the values for **Activated By/Activated Date** fields.
- Don't manually change the values for these fields. They are system fields that are governed by system rules. Any value you attempt to set will get over written.

When to add a State versus a Kanban column

Use both States and Kanban columns to track the status of work. Workflow states are shared across a project while Kanban columns are shared within a team. Only project collection admins can add custom states, while team admins can add Kanban columns.

Add custom states when you want all teams to track the status according to the business workflow adopted by the organization. By customizing the process, you automatically customize the projects and work item types that reference that process.

Adding custom states to support workflow states that multiple teams want to track, helps avoid the resulting confusion of different teams creating queries based on a Kanban column. Because each team can customize the Kanban board columns and swimlanes, the values assigned to work items that appear on different boards might not be the same. The primary workaround for this issue is to maintain single ownership of work items by team area path. Another workaround is to formalize the columns by adding custom states that can be shared across teams.

Auto completion of work items with pull requests

When you link a work item to a pull request (PR), you can automatically complete those work items when you complete the PR. To learn how, see Auto complete work items with pull requests.

Related articles

Inheritance process model

- Customize a workflow for a process
- Apply rules to workflow states (Inheritance process)
- Rules and rule evaluation
- Sample custom rule scenarios

On-premises XML process model

- Change the workflow for a work item type
- ProcessConfiguration XML element reference
- Customize your work tracking experience
- Rules and rule evaluation
- Sample custom rule scenarios

Dashboard widgets

- Lead Time and Cycle Time control charts (widgets)

Use links to view dependencies and track related work

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

By linking work items to other work items, you can track related work, view a hierarchy of work, view dependencies, and more. By linking work items to devops and other objects, you support an auto trail of changes and enable quick navigation to work items and linked objects.

Different link types are used to link to the various objects. For example, you can use *Parent/Child* links to support a hierarchical tree structure of work items. The *Commit* and *Branch* link types support links between work items and Git commits and branches, respectively.

Link work items to support the following goals:

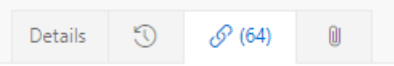
- Track dependencies, related items, and work hierarchies
- Track which work items are tested by test cases and test results
- Support an audit trail of code changes and the work items they support
- Support end-to-end traceability
- Share information by linking work items to a network share, storyboard, or document.

This article describes the link types available for your use. You can link objects from the web portal or Visual Studio Team Explorer. For details on linking work items and deleting links, see Add link to work items.

> **TIP**
> You can set up automatic linking and other settings that link work items to Git commits, pull requests, builds, and more. To learn how, see the following resources:
>
> - Configure repositories and branches to integrate with work tracking
> - Configure pipelines to support work tracking
> - Drive Git development from a work item
> - Link and view work items to builds and deployments.

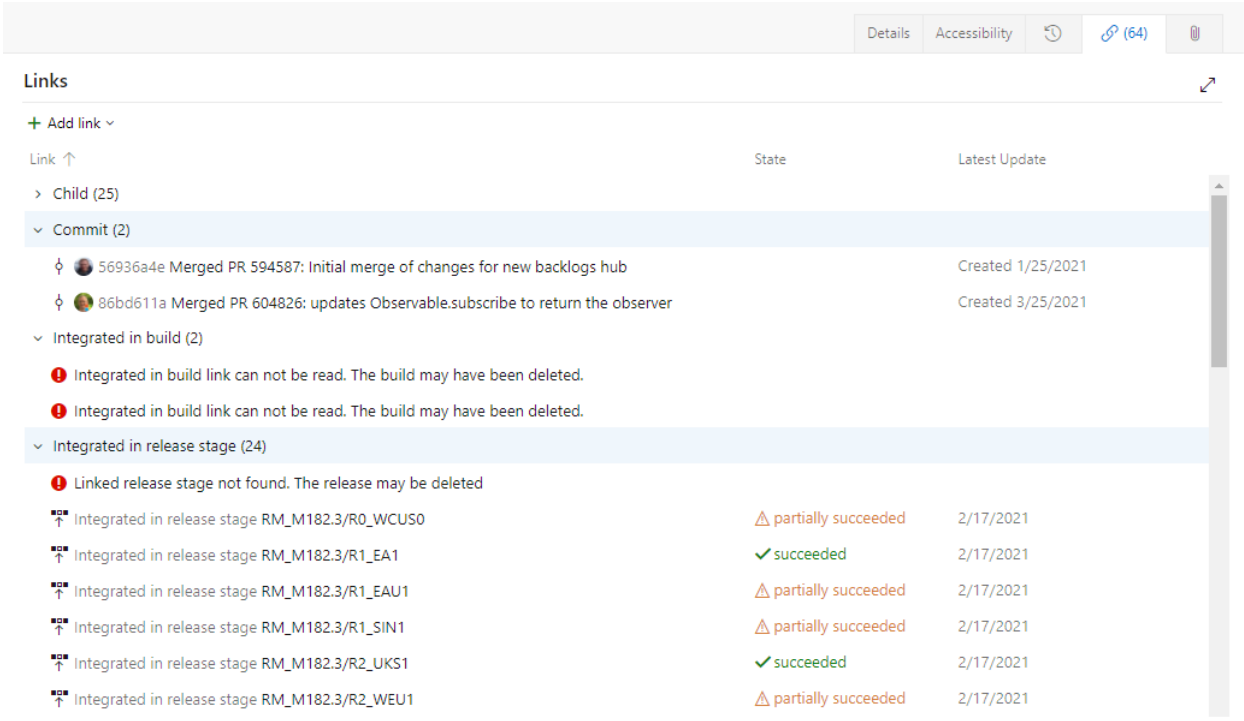
View list of linked objects

To view the list of all objects linked to a work item, open the work item and choose the **Links** tab. The links tab indicates the count of all linked objects.



Linked objects are grouped under their link type, with a count within each group. You can expand or collapse each group, and sort within each group by **State**, **Latest Update**, or **Comment** by choosing the corresponding column title.

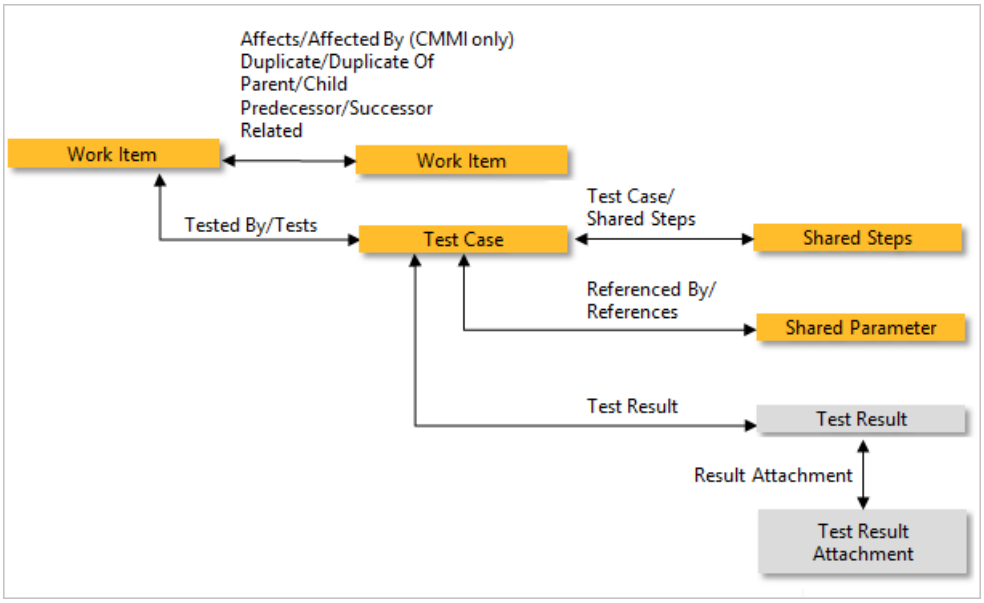
For example, the following **Links** tab shows a portion of the 64 linked objects for a work item.



Links prefaced with the red exclamation mark indicate that the build, release, or other object has been deleted. This is usually due to retention policies which automatically delete these objects after a certain time period has passed.

Work items linked to work items

There are several system link types used to link work items to each other: two tree topology, one dependency topology, and one network. Tree topology links support nested hierarchies, tree queries, and several reports. Dependent links support tracking tasks that must be completed before others can be started. And, the **Related** link type supports connecting work items that are at the same level.



All two-way link types are characterized by a *Forward* and *Reverse* name, such as Parent/Child and Duplicate/Duplicate Of. When you link using one of these names, the linked work item is updated to include a link with the corresponding link type. For example, if you add a Parent link to a work item, the linked work item contains a Child link.

As a quick reference guide, use the following link types as indicated:

- Use the **Duplicate** link type when two work items have been created that essentially capture the same information; close one of the work items and keep the other one active
- Use the **Parent/Child** link types when you want to break down work items into smaller items—for example, break down features into stories, or stories into tasks
- Use **Predecessor-Successor** link types when you want to track tasks that must be completed before others can be started; this link type is most often used when you plan work using Project
- Use the **Related** link type when the work items being linked are at the same level—such as two user stories that define features that overlap one another—or to link work items that are defined in different projects or managed by different teams.

For guidance on choosing link types, review the Link type reference in the related notes section.

You can create links from within a work item form, from a work item that appears in a list of query results, in Microsoft Excel, or in Microsoft Project. You can also use any of the client programs for Team Foundation, such as Team Explorer and the web portal, to create links or attach files.

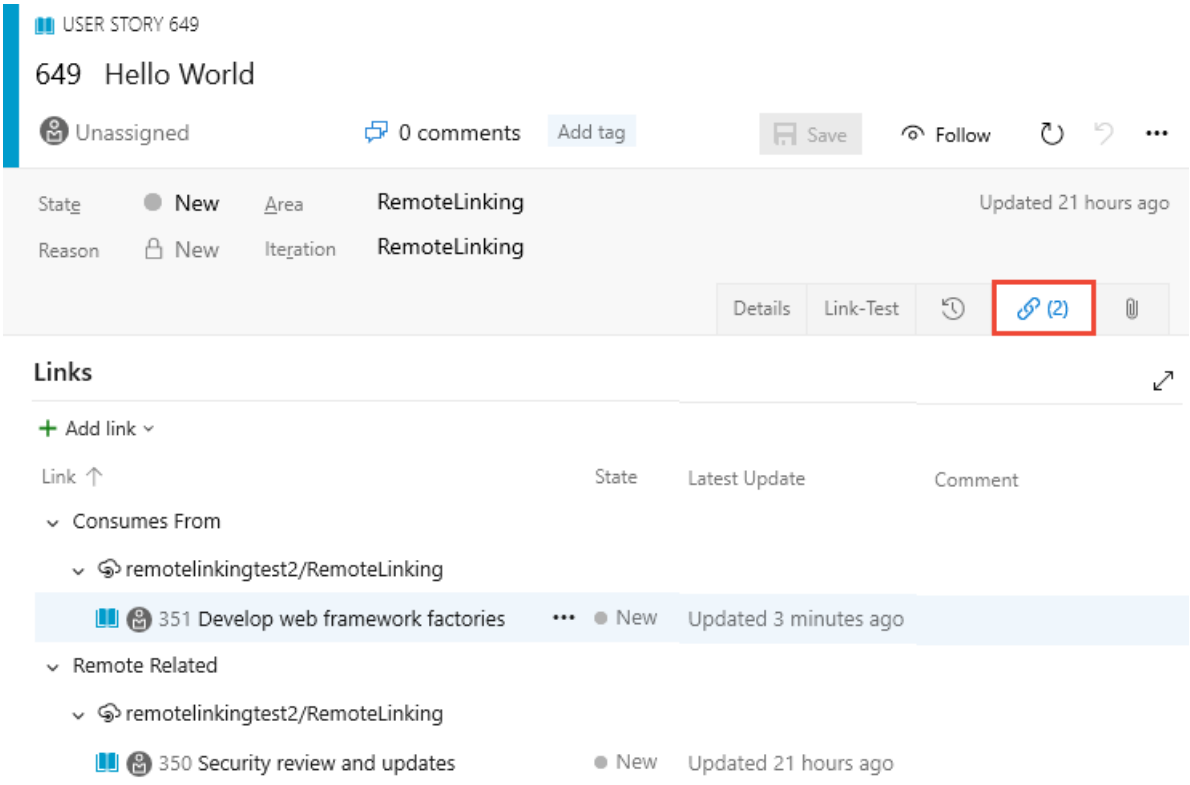
Also, you can use the context menu in the web portal or Team Explorer.

> **NOTE**
> For each work item, you can add a maximum of 1000 links to other work items.

Cross-organization work item linking

Organizations that use Azure Active Directory can link work items that exist in different projects across organizations. Use the following link types as indicated:

- Use the **Consumes From/Produces For** link types when you want to track dependencies of work items that are defined in different organizations and managed by different teams.
- Use the **Remote Related** link type when the work items being linked are defined in different organizations and managed by different teams, but don't have strong inter-dependencies.



Add a work item link

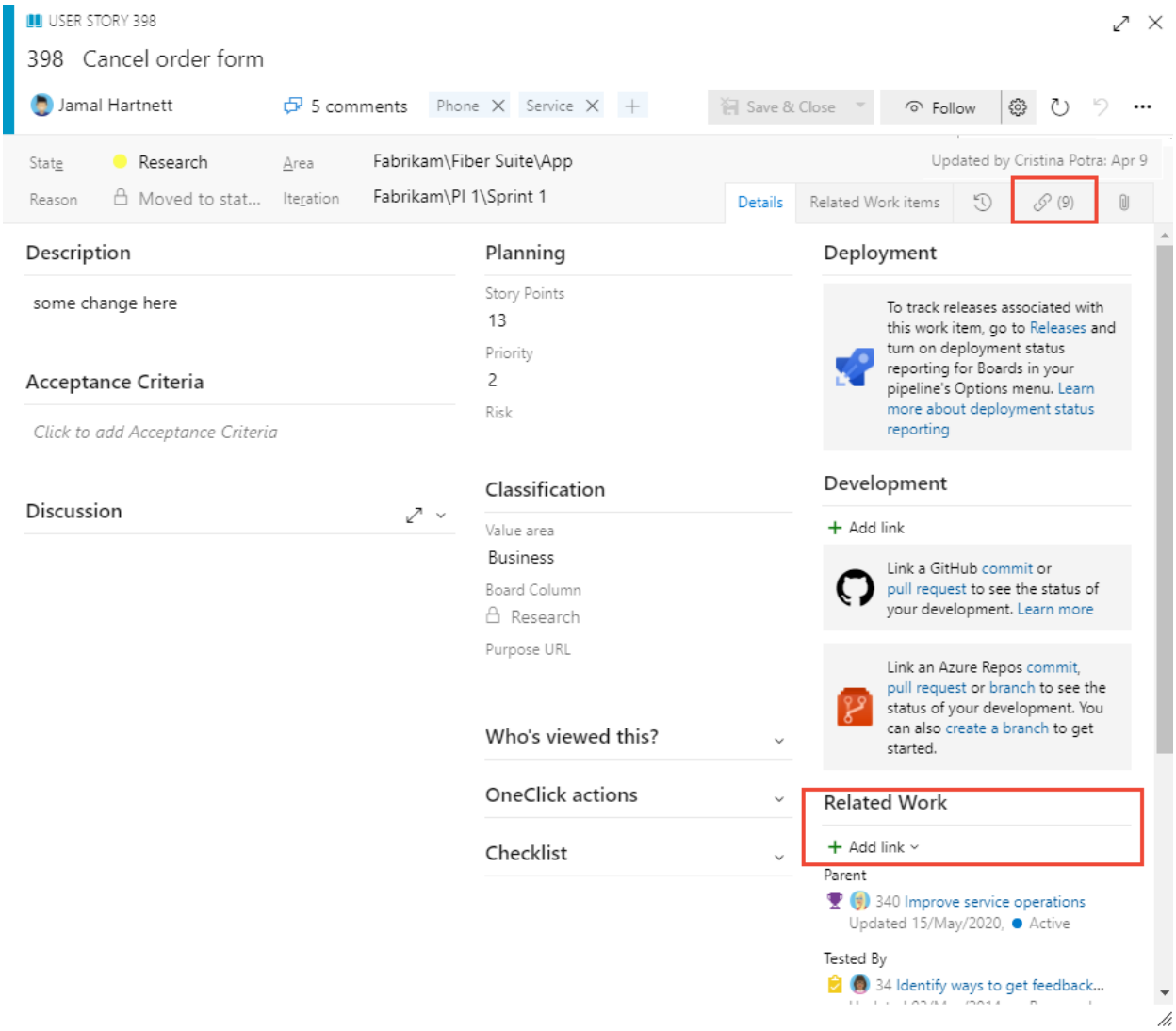
You can create links between work items by using one of the links control tabs within a work item form. The user interface to link a work item differs based on the platform, version, and client you use. To link several work items to a new or existing item, see Add link to work items.

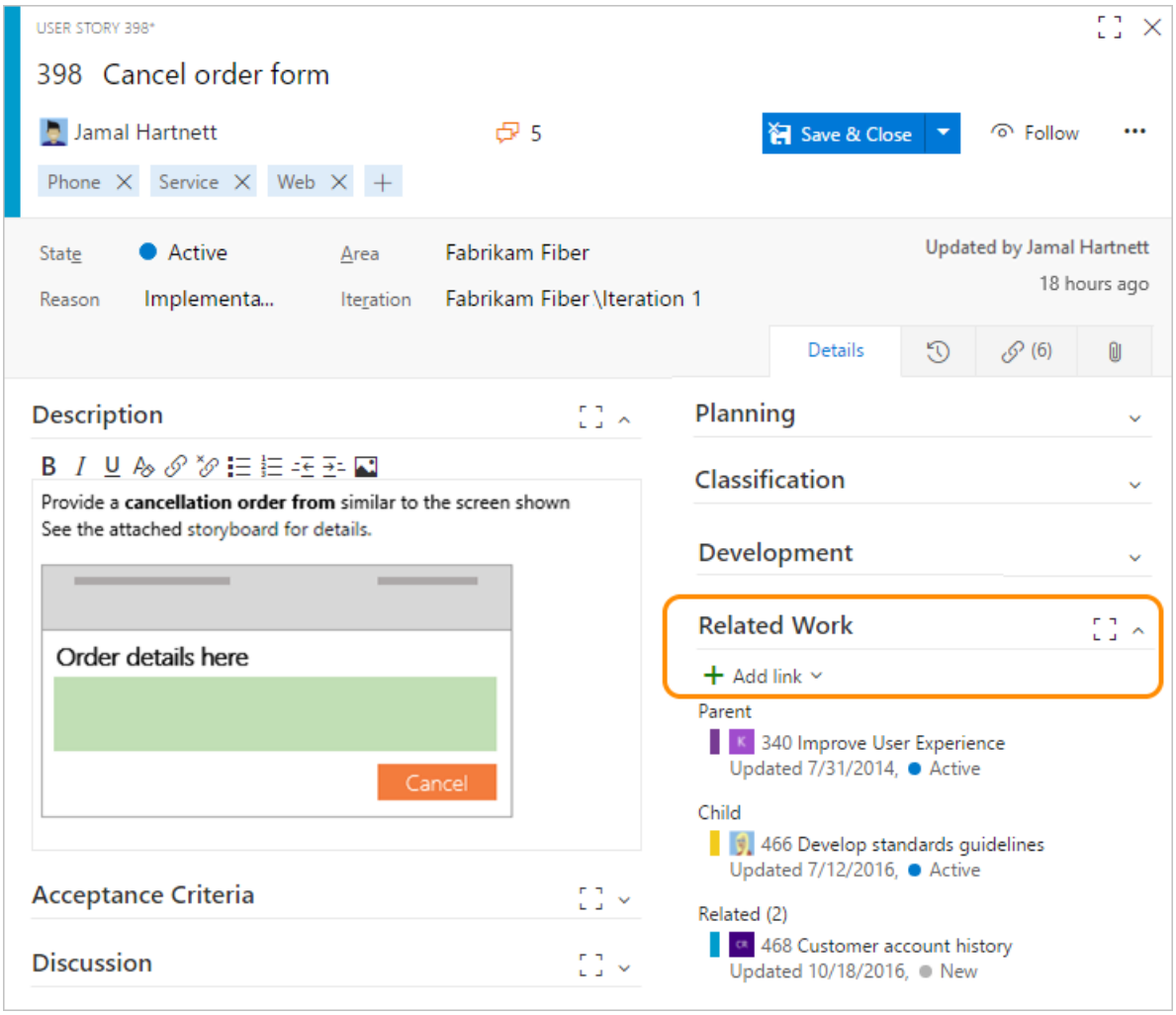
> **NOTE**
> Work item forms and features available to you can differ depending on whether you open the form from the web portal or Visual Studio Team Explorer.

- Browser
- Visual Studio
- Team Explorer Everywhere

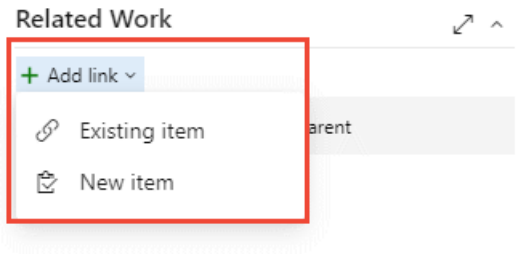
From the work item form, you can add a link using the **Related Work** section or from the **Links** tab.

Open a work item and choose **Add link** or the + plus icon to add a link.





Choose **Existing item** to link to a work item or other object using any supported link type. Choose **New item** to start a link and define a new work item at the same time. For details, see Add link to work items.



From the **Related Work** or **Links** tab, you can also complete these actions:

- View the list of objects linked to the work item
- Open an associated item or object: choose the linked item
- Delete a link: highlight it and choose the ✕ delete icon

From a query results page, you can also complete these actions:

- Link selected items to a new work item
- Link selected items to an existing work item

For details, see Add link to work items.

Parent-child work item links

These features let you quickly link or change links that use the parent-child link type:

- To link backlog items to portfolio backlog items or to change the link structure among these items, use the mapping pane to organize your backlog.
- To create and link tasks to backlog items, use the sprint backlog page; from the web portal you can also drag-and-drop items to change the link structure.
- To indent (), outdent (), and change the link structure of a tree hierarchy, you can re-parent and reorder items from a backlog in the web portal or use a tree query in Team Explorer.

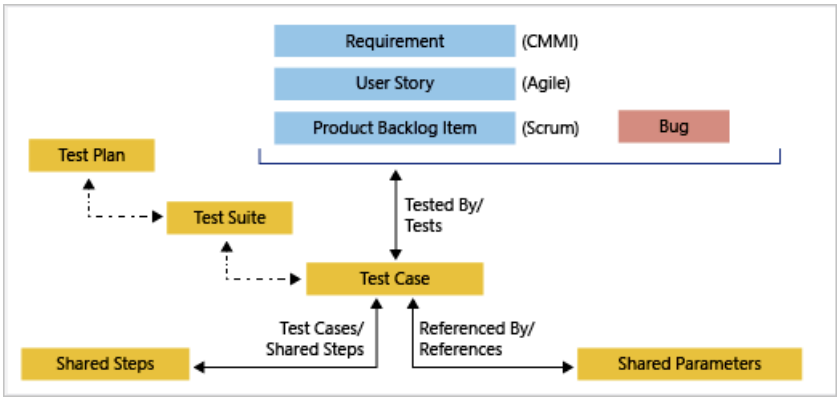
You can also use Excel to change the link structure. See Bulk add or modify work items with Excel.

Test work item links

Test related link types link test case management work items to one another, or to other work items. From the web portal or Microsoft Test Manager, you can view which test cases are defined for a test suite, and which test suites are defined for a test plan. However, these objects aren't linked to each other through link types.

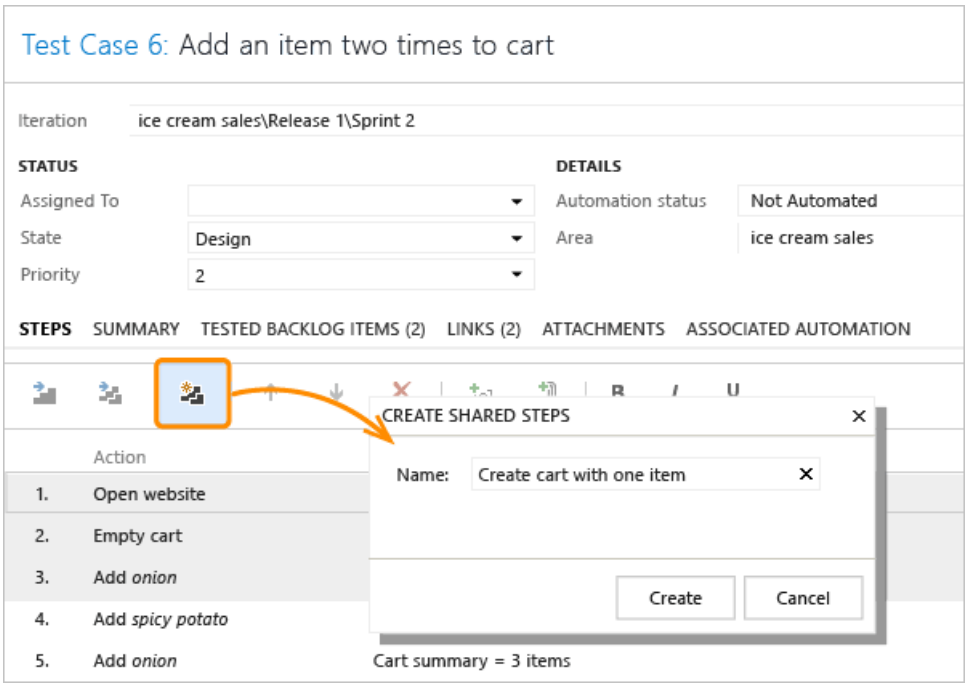
You can link work items to test cases using the **Tested/Tested By** link types. You use the same link controls you use to link work items to other work items as described earlier.

The following image shows the full set of link types used in linking test management work item types. most links between test management artifacts occur by executing a task from the **Test** pages or Microsoft Test Manager.



For example, when you add Shared Steps to a Test Case, they are automatically linked using the **Test Case/Shared Steps** link types. See Share steps between test cases.

Test case work item form (TFS 2015)

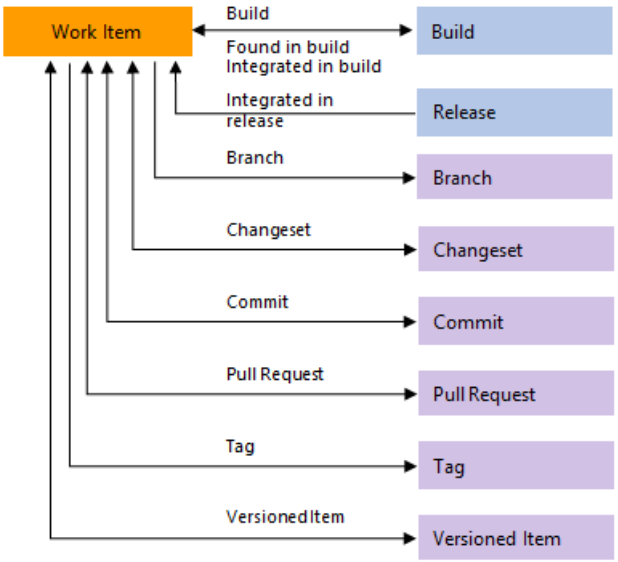


From **Test** you can add test plans, test suites, and test cases—which are linked, but not through a specific link type. Also, the test system creates and manages the associations of test results to test cases and test plans.

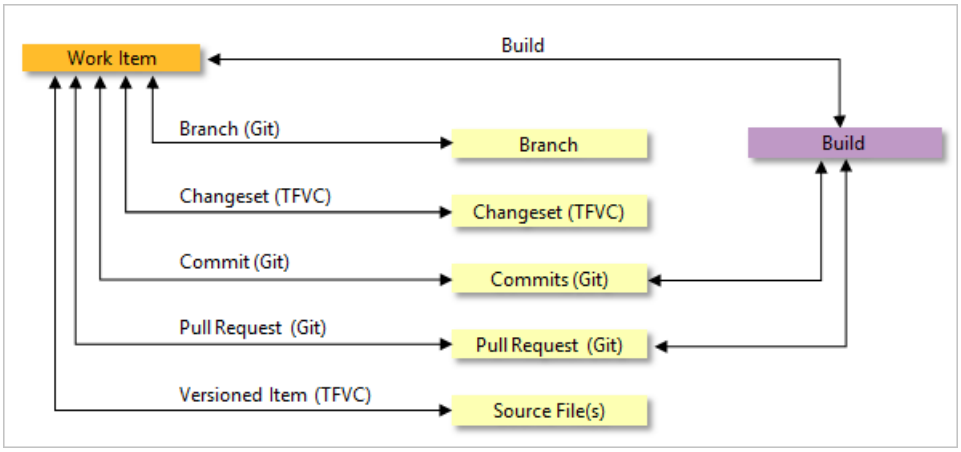
Work items linked to code artifacts and build and release pipelines

As you develop your software, you can capture which code changes and builds support the completion of a work item. In this way, your team can understand what work was done or how a bug was fixed through the audit trail of changes to the code base.

The link types used to construct these links—as illustrated in the following image—are: Branch, Build, Changeset, Commit, Found in build, Integrated in build, Pull Request, Versioned Item, and Integrated in release environment.



The link types used to construct these links—as illustrated in the following image—are: Branch, Build, Changeset, Commit, Pull Request, and Versioned Item.



To learn more about the links control or to customize the Development links control, see LinksControlOptions elements, Development links control.

You can add a link from the work item to the supported artifacts using the method described earlier for linking work items. However, an easier method is to add the work item ID to a commit, pull request, changeset, or other supported Git or TFVC operation at the time you create those items. Also, you can link work items from the **Development** control within the work item form as described in Work items linked to Git code development.

See the following articles for more information:

- Configure repositories and branches to integrate with work tracking
- Configure pipelines to support work tracking
- Drive Git development from a work item
- Link and view work items to builds and deployments
- Link to work items from pull requests, commits, and comments
- Auto complete work items with pull requests.

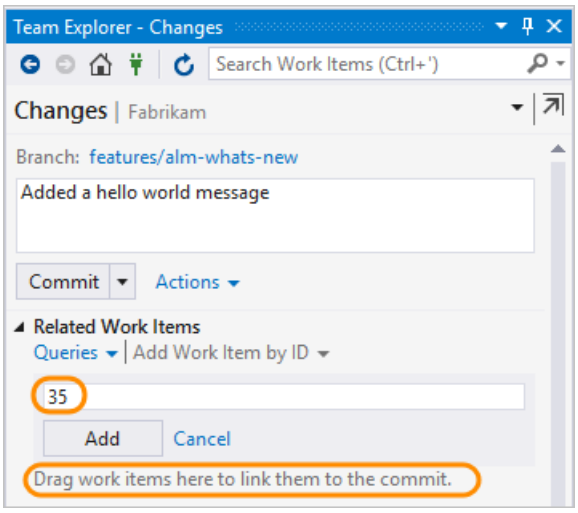
- Configure repositories and branches to integrate with work tracking
- Configure pipelines to support work tracking
- Drive Git development from a work item
- Link to work items from pull requests, commits, and comments
- Auto complete work items with pull requests.

Work items linked to Git code development

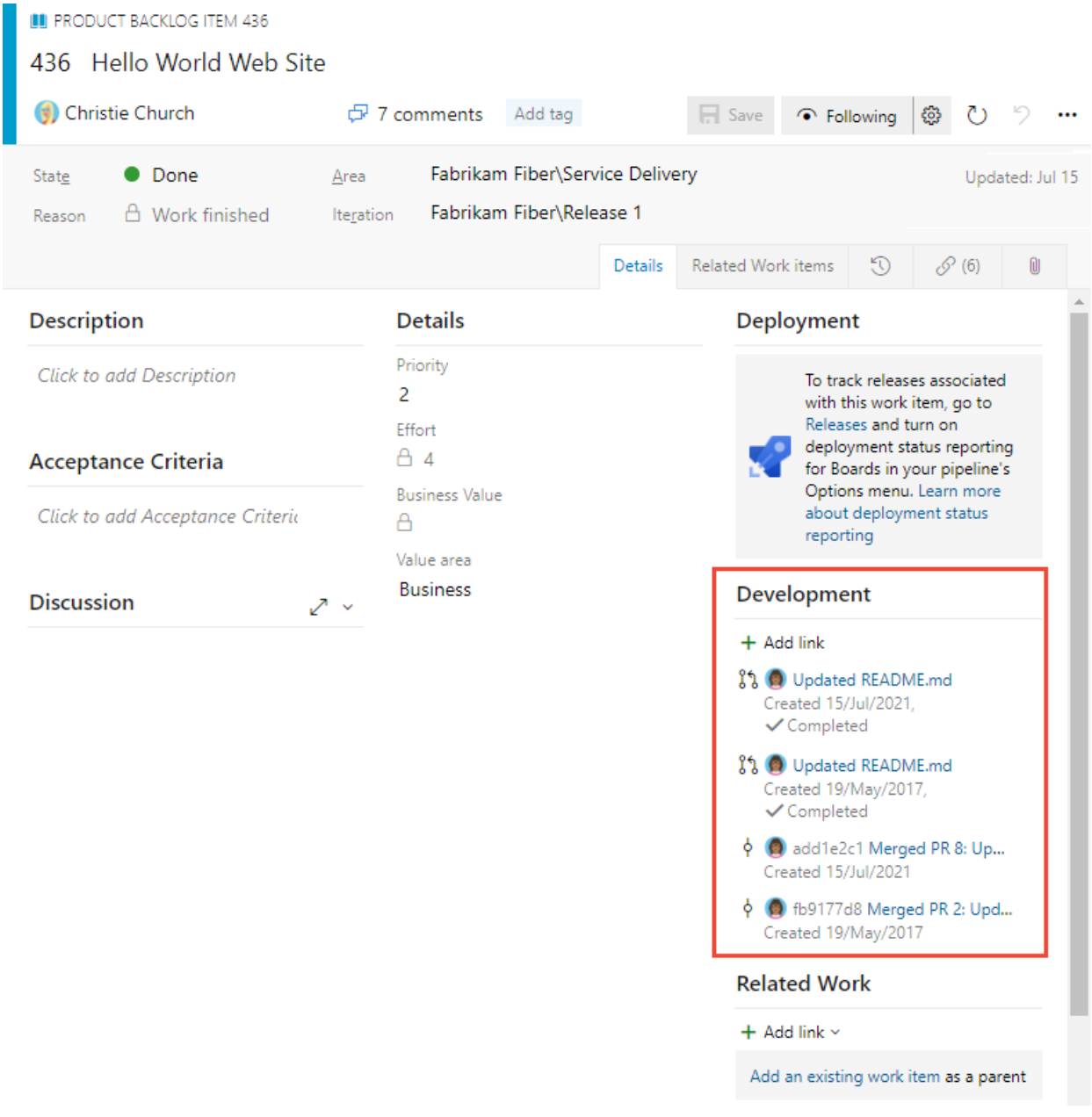
The recommended method is to drive development from the work item or add the work item ID when creating branches, commits, and pull requests.

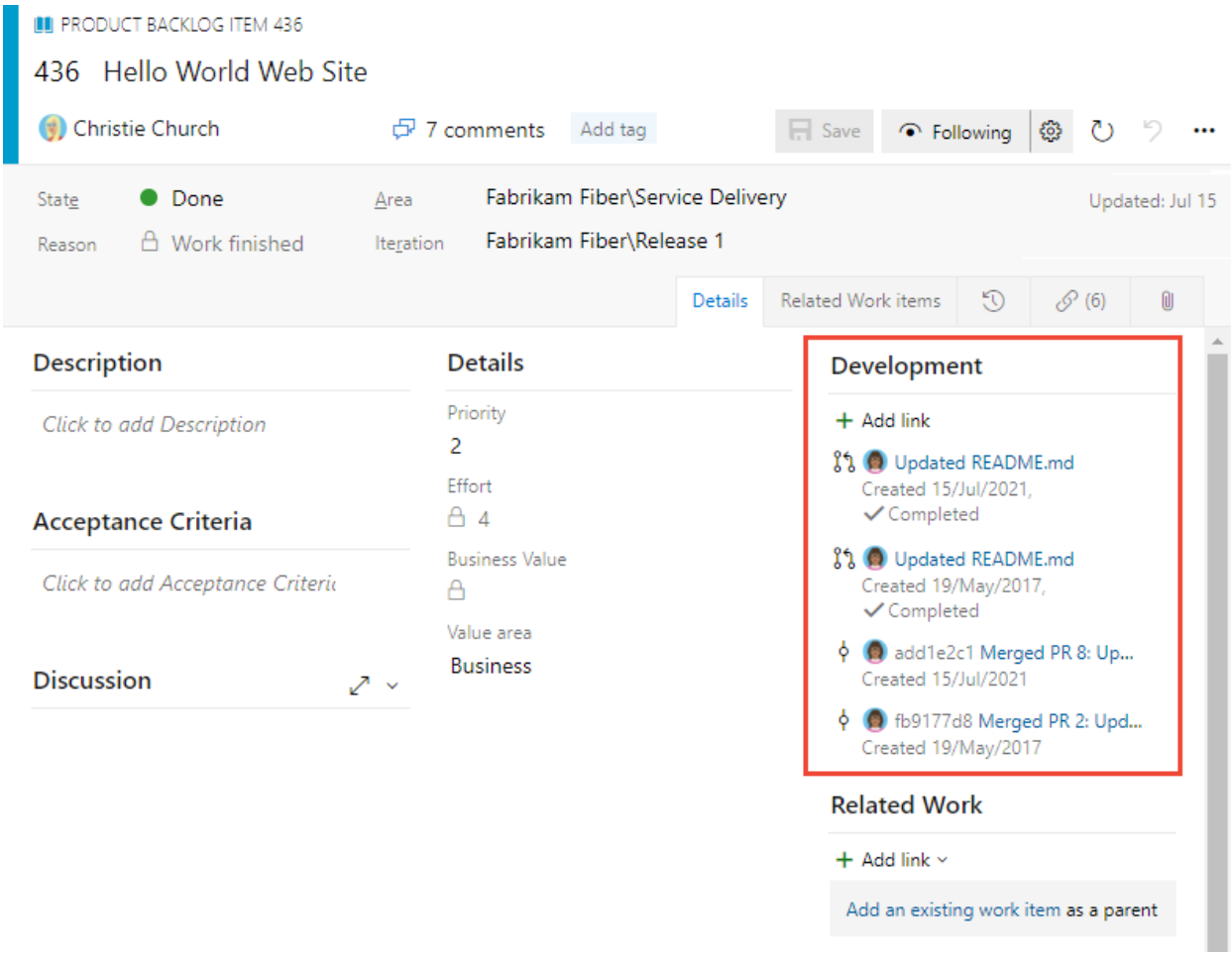
Git lets you link work items to commits by using the **Commit** link type. You can do this in several ways:

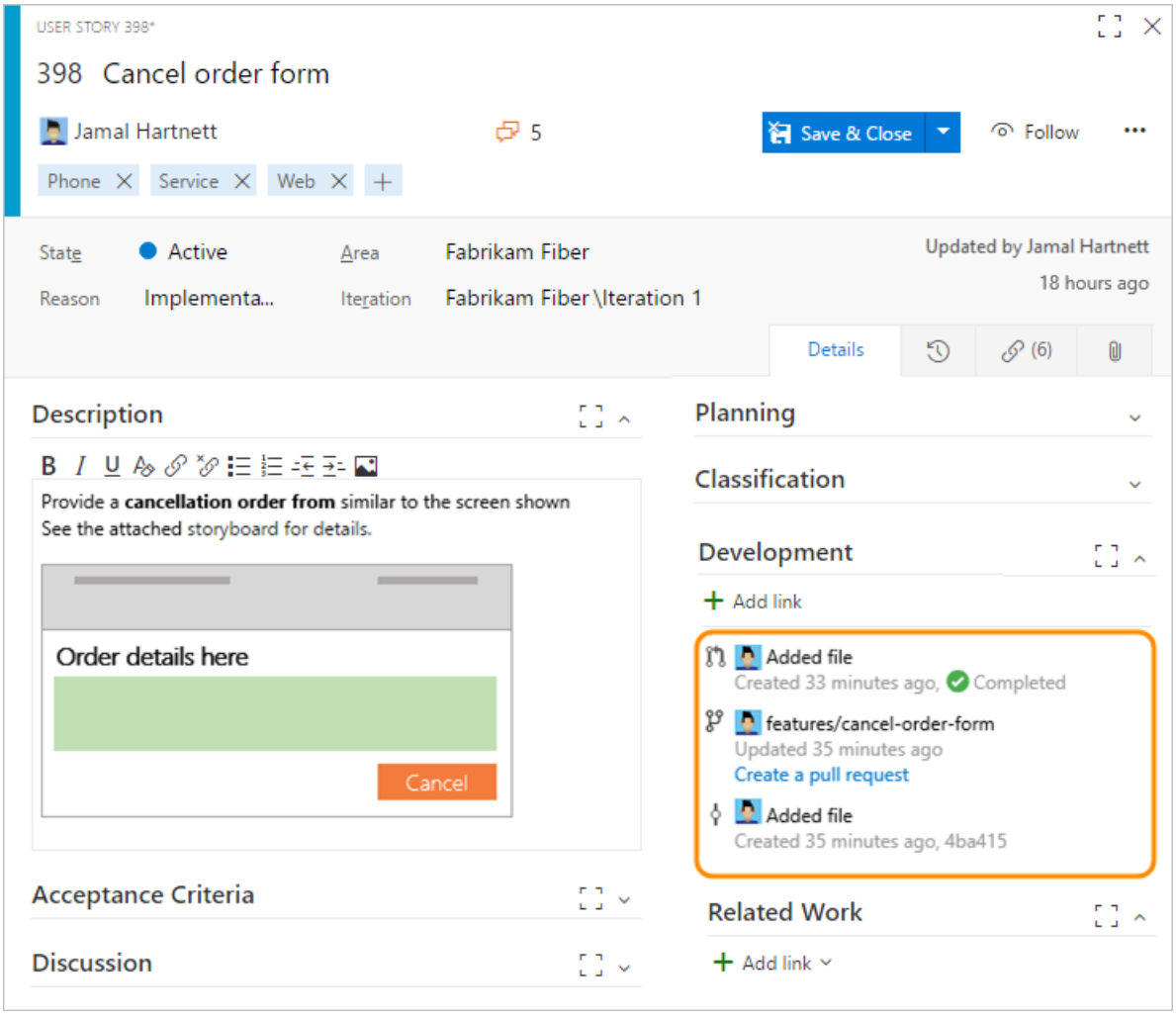
- In Visual Studio Team Explorer, add work item IDs before you commit your changes



- You can use the git-commit command and include the work item ID in your comment. For example, you apply this comment #35 Catch null exception to your commit. When you push the commit, the system creates a Commit link between the commit and work item #35.
- And, with the **Development** control, you can drive your git development from the work item as shown in the following image.




PRODUCT BACKLOG ITEM 436
436 Hello World Web Site
Christie Church
7 comments
Add tag
Save
Following
State
Done
Area
Fabrikam Fiber\Service Delivery
Updated: Jul 15
Reason
Work finished
Iteration
Fabrikam Fiber\Release 1
Details
Related Work items
(6)
Description
Click to add Description
Acceptance Criteria
Click to add Acceptance Criteri
Discussion
Details
Priority
2
Effort
4
Business Value
Value area
Business
Development
Add link
Updated README.md
Created 15/Jul/2021,
Completed
Updated README.md
Created 19/May/2017,
Completed
add1e2c1 Merged PR 8: Up...
Created 15/Jul/2021
fb9177d8 Merged PR 2: Upd...
Created 19/May/2017
Related Work
Add link
Add an existing work item as a parent


USER STORY 398*
398 Cancel order form
Jamal Hartnett
5
Save & Close
Follow
Phone
Service
Web
State
Active
Area
Fabrikam Fiber
Updated by Jamal Hartnett
18 hours ago
Reason
Implementa...
Iteration
Fabrikam Fiber\Iteration 1
Details
(6)
Description
Provide a cancellation order from similar to the screen shown
See the attached storyboard for details.
Order details here
Cancel
Acceptance Criteria
Discussion
Planning
Classification
Development
Add link
Added file
Created 33 minutes ago, Completed
features/cancel-order-form
Updated 35 minutes ago
Create a pull request
Added file
Created 35 minutes ago, 4ba415
Related Work
Add link

Work items linked to GitHub artifacts

By connecting Azure Boards with GitHub repositories, you enable linking between GitHub commits and pull requests to work items. You can use GitHub for software development while using Azure Boards to plan and track your work.

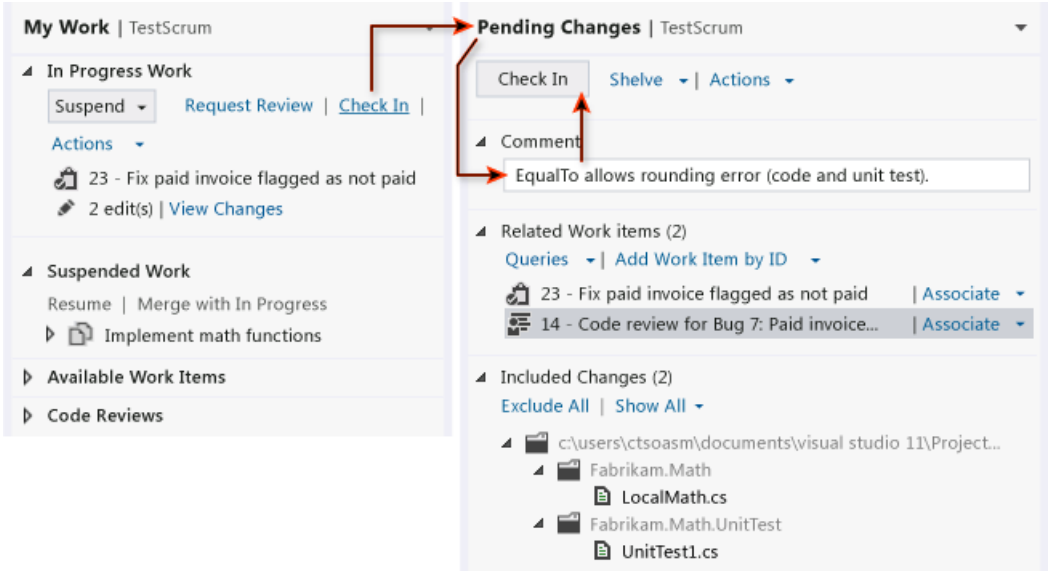
The link types supported include **GitHub Commit**, **GitHub Issue**, and **GitHub Pull Request**.

The link types supported include **GitHub Commit** and **GitHub Pull Request**.

> **IMPORTANT**
> You can only link to GitHub artifacts whose repositories you have connected to Azure Boards. To create that connection, see Connect Azure Boards to GitHub. To learn more about linking to GitHub artifacts, see Link GitHub commits, pull requests, and issues to work items.

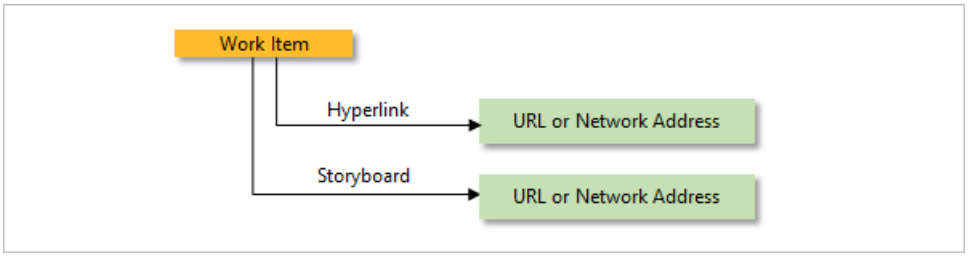
Work items linked to TFVC code development

Team Foundation version control (TFVC) lets you link work items to version control changesets or versioned source code files by using the **Changeset** and **Versioned Item** link types. When you check in pending changes or use My Work to check in changes, work items are automatically linked to your changes.



Work items linked to a Web site, network share, storyboard, or document

You can use the Hyperlinks or Storyboard link type to link a work item to a Web site, network share, or document located on a network share. Both of these link types are one-way links. To add links of this type, you can use the same links controls described earlier for linking work items.



By using the Storyboard link type, you differentiate the link you're adding to specify a storyboard or document that provides work item specifications. Use this link type to provide your team access to the shared file where

they can add their comments.

Query for linked work items

To filter items based on hierarchical links, use the **Tree of work items** query type. To filter items based on all link types, use **Work items and direct links**.

You can search for work items that not only meet criteria for field values but also that are linked to other work items with specific types of links. This kind of query displays a primary set of work items, which meet the field criteria, and a secondary set, which is linked to items in the primary set.

For query examples, see Link and attachment queries.

> **NOTE**
> You can't construct a query that shows a hierarchical view of Test Plans, Test Suites, and Test Cases. These items aren't linked together using Parent/Child or any other link type. You can only view the hierarchy through the **Test > Test Plans** page.

Related articles

You should now have a broad understanding of the various link relationships you can create to track dependencies and create an audit trail for your code development.

Once you've formed a link relationship, you can't edit the link type of that relationship from the web portal, but you can do it from Team Explorer.

For more information, see these articles:

- Add link to multiple work items
- Track dependencies using Delivery Plans
- Share plans, add attachments
- Use mapping to link backlog items to features and epics
- Bulk modify links using Excel
- Link types reference

Visualize related work and other objects

You can view related work items and object within a work item form by installing the Work item visualization extension available from the Visual Studio Marketplace, Azure DevOps tab.

Reference guide for link types used in Azure DevOps and Azure Boards

10/4/2022 • 15 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can use different link types to manage the various relationships between work items and other artifacts, like builds, commits, pull requests, and more.

You can link work items to other work items or artifacts using the following link types.

- **Work link types**: links work items including select test case management work items
- **Hyperlink**: connects a work item to any URL or network share
- **External link types**: connects a work item to an external object, such as a code object, build, or wiki page
- **Remote work link types**: connects work items that are defined in different organizations
- **GitHub link types**: connects a work item to a GitHub repository commit, issue, or pull request.

A specific field maintains a count of links for the first four link types, such as *Related Link Count*, *Hyperlink Count*, *External Link Count*, and *Remote Link Count*.

- **Work link types**: links work items including select test case management work items
- **Hyperlink**: connects a work item to any URL or network share
- **External link types**: connects a work item to an external object, such as a code object, build, or wiki page
- **GitHub link types**: connects a work item to a GitHub repository commit or pull request.

A specific field maintains a count of links for the first three link types, such as *Related Link Count*, *Hyperlink Count*, and *External Link Count*.

- **Work link types**: links work items including select test case management work items
- **Hyperlink**: connects a work item to any URL or network share
- **External link types**: connects a work item to an external object, such as a code object, build, or storyboard.

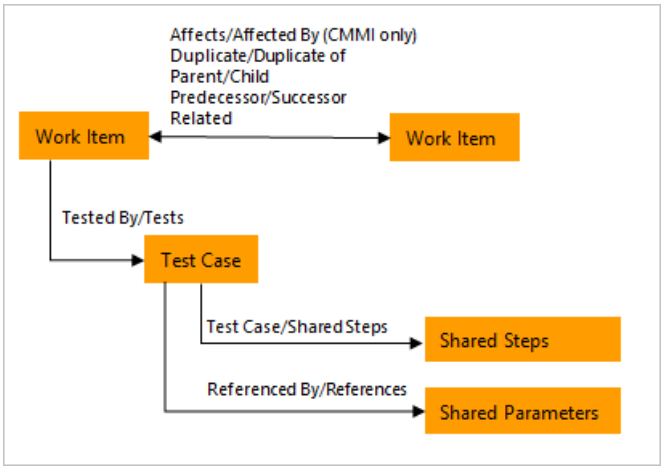
A specific field maintains a count of links for each of these link types, such as *Related Link Count*, *Hyperlink Count*, and *External Link Count*.

Link types you use to link work items are subject to certain restrictions based on their topology. Use the guidance provided in the following tables to choose which link type to use based on the types of queries and reports you'll want to create. To learn more about the different topologies, see Link type topologies and restrictions.

Work link types

Work link types are system-defined, process-defined, or user-defined (custom). The links listed in the following table are system defined.

Each work link type defines the link labels, topology type, and restrictions that are used when links between work items are constructed. For example, the parent-child link type defines two labels: Parent and Child. The link type also supports a hierarchical or tree topology, and prevents circular references from being created between work items.



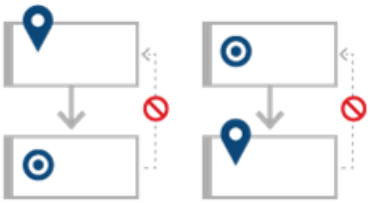
A work item's *Related Link Count* corresponds to the sum of all links defined with a work link type.

The following table describes the work item link types you can specify to scope a links control using the **WorkItemLinksFilter** XML element.

Name

Reference name

Usage



Affects-Affected by (CMMI only)

Microsoft.VSTS.Common.Affects-Forward
Microsoft.VSTS.Common.Affects-Reverse
Topology type: Dependency
Link category: Process-defined

Use this directional link to create links between any set of work items, but not ones that would create closed loops. Typically used to track change requests made to requirements.

Restrictions and recommendations:

- You can link a change request to only one requirement using Affects. You can link requirements to as many child change requests as needed using Affected by.
- Only use Affects-Affected by links to link work items in the same project. This action is recommended if you plan to use Excel or Project to modify or update work item data.

Child-Parent



System.LinkTypes.Hierarchy-Forward
System.LinkTypes.Hierarchy-Reverse
Topology type: Tree
Link category: System-defined

Use this directional link to create one-to-many relationships between a single parent to one or more child items. Use to organize work item within a hierarchy. You can quickly create this hierarchy among backlog items using the mapping function or among backlog items and tasks using the sprint backlog or Taskboard.

Typical uses include:

- Maintain task summary relationships. Parent-child links are created for summary tasks and their subordinate tasks.
- Link tasks to PBIs, user stories, or requirements. Supports Backlog Overview, Stories Overview, and Requirements Overview reports.

Restrictions and recommendations:

- Use Excel to bulk edit both work items and parent-child links. See Bulk add or modify work items with Excel.
- A work item can have only one Parent. A parent work item can have many children.
- Only use parent-child links to link work items in the same project. This action is recommended if you plan to use Excel to modify or update work item data.

Duplicate-Duplicate of [1]



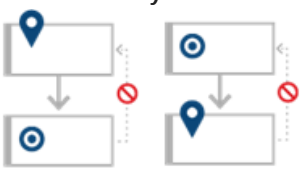
System.LinkTypes.Duplicate-Forward
System.LinkTypes.Duplicate-Reverse
Topology type: Tree
Link category: System-defined

Use this directional link to create one-to-many relationships between a single parent to one or more child items. Use to track tasks, bugs, or other work items that are duplicates of one another.

Restrictions and recommendations:

- A work item can have only one Duplicate.
- Only use Duplicate or Duplicate Of links to link work items in the same project. This action is recommended if you plan to use Excel or Project to modify or update work item data.

Referenced By-References



Microsoft.VSTS.TestCase.
SharedParameterReferencedBy
Topology type: Dependency
Link category: Process-defined

Use to link test cases to shared parameters. Use to link Test Cases to Shared Parameters to support the ability to

repeat a test with different data. In general, you wouldn't add this link type to a scoped links control. To learn more, see Repeat a test with different data.

Related



System.LinkTypes.Related
Topology type: Network
Link category: System-defined

Use this non-directional link to create links between any set of work items. Use to link work items that are at the same level, such as two user stories that define features that overlap one another. The Related link type creates simple relationships with few restrictions.

- Relate work items that are at the same level, such as two user stories that define features that overlap one another.
- Link work items that are defined in different projects and managed by different teams.
- Find and view work items and their related work items in a two-tiered view.
- Create simple relationships with few restrictions.

Successor-Predecessor



System.LinkTypes.Dependency

Topology type: Dependency
Link category: System-defined
Choose **Predecessor** link type when linking to a work item that should be completed before the work item you're linking from. Choose **Successor** link type when linking to a work item that should be completed *after* to the work item you're linking from.

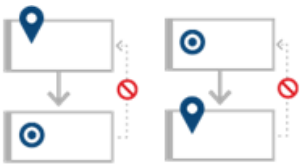
Use this directional link to create links between any set of work items, but not ones that would create closed loops. Use to track tasks that must be completed before others can be started. When you plan work using Microsoft Project, linked tasks are represented as predecessor-successor links in Azure Boards. Typically used to track work that must be completed before beginning work on predecessor items.

- Track tasks that must be completed before others can be started. When you plan work using Project, linked tasks are represented as predecessor-successor links in TFS.
- Supports one-to-many relationships.
- Find and view predecessor work items and their successor work items in a two-tiered, direct links query view.

Restrictions and recommendations:-

- An error appears when you attempt to create links that define circular relationships.
- Create predecessor-successor links only to work items that are within the same project. You can create predecessor-successor links between work items that are defined in different projects. However, if you export a query to Excel or Project, only those work items that are defined for the project for which the query is defined are imported.

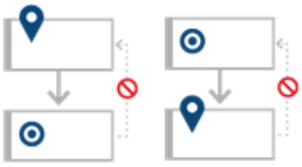
Tested by-Tests



Microsoft.VSTS.Common.TestedBy-Forward
Microsoft.VSTS.Common.TestedBy-Reverse
Topology type: Dependency
Link category: Process-defined

Link test cases to work items, such as bugs, user stories, requirements, and product backlog items. Use to track test cases that test user stories (Agile), product backlog items (Scrum), or requirements (CMMI). Can also link to other work item types such as bugs, issues, or tasks. For on-premises Azure DevOps, there are several SQL reports that depend on these links. See Review team activities to support useful reports.

Test Case-Shared Steps



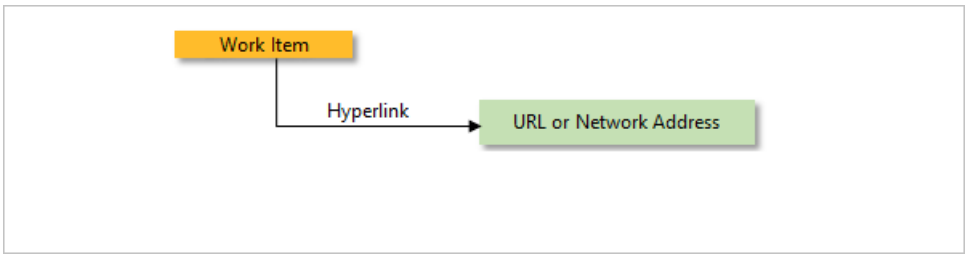
Microsoft.VSTS.TestCase.
SharedStepReferencedBy
Topology type: Dependency
Link category: Process-defined

Use to link test cases with shared steps. You share steps between test cases to avoid having to create multiple entries of the same sequence of steps. To learn more, see Share steps between test cases.

> **NOTE**
> 1. Available from TFS 2017 and later versions.

Hyperlink link type

There's one link type that tracks the number of hyperlinks—*Hyperlink Count*—added to a work item. A hyperlink can link a work item to any URL.



Link name

Tool supported

Artifact type

Usage

Hyperlink

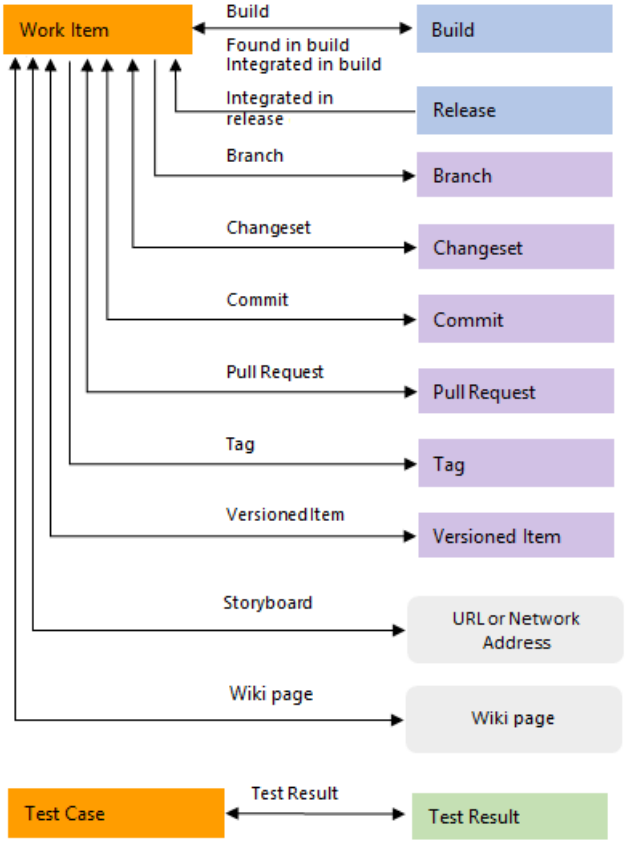
Work item tracking

Hyperlink

Used to link a work item to a URL. **Workitem Hyperlink** is the name of this link type in the Artifact Link Types API.

External link types

External link types are system-defined link types that support linking work items to other objects stored within Azure DevOps as shown in the following image. A work item's *External Link Count* corresponds to the sum of all links defined with an external link type.



> **NOTE**
> You can only use an external link type to link to an Azure DevOps object. To link work items to other objects outside of Azure DevOps, you can use the Hyperlink link type.

The following table describes the external link types you can choose when adding a link type from a work item or test case.

The following table describes the external link types you can choose when adding a link type from a work item or test case. Also, you can use specify one of these link types to scope a links control using the **ExternalLinksFilter** XML element.

Link name

Tool supported

Artifact type

Usage
Branch
Git
Branch
Used to link a work item to a branch.
Pipelines/Build
Build
Build
Used to link a work item to a build.
Changeset (or Fixed in Changeset)
VersionControl
Changeset
Used to link a work item to a changeset.
Commit (or Fixed in Commit)
Git
Commit
Used to link a work item to a commit.
Found in build
Pipelines/Build
Build
Used to link a work item to a build.
Integrated in build
Build
Build pipeline
Used to link a work item to a build.
Integrated in release environment
Release
Release pipeline
Used to link a release to a work item. The system creates a link of this type when a user enables the **Report deployment status to Work** option for a release definition. To learn how to set this option, see Release pipelines, How do I integrate and report release status?
Pull Request
Git
PullRequestId

Used to link a work item to a pull request.

Result attachment

Test Management

TcmResultAttachment

Used to link a work item to an attachment associated with a test result. These links appear when you associate a work item with a test result from **Test** or Microsoft Test Manager.

Source Code File<

VersionControl

LatestItemVersion

Used to link a work item to a file under Team Foundation version control (TFVC).

Storyboard

Requirements

Storyboard

Used to link a work item to a PowerPoint file or other file that contains story boarding information on a network.

Tag

Git

Tag

Used to link a work item to a tag that's been defined for a git commit or git repository. For more information, see Work from the Git command prompt for more information.

Test Result

Test Management

TcmResult

Used to link a work item to a test result. These links appear when you associate a work item with a test result from **Test** or Microsoft Test Manager.

Versioned item

VersionControl

LatestItemVersion

Used to link a work item to a file or changeset defined within a TFVC repository. **Source Code File** is the name of this link type in the Artifact Link Types API.

Wiki

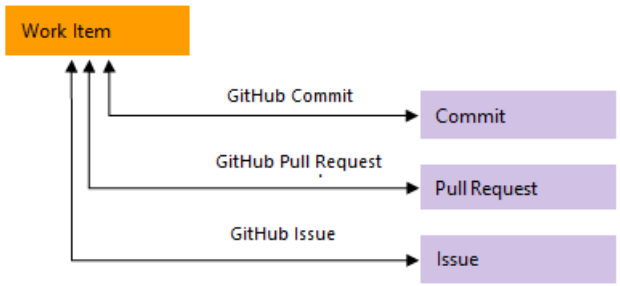
Wiki

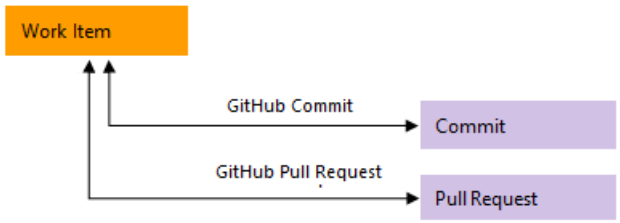
Wiki

Used to link a work item to a wiki page. Supported for TFS 2018.2 and later versions.

GitHub link types

GitHub link types are system-defined link types. They support linking work items to GitHub objects as shown in the following image.





> **IMPORTANT**
> You can only link to GitHub artifacts whose repositories you have connected to Azure Boards. To create that connection, see Connect Azure Boards to GitHub. To learn more about linking to GitHub artifacts, see Link GitHub commits, pull requests, and issues to work items.

The following table describes the GitHub link types you can choose when adding a link type from a work item.

Link name	Artifact type	Usage
GitHub Commit	GitHub repository commit	Used to link a work item to a GitHub commit.
GitHub Issue	GitHub repository issue	Used to link a work item to a GitHub issue.
GitHub Pull Request	GitHub repository pull request	Used to link a work item to a GitHub pull request.

Remote work link types

Remote work link types are system-defined link types that support linking work items defined in different organizations. Organizations must be managed by the same Azure Active Directory as described in Connect your organization to Azure Active Directory.

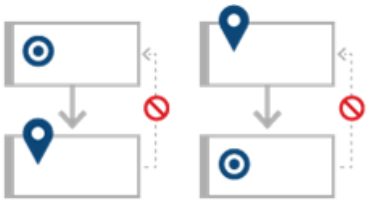
A work item's *Remote Link Count* corresponds to the sum of all links defined with a remote work link type.

Name

Reference name

Usage

Consumes From-Produced For



(Dependency topology)

System.LinkTypes.Remote.Dependency-Forward
System.LinkTypes.Remote.Dependency-Reverse
Topology type: Dependency
Link category: System-defined

Use this directional link to create links between work items that have dependencies and are defined in different organizations. Organizations must be managed by the same Azure Active Directory. Typically used to track change requests made to requirements.

Remote Related



System.LinkTypes.Remote.Related
Topology type: Network
Link category: System-defined

Use this non-directional link to create links between work items defined in different organizations. Organizations must be managed by the same Azure Active Directory.

Custom link types

You can create custom link types; export and import definitions of link types; and delete, activate, deactivate, and reactivate types of links. See the following articles:

- Link type element reference
- Manage link types (witadmin)

List link types

To get a list of link types, you can use one of the supported command-line tools.

az boards work-item relation list-type

You can list link types supported by your organization with the az boards work-item relationlist-type command or the Work Item Relation Types - List REST API command. To get started, see Get started with Azure DevOps CLI.

```
az boards work-item relation list-type [--org]
```

Optional parameters

- **org**: Azure DevOps organization URL. You can configure the default organization using `az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using `git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.

Example

The following command lists the work item link types in table format that are defined for the fabrikam organization. For other formats, see Output formats for Azure CLI commands.

```
az boards work-item relation list-type --org fabrikam --output table
Name                 ReferenceName                                                Enabled    Usage
-------------------  -----------------------------------------------------------  ---------  ------------
Produces For         System.LinkTypes.Remote.Dependency-Forward                   True       workItemLink
Consumes From        System.LinkTypes.Remote.Dependency-Reverse                   True       workItemLink
Duplicate            System.LinkTypes.Duplicate-Forward                           True       workItemLink
Duplicate Of         System.LinkTypes.Duplicate-Reverse                           True       workItemLink
Referenced By        Microsoft.VSTS.TestCase.SharedParameterReferencedBy-Forward  True       workItemLink
References           Microsoft.VSTS.TestCase.SharedParameterReferencedBy-Reverse  True       workItemLink
Tested By            Microsoft.VSTS.Common.TestedBy-Forward                       True       workItemLink
Tests                Microsoft.VSTS.Common.TestedBy-Reverse                       True       workItemLink
Test Case            Microsoft.VSTS.TestCase.SharedStepReferencedBy-Forward       True       workItemLink
Shared Steps         Microsoft.VSTS.TestCase.SharedStepReferencedBy-Reverse       True       workItemLink
Successor            System.LinkTypes.Dependency-Forward                          True       workItemLink
Predecessor          System.LinkTypes.Dependency-Reverse                          True       workItemLink
Child                System.LinkTypes.Hierarchy-Forward                           True       workItemLink
Parent               System.LinkTypes.Hierarchy-Reverse                           True       workItemLink
Related              System.LinkTypes.Related                                     True       workItemLink
Remote Related       System.LinkTypes.Remote.Related                              True       workItemLink
Attached File        AttachedFile                                                 True       resourceLink
Hyperlink            Hyperlink                                                    True       resourceLink
Artifact Link        ArtifactLink                                                 True       resourceLink
```

The default json format provides additional information about the attributes defined for the link types. For example, the information for the link types *Produces For* and *Consumes From* are listed as follows.

```
  {
    "attributes": {
      "acyclic": true,
      "directional": true,
      "editable": false,
      "enabled": true,
      "isForward": true,
      "oppositeEndReferenceName": "System.LinkTypes.Remote.Dependency-Reverse",
      "remote": true,
      "singleTarget": true,
      "topology": "dependency",
      "usage": "workItemLink"
    },
    "name": "Produces For",
    "referenceName": "System.LinkTypes.Remote.Dependency-Forward",
    "url": "https://dev.azure.com/mseng/_apis/wit/workItemRelationTypes/System.LinkTypes.Remote.Dependency-
Forward"
  },
  {
    "attributes": {
      "acyclic": true,
      "directional": true,
      "editable": false,
      "enabled": true,
      "isForward": false,
      "oppositeEndReferenceName": "System.LinkTypes.Remote.Dependency-Forward",
      "remote": true,
      "singleTarget": true,
      "topology": "dependency",
      "usage": "workItemLink"
    },
    "name": "Consumes From",
    "referenceName": "System.LinkTypes.Remote.Dependency-Reverse",
    "url": "https://dev.azure.com/mseng/_apis/wit/workItemRelationTypes/System.LinkTypes.Remote.Dependency-
Reverse"
  },
```


`witadmin listlinktypes`

You can list link types supported for your project collection using the **witadmin listlinktypes** command-line tool or the Work Item Relation Types - List REST API command.

Here we list the link types for the fabrikam-sever default collection:

```
C:\Program Files (x86)\Microsoft Visual
Studio\2019\Community\Common7\IDE\CommonExtensions\Microsoft\TeamFoundation\Team Explorer>witadmin
listlinktypes /collection:http://fabrikam-server/DefaultCollection

Reference Name: Microsoft.VSTS.TestCase.SharedParameterReferencedBy
Names: Referenced By, References
Topology: Dependency
Is Active: True

Reference Name: Microsoft.VSTS.Common.TestedBy
Names: Tested By, Tests
Topology: Dependency
Is Active: True

Reference Name: Microsoft.VSTS.TestCase.SharedStepReferencedBy
Names: Test Case, Shared Steps
Topology: Dependency
Is Active: True

Reference Name: System.LinkTypes.Duplicate
Names: Duplicate, Duplicate Of
Topology: Tree
Is Active: True

Reference Name: System.LinkTypes.Dependency
Names: Successor, Predecessor
Topology: Dependency
Is Active: True

Reference Name: System.LinkTypes.Hierarchy
Names: Child, Parent
Topology: Tree
Is Active: True

Reference Name: System.LinkTypes.Related
Name: Related
Topology: Network
Is Active: True
```

Link type attributes

The following table provides descriptions for each of the link type attributes returned by **Azure Boards** CLI or the REST API.

Attribute	Description
Names, `name`	Specifies the friendly name assigned to the link type(s). Directional links are defined in pairs, therefore include a forward and reverse name.
Reference name, `referenceName`	Specifies the name assigned to the link type or link type pair.
`acyclic`	Indicates whether the link type allows or (`true`) or restricts (`false`) circular relationships. For example, tree type links restrict circular relationships. For more information, see LinkTypes elements reference.
`directional`	

Indicates whether the link type is directional (`true`) or not (`false`). Directional link types are defined in pairs with a forward and reverse component. For more information, see LinkTypes elements reference.

`editable`

Indicates whether the link type is editable (`true`) or not (`false`). You can only add and edit custom link types for on-premises deployments using **witadmin** Manage link type command-line tool. System link types always have `editable=false` .

Is Active, `enabled`

Indicates whether the link type is active (`true`) or not (`false`). You can only custom link types for on-premises deployments using the **witadmin** Manage link type command-line tool.

`isForward`

Indicates whether the link type specifies the forward link (`true`) or not (`False`) within a link type pair.

`oppositeEndReferenceName`

Specifies the reference name of the link type that defines the link in the opposite direction of a link type pair.

`remote`

Indicates whether the link type supports linking to a remore work item (`true`) or not (`False`). Link types with `remote=false` require that the target work item resides in the same organization or collection as the origin work item.

`singleTarget`

Indicates whether the link type allows for more than one target (`false`) or is restricted to a single target (`true`).

`topology`

Specifies the topology type—dependency `,` network `, and` tree`. For descriptions, see Link type topologies and restrictions.

`usage`

Specifies the usage type—resourceLink or `workItemLink` . The workItemLink `value indicates a link type that links two work items. The` resourceLink` value indicates a link type used to link a work item to a resource, such as a URL or attachment.

`url`

Lists the attributes of the link type in json format.

Related articles

- Link work items to track dependencies
- Add link to multiple work items
- Query FAQs
- Track dependencies using Delivery Plans
- Use mapping to link backlog items to features and epics
- Bulk modify links using Excel
- Link type topologies and restrictions
- Artifact Link Types API

Work tracking, process, and project limits

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This article defines operational and object limits placed on work tracking operations and work tracking customization. In addition to the specified hard limits on select objects, certain practical limits apply. When you customize work item types (WITs), consider the limits placed on objects.

Work items and queries

When defining work items or running queries, the following operational limits apply.

OBJECT	LIMIT
Attachments added to a work item	100
Attachment size	60 MB
Long text field	1 M characters
Query execution time	30 seconds
Query results	20,000 items
Query length	32,000 characters
Shared queries under a folder	999 queries
Work item links assigned to a work item	1,000
Work item tags assigned to a work item	100
Work item revisions (REST API)	10,000

A work item revision limit of 10,000 is in effect for updates made through the REST API for Azure DevOps Services. This limit restricts updates from the REST API, however, updates from the web portal are not affected.

OBJECT	LIMIT
Long text field	1 M characters
Work item tags assigned to a work item	100
Work item links assigned to a work item	1,000
Attachments added to a work item	100
Attachment size	4 MB to 2 GB

OBJECT	LIMIT
Query execution time	6 minutes
Query results	20,000 items
Query length	32,000 characters
Shared queries under a folder	999 queries

The default maximum attachment size is 4 MB. You can change the maximum size up to 2 GB.

To improve query performance, see Guidance to create high-performing queries.

Backlogs, boards, dashboards, and teams

When working with teams, work item tags, backlogs, and boards, the following operational display and object limits apply.

USER INTERFACE	LIMIT
Backlogs	10,000 work items
Boards	1,000 cards (excluding those cards in the *Proposed* and *Completed* workflow state categories)
Taskboard	1,000 tasks
Area Paths	10,000 per organization
Area Path Depth	14
Area Paths per team	300
Iteration Paths	10,000 per organization
Iteration Path Depth	14
Iteration Paths per team	300
Dashboards per project	500
Teams	5,000 per organization
Work item tags	150,000 tag definitions per organization or collection
Delivery plans per project	1,000

Each backlog can display up to 10,000 work items. This is a limit on what the backlog can display, not a limit on the number of work items you can define. If your backlog exceeds this limit, then you may want to consider adding a team and moving some of the work items to the other team's backlog.

Additional notes:

- Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is

greater than a year old. You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock for display.

- Avoid nesting backlog items of the same type. To learn more, see Fix reordering and nesting issues.
- Avoid assigning the same area paths to more than one team. To learn more, see Limitations of multi-team Kanban board views.
- By default, work item limits might be initially configured to lower values.

When working with teams, work item tags, backlogs, and boards, the following operational limits apply. Default and maximum limits.

USER INTERFACE	LIMIT
Backlogs	999 work items
Boards	400 cards
Dashboards per project	500
Taskboard	800 work items
Teams	5,000 per project
Work item tags	150,000 tag definitions per project

Each backlog can display up to 999 work items. If your backlog exceeds this limit, then you may want to consider adding a team and moving some of the work items to the other team's backlog.

Additional notes:

- Avoid nesting backlog items of the same type. To learn more, see Fix reordering and nesting issues.
- Avoid assigning the same area paths to more than one team. To learn more, see Limitations of multi-team Kanban board views.

For the On-premises XML process model, you can modify the backlog and taskboard limits by editing the ProcessConfiguration.xml file. For details, see Process configuration XML element reference.

Projects

Azure DevOps Services limits each organization to 1000 projects per organization, an increase over the previous limit of 300 projects.

> **NOTE**
> Above 300 projects certain experiences, such as connecting to a project from Visual Studio, may start to degrade. For on-premises Azure DevOps Server, there are no hard limits to the number of projects. However, you may find performance issues if the number of projects approaches 300. If you plan to migrate your on-premises collection to Azure DevOps Services, you'll need to observe the maximum limit of 1000 projects. If your collection has more than 1000 projects, you'll either need to split the collection or delete older projects.

For more information, see Migrate data from Azure DevOps Server to Azure DevOps Services.

Process customization

A number of limits are imposed on the number of objects that you can define for a process. To learn about process models, see Customize your work tracking experience.

The following table lists the maximum number of objects that you can define for the Inheritance and Hosted XML process models. While these represent hard limits, practical limits may also apply.

OBJECT	INHERITANCE	HOSTED XML
Number of processes you can have in an organization	128	64
Work item types defined for a process	64	64
Fields defined for an organization	8192	8192
Fields defined for a process	1024	1024
Fields defined for a work item type	1024	1024
Picklists defined for an organization or collection	2048	-
Picklist items defined for a list	2048	2048
Picklist item character length	256	-
Workflow states defined for a work item type	32	16
Rules defined for a work item type	1024	1024
Actions defined for a rule	10	10
Portfolio backlog levels defined for a process	5	5
Categories defined for a process	-	32
Global lists defined for a process	-	256
List items defined within a global list	-	1024
Work item attachment size	60 MB	60 MB

For additional restrictions and conformance requirements of the Hosted XML process model, see Customize a process when using Hosted XML.

> **NOTE**
> For the Hosted XML process model, you can define an approximate total of 10K items for all global lists specified across all WITs.

The following table lists the maximum number of objects that you can define for the Inheritance and On-premises XML process models. While these represent hard limits, practical limits may also apply.

OBJECT	INHERITANCE	ON-PREMISES XML
Number of processes you can have in an organization	64	64
Work item types defined for a process	64	64
Fields defined for a collection	8192	1024
Fields defined for a process	1024	1024
Fields defined for a work item type	1024	1024
Picklists defined for a collection	1024	N/A
Picklist items defined for a list	2048	2048
Picklist item character length	256	N/A
Workflow states defined for a work item type	32	16
Rules defined for a work item type	1024	1024
Portfolio backlog levels defined for a process	5	5
Categories defined for a process	N/A	32
Global lists defined for a process	N/A	256
List items defined within a global list	N/A	1024

> **NOTE**
> For the On-premises XML process model, you can define an approximate total of 10K items for all global lists specified across all WITs.

The following table lists the maximum number of objects that you can define for the ON-premises XML process model. While these represent hard limits, practical limits may apply.

OBJECT	ON-PREMISES XML
Number of processes you can have in an organization	64
Work item types defined for a process	64
Fields defined for a collection	1024
Fields defined for a process	1024
Fields defined for a work item type	1024

OBJECT	ON-PREMISES XML
Picklists defined for a collection	N/A
Picklist items defined for a list	2048
Picklist item character length	N/A
Workflow states defined for a work item type	16
Rules defined for a work item type	1024
Portfolio backlog levels defined for a process	5
Categories defined for a process	32
Global lists defined for a process	256
List items defined within a global list	1024
Size of imported process template	2 GB

> **NOTE**
> For the On-premises XML process model, you can define an approximate total of 10K items for all global lists specified across all WITs.

Practical limits

We recommend that you consider the following guidance in order to minimize performance issues.

- Minimize the number of custom fields you define. All custom fields contribute to the total allowed for a process, collection, or organization. Note that you can specify different behavior for the same field in a different WIT. That is, you can specify different rules, picklists, and more.
- Minimize the number of rules you define for a WIT. While you can create multiple rules for a WIT, addition rules can negatively impact performance when a user adds and modifies work items. When users save work items, the system validates all rules associated with the fields for its work item type. Under certain conditions, the rule validation expression is too complex for SQL to evaluate.
- Minimize the number of custom WITs you define.

- Minimize the number of custom fields you define. All custom fields contribute to the total allowed for a process, collection, or organization. Note that you can specify different behavior for the same field in a different WIT. That is, you can specify different rules, picklists, and more.
- Minimize the number of rules you define for a WIT. While you can create multiple rules for a WIT, addition rules can negatively impact performance when a user adds and modifies work items. When users save work items, the system validates all rules associated with the fields for its work item type. Under certain conditions, the rule validation expression is too complex for SQL to evaluate.
- Minimize the number of custom WITs you define.

- Minimize the number of reportable fields you define. Reportable fields impact performance of your data warehouse.

> **NOTE**
>
> **Work Item Rules Validation Exceeds SQL Limits**: A single SQL expression is defined per project to validate work items whenever they are created or updated. This expression grows with the number of rules you specify for all work item types defined for the project. Each behavioral qualifier specified for a field results in an increase in the number of sub-expressions. Nested rules, rules that apply only on a transition or conditioned on the value of some other field, cause more conditions to be added to an IF statement. Once the expression reaches a certain size or complexity, SQL can't evaluate it any more and generates an error. Removing some WITs or eliminating some rules, can resolve the error.

Rate limits

To reduce costs and to enhance scalability and performance, Azure DevOps Services, like many Software-as-a-Service solutions, uses multi-tenancy. To ensure good performance and reduce the likelihood of outages, Azure DevOps Services limits the resources individuals can consume and the number of requests they can make to certain commands. When these limits are exceeded, subsequent requests may be either delayed or blocked.

Most rate limits are reached through REST API calls or non-optimized queries. To learn more, see the following articles:

- Rate limits
- Best practices (to avoid hitting rate limits)

Migrate and import limits

When determining to migrate from on-premises to Azure DevOps Services, there are several size limits that you may encounter. These limits include:

- Database size is above the recommended size
- Largest table size is above the recommended size
- The database metadata size is above the supported size

To learn more, see Migrate data from Azure DevOps Server to Azure DevOps Services and Troubleshoot import and migration errors.

Related articles

- Guidance to create high-performing queries
- About process customization and inherited processes
- Create an Inheritance process
- Best practices
- Naming restrictions and conventions

- Guidance to create high-performing queries
- Customize your work tracking experience
- About process customization and inherited processes
- On-premises XML process customization
- Rules and rule evaluation
- Naming restrictions and conventions

- Guidance to create high-performing queries
- Customize your work tracking experience
- On-premises XML process customization
- Rules and rule evaluation

- Naming restrictions and conventions

Related resources

- Tags Manager
- WIQL Editor
- Process Template Editor

Promote an Agile culture within your team

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

> **NOTE**
> Are you new to Agile? Learn more about Agile Culture and Scaling Agile to Large Teams.

As your team grows, you want your tools to grow with it. And, if you're an enterprise adopting Agile methodologies, you want your Agile tools to support the business goals of your enterprise.

However, to successfully scale Agile requires addressing both the culture and tools within your organization.

Alignment enables autonomy

Organizations that aspire to be agile need to consider the twin obligations of creating alignment across the enterprise and supporting team autonomy. Teams need autonomy to be efficient. And, enterprises need alignment across teams and the organization to be efficient.

Too much alignment with insufficient team autonomy leads doesn't support innovation or agility of teams to get things done. Too little alignment with each team running their own program doesn't provide the insight and coordination required to meet business goals.

With the right level of alignment across the organization and team autonomy, individuals are empowered to innovate and inspired to collaborate to meet business goals.

Create alignment and support autonomy

As you plan how you want to grow your Agile tool set, consider the following areas. These areas are key to creating enterprise alignment while developing team autonomy.

Area

Create alignment

Support autonomy

Product vision

The organization defines the goals and roadmap for the organization. Goals can be defined as epics and features that show up on the portfolio backlog.

Teams determine how to best meet the roadmap. Teams break down goals into user stories or product backlog items using their team backlogs.

Team structure

Based on business goals, organizations determine the number and size of teams. Vertically structured feature teams lead to greater autonomy and efficiency.

With teams, there should be some established roles, such as product owner and development leads, but also room to rotate roles. For example, team members can take turns acting as Scrum Master, developing sprint demos, running sprint retrospectives, or crafting sprint emails.

Development cadence

Agile organizations need to release products and feature updates at regular intervals. Establishing regular release and sprint schedules promotes the rhythm of the business.
Each sprint--a time boxed iteration of constant duration between two and four weeks—includes planning, executing, delivering value, reflecting, and engaging in continuous improvement.

All teams manage their work within the set sprint cadence. Teams provide input into the length of sprint that works best for them.
Teams choose the Agile methods that work for them, Scrum, Kanban, or a mix of both. Teams also take ownership of starting and acting on their own set of continuous improvement practices.
It's possible for some teams to execute in shorter sprints. For example, if an organization sets a 2-week sprint cadence, some teams may choose to operate in 1-week sprints, while still aligning with the organizational schedule.

Communication cadence

Just as sprints bring a natural rhythm to the flow of work, so too do regular communications. By setting expectations for the types of communications they want to see to stay aligned and how often they occur, organizations naturally create alignment across teams and the enterprise.
Team sprint emails, bug bar status, and release team feature delivery status are examples of such regular communications.

Teams determine the details that they communicate and who will develop the communication. Their sprint emails may contain a summary of previous sprint accomplishments and next sprint plans or include a demo of recently completed features.

Quality

Each organization needs to set the criteria and standards by which they assess quality and set expectations for quality standards. A few ways they define the criteria is to set exit criteria for new feature development, standards to manage technical debt, and bug caps for teams or individuals.
Also, they can monitor bug status and trends by creating bug dashboards.

Teams choose how they meet the quality standards. They may stage bug bashes for new features or at the end of each sprint. They may choose an individual to act as a bug shield on a rotating basis.

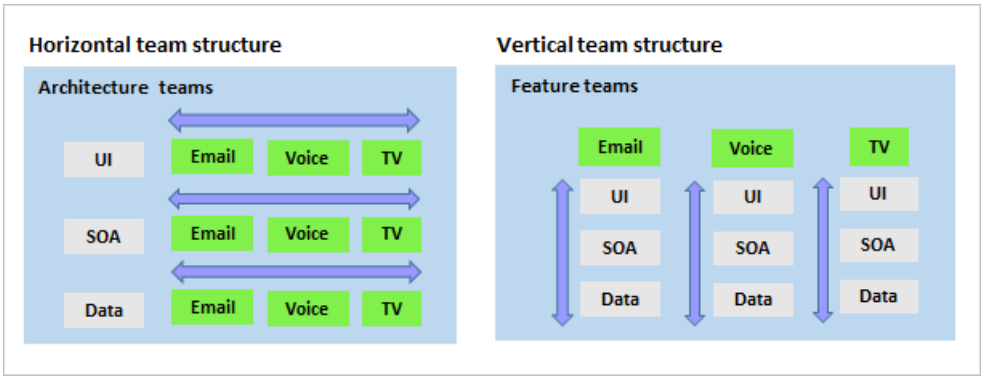
Manage risk, track work

The organization determines how each functional unit will communicate status and risk. They establish a "contract of communication" as to the minimum required information the organization needs.
Also, the organization provides the infrastructure to reduce risks. The organization owes the teams anything they can do to reduce risks that are common across teams.

Beyond meeting the needs set by the organization, teams determine any other details they need to manage and track to reduce their risks. Whether they use a white board with sticky notes or a full Gantt chart, they manage the details.
For example, teams may add a backlog item to track a dependency they have on another team. Or, they may track their risks via a list of issues or impediments. Also, teams regularly contribute to improving the process and infrastructure to support the organizations ability to manage risk and gain insights.

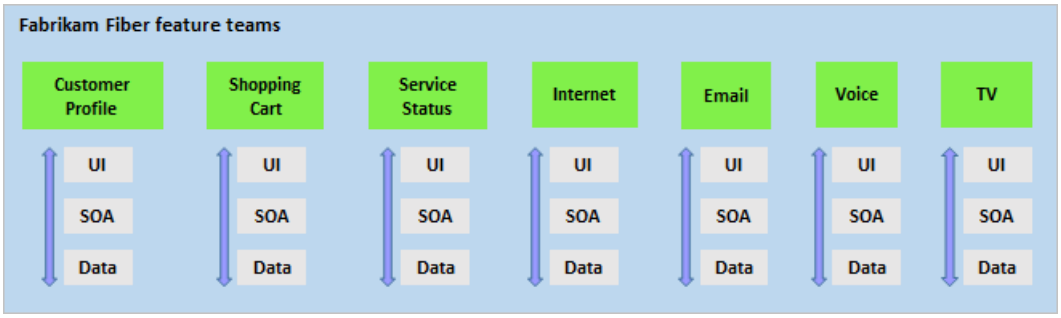
Feature teams

As you scale, one of the most important tasks to consider is how you structure your teams. Traditionally, horizontal team structures divide teams according to the software architecture: user interface, service-oriented architecture, and data teams.



However, with the adoption of Agile practices, vertical team structures that span the architecture have been shown to provide greater team autonomy. Vertical teams can deliver on the features they own by working across the software architecture. They also spread the knowledge needed to work at all architecture levels throughout all the teams.

Configure your teams along the value streams your organization wants to deliver. For example, Fabrikam Fiber, organizes their teams into the following seven feature teams.



Each team plans the features they'll deliver. They have the autonomy to determine how they'll structure the data, architect the services, and design of the web and mobile user interfaces. They plan in adherence with the quality standards set by the organization and to which all teams contribute.

Configure your Agile tools to scale

As your organization grows, you can scale your Agile tools in the following ways.

- **Add teams and filtered backlog views**: You add teams to support team autonomy and provide them with the tools they can configure and manage that supports how they want to work. These tools include product backlogs, Kanban boards, sprint backlogs and taskboards, and many others.

  Also, you can configure teams to support a hierarchy of backlogs and portfolio backlogs so that portfolio managers can review priority and progress across several teams.

- **Set up sprints and releases**: You can structure your iterations to support a flat set of sprints, or a set of sprints embedded within scheduled releases. Each team activates the set of sprints and releases that they need to participate in.

- **Manage portfolios**: by setting up a hierarchy of teams and backlogs and activating portfolio backlogs. Feature teams focused on a subset of the product backlog can remain focused on just their backlog. Portfolio managers who want to view and organize backlogs to track progress and dependencies can manage portfolio backlogs of Features and Epics.

  If other portfolio backlogs are needed--for example, Scenarios or Initiatives--you can add them as well.

- **Configure dashboards**: With team dashboards, you can configure many charts that track progress within a team or across teams. Specifically, you can add status and trend charts based on queries you create.

- **Group or categorize work:** There are several ways to group work that you want to track. Backlogs filter work items based on team area assignments. And, portfolio backlogs allow you to group backlog items under Features and Epics.

  If you want to track and report on work items based on other groupings, you can. You can add tags to work items and then filter backlogs or queries based on tags. Also, you can add subarea paths to represent more granular feature areas.

- **Add folders and use team favorites:** As your teams grow, you'll see a growing list of work item queries, build definitions, and source code folders. By using folders, subfolders, and team favorites, you can manage many of these lists more easily. You can add team favorites for shared queries, source code, and build definitions.

Related articles

Before you can create or work with any of the Agile tools, you need a project. If you don't have one yet, you can create one.

If you're ready to move from one team to two teams, or configure several teams, see Add teams. To add a team administrator or configure team assets, see Manage teams and configure team tools.

For more information, see these articles:

- Practices that scale
- Visibility across teams
- Review team delivery plans
- Implement Scaled Agile Framework® to support epics, release trains, and multiple backlogs.

Agile culture industry resources

- Nexus, the Exoskeleton of Scaled Scrum
- Culture over process
- The Culture Game - Tools for the Agile Manager
- Scaled Agile Framework (SAFe)
- Scaling Agile Software Development, - Disciplined Agility at Scale (White Paper)

Scale with teams and not projects

Often, organizations look at adding a project for each software development project.

Adding teams to scale your tools rather than adding projects is recommended for the following reasons:

- **Visibility:** It's much easier to view progress across all teams
- **Tracking and auditing:** It's easier to link work items and other objects for tracking and auditing purposes
- **Maintainability:** You minimize the maintenance of security groups and process updates.

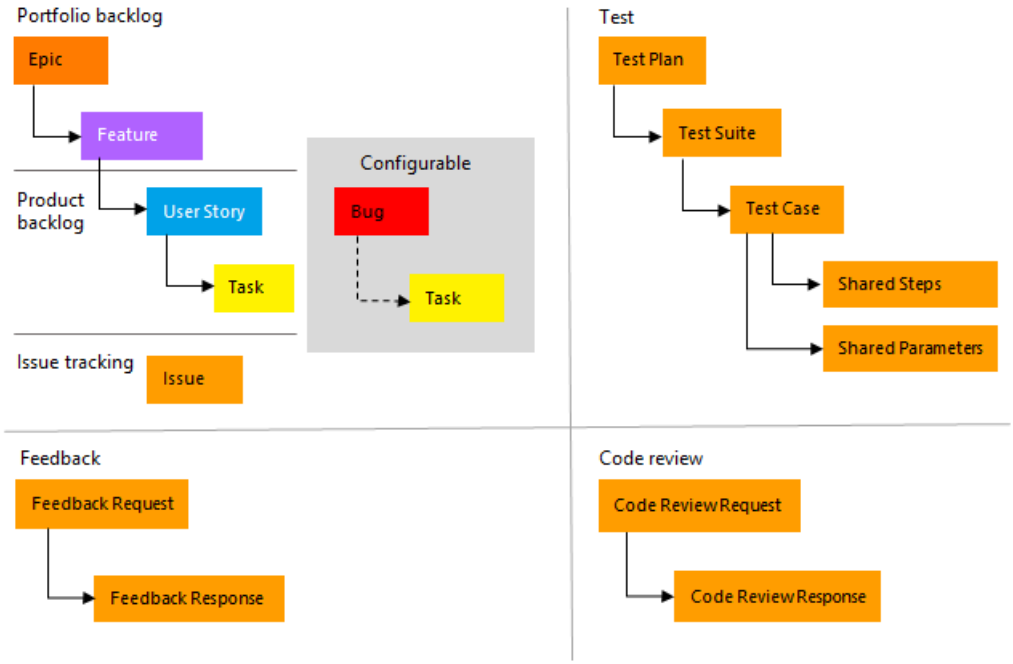
To learn more, see About projects and scaling your organization.

Agile process work item types

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Agile process supports the following work item types (WITs) to plan and track work, tests, feedback, and code review. With different WITs you can track different types of work—such as features, user stories, and tasks. These artifacts are created when you create a project using the Agile process. They're based on Agile principles and values.



Along with the WITs, teams have access to a set of work item queries to track information, analyze progress, and make decisions.

NOTE

You can customize the work tracking system for your project by creating and customizing an inherited process and applying that process to your project. To learn more, see Inheritance process model.

NOTE

You can customize the work tracking system for your project by customizing an Inherited process or an On-premises XML process. To learn more, see Inheritance process model or On-premises XML process customization.

The latest version of each process uploads automatically when you install or upgrade to the latest version of Azure DevOps Server. Additional artifacts, such as SQL Server reports are only available when you connect to a project. Other resource requirements apply.

NOTE

You can customize the work tracking system for your project by customizing an On-premises XML process. To learn more, see On-premises XML process customization.

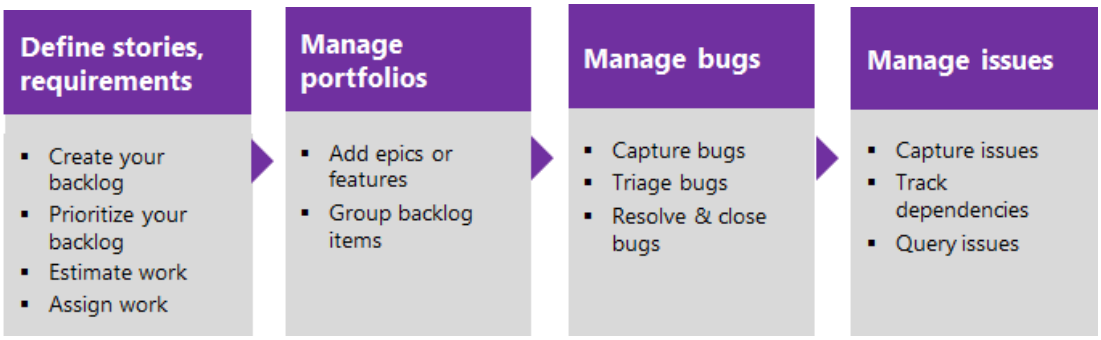
The latest version of each process uploads automatically when you install or upgrade to the latest version of Azure DevOps Server. Additional artifacts, such as SQL Server reports are only available when you connect to a project. Other resource requirements apply.

Plan and track work with Agile

You build your project plan by creating a backlog of user stories that represent the work you want to develop and ship. You track bugs, tasks, and blocking issues using the bug, task, and issue WITs. To support portfolio management, teams create features and epics to view a roll up of user stories within or across teams. For details about using Agile WITs, see Agile process work item types and workflow.

The essential flow for getting started is as shown. To get started using Scrum or Kanban tools, see Get started with Agile tools to plan and track work.

Click one of the following images to go to the linked article.



NOTE

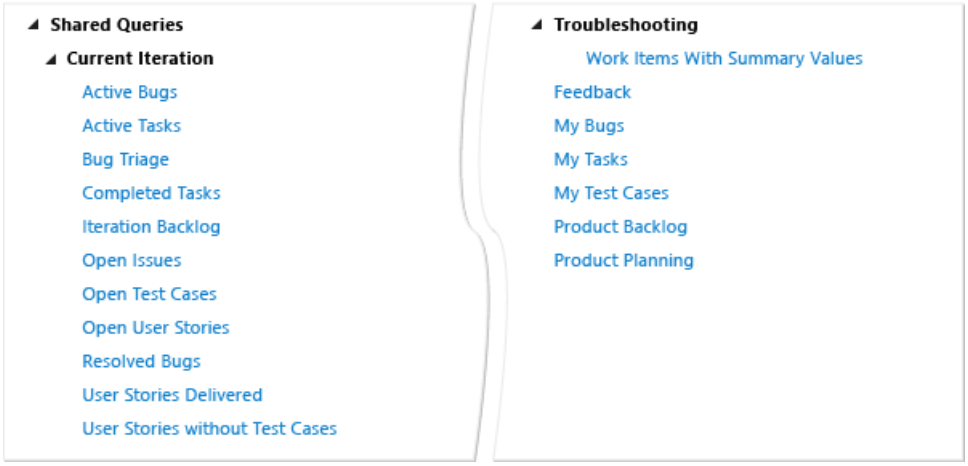
A work item is a database record that contains the definition, assignment, priority, and state of work. Work item types define the template of fields, workflow, and form for each type. Work items can be linked to each other to support tracking dependencies, roll up of work, and reports.

List work items by using queries

You can manage your workload more effectively by frequently reviewing the status of user stories and tasks. You can use the shared work item queries to list work items for a current sprint or the product backlog.

NOTE

New projects no longer define a default set of Shared Queries at the time of project creation. The definitions for Shared Queries have been removed from the process template. For on-premises deployments, you can add them to a custom process template as described in Add work item queries to a process template.



Descriptions of predefined queries are listed later in this article.

You can view and run queries from the web portal or from the Team Explorer plug-in to Visual Studio. You can modify a query using the query editor to apply different filter criteria. Also, you can add queries to team dashboards.

Quick tips on shared queries

If you are new to Azure Boards, work tracking, and shared queries, review these tips to learn how you can manage work more effectively:

- To find work items that are assigned to you, add **@Me** as the value for the Assigned To field in one of the query clauses.
- All valid users with standard access can create queries and folders under the **My Queries** area. To create queries and query folders under **Shared Queries**, you must have the Contribute permission set and have been assigned Basic access or greater. For more information, see Set permissions on queries.
- You can modify any query by adding criteria to focus on a product area, an iteration, or another field. To modify a query, open the query editor.
- You can open any query in Excel where you can update the fields of one or more work items and publish your changes to the database for tracking work items.
- You can visualize status or progress by creating a pie-chart, column chart, or trend chart for flat-list queries.

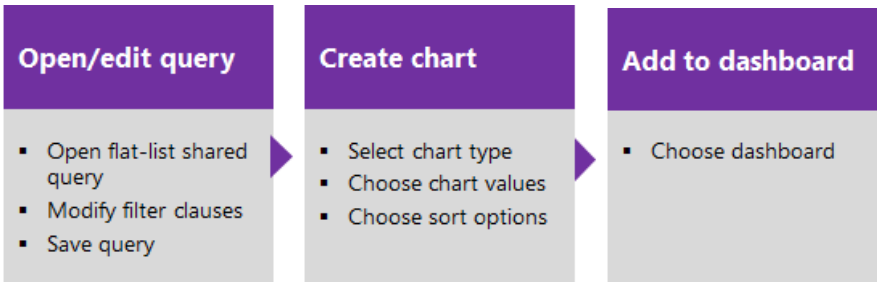
> **IMPORTANT**
> Starting with Visual Studio 2019, the Azure DevOps plug-in for Office has deprecatedsupport for Microsoft Project. Project integration and the **TFSFieldMapping** command is not supported for Azure DevOps Server 2019 and later versions, including Azure DevOps Services. You can continue to use Microsoft Excel.

Monitor progress

All processes—Agile, Scrum, and CMMI—support building status and trend charts and dashboards. Also, several charts are automatically built based on the Agile tools you use. These charts display within the web portal.

Create light-weight charts

To get started, you can define a shared flat query and create a chart based on your tracking interests. Chart types include status—pie, bar, column, stacked bar, and pivot—and trend—stacked area, line, and area—charts.



Analytics widgets and Power BI reports

The Analytics Service can answer quantitative questions about the past or present state of your projects. You can add Analytics widgets to a dashboard or use Power BI to create charts and reports.

To learn more, see What is the Analytics Service?

SQL Server reports

If your project collection and the project are configured with SQL Server Analysis Services and Reporting Services, you'll have access to many Agile reports. For these reports to be useful, teams must complete certain activities, such as define build processes, link work items, and update status or remaining work.

If you need to add reporting services or update reports to the latest versions, see Add reports to a project.

Related articles

Before you start tracking work, you must have a project. To create one, see Create a project.

If you have a project, start tracking work:

- Add work items to manage a project - to gain more familiarity with the work item form features
- Create a backlog - to develop your product backlog
- Kanban - to start working in Kanban
- Plan a sprint - to start working in Scrum
- Excel.

For more information on Agile tools:

- Team assets
- Best tool to add, update, and link work items

Agile process versions

As updates are made to the Agile process template, the version number is updated. The following table provides a mapping of the versioning applied as updates are made to the Azure DevOps on-premises process templates. For Azure Boards, the latest version is always used. Each template provides a `version` element. This element specifies a major and minor version.

ON-PREMISES VERSION	AGILE PROCESS NAME	MAJOR VERSION
Azure DevOps Services Azure DevOps Server 2022	Agile	18
Azure DevOps Server 2020 Azure DevOps Server 2019	Agile	17
TFS 2018	Agile	16

For a summary of updates made to process templates, see Changes made to process templates.

Agile process predefined queries

Product backlog and feedback queries

Product owners can use the shared queries that are defined in the following table to plan and track user stories that compose the product backlog.

Shared query	Description
Product Backlog	Provides a tree list of all user stories that are in a New, Active or Resolved state and sorts them by rank.
Product Planning	Provides a flat list of all user stories that are not in a Removed state, and have not been closed in the last 90 days.
Feedback	Lists all feedback responses that are in an Active state.

Iteration planning queries

The following table describes the shared queries that are listed under the **Current Iteration** folder. These queries find work items that are assigned to a specified iteration. As you plan more iterations, you can modify these queries to specify a different iteration and then save them to other folders that you create, such as **Iteration 2** or **Iteration 3**.

The project administrator for each project defines area and iteration paths for that project so that the team can track progress by those designations.

Shared query	Description
Active Bugs	Lists all active bugs and sorts them by rank, priority, and severity.
Active Tasks	Lists all active tasks and sorts them by rank, priority, and severity.
Bug Triage	Lists all active bugs that aren't assigned to a team member. The Triage Workbook references this query.
Completed Tasks	Lists all tasks that have been closed and sorts them by rank, priority, and severity.
Iteration Backlog	Lists all user stories and their linked tasks and sorts the stories by rank and priority.
Open Issues	Lists all issues under the specified iteration path that aren't closed and any tasks that are linked to the issues and

then sorts the issues by rank and priority.

The Issues Workbook references this query.

Open Test Cases

Lists all test cases that aren't closed and sorts them by priority.

Open User Stories

Lists all active user stories and sorts them by their stack rank.

Resolved Bugs

Lists all resolved bugs and sorts them by rank, priority, and severity.

User Stories

Lists all user stories that aren't closed and sorts them by priority and then ID,

User Stories without Test Cases

Lists all user stories that don't have a link to a test case. Stories are sorted by ID.

> **TIP**
> Queries listed under the **Current Iteration** folder do not automatically update when a new iteration becomes current. The current iteration is based on the dates that you assign to your sprint schedules. You must manually update the iteration path of each query to have it point to the iteration path that corresponds to the current iteration. Or, you can edit the shared query to use the **@CurrentIteration** macro.

Find tasks with summary values

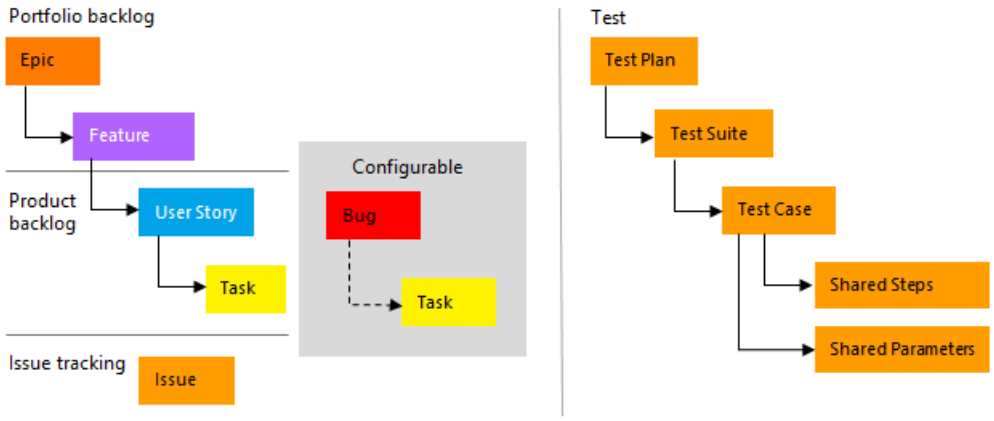
The **Work Items With Summary Values** shared query, which is located in the **Troubleshooting** folder, lists all tasks that have child tasks and that contain non-zero values in the Remaining Work or Completed Work fields. This query is designed to find tasks that report work effort that is already accounted for in their child tasks. For the hours to be counted only once, summary tasks shouldn't be assigned any hours. For more information, see Address inaccuracies published for summary values.

Agile process work item types for workflow management in Azure Boards

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Teams use the work item types (WITs) provided with the Agile process. Work item types help your team to plan and track progress of software projects. You define user stories to manage the backlog of work. Then, using the Kanban board, you track progress by updating the status of those stories.



To gain insight into a portfolio of features, scenarios, or user experiences, product owners and program managers map user stories to features. When a team works in sprints, they define tasks that automatically link to user stories. If you are new to the Agile process, review the section Plan and track work with Agile to get started.

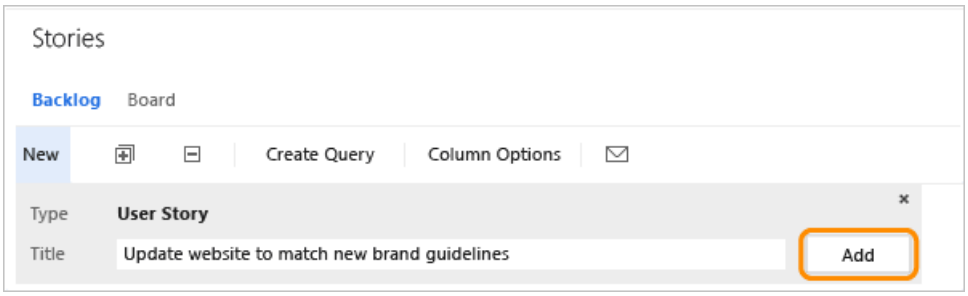
Using the web portal or Microsoft Test Manager, testers can create and run test cases. Bugs and issues are used to track code defects and blocking issues.

> **NOTE**
>
> Work item tracking forms and features available to you differ depending on whether you open the form from the web portal or Visual Studio Team Explorer. Forms and guidance provided in this article reflect those available with the new form experience (Azure DevOps Services and TFS 2017 and later versions).
>
> For guidance on the work item form for TFS 2015 or earlier versions or Visual Studio Team Explorer, see Add work items and select the earlier TFS version from the content version selector (above the table of contents).

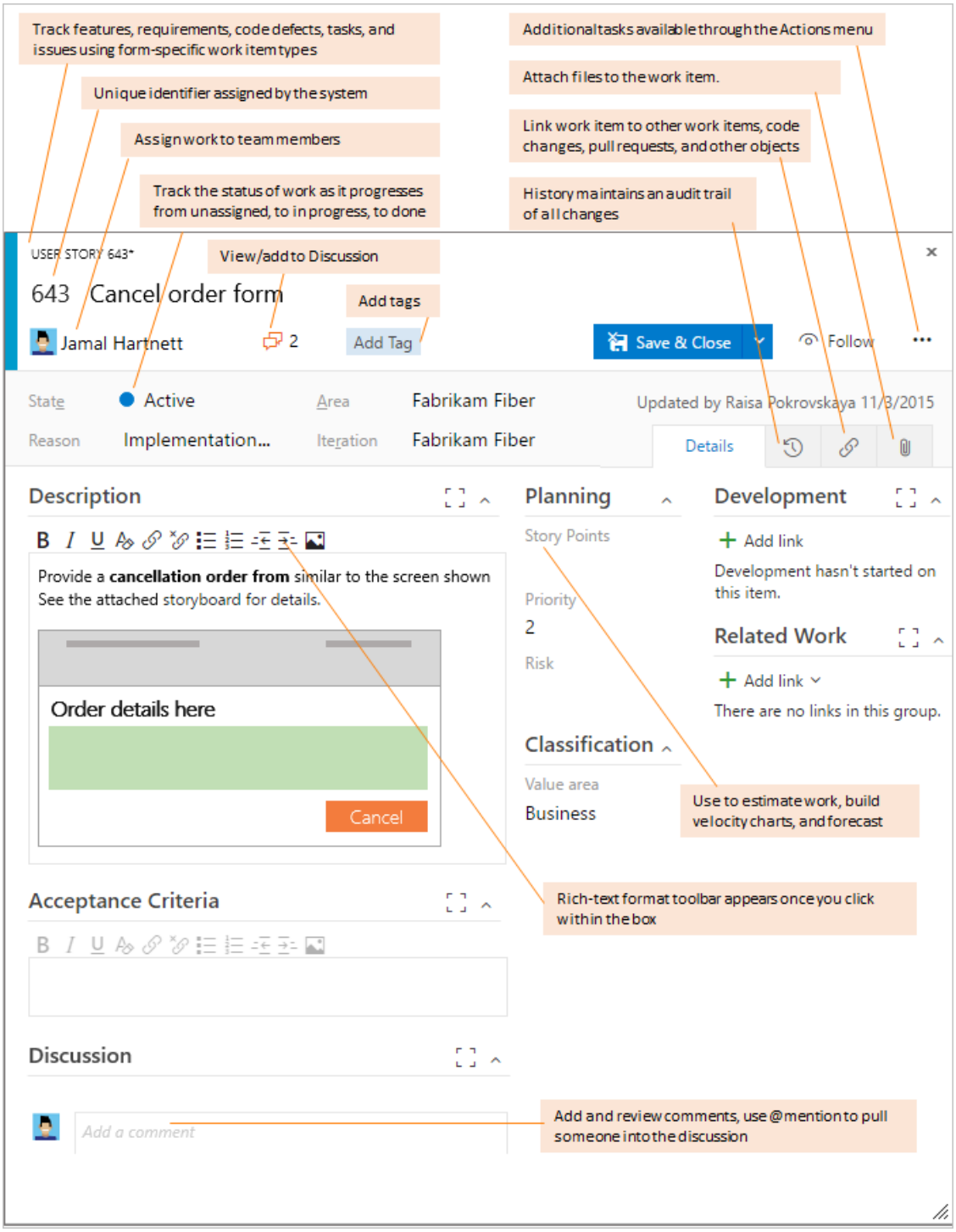
Define user stories

User stories define the applications, requirements, and elements that teams need to create. Product owners typically define and stack rank user stories. The team then estimates the effort and work to deliver the highest priority items.

Create user stories from the quick add panel on the product backlog page. From that page, you can also drag-and-drop items to reorder them or map them to features.



Later, you can open each user story to provide more details and estimate the story points.



By defining the **Story Points**, your team can use the forecast feature and velocity charts to estimate future sprints or work efforts. By prioritizing the user stories on the backlog page (that's captured in the Stack Rank field), product owners can indicate which items should be given higher priority.

Use the following guidance and that provided for fields used in common across work item types when filling out the form.

Field/tab

Usage

Description

For user stories, provide enough detail for estimating how much work will be required to implement the story. Focus on who the feature is for, what users want to accomplish, and why. Don't describe how the feature should be developed. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

Acceptance Criteria

Provide the criteria to be met before the bug or user story can be closed. Before work begins, describe the customer acceptance criteria as clearly as possible. Conversations between the team and customers to define the acceptance criteria will help ensure that your team understands your customers' expectations. You can use the acceptance criteria as the basis for acceptance tests to more effectively evaluate whether an item is satisfactorily completed.

Value Area

The area of customer value addressed by the epic, feature, requirement, or backlog item. Values include:

- **Architectural**: Technical services to implement business features that deliver solution
- **Business**: Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default)

Story Points

Estimate the amount of work required to complete a user story using any numeric unit of measurement your team prefers.

Agile velocity charts and forecast tools reference the values in this field. For more information, see the Estimating white paper.

Priority

A subjective rating of the user story, feature, or requirement as it relates to the business. Allowed values are:

- **1**: Product can't ship without the feature.
- **2**: Product can't ship without the feature, but it doesn't have to be addressed immediately.
- **3**: Implementation of the feature is optional based on resources, time, and risk.

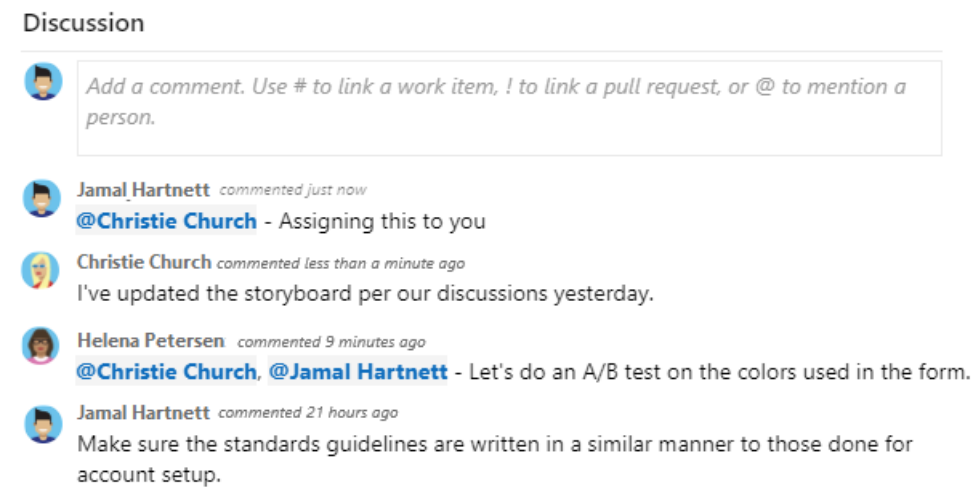
Risk

A subjective rating of the relative uncertainty around the successful completion of a user story. Allowed values are:

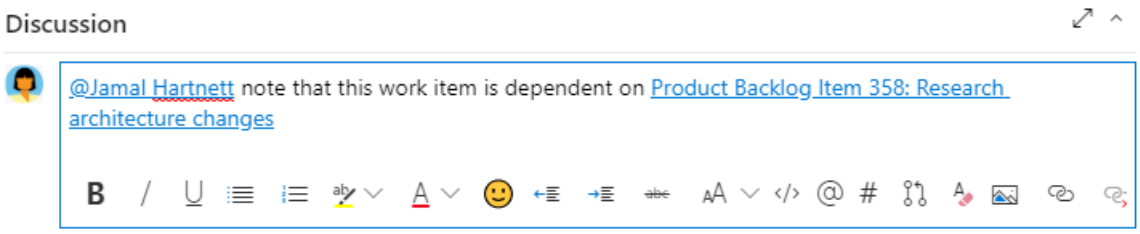
- **1 - High**
- **2 - Medium**
- **3 - Low**

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



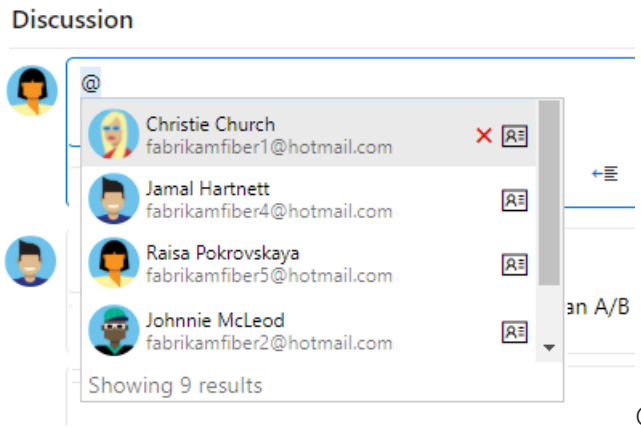
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the History field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons —@ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.



@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon

and then choose **Delete**.



> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

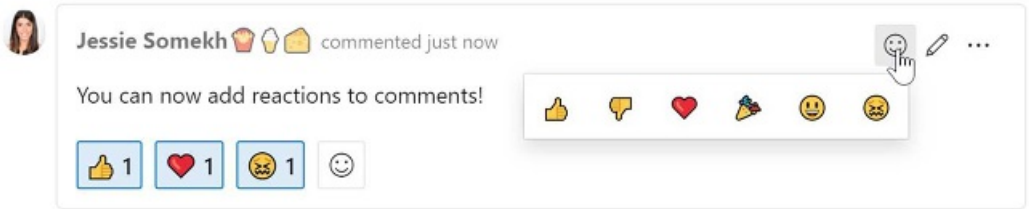
To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

You can't edit or delete comments once you've entered them.

> **IMPORTANT**
> For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

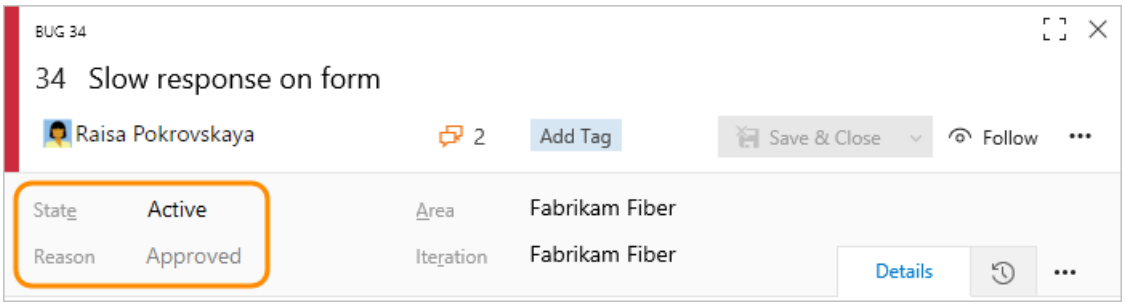
Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



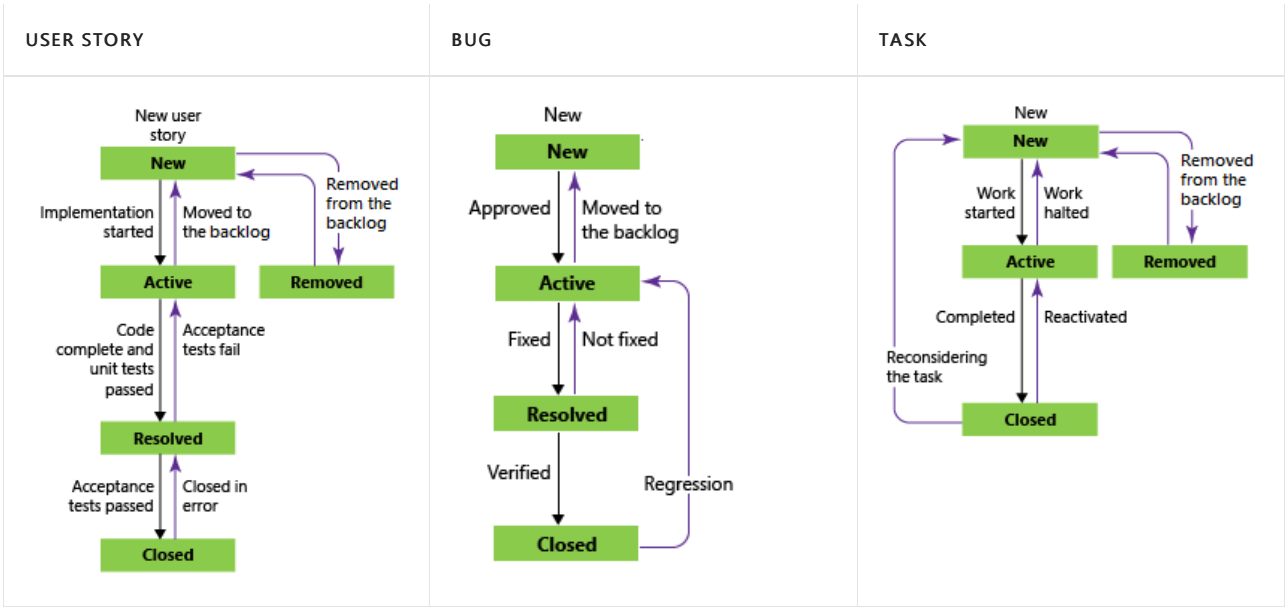
Track progress

As work progresses, you change the State field to update the status. Optionally, you can specify a reason. The state and reason fields appear on the work item form in the header area.



Agile workflow states

By updating the workflow, teams know which items are new, in progress, or completed. Most WITs support transition both forward and backward from each workflow state. These diagrams show the main progression and regression states of the user story, bug, and task WITs.

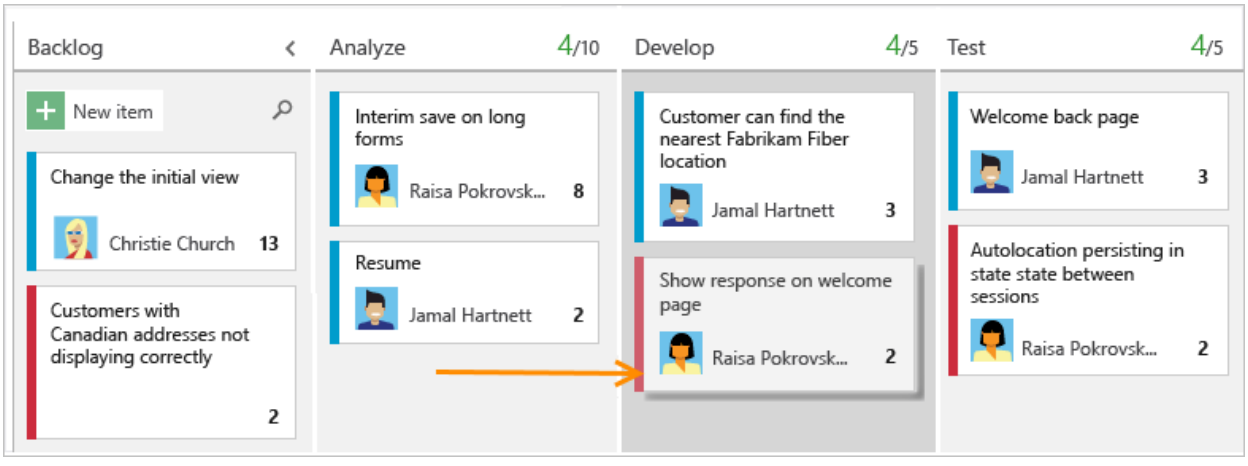


A typical workflow progression for a user story follows:

- The product owner creates a user story in the **New** state with the default reason, **New user story**
- The team updates the status to **Active** when they decide to complete the work during the sprint
- A user story is moved to **Resolved** when the team has completed all its associated tasks and unit tests for the story pass
- A user story is moved to the **Closed** state when the product owner agrees that the story has been implemented according to the Acceptance Criteria and acceptance tests pass.

Update status with Kanban or taskboards

Teams can use the Kanban board to update the status of requirements, and the sprint taskboard to update the status of tasks. Dragging items to a new state column updates both the State and Reason fields.



You can customize the Kanban board to support more swim lanes or columns. For more customization options, see Customize your work tracking experience.

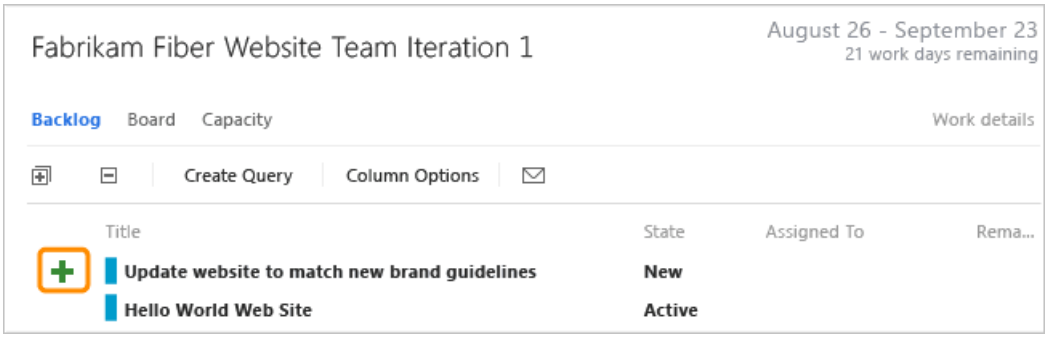
Map user stories to features

When you manage a suite of products or user experiences, you might want to view the scope and progress of work across the product portfolio. You can view the scope and progress of work by defining features and mapping user stories to features.

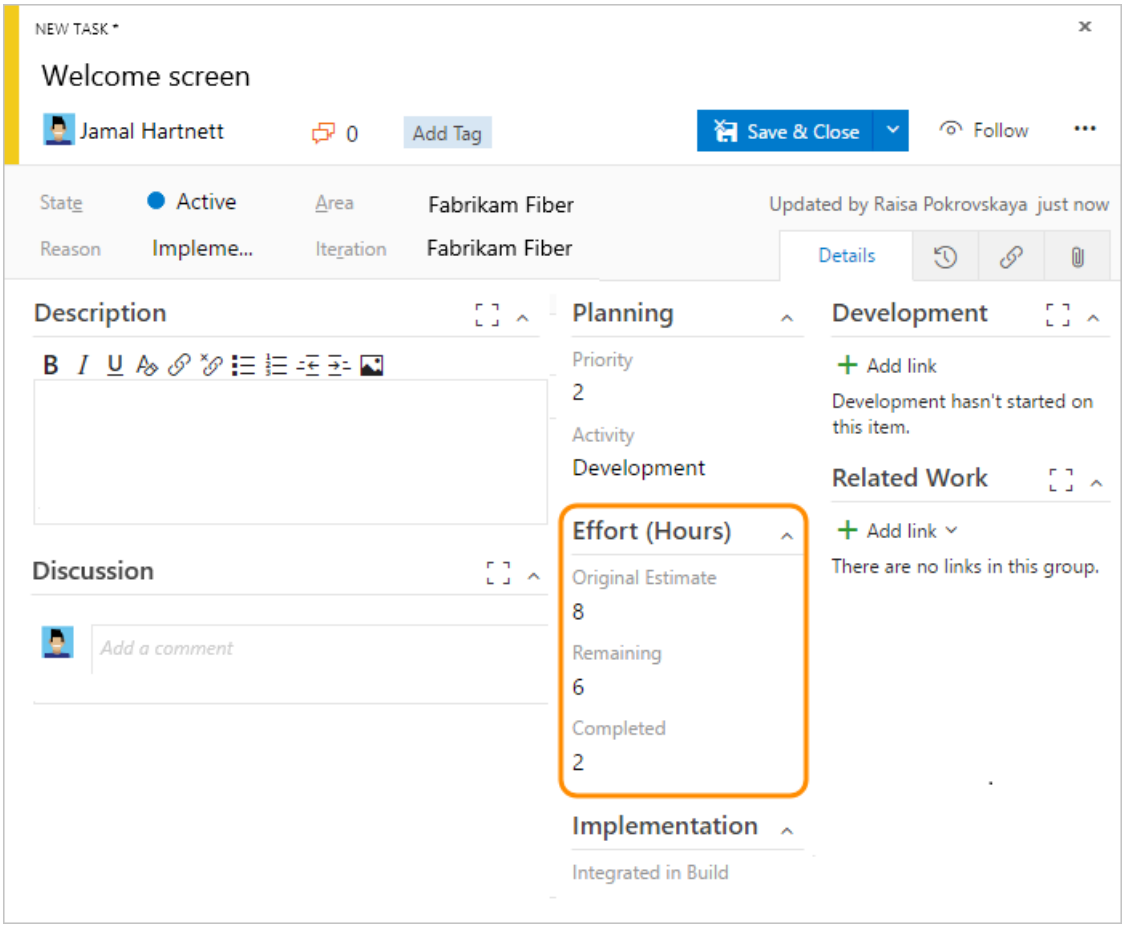
Using portfolio backlogs, you can drill down from one backlog to another to view the level of detail you want. Also, use portfolio backlogs to view a rollup of work in progress across several teams when you setup a hierarchy of teams.

Define tasks

When your team manages their work in sprints, they can use the sprint backlog page to break down the work to be accomplished into distinct tasks.



Name the task and estimate the work it will take.



Using Agile processes, teams forecast work and define tasks at the start of each sprint. Each team member then

performs a subset of those tasks. Tasks can include development, testing, and other kinds of work. For example, a developer defines tasks to implement user stories, and a tester defines tasks to write and run test cases.

When teams estimate work using hours or days, they define tasks and the **Remaining Work** and **Activity** (optional) fields.

Field/tab

Usage

Original Estimate

The amount of estimated work required to complete a task. Typically, this field doesn't change after it's assigned.

You can specify work in hours or in days. There are no inherent time units associated with this field.

Remaining Work

The amount of work remaining to complete a task. As work progresses, update this field. It's used to calculate capacity charts, the sprint burndown chart, and the following (TFS only) reports: Burndown and Burn Rate, Remaining Work, and Status on All Iterations.

If you divide a task into subtasks, specify hours for the subtasks only. You can specify work in any unit of measurement your team chooses.

Completed Work

The amount of work spent implementing a task.

Activity

Select the type of activity this task represents when your team estimates sprint capacity by activity.

Integrated in Build

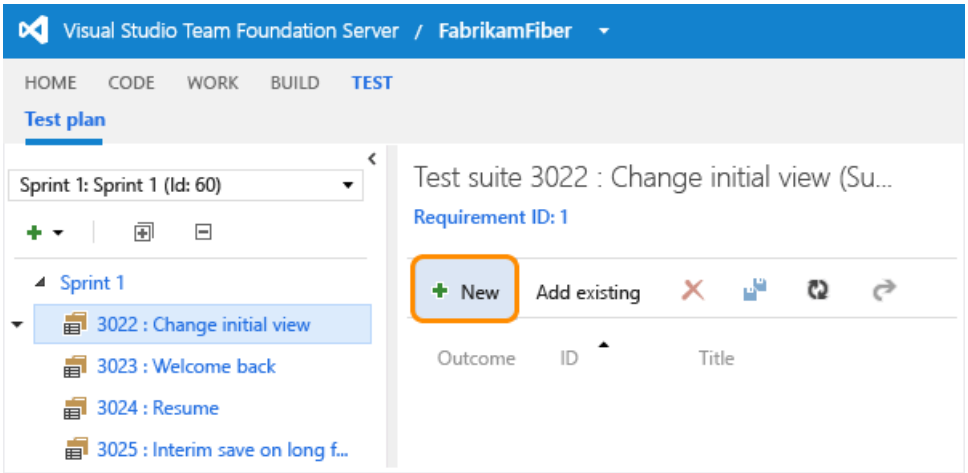
Product build number that incorporates the code or fixes a bug.

If you use Microsoft Project to assign resources and track a schedule.

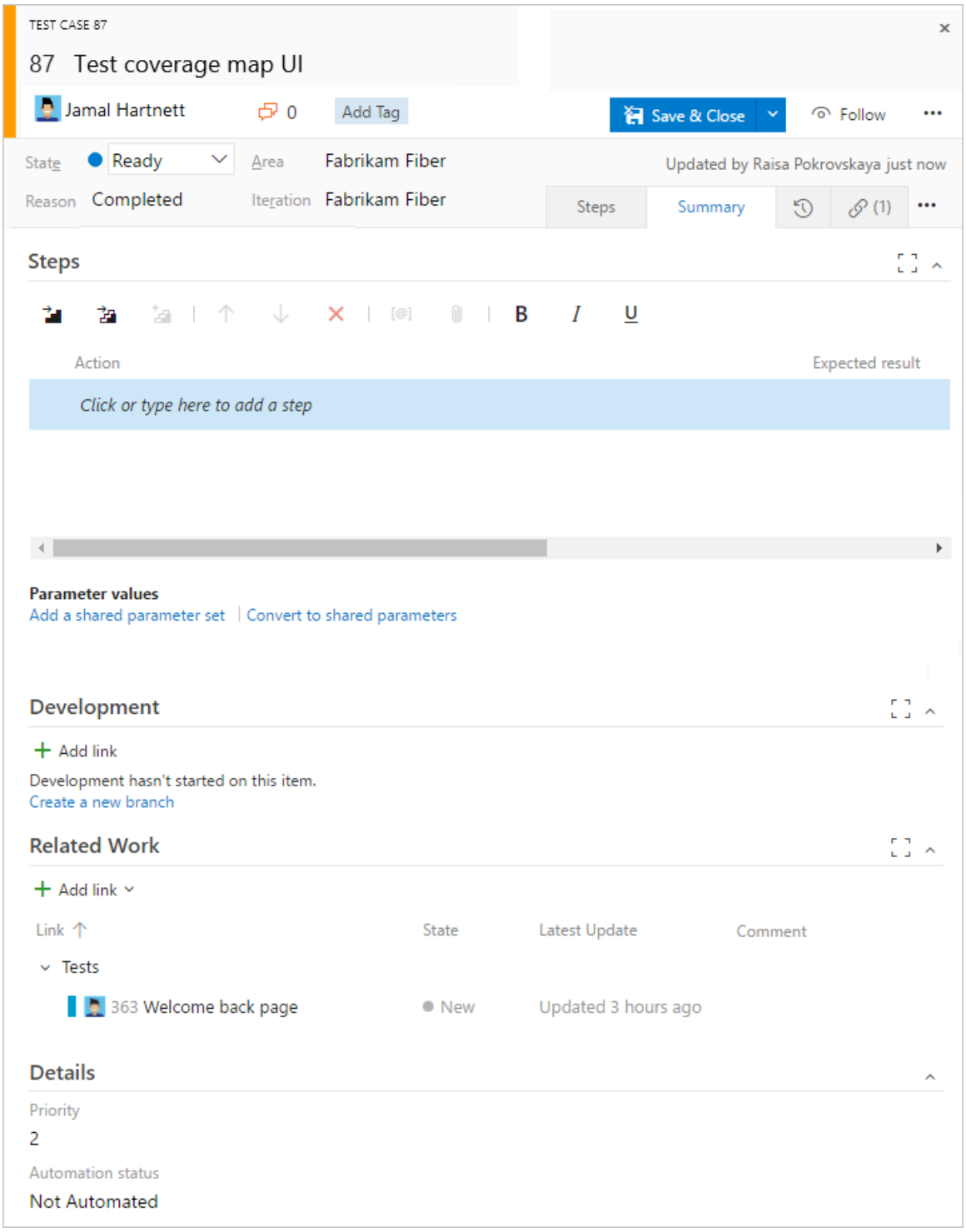
Track test progress

Test user stories

From the web portal or Test Manager, you can create test cases that automatically link to a user story or bug. Or, you can link a user story to a test case from the (links tab).



The test case contains multiple fields, many of which are automated and integrated with Test Manager and the build process. For a description of each field, see Query based on build and test integration fields.



The (links tab) captures the links to user stories and bugs in a test case. By linking user stories and bugs to test cases, the team can track the progress made in testing each item. By defining these links, you support information that appears in the Stories Overview Report report.

Track code defects

You can create bugs from the web portal, Visual Studio, or when testing with Test Manager.

Definitions for common work tracking fields

The following fields and tabs appear in most work items. Each tab is used to track specific information, such as history, links, or attachments. These three tabs provide a history of changes, view of linked work items, and ability to view and attach files.

The only required field for all work item types is **Title**. When the work item is saved, the system assigns it a

unique **ID**. The form highlights required field in yellow. For information about other fields, see Work item field index.

Field/tab

Usage

Title

Enter a description of 255 characters or less. You can always modify the title later.

Assigned To

Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu will list only team members or contributors to the project.

State

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current state.

Reason

Use the default first. Update it when you change state. Each State is associated with a default reason.

Area

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting.

To change the dropdown list of areas, see Add and modify area and iteration paths.

Iteration

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later, during a planning meeting.

To change the drop-down list of iterations, see Add and modify area and iteration paths.

 (History)

Review the audit trail that the system captures and capture additional information.

Every time that the work item is updated, information is appended to the history. History includes the date of the change, who made the change, and which fields were changed. You can also add formatted text to the history field.

 (Links)

Add all types of links, such as hyperlinks, changesets, source files, and so on.

This tab also lists all links defined for the work item.

 (Attachments)

Share more detailed information by adding files to the work item, such as email threads, documents, images, log files, or other file types.

Customize work item types

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance

process.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process or Customize the On-premises XML process model depending on the process model used by your project.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize the On-premises XML process model.

Related articles

Before you start tracking work, you must have a project. To create one, see Create a project.

If you have a project, start tracking work:

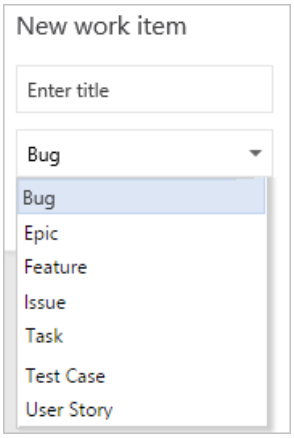
- Add work items to manage a project - to gain more familiarity with the work item form features
- Create a backlog - to develop your product backlog
- Kanban - to start working in Kanban
- Plan a sprint - to start working in Scrum
- Excel.

For more information on Agile tools:

- Team assets
- Best tool to add, update, and link work items

Track issues

Issues are used to track events that may block progress or shipping a user story. Bugs, on the other hand, are used to track code defects. You can add an issue from the New work item widget added to a team dashboard, or from the **New** menu on the Queries page.



Work items you add from the widget are automatically scoped to your team's default area and iteration paths. To change the team context, see Switch team context.

Track business value

You can use the Priority field to differentiate the value of various stories. Or, you can add a custom field to the User Story WIT that tracks the relative value of stories. To learn how, see Customize a field for a process.

Backlog list order

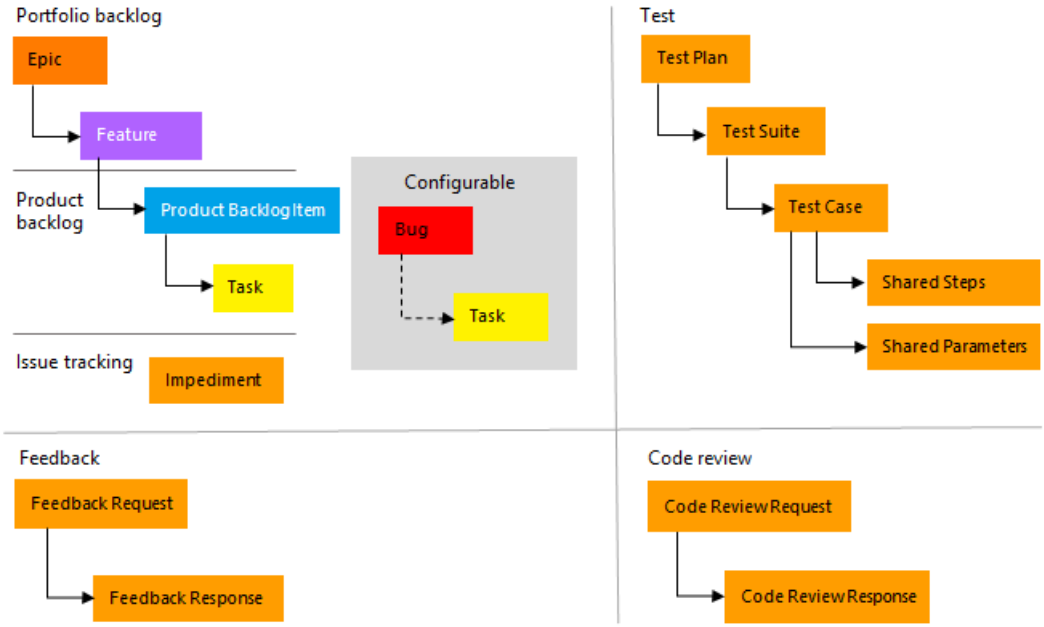
The Stack Rank field is used to track the relative ranking of user stories, however by default it doesn't appear on the work item form. The sequence of items on the backlog page is determined according to where you've added the items or moved the items on the page. As you drag items, a background process updates this field.

Manage your Scrum process template artifacts

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Scrum process supports the following work item types (WITs) to plan and track work, tests, feedback, and code review. With different WITs you can track different types of work—such as product backlog items, tasks, bugs, and more. These artifacts are created when you create a project using the Scrum process. They're based on Scrum principles and values.



Along with the WITs, teams have access to a set of work item queries to track information, analyze progress, and make decisions.

> **NOTE**
> You can customize the work tracking system for your project by creating and customizing an inherited process and applying that process to your project. To learn more, see Inheritance process model.

> **NOTE**
> You can customize the work tracking system for your project by customizing an Inherited process or an On-premises XML process. To learn more, see Inheritance process model or On-premises XML process customization.
>
> The latest version of each process uploads automatically when you install or upgrade to the latest version of Azure DevOps Server. Additional artifacts, such as SQL Server reports are only available when you connect to a project. Other resource requirements apply.

NOTE

You can customize the work tracking system for your project by customizing an On-premises XML process. To learn more, see On-premises XML process customization.

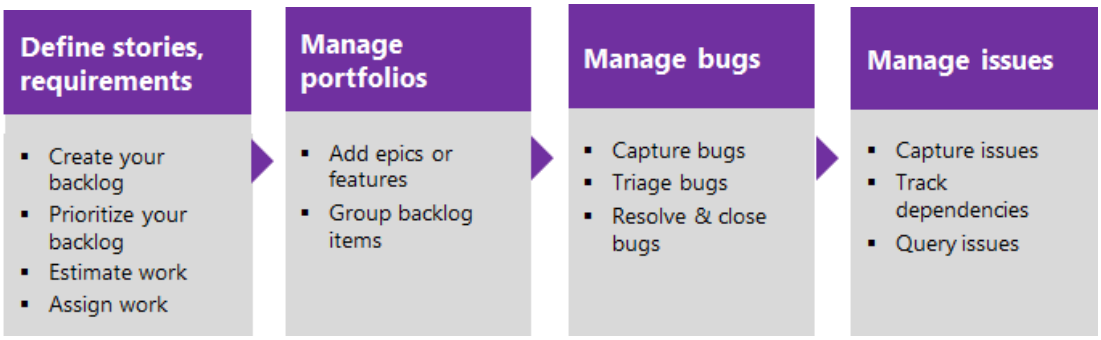
The latest version of each process uploads automatically when you install or upgrade to the latest version of Azure DevOps Server. Additional artifacts, such as SQL Server reports are only available when you connect to a project. Other resource requirements apply.

Plan and track work with Scrum processes

You build your project plan by creating a backlog of work items that represent the features, requirements, user stories, or other work to do. You track bugs, tasks, and blocking issues using the bug, task, and impediment WITs. To support portfolio management, teams create features and epics to view a roll up of their product backlog items within or across teams. For details about using Scrum WITs, see Scrum process work item types and workflow.

The essential flow for getting started is as shown. To get started using Scrum or Kanban tools, see Get started with Agile tools to plan and track work.

Click one of the following images to go to the linked article.



NOTE

A work item is a database record that contains the definition, assignment, priority, and state of work. Work item types define the template of fields, workflow, and form for each type. Work items can be linked to each other to support tracking dependencies, roll up of work, and reports.

Scrum work item types and workflow provides more details about using these WITs.

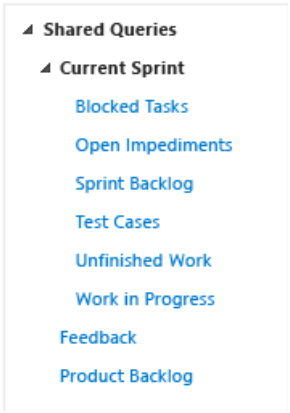
List work items

Define work item queries to list work items for a current sprint or the product backlog.

NOTE

New projects no longer define a default set of Shared Queries at the time of project creation. The definitions for Shared Queries have been removed from the process template. For on-premises deployments, you can add them to a custom process template as described in Add work item queries to a process template.

Or, use the shared queries that the Scrum process provides.



Descriptions of predefined queries are listed later in this article.

> **TIP**
> Queries listed under the **Current Iteration** folder do not automatically update when a new iteration becomes current. The current iteration is based on the dates that you assign to your sprint schedules. You must manually update the iteration path of each query to have it point to the iteration path that corresponds to the current iteration. Or, you can edit the shared query to use the **@CurrentIteration** macro.

You can view and run queries from the web portal or from the Team Explorer plug-in to Visual Studio. You can modify a query using the query editor to apply different filter criteria. Also, you can add queries to team dashboards.

Quick tips on shared queries

If you are new to Azure Boards, work tracking, and shared queries, review these tips to learn how you can manage work more effectively:

- To find work items that are assigned to you, add **@Me** as the value for the Assigned To field in one of the query clauses.
- All valid users with standard access can create queries and folders under the **My Queries** area. To create queries and query folders under **Shared Queries**, you must have the Contribute permission set and have been assigned Basic access or greater. For more information, see Set permissions on queries.
- You can modify any query by adding criteria to focus on a product area, an iteration, or another field. To modify a query, open the query editor.
- You can open any query in Excel where you can update the fields of one or more work items and publish your changes to the database for tracking work items.
- You can visualize status or progress by creating a pie-chart, column chart, or trend chart for flat-list queries.

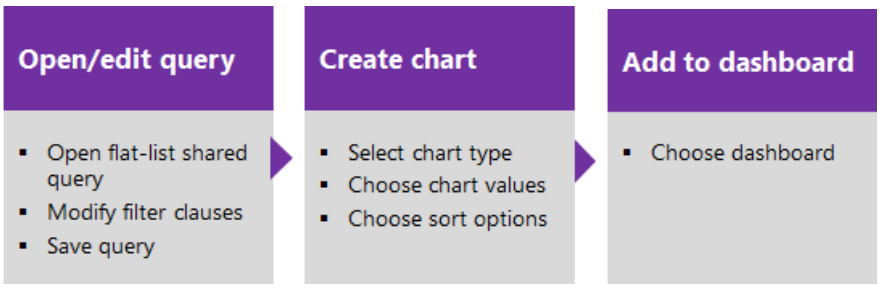
> **IMPORTANT**
> Starting with Visual Studio 2019, the Azure DevOps plug-in for Office has deprecatedsupport for Microsoft Project. Project integration and the **TFSFieldMapping** command is not supported for Azure DevOps Server 2019 and later versions, including Azure DevOps Services. You can continue to use Microsoft Excel.

Monitor work progress

All processes—Agile, Scrum, and CMMI—support building status and trend charts and dashboards. Also, several charts are automatically built based on the Agile tools you use. These charts display within the web portal.

Create light-weight charts

To get started, you can define a shared flat query and create a chart based on your tracking interests. Chart types include status—pie, bar, column, stacked bar, and pivot—and trend—stacked area, line, and area—charts.



Analytics widgets and Power BI reports

The Analytics Service can answer quantitative questions about the past or present state of your projects. You can add Analytics widgets to a dashboard or use Power BI to create charts and reports.

To learn more, see What is the Analytics Service?

Scrum process and SQL Server reports

If your project collection and the project are configured with SQL Server Analysis Services and Reporting Services, you'll have access to many Scrum reports. For these reports to be useful, teams must complete certain activities, such as define build processes, link work items, and update status or remaining work.

If you need to add reporting services or update reports to the latest versions, see Add reports to a project.

Related articles

Before you start tracking work, you must have a project. To create one, see Create a project.

If you have a project, start tracking work:

- Add work items to manage a project - to gain more familiarity with the work item form features
- Create a backlog - to develop your product backlog
- Kanban - to start working in Kanban
- Plan a sprint - to start working in Scrum
- Excel.

For more information on Agile tools:

- Team assets
- Best tool to add, update, and link work items

Scrum process versions

As updates are made to the Scrum process template, the version number is updated. The following table provides a mapping of the versioning applied as updates are made to the Azure DevOps on-premises process templates. For Azure Boards, the latest version is always used. Each template provides a `version` element. This element specifies a major and minor version.

TFS VERSION	SCRUM PROCESS NAME	MAJOR VERSION
Azure DevOps Services Azure DevOps Server 2022	CMMI	18

TFS VERSION	SCRUM PROCESS NAME	MAJOR VERSION
Azure DevOps Server 2020 Azure DevOps Server 2019	Scrum	17
TFS 2018	Scrum	16

For a summary of updates made to process templates, see Changes made to process templates.

Scrum process predefined queries

Your product owner can plan and track product backlog items and bugs in the product backlog by using the **Product Backlog** query. You can find work items that are assigned to the current sprint by using the shared queries that are listed under the **Current Sprint** folder. These queries find work items that are assigned to a specified iteration or sprint. As you plan more sprints, you can modify these queries to specify the latest sprint and save them to other folders that you create, such as **Sprint 2** or **Sprint 3**.

The project administrator for each project defines area paths and iteration paths for that project so that the team can track progress by those designations.

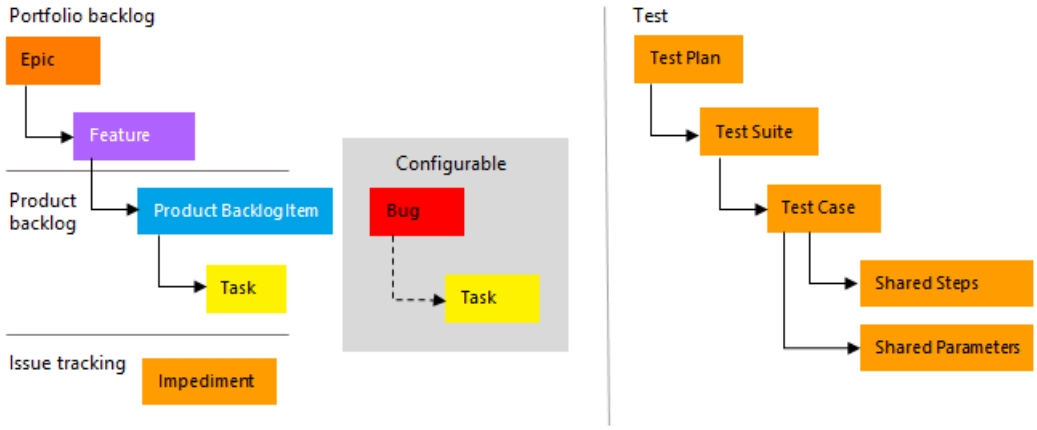
SHARED QUERIES	DESCRIPTION
Blocked Tasks	Lists all tasks in the current sprint that have been marked as Blocked.
Open Impediments	Lists all open impediment work items in the current sprint.
Sprint Backlog	Lists all product backlog items, bugs, and their linked tasks that your team has committed to complete in the current sprint.
Test Cases	Lists all test cases in the current sprint and sorts them by priority.
Unfinished Work	Lists all product backlog items, bugs, and their linked tasks that have not been marked as Done in the current sprint.
Work in Progress	Lists all tasks in the current sprint that are marked as In Progress.
Feedback	Lists all feedback responses that are in an Active state.
Product Backlog	Lists all product backlog items and bugs that are assigned to the root iteration. Product backlog items and bugs are sorted by backlog priority.

Manage your Scrum process work item types and workflow

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To plan a software project and track software defects using Scrum, teams use the product backlog item (PBI) and bug work item types (WITs). To gain insight into a portfolio of features, scenarios, or user experiences, product owners, and program managers can map PBIs and bugs to features. When teams work in sprints, they define tasks that automatically link to PBIs and bugs.



> **NOTE**
> If you are new to the Scrum process, review About Sprints, Scrum and project management to get started.

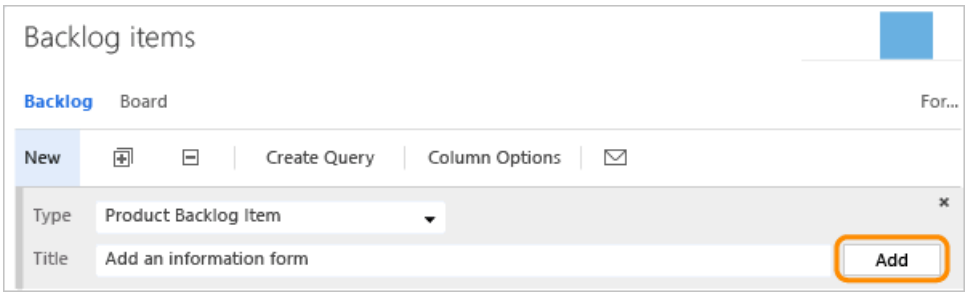
Using the web portal or Microsoft Test Manager, testers can create and run test cases and create bugs to track code defects. Impediments track blocking issues.

> **NOTE**
> Work item tracking forms and features available to you differ depending on whether you open the form from the web portal or Visual Studio Team Explorer. Forms and guidance provided in this article reflect those available with the new form experience (Azure DevOps Services and TFS 2017 and later versions).
>
> For guidance on the work item form for TFS 2015 or earlier versions or Visual Studio Team Explorer, see Add work items and select the earlier TFS version from the content version selector (above the table of contents).

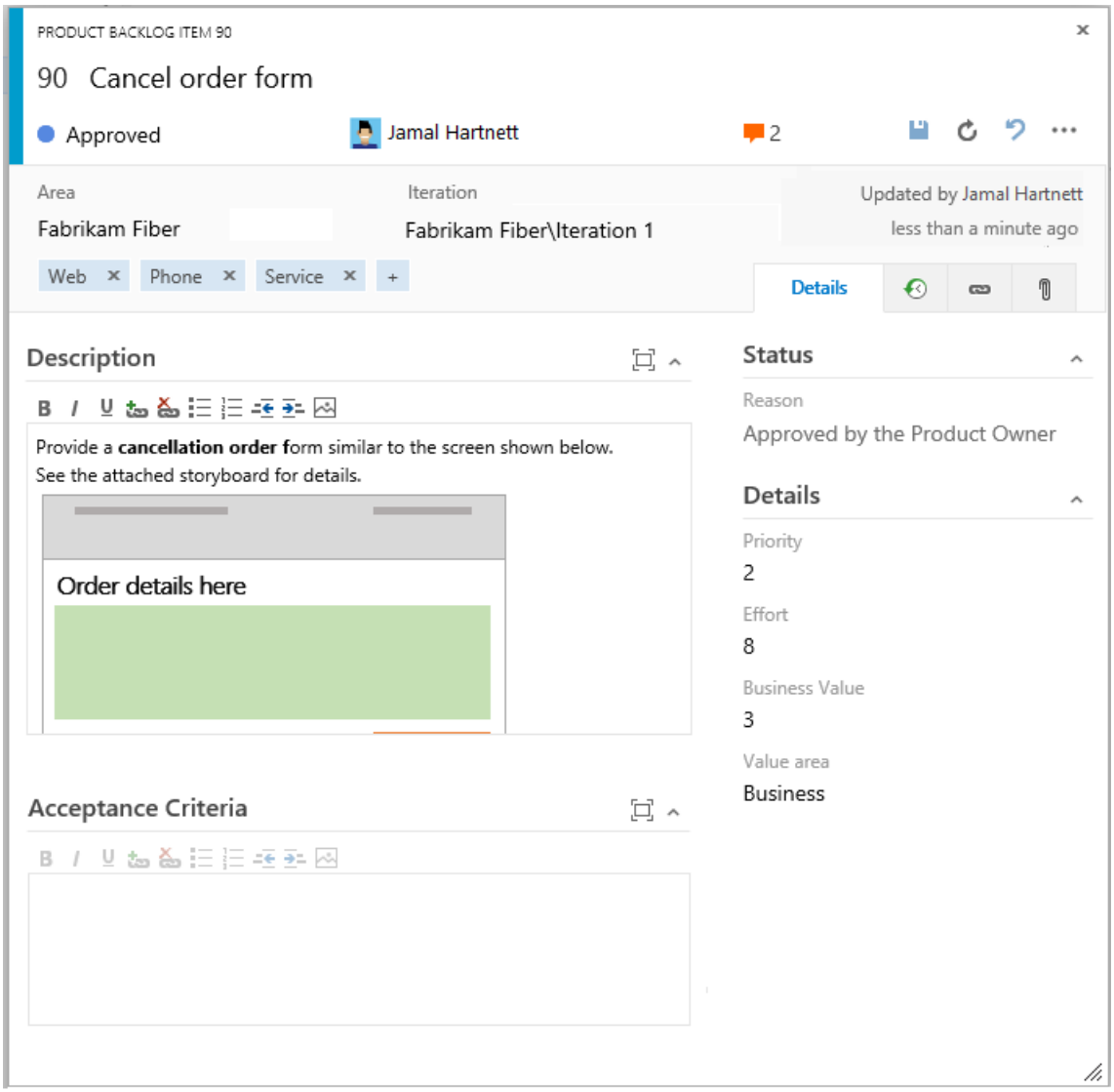
Define PBIs and bugs

When you define a product backlog item, you want to focus on the value that your customers will receive and avoid descriptions of how your team will develop the feature. The product owner can prioritize your product backlog based on each item's business value, effort, and relative dependency on other backlog items. As your business requirements evolve, so does your product backlog. Typically, teams specify details only for the highest priority items, or those items assigned to the current and next sprint.

You can create PBIs and bugs from the quick add panel on the product backlog page.



Later, you can open each PBI or bug to provide more details and estimate the effort. Also, by prioritizing the PBIs and bugs on the backlog page (which is captured in the Backlog Priority field), product owners can indicate which items should be given higher priority.



By defining the **Effort** for PBIs and bugs, teams can use the forecast feature and velocity charts to estimate future sprints or work efforts. By defining the **Business Value**, product owners can specify priorities separate from the changeable backlog stack ranking.

Use the following guidance and that provided for fields used in common across work item types when filling out the form. For details about creating bugs, see Manage bugs.

Field/tab

Usage

Effort

Estimate the amount of work required to complete a PBI using any unit of measurement your team prefers, such

as story points or time. A numeric value is required.

Agile velocity charts and forecast tools reference the values in this field. For more information, see the Estimating white paper.

Business Value

Specify a number that captures the relative value of a PBI compared to other PBIs. The higher the number, the greater the business value.

Description

Provide enough detail for estimating how much work will be required to implement the item. Focus on who the feature is for, what users want to accomplish, and why. Don't describe how the feature should be developed. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

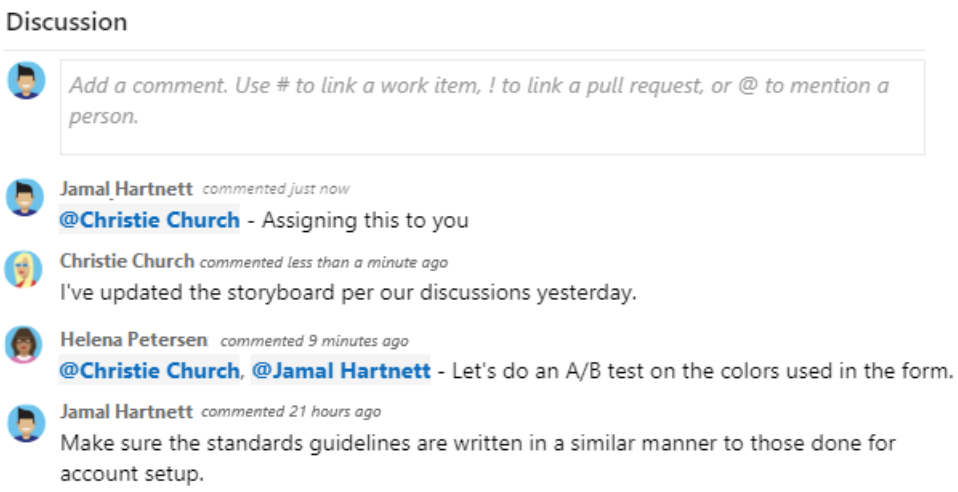
Acceptance Criteria

Define what "Done" means by describing the criteria that the team should use to verify whether the PBI or the bug fix has been fully implemented.

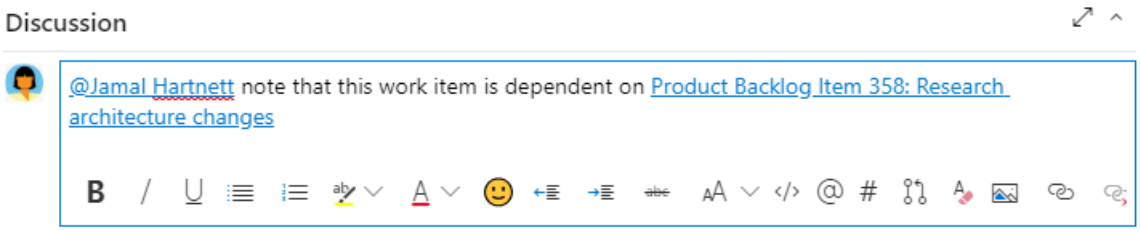
Before work begins on a PBI or bug, describe the criteria for customer acceptance as clearly as possible. Conversations between the team and customers to determine the acceptance criteria helps ensure a common understanding within the team to meet customers' expectations. The acceptance criteria can be used as the basis for acceptance tests so that the team can more effectively evaluate whether an item has been satisfactorily completed.

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



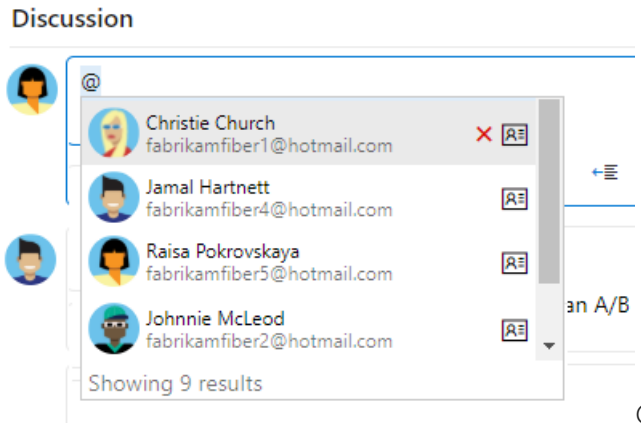
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the **History** field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons —@ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.

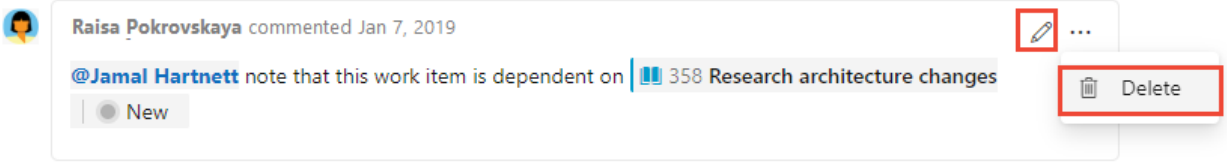


@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon and then choose **Delete**.



> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently

referenced will appear from which you can select.

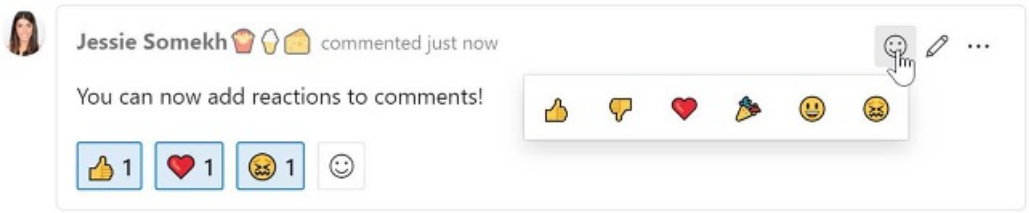
To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

You can't edit or delete comments once you've entered them.

> **IMPORTANT**
> For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

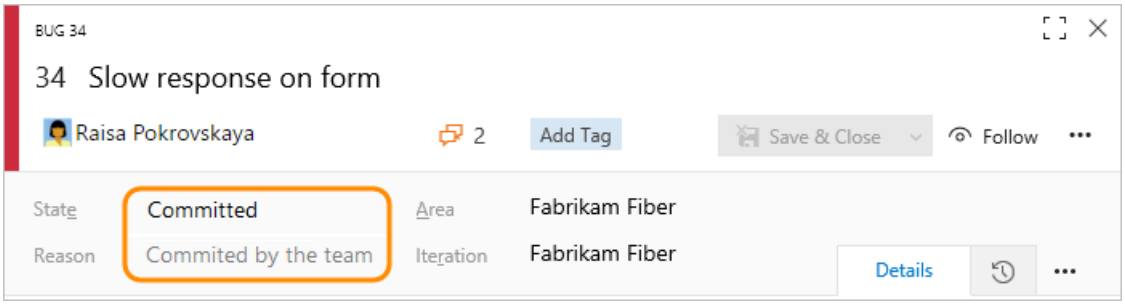
Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



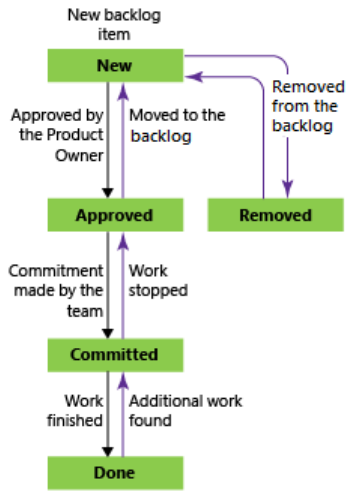
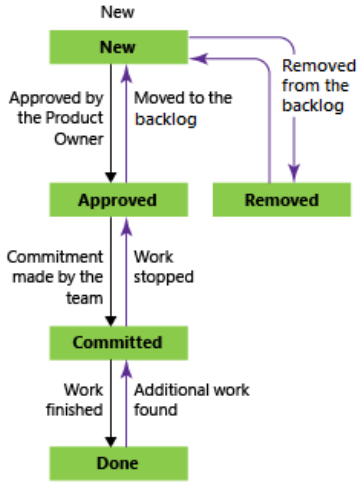
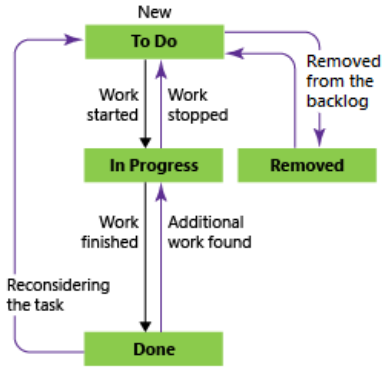
Track progress

As work progresses, you change the State field to update the status. Optionally, you can specify a reason. The state and reason fields appear on the work item form in the header area.



Scrum workflow states

By updating the State, teams know which items are new, in progress, or completed. Most WITs support transition both forward and backward from each workflow state. These diagrams show the main progression and regression states of the PBI, bug, and task WITs.

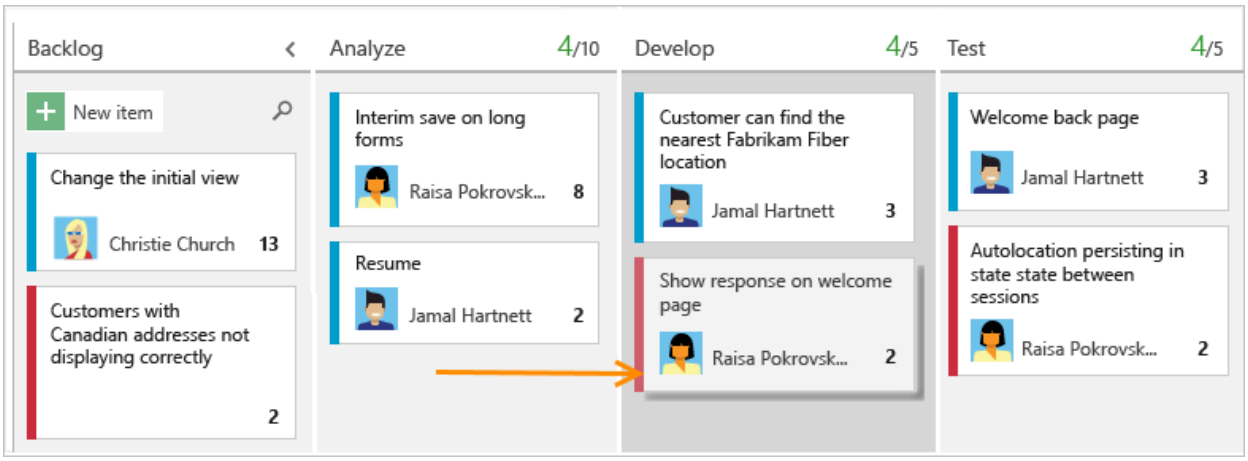
PRODUCT BACKLOG ITEM	BUG	TASK
		

PBIs and bugs follow this typical workflow progression:

- The product owner creates a PBI or a tester creates a bug in the **New** state with the default reason, **New backlog item**
- The product owner moves the item to **Approved** after it's sufficiently described and ready for the team to estimate the level of effort. Most of the time, items near the top of the Product Backlog are in the Approved state, while items toward the middle and bottom are in a New state
- The team updates the status to **Committed** when they decide to commit to working on it during the sprint
- The item is moved to the **Done** state when the team has completed all its associated tasks and the product owner agrees that it has been implemented according to the Acceptance Criteria.

Update status with Kanban or taskboards

Teams can use the Kanban board to update the status of PBIs, and the sprint taskboard to update the status of tasks. Dragging items to a new state column updates both the State and Reason fields.



You can customize the Kanban board to support more swim lanes or columns. For other customization options, see Customize your work tracking experience.

Map PBIs to features

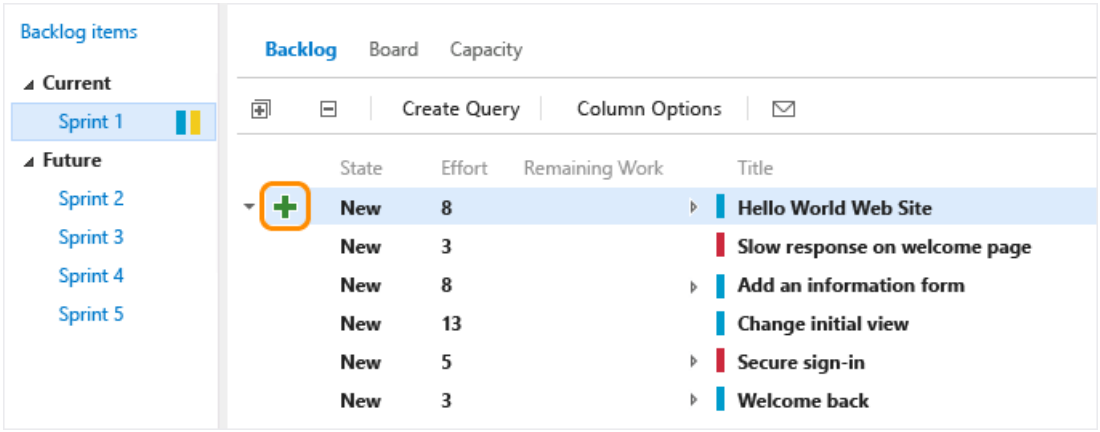
When you manage a suite of products or user experiences, you might want to view the scope and progress of work across the product portfolio. You can do this by defining features and mapping PBIs to features.

Using portfolio backlogs, you can drill down from one backlog to another to view the level of detail you want. Also, you can use portfolio backlogs to view a rollup of work in progress across several teams when you setup a

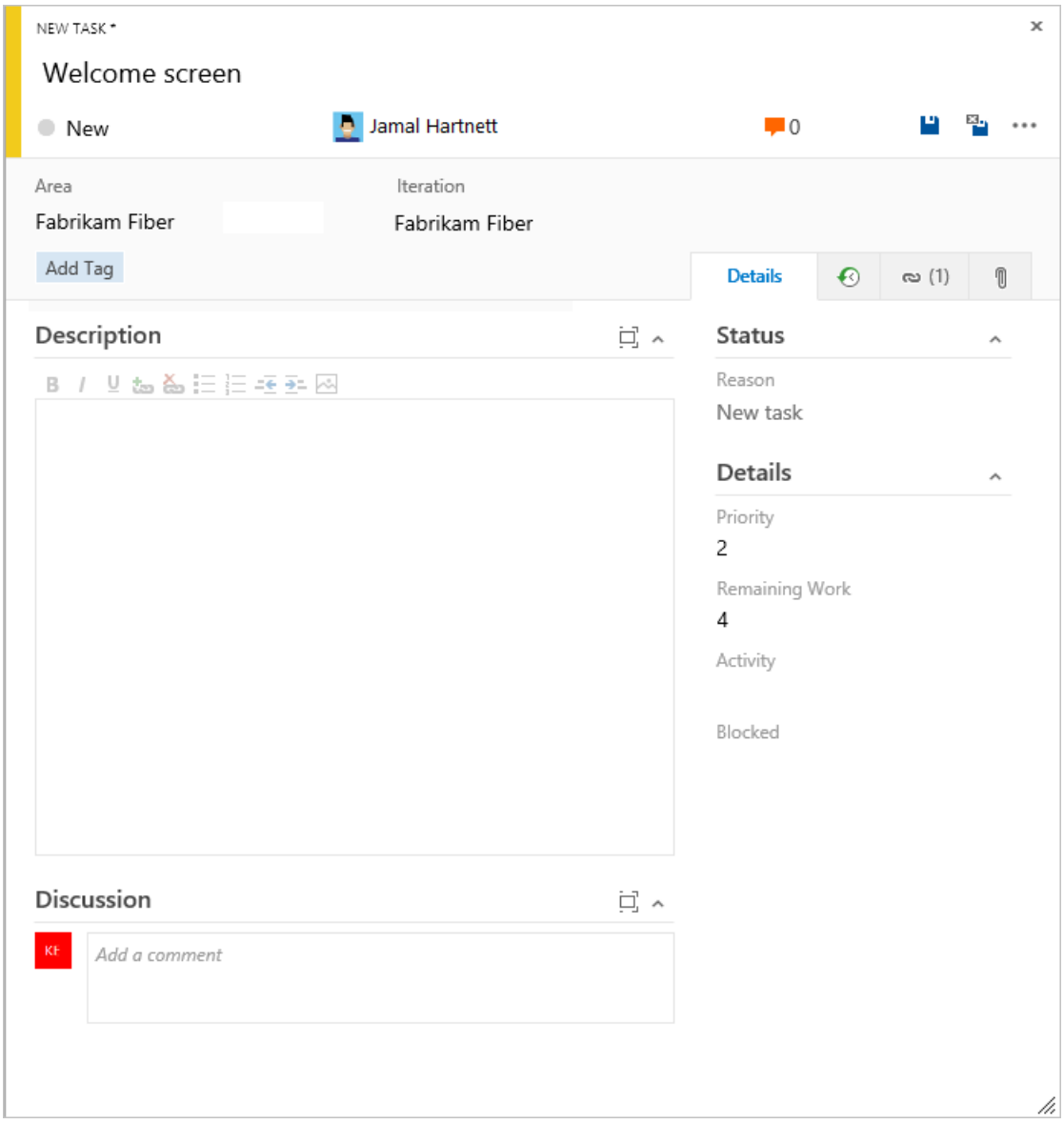
hierarchy of teams.

Define tasks

When your team manages their work in sprints, they can use the sprint backlog page to break down the work to be accomplished into distinct tasks.



Name the task and estimate the work it will take.



Using Scrum, teams forecast work and define tasks at the start of each sprint, and each team member completes a subset of those tasks. Tasks can include development, testing, and other kinds of work. For example, a developer can define tasks to implement PBIs, and a tester can define tasks to write and run test cases.

When teams estimate work using hours or days, they define tasks and the **Remaining Work** and **Activity** (optional) fields.

Field/tab

Usage

Remaining Work

Indicate how many hours or days of work remain to complete a task. As work progresses, update this field. It's used to calculate capacity charts, the sprint burndown chart, and the Sprint Burndown (Scrum) report.
If you divide a task into subtasks, specify Remaining Work for the subtasks only. You can specify work in any unit of measurement your team chooses.

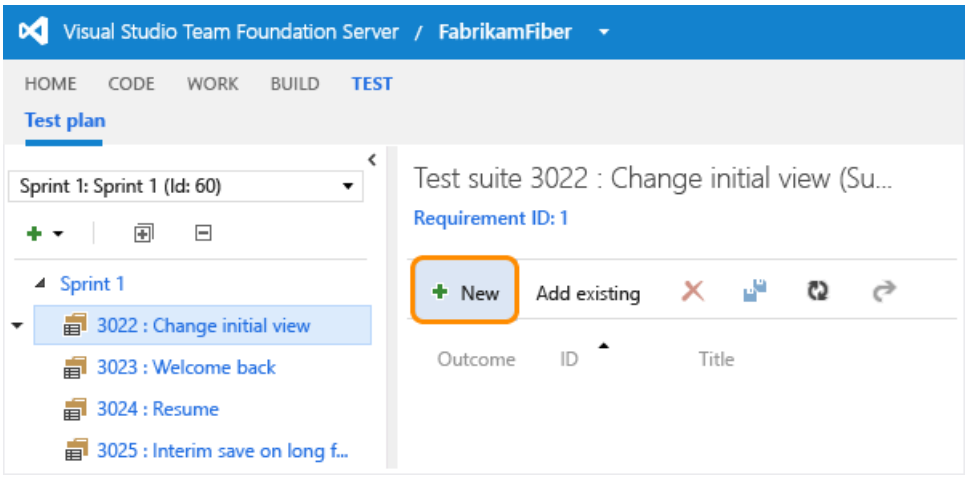
Activity

Select the type of activity this task represents when your team estimates sprint capacity by activity.

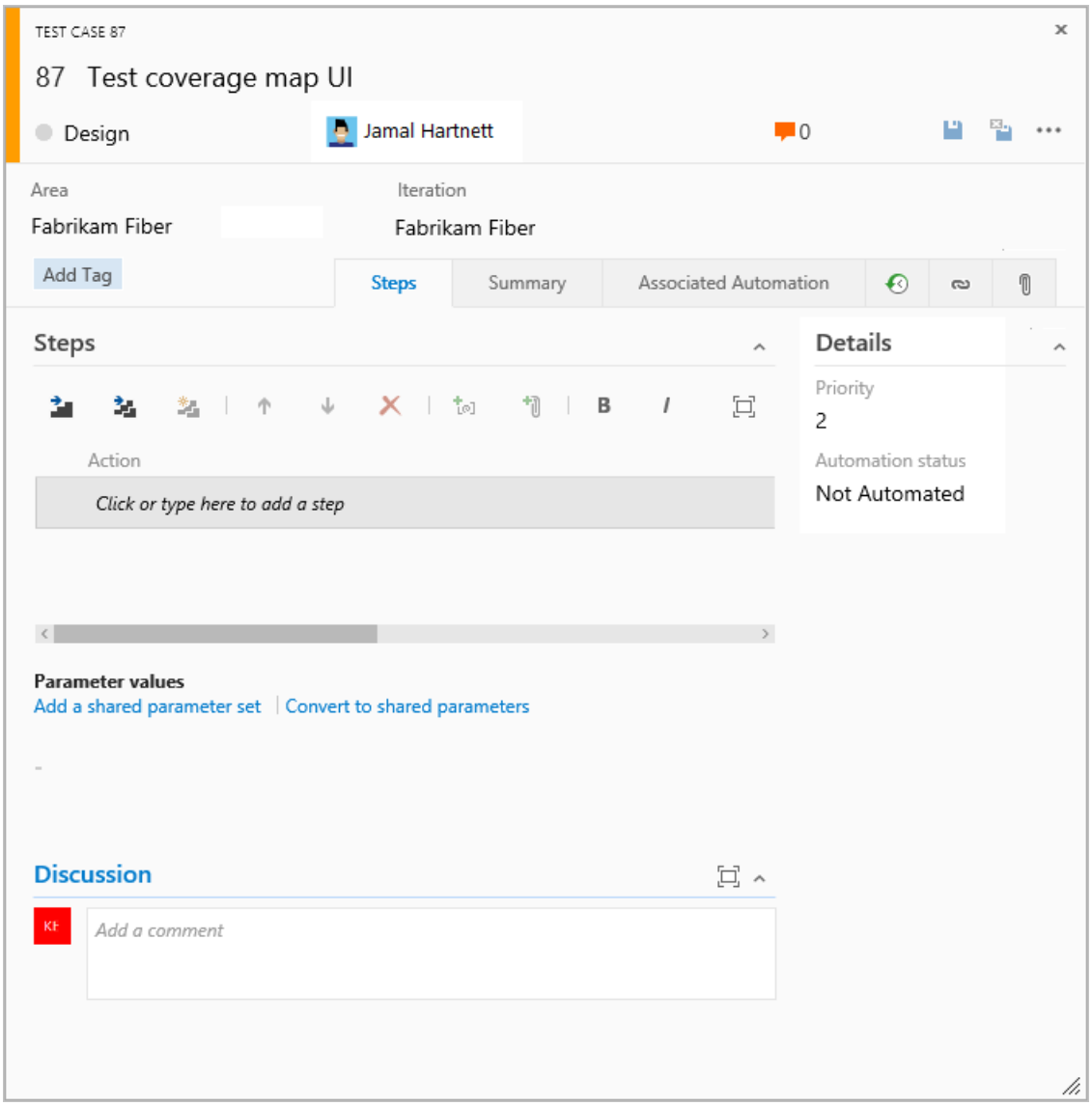
Track test progress

Test PBIs

From the web portal or Test Manager, you can create test cases that automatically link to a PBI or bug. Or, you can link a PBI or bug to a test case from the (links tab).



The test case contains many fields, many of which are automated and integrated with Test Manager and the build process. For a description of each field, see Query based on build and test integration fields.



The (links tab) captures the links to all the PBIs and bugs in a test case. By linking PBIs and bugs to test cases, the team can track the progress made in testing each item.

Track code defects

You can create bugs from the web portal web portal, Visual Studio, or when testing with Test Manager.

Definitions for common work tracking fields

The following fields and tabs appear in most work items. Each tab is used to track specific information, such as history, links, or attachments. These three tabs provide a history of changes, view of linked work items, and ability to view and attach files.

The only required field for all work item types is **Title**. When the work item is saved, the system assigns it a unique **ID**. The form highlights required field in yellow. For information about other fields, see Work item field index.

Field/tab

Usage

Title

Enter a description of 255 characters or less. You can always modify the title later.

Assigned To

Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu will list only team members or contributors to the project.

State

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current state.

Reason

Use the default first. Update it when you change state. Each State is associated with a default reason.

Area

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting.

To change the dropdown list of areas, see Add and modify area and iteration paths.

Iteration

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later, during a planning meeting.

To change the drop-down list of iterations, see Add and modify area and iteration paths.

 (History)

Review the audit trail that the system captures and capture additional information.

Every time that the work item is updated, information is appended to the history. History includes the date of the change, who made the change, and which fields were changed. You can also add formatted text to the history field.

 (Links)

Add all types of links, such as hyperlinks, changesets, source files, and so on.

This tab also lists all links defined for the work item.

 (Attachments)

Share more detailed information by adding files to the work item, such as email threads, documents, images, log files, or other file types.

Customize work item types

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process or Customize the On-premises XML process model depending on the process model used by your project.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize the On-premises XML process model.

Related articles

Before you start tracking work, you must have a project. To create one, see Create a project.

If you have a project, start tracking work:

- Add work items to manage a project - to gain more familiarity with the work item form features
- Create a backlog - to develop your product backlog
- Kanban - to start working in Kanban
- Plan a sprint - to start working in Scrum
- Excel.

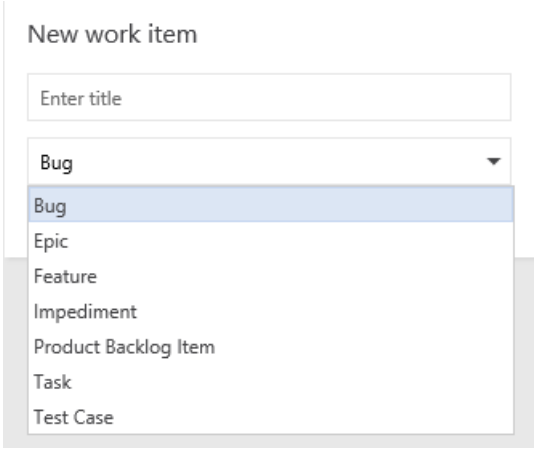
For more information on Agile tools:

- Team assets
- Best tool to add, update, and link work items

Track impediments

Use the impediment WIT to track events that may block progress or ship a PBI. Use the Bug WIT exclusively to track code defects.

You can add an impediment from the New work item widget added to a team dashboard, or from the **New** menu on the Queries page.



Work items you add from the widget are automatically scoped to your team's default area and iteration paths. To change the team context, see Switch team context.

Backlog list order

The Backlog Priority field is used to track the relative ranking of PBIs, bugs, features, or epics. However, by default it doesn't appear on the work item form. The sequence of items on the backlog page is determined according to where you've added the items or moved the items on the page. As you drag items, a background process updates this field.

Links control, client work item form

Work item forms displayed in a client and the web portal for TFS 2015 and earlier versions display link tabs and link control restrictions as described in the following table.

Tab name

Work item type

Link restrictions

All Links

Feedback Request

Feedback Response

- No restrictions.

Links

Product Backlog Item

Bug

Impediment

Shared steps

Task

Test Case

- No restrictions.

Links

Code Review Request

- Allows only **Parent** and **Child** links to Code Review Response work items.
- Excludes links to work items in other projects.

Stories

Feedback Response

- Allows only **Related** links.
- Allows links to Bug and Product Backlog Items.
- Excludes links to work items in other projects.

Storyboards

Product Backlog Item

- Allows only **Storyboard** links.

Tasks

Product Backlog Item

- Allows only **Child** links to Tasks.
- Excludes links to work items in other projects.

Test Cases

Product Backlog Item

Bug

- Allows only **Tested By** links.
- Allows links only to test cases.
- Excludes links to work items in other projects.

Tested Backlog Items

Test case

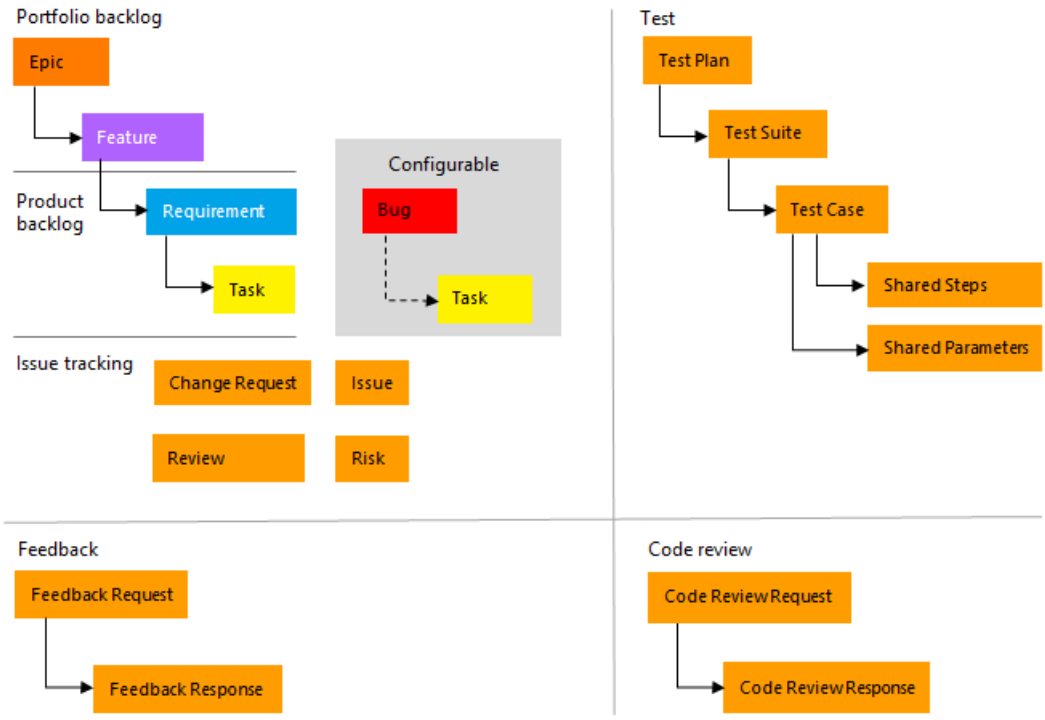
- Allows only **Tests** links.
- Allows links to Bug and Product Backlog Items.
- Excludes links to work items in other projects.

Understand CMMI process template artifacts

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The CMMI process supports the following work item types (WITs) to plan and track work, tests, feedback, and code review. With different WITs you can track different types of work—such as requirements, change requests, tasks, bugs and more. These artifacts are created when you create a project using the CMMI process. They're based on the Capability Maturity Model Integration (CMMI) process.



Along with the WITs, teams have access to a set of work item queries to track information, analyze progress, and make decisions.

> **NOTE**
> You can customize the work tracking system for your project by creating and customizing an inherited process and applying that process to your project. To learn more, see Inheritance process model.

> **NOTE**
> You can customize the work tracking system for your project by customizing an Inherited process or an On-premises XML process. To learn more, see Inheritance process model or On-premises XML process customization.
>
> The latest version of each process uploads automatically when you install or upgrade to the latest version of Azure DevOps Server. Additional artifacts, such as SQL Server reports are only available when you connect to a project. Other resource requirements apply.

NOTE

You can customize the work tracking system for your project by customizing an On-premises XML process. To learn more, see On-premises XML process customization.

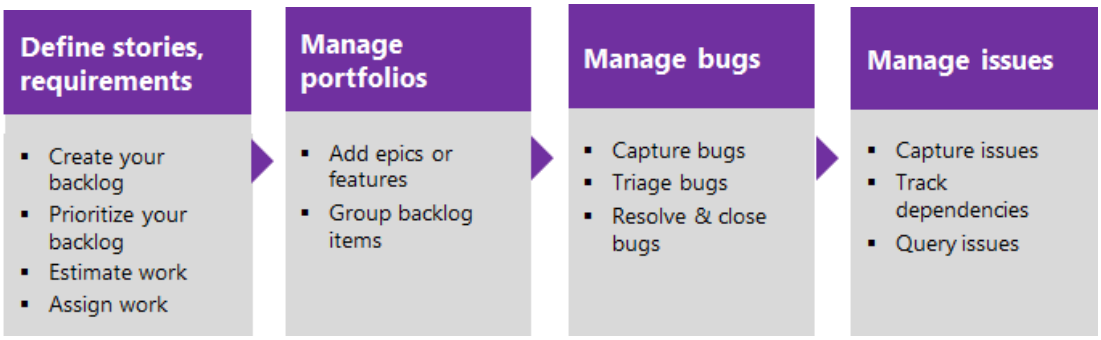
The latest version of each process uploads automatically when you install or upgrade to the latest version of Azure DevOps Server. Additional artifacts, such as SQL Server reports are only available when you connect to a project. Other resource requirements apply.

Plan and track work with CMMI

Teams plan their project by capturing features and requirements. When teams work in sprints, they define tasks and link them to requirements. To gain insight into a rollup of requirements across teams, program managers link requirements to a feature. Blocking issues are tracked using issues. For details on using these WITs, see CMMI process work item types and workflow

The essential flow for getting started is as shown. To get started using Scrum or Kanban tools, see Get started with Agile tools to plan and track work.

Click one of the following images to go to the linked article.



NOTE

A work item is a database record that contains the definition, assignment, priority, and state of work. Work item types define the template of fields, workflow, and form for each type. Work items can be linked to each other to support tracking dependencies, roll up of work, and reports.

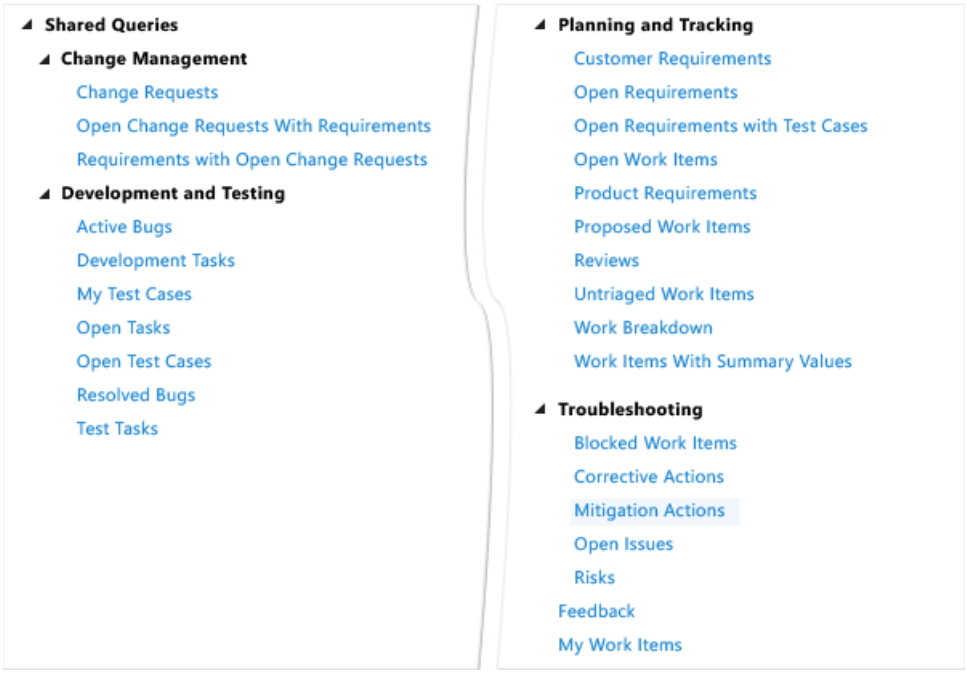
List work items with queries

You can use work item queries to list work items based on their type, such as change requests, bugs, tasks, and requirements.

NOTE

New projects no longer define a default set of Shared Queries at the time of project creation. The definitions for Shared Queries have been removed from the process template. For on-premises deployments, you can add them to a custom process template as described in Add work item queries to a process template.

Or, use one of the shared queries that the CMMI process provides.



Descriptions of predefined queries are listed later in this article.

You can view and run queries from the web portal or from the Team Explorer plug-in to Visual Studio. You can modify a query using the query editor to apply different filter criteria. Also, you can add queries to team dashboards.

Quick tips on shared queries

If you are new to Azure Boards, work tracking, and shared queries, review these tips to learn how you can manage work more effectively:

- To find work items that are assigned to you, add **@Me** as the value for the Assigned To field in one of the query clauses.
- All valid users with standard access can create queries and folders under the **My Queries** area. To create queries and query folders under **Shared Queries**, you must have the Contribute permission set and have been assigned Basic access or greater. For more information, see Set permissions on queries.
- You can modify any query by adding criteria to focus on a product area, an iteration, or another field. To modify a query, open the query editor.
- You can open any query in Excel where you can update the fields of one or more work items and publish your changes to the database for tracking work items.
- You can visualize status or progress by creating a pie-chart, column chart, or trend chart for flat-list queries.

> **IMPORTANT**
> Starting with Visual Studio 2019, the Azure DevOps plug-in for Office has deprecatedsupport for Microsoft Project. Project integration and the **TFSFieldMapping** command is not supported for Azure DevOps Server 2019 and later versions, including Azure DevOps Services. You can continue to use Microsoft Excel.

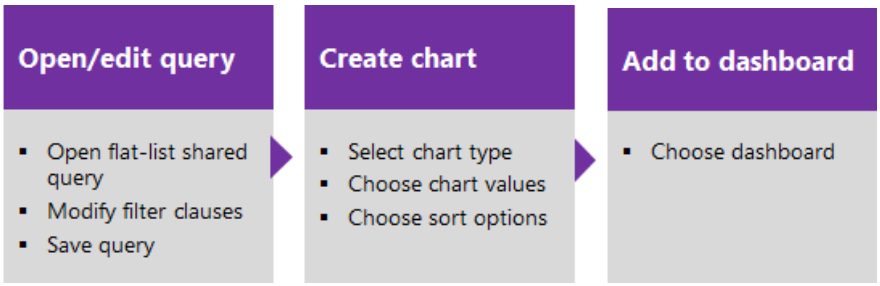
Monitor progress

All processes—Agile, Scrum, and CMMI—support building status and trend charts and dashboards. Also, several charts are automatically built based on the Agile tools you use. These charts display within the web portal.

Create light-weight charts

To get started, you can define a shared flat query and create a chart based on your tracking interests. Chart

types include status—pie, bar, column, stacked bar, and pivot—and trend—stacked area, line, and area—charts.



Analytics widgets and Power BI reports

The Analytics Service can answer quantitative questions about the past or present state of your projects. You can add Analytics widgets to a dashboard or use Power BI to create charts and reports.

To learn more, see What is the Analytics Service?

SQL Server reports

If your project collection and the project are configured with SQL Server Analysis Services and Reporting Services, you'll have access to many CMMI reports. For these reports to be useful, teams must do certain activities, such as define build processes, link work items, and update status or remaining work.

If you need to add reporting services or update reports to the latest versions, see Add reports to a project.

Related notes

Before you start tracking work, you must have a project. To create one, see Create a project.

If you have a project, start tracking work:

- Add work items to manage a project - to gain more familiarity with the work item form features
- Create a backlog - to develop your product backlog
- Kanban - to start working in Kanban
- Plan a sprint - to start working in Scrum
- Excel.

For more information on Agile tools:

- Team assets
- Best tool to add, update, and link work items

CMMI process versions

As updates are made to the CMMI process template, the version number is updated. The following table provides a mapping of the versioning applied as updates are made to the Azure DevOps on-premises process templates. For Azure Boards, the latest version is always used. Each template provides a `version` element. This element specifies a major and minor version.

TFS VERSION	CMMI NAME	MAJOR VERSION
Azure DevOps Services Azure DevOps Server 2022	CMMI	18
Azure DevOps Server 2020 Azure DevOps Server 2019	CMMI	17

TFS VERSION	CMMI NAME	MAJOR VERSION
TFS 2018	CMMI	16

For a summary of updates made to process templates, see Changes made to process templates.

More CMMI guidance

The situations and working practices of development teams vary widely, and most companies will have their own well-established processes. For these reasons, the guidance given here doesn't attempt to prescribe a development process in full. Instead, we describe just the activities that are relevant to making best use of the CMMI process.

- Background to CMMI : Provides an overview of CMMI and the six capability levels that are intrinsic to the model.
- Project management : Provides guidance to help you better understand how to manage, plan, and coordinate the development and maintenance of software products working with the CMMI model.
- Engineering : Addresses the value-added activities for discovering the information that is required to design and build software products

Using the CMMI template and guidance can help you achieve the aims of CMMI if you use it as part of a process improvement program. Adapt this guidance to your own situation, which will depend on the type and history of the product that you're developing, the project's scale, the background of the team members, and accepted practices in your organization.

This guidance was developed in partnership with David Anderson. For more information, see the following Web page: David J Anderson & Associates.

CMMI process predefined queries

Query for items assigned to you

You can find the work items that are assigned to you by using one of the shared queries that the following table describes.

SHARED QUERY	DESCRIPTION
My Test Cases	Lists all test cases that are not closed and that are assigned to the team member who is running the query. Test cases are sorted by priority and then ID.
My Work Items	Lists all work items, excluding shared steps, that are not closed and that are assigned to the team member who is running the query. Work items are sorted by rank, priority, type, and ID.

Development and test queries

Team members can use the shared queries that are described in the following table to track the status of development and test tasks and active and resolved bugs.

SHARED QUERY	DESCRIPTION
Active Bugs	Lists all active bugs and sorts them by rank, priority, and severity.

SHARED QUERY	DESCRIPTION
Development Tasks	Lists all tasks whose **Discipline** is set to **Development**. Tasks are sorted by ID.
My Test Cases	Lists all test cases that are not closed and that are assigned to the team member who is running the query. Test cases are sorted by priority and then ID.
Open Tasks	Lists all tasks that are not closed, sorted by rank, priority, and then ID.
Open Test Cases	Lists all test cases that are not closed, sorted by priority and then ID.
Resolved Bugs	Lists all resolved bugs that are defined for the project, sorted by rank, priority, and severity.
Test Tasks	Lists all tasks whose **Discipline** is set to **Test**, sorted by ID.

Backlog management queries

Product owners can use the shared queries that are described in the following table to track the status of requirements and untriaged work.

SHARED QUERY	DESCRIPTION
Customer Requirements	Lists all requirements, sorted by ID, that have been identified as Scenario or Quality of Service work items.
Product Requirements	Lists all requirements, sorted by ID, that have been identified as Functional, Operational, Security, Safety, or a Feature.
Open Requirements	Lists all requirements that are not closed, sorted by iteration ID, priority, and then work item ID.
Open Requirements without Test Cases	Lists all requirements that are not closed and that do not have a Tested By link to a test case, sorted by work item ID.
Open Work Items	Lists all work items except shared steps that are not closed. Work items are sorted by rank, priority, type, and then ID.
Proposed Work Items	Lists all proposed work items, sorted by rank, priority, iteration, area, triage, and then work item ID.
Reviews	Lists all reviews, sorted by work item ID.
Untriaged Work Items	Lists all requirements, tasks, change requests, bugs, and issues that have not been closed or triaged. The Triage field for these work items is set to Pending, More Info, or Info Received. Work items are sorted by state, triage, rank, priority, iteration, and area.
Work Breakdown	Lists all requirements that are not closed and their child requirements or tasks.

SHARED QUERY	DESCRIPTION
Work Items With Summary Values	Lists all tasks that have child tasks and that contain non-zero values for the Remaining Work or Completed Work fields. This query is designed to find tasks that report work effort that is already accounted for in their child tasks. For the hours to be counted only once, summary tasks should not be assigned any hours. For more information, see Address inaccuracies published for summary values.

Change management queries

Product owners can use the shared queries that are described in the following table to track change requests and dependencies that have been identified between change requests and requirements.

SHARED QUERY	DESCRIPTION
Change Requests	Lists all change requests, sorted by ID.
Open Change Requests with Requirements	Lists change requests that are not closed and their linked requirements, sorted by ID. Only change requests that are linked to a requirement with a link type of Affects appears in the list.
Requirements with Open Change Requests	Lists requirements and the change requests that are not closed and that depend on them, sorted by ID. Only requirements that are linked to a change request with a link type of Affected By are listed.

Troubleshooting queries

Product owners can use the shared queries that are described in the following table to troubleshoot issues and risks to the product schedule.

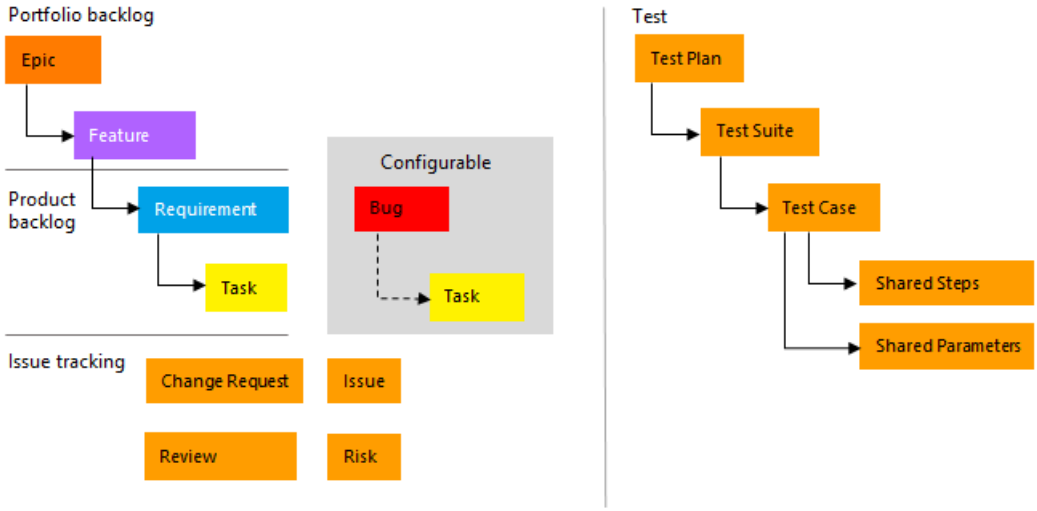
SHARED QUERY	LISTS
Blocked Work Items	Lists all work items where the **Blocked** field is set to **Yes**. Only requirements, tasks, bugs, issues, and change requests can be blocked.
Corrective Action Status	Lists all tasks whose **Task Type** is set to **Corrective Action**.
Mitigation Actions	Lists all tasks whose **Task Type** is set to **Mitigation Action**.
Open Issues	Lists all issues that are not closed. The Issues workbook references this query.
Risks	Lists all risks, sorted by ID.

CMMI process work item types and workflow in Azure Boards

10/4/2022 • 15 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Teams use the work item types (WITs) provided with the MSF for CMMI Process Improvement 2015 (CMMI) process to plan and track progress of software projects. Teams define requirements to manage the backlog of work and then, using the Kanban board, track progress by updating the status of requirements.



To gain insight into a portfolio of requirements, product owners can map requirements to features. When teams work in iterations, they define tasks that automatically link to requirements.

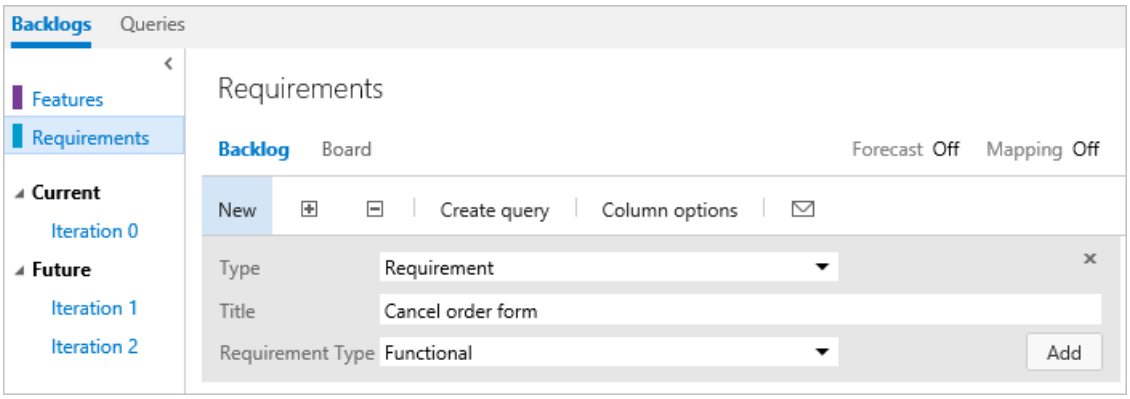
Using Microsoft Test Manager and the web portal, testers create and run test cases and define bugs to track code defects.

To support other CMMI processes, teams can track change requests, risks, issues, and notes captured in review meetings. If you're new to the CMMI process, review the section Plan and track work with CMMI to get started.

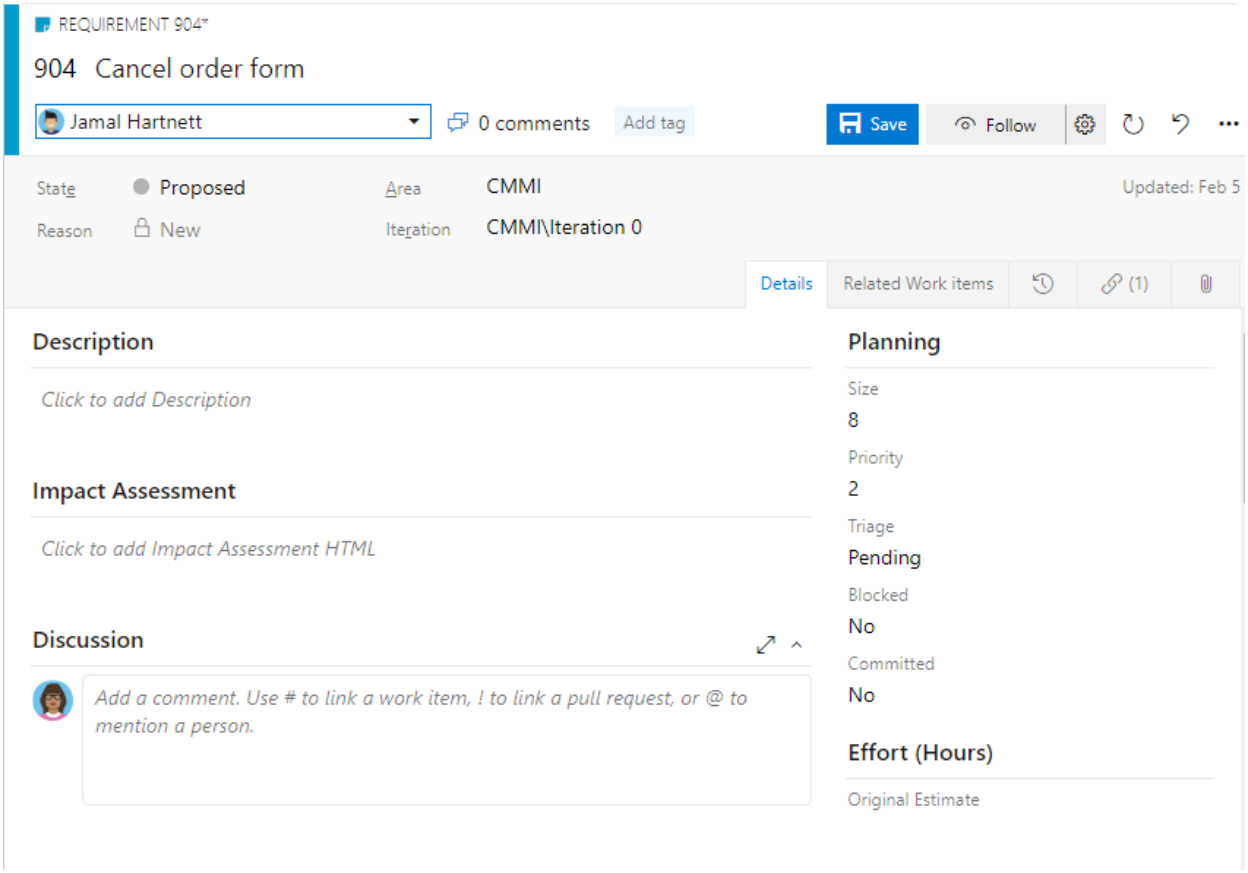
> **NOTE**
>
> Work item tracking forms and features available to you differ depending on whether you open the form from the web portal or Visual Studio Team Explorer. Forms and guidance provided in this article reflect those available with the new form experience (Azure DevOps Services and TFS 2017 and later versions).
>
> For guidance on the work item form for TFS 2015 or earlier versions or Visual Studio Team Explorer, see Add work items and select the earlier TFS version from the content version selector (above the table of contents).

Define requirements

Create requirements from the quick add panel on the product backlog page.



Later, you can open each requirement to provide more details and estimate its size.



Instead, you can bulk add requirements using Excel.

Instead, you can bulk add requirements using Excel or Project.

> **IMPORTANT**
> Microsoft Project Integration and the **TFSFieldMapping** command are not supported for:
>
> - Visual Studio 2019 and Azure DevOps Office® Integration 2019
> - Azure DevOps Server 2019 and later versions, including Azure DevOps Services.
>
> However, full support for Microsoft Excel integration is maintained and supports bulk import and update of work items. Alternatives to using Microsoft Project include the following:
>
> - Delivery plans
> - A Marketplace extension such as Project Connect or GANTT chart.

Requirements specify the functions and product elements that teams need to create. Product owners typically define and stack rank requirements on the product backlog page. The team then scopes the size of the effort to deliver the highest priority items.

Use the following guidance and that provided for fields used in common across work item types when filling out the form. For more information, see Plan a project.

Field

Usage

Description

Provide enough detail for estimating how much work will be required to implement the requirement. Focus on who the requirement is for, what users want to accomplish, and why. Don't describe how the requirement should be developed. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

In HTML fields, you can add rich text and images.

Impact Assessment

The customer impact of not implementing this requirement. You might include details from the Kano model about whether this requirement is in the surprise, required, or obvious categories. You capture this information in the rich-text HTML field that corresponds to Impact Assessment.

Requirement Type (Required)

The kind of requirement to implement. You can specify one of the following values:

- **Business Objective**
- **Feature** (default)
- **Functional**
- **Interface**
- **Operational**
- **Quality of Service**
- **Safety**
- **Scenario**
- **Security**

Value area

The area of customer value addressed by the epic, feature, requirement, or backlog item. Values include:

- **Architectural**: Technical services to implement business features that deliver solution
- **Business**: Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default)

Size

Estimate the amount of work required to complete a requirement using any numeric unit of measurement your team prefers. By defining the **Size** for requirements, teams can use the Agile velocity charts and forecast tools to estimate future iterations or work efforts. The Kanban Cumulative Flow Diagram references the values in this field. For more information, see the Estimating white paper.

Original Estimate

The amount of estimated work required to complete a task. Typically, this field doesn't change after it's assigned.

You can specify work in hours or in days. There are no inherent time units associated with this field.

Start Date/Finish Date

The target dates for when the work will start or finish.

Priority (Required)

A subjective rating of the requirement as it relates to the business. Allowed values are:

- **1**: Product can't ship without the item.
- **2**: (default) Product can't ship without the item, but it doesn't have to be addressed immediately.
- **3**: Implementation of the item is optional based on resources, time, and risk.

Triage (Required)

Indicates the type of triage decision that is pending for the work item. Use this field when the work item is in the Proposed state and specify one of the following values: **Pending** (default), **More Info**, **Info Received**, and **Triaged**.

Blocked

Indicates whether a team member is prevented from making progress toward implementing a requirement or task or resolving a bug, change request, or risk. If an issue has been opened to track a blocking problem, you can create a link to the issue. You can specify **Yes** of **No**.

Committed (Required)

Indicates whether the requirement is committed in the project or not. You can specify **Yes** or **No** (default).

Integrated In

Product build number that incorporates the requirement, change request, or fixes a bug.

User Acceptance Test (Required)

The status of the user acceptance test for a requirement. You can specify one of the following values:

- **Pass**
- **Fail**
- **Not Ready** (default)
- **Ready**
- **Skipped**
- **Info Received**

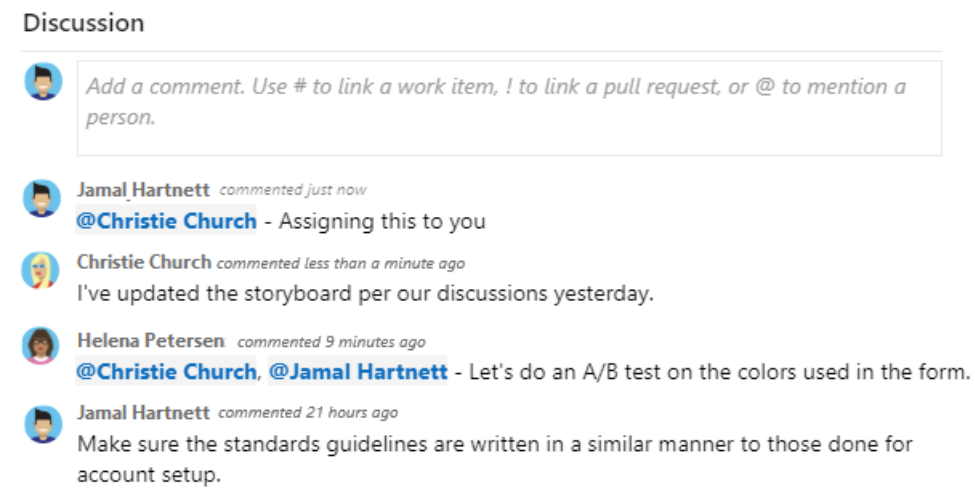
You specify Not Ready when the requirement is in the Active state, and you specify Ready when the requirement is in the Resolved state.

Subject Matter Experts

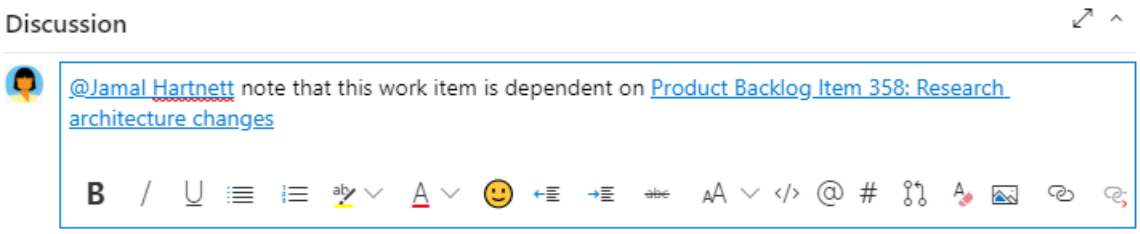
The names of team members who are familiar with the customer area that this requirement represents.

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



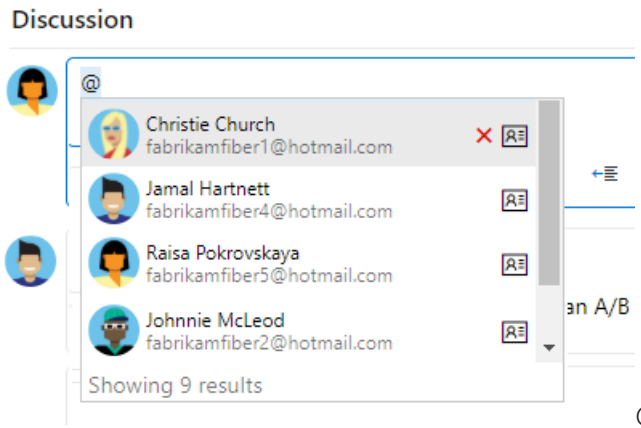
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the History field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons —@ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.



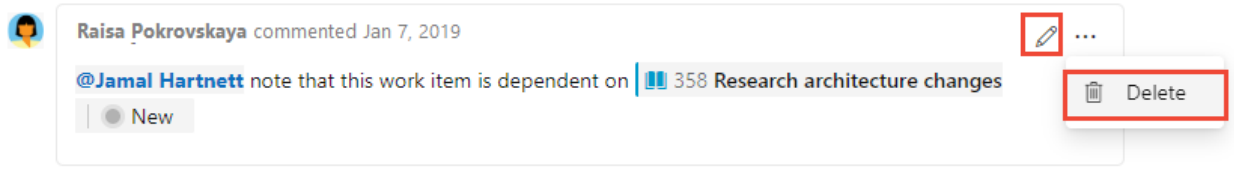
@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon

and then choose **Delete**.



> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

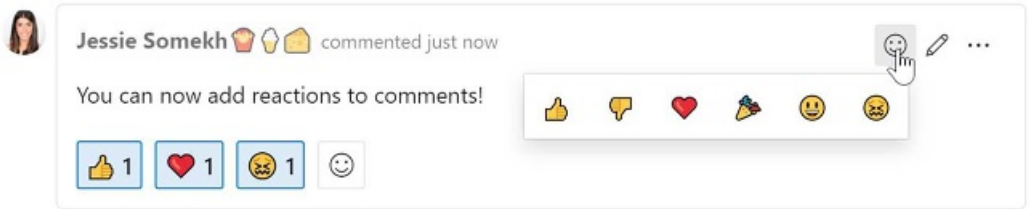
To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

You can't edit or delete comments once you've entered them.

> **IMPORTANT**
> For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

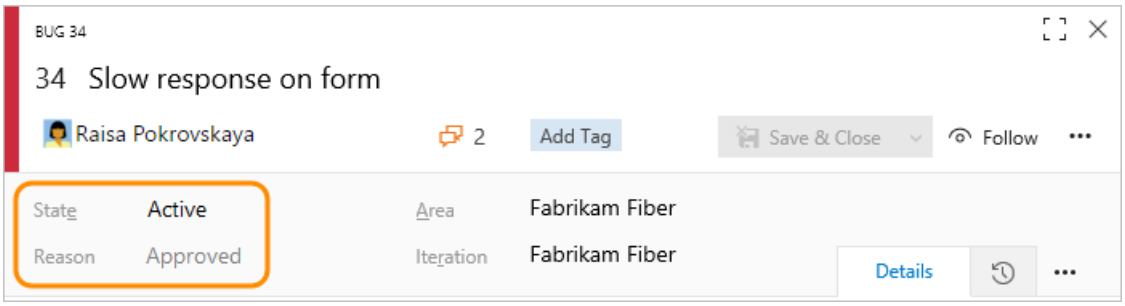
Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



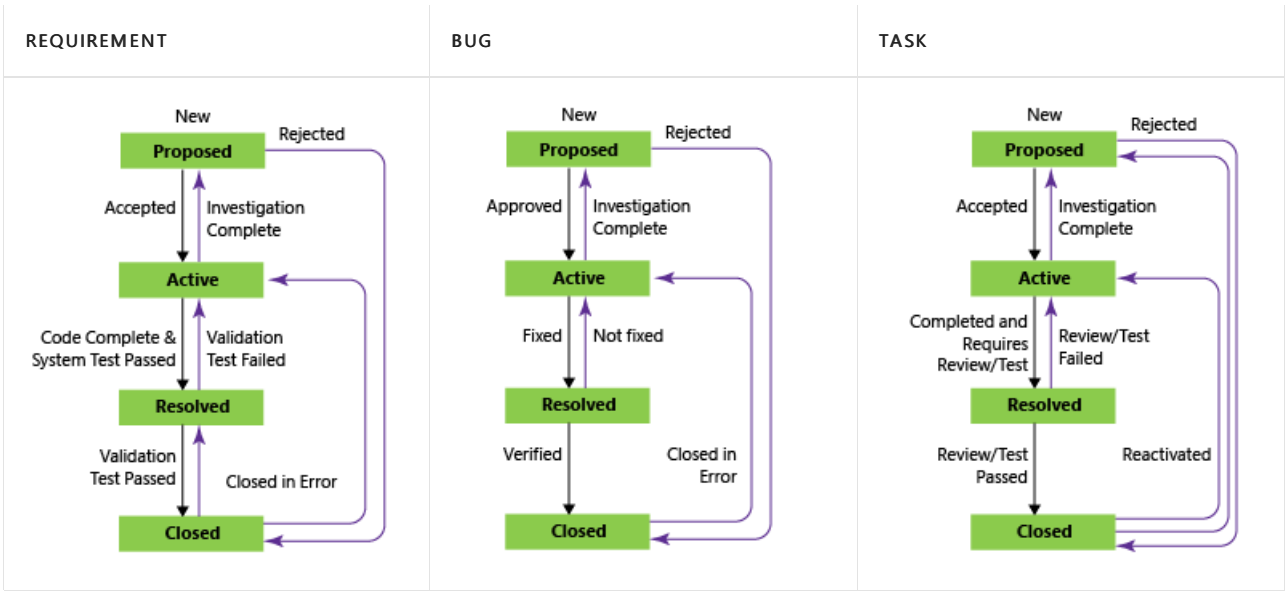
Track work progress

As work progresses, you change the State field to update the status. Optionally, you can specify a reason. The state and reason fields appear on the work item form in the header area.



CMMI workflow states

These diagrams show the main progression and regression states for the Requirement, Bug, and Task WITs.

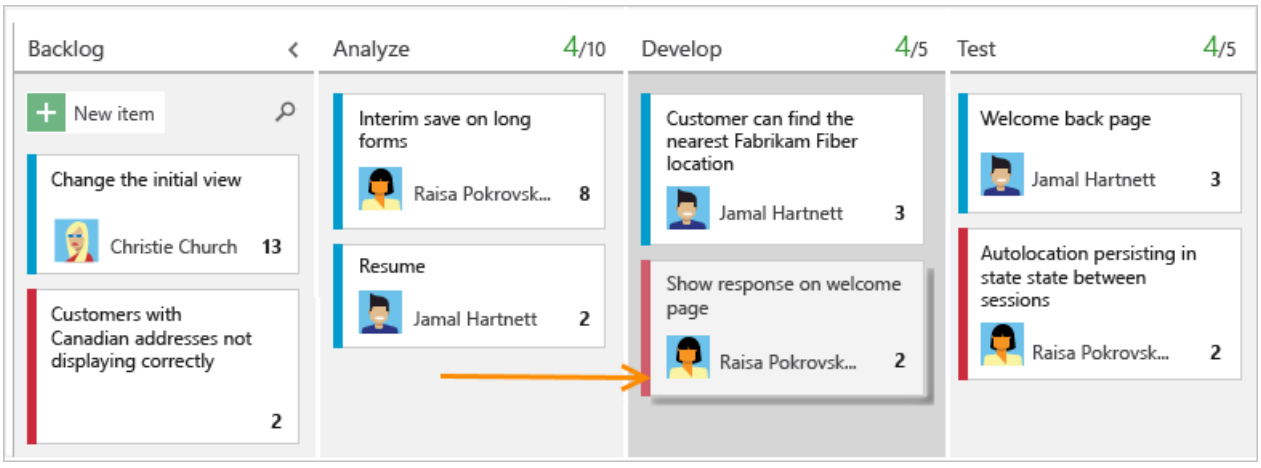


The typical workflow progression for a requirement is:

- The product owner creates a requirement in the **Proposed** state with the default reason, **New requirement**.
- The product owner updates the status to **Active** when they begin work to implement it.
- The team updates the status to **Resolved** when code development is finished and system tests have passed.
- Lastly, the team or product owner moves the requirement to **Closed** when the product owner agrees that it has been implemented according to the Acceptance Criteria and passed all validation tests.

Update work status with Kanban or taskboards

Teams can use the Kanban board to update the status of requirements, and the sprint taskboard to update the status of tasks. Dragging items to a new state column updates both the State and Reason fields.



You can customize the Kanban board to support more swim lanes or columns. For more customization options, see Customize your work tracking experience.

Map requirements to features

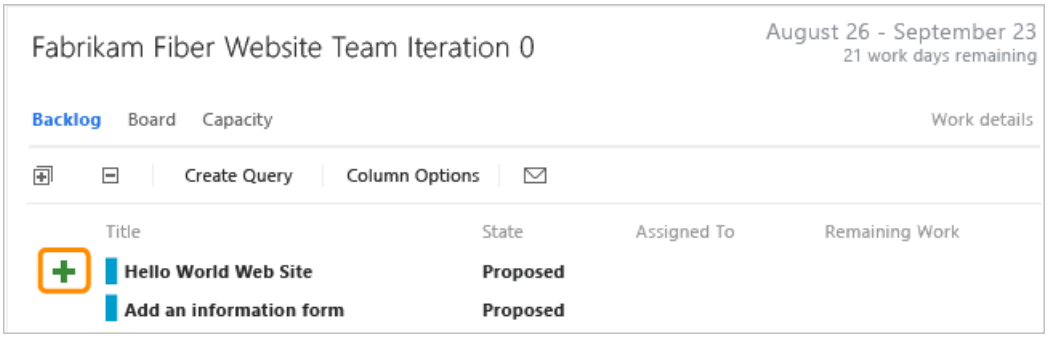
When you manage a suite of products or user experiences, you might want to view the scope and progress of work across the product portfolio. You can do view scope by defining features and mapping requirements to features.

Using portfolio backlogs, you can drill down from one backlog to another to view the level of detail you want. Also, you can use portfolio backlogs to view a rollup of work in progress across several teams when you setup a hierarchy of teams.

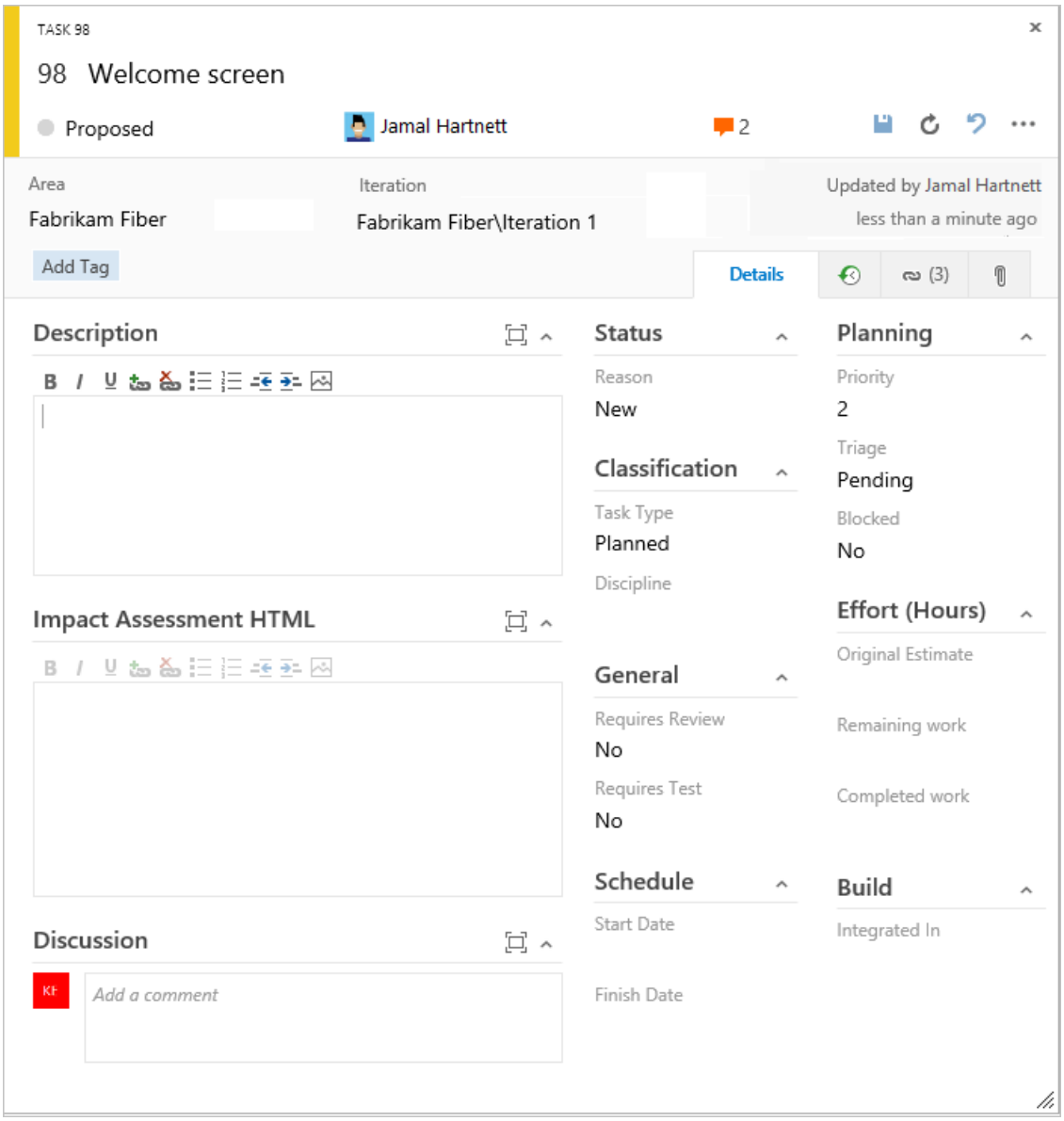
The feature work item contains similar fields provided for requirements and includes other fields, as the following table describes.

Define tasks

When your team manages their work in sprints, they can use the sprint backlog page to break down the work to be accomplished into distinct tasks.



Name the task and estimate the work it will take.



When teams estimate work, they define tasks and estimate the hours or days to complete tasks. Teams forecast work and define tasks at the start of an iteration, and each team member does a subset of those tasks. Tasks can include development, testing, and other kinds of work. For example, a developer can define tasks to implement requirements, and a tester can define tasks to write and run test cases. By linking tasks to requirements and bugs, they see the progress made on these items. For more information, see Iteration activities.

Field

Usage

Task Type

Select the kind of task to implement from the allowed values:

- **Corrective Action**
- **Mitigation Action**
- **Planned**

Discipline

Select the discipline this task represents when your team estimates sprint capacity by activity.

- **Analysis**

- Development
- Test
- User Education
- User Experience

This field is also used to calculate capacity by discipline. It's assigned to `type="Activity"` in the ProcessConfiguration file. (2)

For more information, see Implement development tasks.

Original Estimate

The amount of estimated work required to complete a task. Typically, this field doesn't change after it's assigned.

Remaining Work

The amount of work remaining to complete a task. As work progresses, update this field. It's used to calculate capacity charts, the sprint burndown chart, and the Sprint Burndown report. If you divide a task into subtasks, specify hours for the subtasks only. You can specify work in any unit of measurement your team chooses.

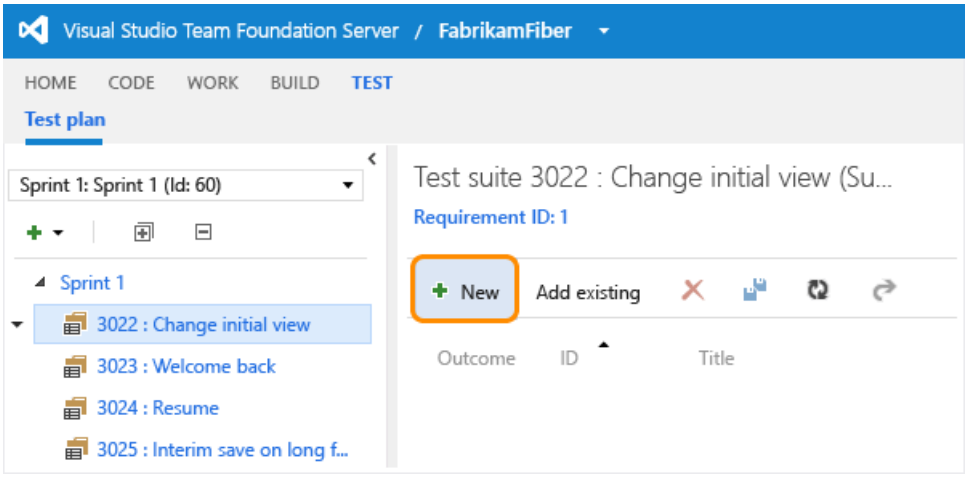
Completed Work

The amount of work that has been spent implementing a task.

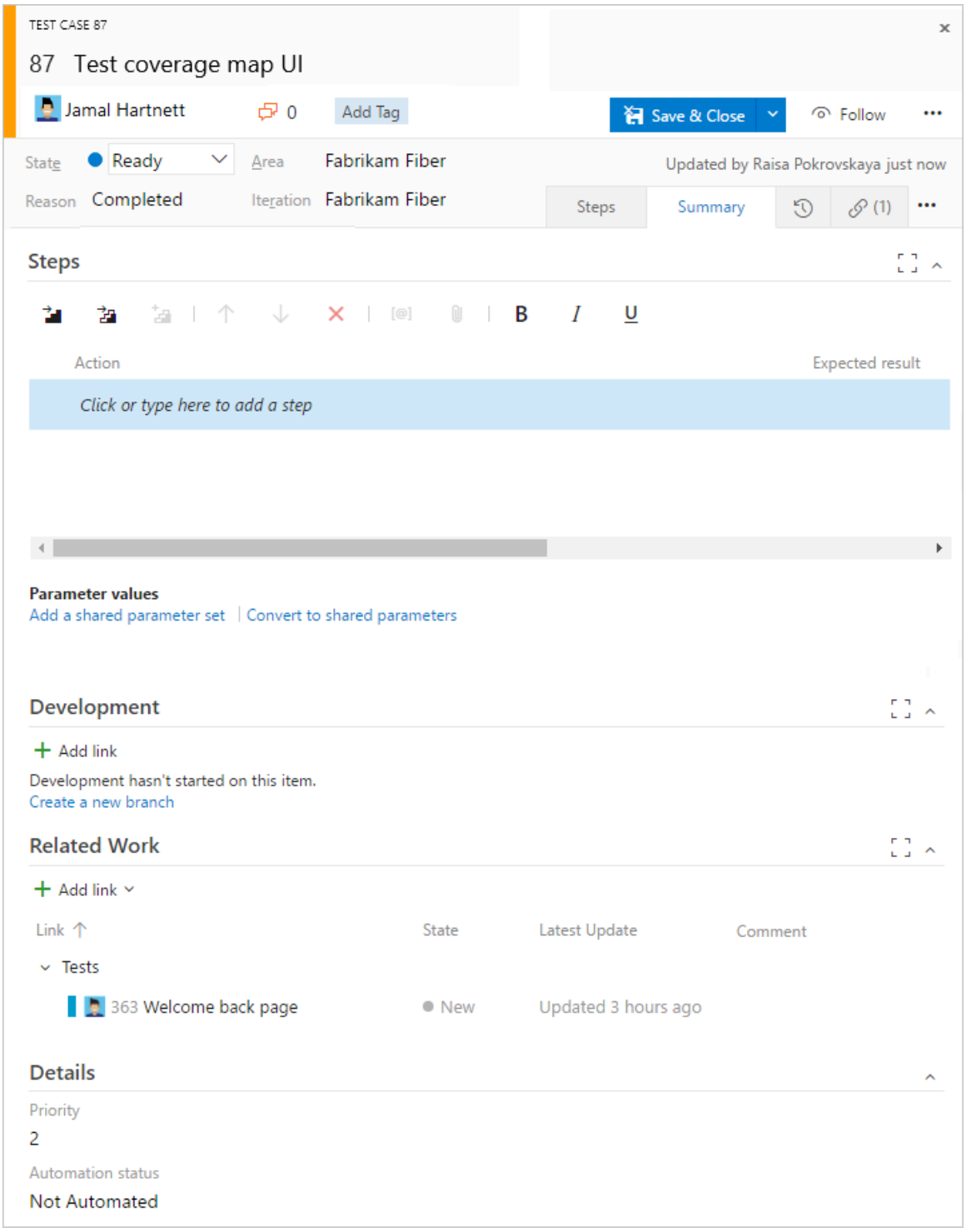
Track test progress

Test requirements

From the web portal or Test Manager, you can create test cases that automatically link to a requirement or bug. Or, you can link a requirement to a test case from the (links tab).



The test case contains many fields, many of which are automated and integrated with Test Manager and the build process. For a description of each field, see Query based on build and test integration fields.



The (links tab) lists all the requirements and bugs in a test case. By using linking, the team can track the progress made in testing each item and supports information that appears in the Requirements Overview Report report.

Track code defects

You can create bugs from the web portal web portal, Visual Studio, or when testing with Test Manager.

Track change requests, risks, issues, and notes captured in review meetings

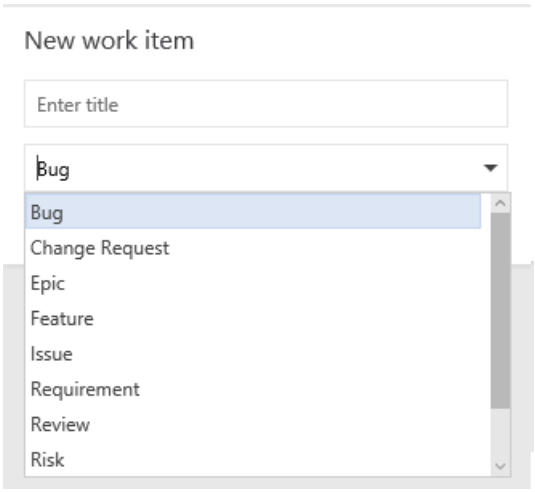
Along with the requirement, feature, task, and bug WITs, you can track information recommended by the CMMI process with the following WITS.

- Change request to manage proposed changes to any work product that is under change control.
- Issue to track an event or situation that might block work or is blocking work on the product. Issues differ

from risks in that teams identify issues spontaneously, generally during daily team meetings.

- Risk to track the probability and degree of variance between actual and desired outcomes. When you manage risks, you strategically minimize the variance between the outcome that you want and the actual outcome.
- Review to document the results of a design or code review. Team members can capture the details of how the design or code meets standards in areas of name correctness, code relevance, extensibility, code complexity, algorithmic complexity, and code security.

  You can add an issue from the New work item widget added to a team dashboard, or from the **New** menu on the Queries page.



Work items you add from the widget are automatically scoped to your team's default area and iteration paths. To change the team context, see Switch team context.

Definitions for common work tracking fields

The following fields and tabs appear in most work items. Each tab is used to track specific information, such as history, links, or attachments. These three tabs provide a history of changes, view of linked work items, and ability to view and attach files.

The only required field for all work item types is **Title**. When the work item is saved, the system assigns it a unique **ID**. The form highlights required field in yellow. For information about other fields, see Work item field index.

Field/tab

Usage

Title

Enter a description of 255 characters or less. You can always modify the title later.

Assigned To

Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu will list only team members or contributors to the project.

State

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current state.

Reason

Use the default first. Update it when you change state. Each State is associated with a default reason.

Area

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting.

To change the dropdown list of areas, see Add and modify area and iteration paths.

Iteration

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later, during a planning meeting.

To change the drop-down list of iterations, see Add and modify area and iteration paths.

 (History)

Review the audit trail that the system captures and capture additional information.

Every time that the work item is updated, information is appended to the history. History includes the date of the change, who made the change, and which fields were changed. You can also add formatted text to the history field.

 (Links)

Add all types of links, such as hyperlinks, changesets, source files, and so on.

This tab also lists all links defined for the work item.

 (Attachments)

Share more detailed information by adding files to the work item, such as email threads, documents, images, log files, or other file types.

Customize work item types

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process or Customize the On-premises XML process model depending on the process model used by your project.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize the On-premises XML process model.

Related articles

Before you start tracking work, you must have a project. To create one, see Create a project.

If you have a project, start tracking work:

- Add work items to manage a project - to gain more familiarity with the work item form features
- Create a backlog - to develop your product backlog
- Kanban - to start working in Kanban

- Plan a sprint - to start working in Scrum
- Excel.

For more information on Agile tools:

- Team assets
- Best tool to add, update, and link work items

Backlog list order

The Stack Rank field is used to track the relative ranking of requirements, features, or epics. However, by default it doesn't appear on the work item form. The sequence of items on the backlog page is determined according to where you've added the items or moved the items on the page. As you drag items, a background process updates this field.

This field doesn't appear on the work item form.

Background to Capability Maturity Model Integration (CMMI)

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The definitive guide to the Capability Maturity Model Integration (CMMI) for Development is published by the Software Engineering Institute as "CMMI: Guidelines for Process Integration and Product Improvement." This book specifically describes the CMMI for Development (CMMI-DEV) version 1.3, which is one of the models within the CMMI product suite. You may also find "CMMI Distilled: A Practical Introduction to Integrated Process Improvement" to be a useful and accessible book about CMMI.

> **NOTE**
> The guidance provided here is based on version 1.3 for CMMI and supports the CMMI process available with Azure DevOps. No plans exist at this time to update this content to support later versions.

Historical notes

The CMMI began in 1987 as the Capability Maturity Model (CMM), a project at the Software Engineering Institute (SEI). SEI is a research center at Carnegie-Mellon University, which was established and funded by the United States Department of Defense. First published in 1991, the CMM for Software began as a checklist of critical success factors. The model also built upon research at International Business Machines (IBM) Corporation and 20th-century quality assurance leaders such as Philip Crosby and W. Edwards Deming. Both the name, *Capability Maturity Model*, and the Staged Representation five levels were inspired by Crosby's Manufacturing Maturity Model. Applied mainly to defense programs, CMM achieved considerable adoption and underwent several revisions. Its success led to the development of CMMs for a variety of subjects beyond software. The proliferation of new models was confusing. In response, the government funded a two-year project to create a single, extensible framework that integrated systems engineering, software engineering, and product development. This effort involved more than 200 industry and academic experts. The result was CMMI.

CMMI-DEV is a model. It isn't a process, nor a prescription to be followed. Instead, CMMI-DEV provides a set of organizational behaviors that have proven to be of use in software development and systems engineering. Why use such a model? What is its purpose? And how best should it be used? These critical questions are perhaps the most misunderstood issues with CMMI.

Why use a model?

Improvement efforts require a model of how your organization works, which functions they need, and how those functions interact. A model gives you an understanding of organizational elements and assists in discussions of how and what can and should be improved.

A model offers the following benefits:

- Provides a common framework and language to help communicate
- Leverages years of experience
- Helps users consider the large picture while focusing n improvement
- Is often supported by trainers and consultants

- Can help solve disagreements by providing agreed-upon standards

What is the purpose of the CMMI model?

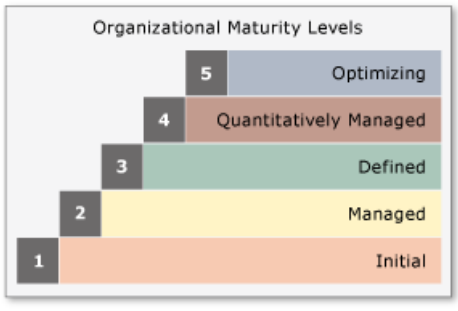
The purpose of the CMMI model is to assess the maturity of an organization's processes and to provide guidance on improving processes, with a goal of improved products. Also, CMMI is a model for risk management and provide a way to measure an organization's ability to manage risk. The ability to manage risk factors factors into an organizations ability to deliver high-quality products. Another perspective on managing risk is how well an organization will perform under stress. A high maturity, high capability organization can easily respond to unexpected, stressful events. A low maturity and lower capability organization tends to panic under stress, blindly follow obviated procedures, or throw out all process altogether and retrench back to chaos.

The CMMI, however, isn't a proven indicator of the economic performance of an organization. Although higher maturity organizations may manage risk better and be more predictable, evidence exists that higher maturity firms tend to be risk-averse. Risk aversion can lead to a lack of innovation or evidence of greater bureaucracy that results in long lead times and a lack of competitiveness. Lower maturity firms tend to be more innovative and creative but chaotic and unpredictable. When results are achieved, they are often the result of heroic effort by individuals or managers.

What's the best way to use the CMMI model?

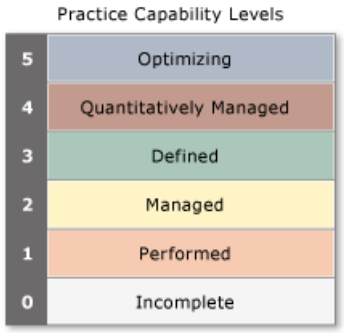
The model was designed to be used as the basis for a process improvement initiative, with its use in assessment only a support system for measuring improvement. There has been mixed success with this usage. It is all too easy to mistake the model for a process definition and try to follow it, instead of a map that identifies gaps in existing processes that may need to be filled. The fundamental building block of CMMI is a process area that defines goals and several activities that are often used to meet them. One example of a process area is Process and Product Quality Assurance. Another is Configuration Management. It is important to understand that a process area is not a process. A single process may cross multiple process areas, and an individual process area may involve multiple processes.

The CMMI-DEV is really two models that share the same underlying elements. The first and most familiar is the Staged Representation, which presents the 22 process areas mapped into one of five organizational maturity levels. An appraisal of an organization would assess the level at which it was operating, and this level would be an indicator of its ability to manage risk and, as, deliver on its promises.



Levels 4 and 5 are often referred to as higher maturity levels. There is often a clear difference between higher maturity organizations, which demonstrate the quantitative management and optimizing behaviors, and lower maturity organizations, which are merely managed or following defined processes. Higher maturity organizations show lower variability in processes and often use leading indicators as part of a statistically defensible management method. As a result, higher maturity organizations tend to be both more predictable and faster at responding to new information, assuming that other bureaucracy doesn't get in the way. Where low maturity organizations tend to demonstrate heroic effort, high maturity organizations may blindly follow processes when under stress and fail to recognize that a process change may be a more appropriate response.

The Continuous Representation models process capability within each of the 22 process areas individually, which allows the organization to tailor their improvement efforts to the processes that offer the highest business value. This representation is more in line with Crosby's original model. Appraisals against this model result in profiles of capability rather than a single number. Because the organizational maturity level is the level that most managers and executives understand, there are ways of mapping the results of a continuous model assessment into the five stages.



Using the staged model as a basis for a process improvement program can be dangerous when implementers forget that the CMMI isn't a process nor a workflow model. Instead, the CMMI is designed to provide goals for process and workflow to achieve. Meeting such goals will improve the maturity of the organization and the likelihood that events unfold as planned. Perhaps the biggest failure mode is making achieving a level the goal and then creating processes and infrastructure simply to pass the appraisal. The goal of any process improvement activity should be measurable improvement, not a number.

The Continuous model has enjoyed success as a guide to process improvement. Some consulting firms choose only to offer guidance around the Continuous model. The most obvious difference is that a process improvement program that is designed around the Continuous model doesn't have artificial goals that are determined by maturity levels. The Continuous model also lends itself to applying process improvement in areas where it is most likely to leverage an economic benefit for the organization. Therefore, those who follow the Continuous model are more likely to receive positive feedback from an initiative that is based on the CMMI model. Moreover, positive feedback is more likely to lead to the development of a virtuous cycle of improvements.

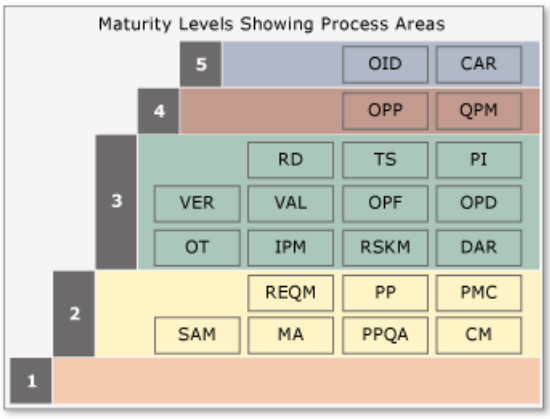
Elements of the CMMI model

The following table lists the 22 process areas that comprise the CMMI model (version 1.3):

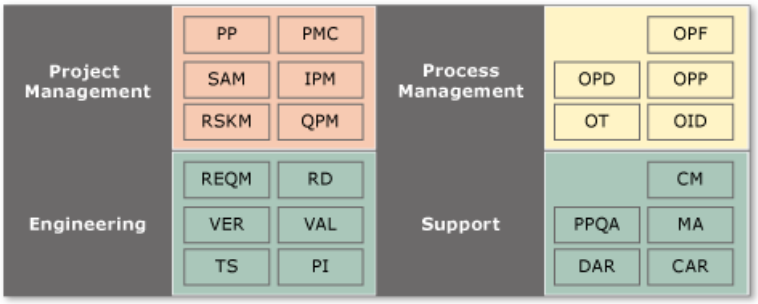
ACRONYM	PROCESS AREA
CAR	Causal Analysis & Resolution
CM	Configuration Management
DAR	Decision Analysis & Resolution
IPM	Integrated Project Management
MA	Measurement & Analysis
OID	Organizational Innovation & Deployment
OPD	Organizational Process Definition

ACRONYM	PROCESS AREA
OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training
PI	Product Integration
PMC	Project Monitoring & Control
PP	Project Planning
PPQA	Process & Product Quality Assurance
QPM	Quantitative Project Management
RD	Requirements Definition
REQM	Requirements Management
RSKM	Risk Management
SAM	Supplier Agreement Management
TS	Technical Solution
VER	Verification
VAL	Validation

In the Staged Representation, the process areas are mapped against each stage, as shown in the following illustration.



In the Continuous Representation, the process areas are mapped into functional groupings, as shown in the following illustration.



Each process area is made up of required, expected, and informative components. Only the required components are actually required to satisfy an appraisal against the model. The required components are the specific and generic goals for each process area. The expected components are the specific and generic practices for each specific or generic goal. Note that, because an expected component is merely expected and not required, this indicates that a specific or generic practice can be replaced by an equivalent practice. The expected practices are there to guide implementers and appraisers. If an alternative practice is chosen, it will be up to the implementer to advise an appraiser and justify why an alternative practice is appropriate. Informative components provide details that help implementers get started with a process improvement initiative that is guided by the CMMI model. Informative components include subpractices of generic and specific practices and typical work products.

Only generic and specific goals are required. Everything else is provided as a guide. For examples of expected and informative components, the CMMI literature pulled data from large space and defense-systems projects. These projects might not reflect the type of projects that are undertaken in your organization, nor may they reflect more recent trends in the industry, such as the emergence of Agile software development methods.

Related articles

- CMMI process
- Software Engineering Institute Releases Version 1.3 of CMMI Product Suite
- CMMI for Development: Guidelines for Process Integration and Product Improvement, Third Edition
- CMMI for Development: Guidelines for Process Integration and Product Improvement (SEI Series in Software Engineering)
- What is Agile Development?

Project management process in Azure Boards

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

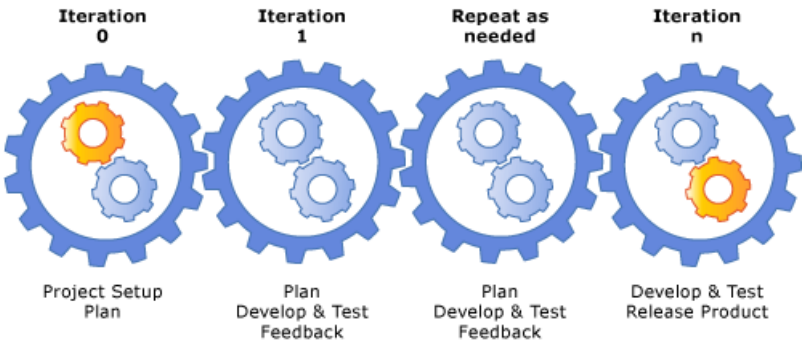
You can use the Project Management section of the MSF for Capability Maturity Model Integration (CMMI) process improvement guidance to better understand how to manage, plan, and coordinate the development and maintenance of software products. For more information about CMMI, see Background to CMMI.

The Project Management grouping of process areas in the CMMI includes Project Planning, Project Monitoring and Control, Supplier Agreement Management, Integrated Project Management, Risk Management, and Quantitative Project Management. All but Quantitative Project Management are part of model levels 2 or 3. Quantitative Project Management is a model level 4 activity that reflects how high-maturity organizations use quantitative, statistically defensive, unbiased data to make management decisions and to steer projects to a successful and predictable outcome.

The project management activities represent economic costs on value-added engineering activities. These activities are necessary and important to manage risk, coordinate successful engineering efforts, and set customer expectations appropriately. However, you should minimize the effort that is used on these activities. "Little and often" is a good mantra. Smaller batches reduce complexity and coordination costs. When you define and tailor your process definition, you should keep in mind that your project management activities should be as minimal as possible while satisfying the risk profile of your project.

Implement iterative development

Team Foundation together with the MSF for CMMI process template supports iterative work. Iterative development manages risk by delivering demonstrable and tested software at set intervals throughout the project.



The project schedule is organized into a series of iterations that are typically four to six weeks long. Each iteration ends with a demonstration of usable, tested software. To schedule sprints, see Schedule sprints.

- The *project plan* states what feature requirements will be developed in each iteration. The project plan is developed in Iteration 0 and reviewed at the start of each iteration. To create and view the project plan, see Create backlog.
- Each *iteration plan* states what tasks will be performed during that iteration. Most tasks are development and test work that fulfills the feature requirements in that iteration. The iteration plan can be viewed through the sprint backlog page.

  Iterative work doesn't automatically manage risks. To minimize risk, you must arrange the project plan in increments. Early iterations should provide an "end-to-end thin slice," that is, a minimal version of the

most important behaviors of the product. Later iterations add more functionality.

By contrast, it would be much less useful to schedule the sales part of a shopping Web site for the first third of the project, the warehouse system in the second third, and the payments system in the last third. This schedule would risk producing an attractive and feature-rich sales Web site that has no means for the business to take money from its customers. It's iterative without being incremental.

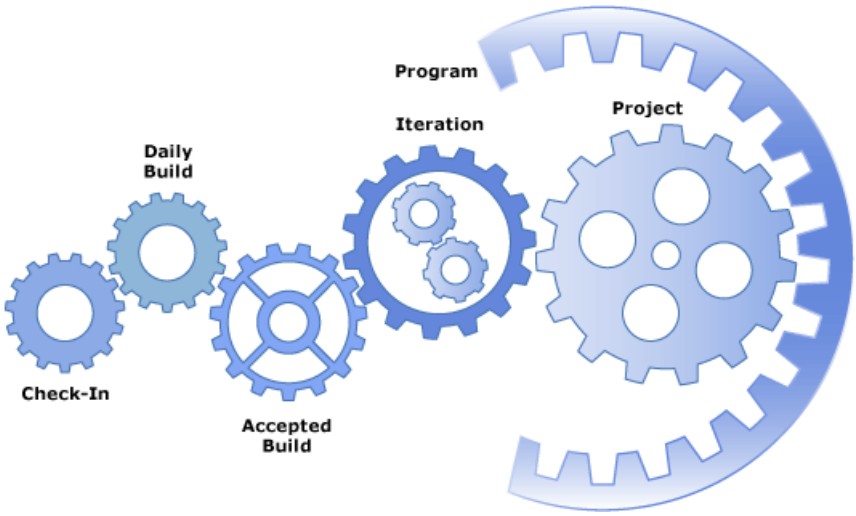
Incremental development has the following benefits:

- Meets the true requirements. Stakeholders have the opportunity to try out the product, which always results in improvements to their stated requirements.
- Tunes the architecture. Allows the development team to discover and address any difficulties that occur with their platform or potential improvements to their design.
- Ensures results. Stakeholders know that, even if project resources are cut part-way through, the expenditure to date hasn't been not wasted. The same is true if the development estimates prove to have been optimistic and you must drop the less important features.

  For more information about how to express the requirements in an appropriate form for incremental development, see Develop requirements.

Larger and smaller cycles

The project and the iteration aren't the only cyclic aspects of software development. For example, in an iteration, team members start and complete tasks and check in code. The build system builds the product on a continuous or nightly basis. The team holds a brief daily review of progress on the iteration tasks.



Large projects

A project in which a team works through a series of iterations may be part of a larger project or program. A large project has several teams that work in parallel. Each team typically has four to 16 people.

Open a separate version control branch for each team. Each team should integrate with the main branch at the end of each iteration. For more information, see Use branches.

Reserve the main branch for integration and tests. The build machine should run a complete set of tests after an integration.

Assign an area to each team so that its work items can be easily separated from the others. For more information, see Customize area and iteration paths.

The teams can share a series of integrations, but sharing isn't always necessary. If the teams don't synchronize integrations, each team must have its own prefix for its iteration names.

Related articles

- **The Project Cycle/Project activities**: Start the project, gather requirements, create a project plan, divide it into iterations, and deliver releases. Manage risks, and manage changes to the plan.
- **The Iteration Cycle/Iteration activities**: Review and update requirements, plan the tasks to implement requirements, and manage issues as they occur.
- CMMI process

Manage project activities in Azure Boards

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To make the most effective use of MSF for CMMI Process Improvement, you should organize your project into a series of iterations, typically between four and eight weeks long. This organization helps you reduce the risks to your project that stem from shifting requirements and implementation costs. Iterative project structure is an important contribution to meeting the risk management requirements of CMMI.

At the start of the project

Project initiation

Initiation includes defining the project vision, which states what users can do when the project releases its product.

It also includes setting up the team, the infrastructure, and other resources and determining the development process.

For more information, see Project inception.

Initial project planning

Project planning includes the following activities:

- Analyzing the requirements in sufficient detail to enable you to form a plan. This analysis can include the use of requirements models, storyboards, and other tools that help envisage the working system.
- Devising an overall design or architecture for the system. If this design involves working on a platform that is new to team members, some time must be assigned to experimenting with it. Development will be slow in the earlier iterations.
- Casting the requirements as a set of incremental product requirements whose development can be estimated. The difference between general requirements and product requirements is an important one, and this activity is significant. For more information, see Develop requirements.
- Making an initial assignment of product requirements to iterations.
- Setting dates for releases.

  The plan and requirements models will be revisited and refined throughout the project. Part of the purpose of iterative development is to allow improvements in the requirements that stem from demonstrating working software at an early stage.

  Initial project planning is done in Iteration 0.

  For more information, see Plan a project.

Exploring an existing product

The goal of your project might be to update a product that already exists. In this case, if the team is unfamiliar with the product, exploration of the code is an activity for Iteration 0. Each development task in later iterations will also involve understanding the code in a particular locality and tracing the consequences of changing it.

For more information, see Visualize code.

During the project

The plan is reviewed and subject to change throughout the project.

Several activities that are related to the project plan are done regularly throughout the project, usually toward the end of an iteration.

Validation

Demonstrate to your customers or business stakeholders the software that has been developed during the iteration. Where feasible, release it to them so that they can experiment with it or use it to a degree in a practical context.

After a sufficient interval, arrange a meeting to review user feedback. The feedback should be used to generate change requests.

Risk management

Review the probability and impact of potential adverse events, and take steps to reduce the risks. For more information, see Manage risks.

Change management

You can use change request work items to record changes in the requirements that are stated by the business stakeholders. They can stem from changes in the business context but also from demonstration and trials of early versions of the product. These changes should be welcomed because they improve the fitness of your product to its business purpose. This effect is part of the goal of incremental development.

Some project teams adjust the product requirements work items when changes are requested, without using a separate work item. But the advantage of the change request work item is that, in the later part of the project, you can review the number and nature of the changes that were made. You can use that information to improve your process or architecture for the future.

Change requests should be used as input to the Product Plan Review.

For more information, see Manage change.

Product plan review

Hold a Product Plan Review before you plan each iteration. The project plan assigns product requirements to iterations.

The plan will change for two principal reasons:

- Changes in requirements.
- Changes in the estimates that the developers made. As the project progresses, the development team can make more reliable estimates of the work that will be required to implement future features. In some cases, some functionality might have been postponed from a previous iteration, which adds a feature to the plan.

  Both types of change become less frequent in later iterations.

  Revise the requirements models from which the product requirements are derived.

  Revise the assignment of requirements to iterations. As in the initial planning activity, the business stakeholders provide the priorities, the development team provides the estimates, and the meeting juggles the features between iterations.

  For more information, see Plan a project.

Before major releases of the product

The activities that are involved in deployment of a product vary according to its type and aren't dealt with here.

Consider the following points in respect to the later iterations of software development:

- Exclude major changes to the design to avoid the chance of unforeseen problems.
- Raise the bar for changes and bugs in triage meetings. Proposed changes and bug fixes should be rejected unless they have significant effects on the usability and fitness for purpose of the product.
- Devote resources to increasing test coverage and to doing manual tests.

Related articles

- Background to CMMI
- CMMI process

Project initiation

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You arrange the basic resources of the project in an initial stage that is named Project Inception.

Hold a planning meeting

At an early stage in the project, several stakeholders and subject matter experts should be convened to discuss the project and make the product plan. Choose stakeholders based on the nature and complexity of the project and its product deliverable.

Depending on the size of the project and its complexity, the meeting may take several days or weeks.

Practice iterative development

An important technique in risk management is planning your project in iterations, typically of four to six weeks. An iteration plan is a list of features that the project team will develop and test. Each feature specifies a task or an improved variant of a task that the user can do by using the product. At the end of each iteration, the planned features are demonstrated. At the end of some iterations, the partly completed product is released for trial by a limited set of users.

The feedback from those demonstrations and trials is used to review the plan.

The product plan is arranged so that the principal user scenarios and the main components of the system are exercised at an early stage, even if only in a simplified manner.

One of the most significant risks in most projects is misunderstood requirements. Requirements can be misunderstood not only by the development team but also by the end users and stakeholders. They can find it difficult to envisage how their business activities will be affected by the installation of the new system.

Also, the business context may change during the lifespan of the project, leading to a change in the product requirements.

An iterative process provides assurance that any adjustment in the requirements that is found by demonstrating the product can be accommodated before the end of the project, without incurring the costs of significant rework.

Another significant risk is poorly estimated development costs. It can be difficult for developers who work in a new area, and perhaps on a new platform, to estimate development costs in advance of the project. In some cases, it can be difficult to determine whether a particular implementation strategy will perform sufficiently well. But by reviewing the plan at the end of each iteration, the team can consider the experience of the previous iterations. This activity is one reason why a good product plan schedules some work on every principal component at an early stage.

Is this project scope driven or date driven?

Some projects require that all the requirements must function before delivery. These kinds of projects are unusual in a software context. A real-world example might be building a bridge. A half-finished span is useless. Conversely, a half-completed but properly planned software project should be deployable and usable for a limited set of users. It can then be completed incrementally over the course of several upgrades.

First determine whether your project is truly scope-driven. If it is, you must wait to determine an end date until you have detailed estimates and a detailed plan. You pay a price for waiting. Planning overhead is increased, and schedule buffering as a contingency against poor estimation will push the delivery date out more, which increases costs. As such, before you decide that you have a scope-driven project, be sure. It's more probable in a complex systems-engineering environment than in a pure software product or service situation.

Most software projects are date-driven because they can be delivered incrementally. For example, if a computer game is to be released for the holiday season in the United States, it must be ready by October. Failure to deliver in October will severely affect sales between Halloween and Christmas, and, if the schedule slips by two months, the window of opportunity may be lost altogether.

Plan project resources

A project should be staffed so that it can be delivered by the desired date. Historical data from previous projects should be used to inform a discussion about sufficient resources.

After you understand your staff requirements, create a project organization chart that clearly identifies the project team structure, resourcing levels, and geographic distribution, if appropriate. Save all staffing information to your project portal.

Define roles and responsibilities

Describe each project role and its responsibilities, and publish them in the project plan. Each person who joins the project should understand their role and responsibilities in the project.

Define a communication plan

It's important to define a communication plan for the project. Paths of communication help manage the coordination costs on the project. It is important to define who should attend meetings, how often meetings are held, paths of communication, and how to escalate issues that cannot be resolved by the usual attendees of any one meeting.

The goal of a good communication plan is to make sure that coordination activities on the project run as smoothly as possible and to avoid wasted effort through miscommunication.

The communication plan should be published to the project portal and maintained as it's required. A communication plan is a useful tool for all staff, new members. It helps them understand how a larger team works and how to get things done by communicating appropriately in different ways, with different team members, and for different purposes.

Identify stakeholders

Identify all relevant project stakeholders. Along with the core team members, the list should include business people and technical people who have an interest in the successful implementation of the project or the effect that the product might have after it enters service. These stakeholders may be upstream or downstream of the software engineering activity.

Outline the project plan

Create a sketch version of the first project plan, which can be revised when development starts. The purpose of this version is to help discuss resources and timescales with project sponsors. It should outline the major features and their estimated delivery dates. For more information, see Plan a project.

Review the project plan

Publish the outline of the project plan on the project portal. Although much work has gone into the plan, it's still a high-level plan that defers many detailed scheduling decisions. This deference is intentional. Too much detail now will generate waste later.

Where requirements are uncertain, plan them in outline only, and detail should be deferred until more information is available. Plan to obtain that information.

Schedule a review meeting with all stakeholders. Face-to-face meetings are always best for this kind of activity. Be sure to schedule enough time to enable a full review and to allow dissenting opinions to be heard.

Obtain Project Commitments

Now that the project plan is agreed upon with the project stakeholders, obtain commitments from each stakeholder to approve the project plan.

Collect the commitments, and archive the details in the project portal.

Other Resources

For more information, see the following Web resources:

- A Practical Guide to Feature Driven Development, Stephen R. Palmer and John Malcolm Felsing; Prentice Hall PTR, 2002.
- The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement, Manfred Bundschuh and Carol Dekkers; Springer, 2008.

Plan a project using the Capability Maturity Model Integration (CMMI) process in Azure Boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The expected outcome of planning a project is a plan that includes a scope, a schedule, a budget, a risk management plan, and a commitment and approval from all stakeholders. With an agreed-upon project plan, you want to progress with analysis, design, development, testing, and eventually delivery.

You can reduce risk by using an iterative development method. Iterations let you demonstrate a partly working product at the end of each iteration and act on feedback from that demonstration. So, the plan provides an overall shape and is subject to review and refinement before the start of each iteration.

Gather and model the requirements

This activity is about discussing what the system should do, with business stakeholders, prospective users, and subject matter experts. It's important to understand the business context. If you've been asked to write an application for police officers, it helps to understand their jargon, procedures, and rules.

UML models are a useful tool for expressing and thinking about complex relationships. You can draw them in Visual Studio and link them to other documents and to Team Foundation work items. For more information, see Model user requirements.

Update and refine the requirements model throughout the project. As each iteration approaches, add more detail to the aspects of the model that are relevant to that iteration. From the model, you can derive verification tests.

For more information, see Develop requirements and Develop tests from a model.

Create incremental product requirements

The requirements as you've gathered them from your customers aren't directly appropriate for scheduling incremental development. For example, to clarify the procedure when a user buys something from a Web site, you might have written a detailed series of steps: customer browses catalog, adds item to cart, checks out cart, supplies address, and pays. Then, the warehouse schedules delivery and the procedure continue. These steps, or an equivalent activity diagram, aren't incremental requirements.

Instead, the first increment of your system may offer only one item for sale, deliver to only one address, and run only a test transaction with the payment service. The second increment might provide a catalog that consists of a simple list. Later increments could add the option of gift wrapping the purchase or of requesting catalogs that are provided by different vendors. Some increments might be about quality of service, such as the ability to handle 1,000 customers instead of only one.

In other words, the early increments should exercise the major use cases end-to-end and gradually add functionality throughout.

The principle is the same if you work with an existing product, but you start from the existing functionality. If you're unfamiliar with its internal design, the cost of updates can be difficult to estimate. It's worth being liberal with your estimates for the earlier changes.

For more information, see Develop requirements.

Enter and edit product requirements

Record the incremental product requirements as requirement work items in Team Foundation, and set the requirements type to Feature. You can create a backlog of requirements using the web portal or Team Explorer. If you have several work items that you want to create at the same time, you can use the Excel.

Estimate the product requirements

The development team should estimate the work that is required to develop each product requirement. The estimate should be entered in hours, in the Original Estimate field of the work item.

Early in the project, a rough estimate is all that is needed.

Break large product requirements into smaller ones. Ideally, each product requirement will take only a few days of development time.

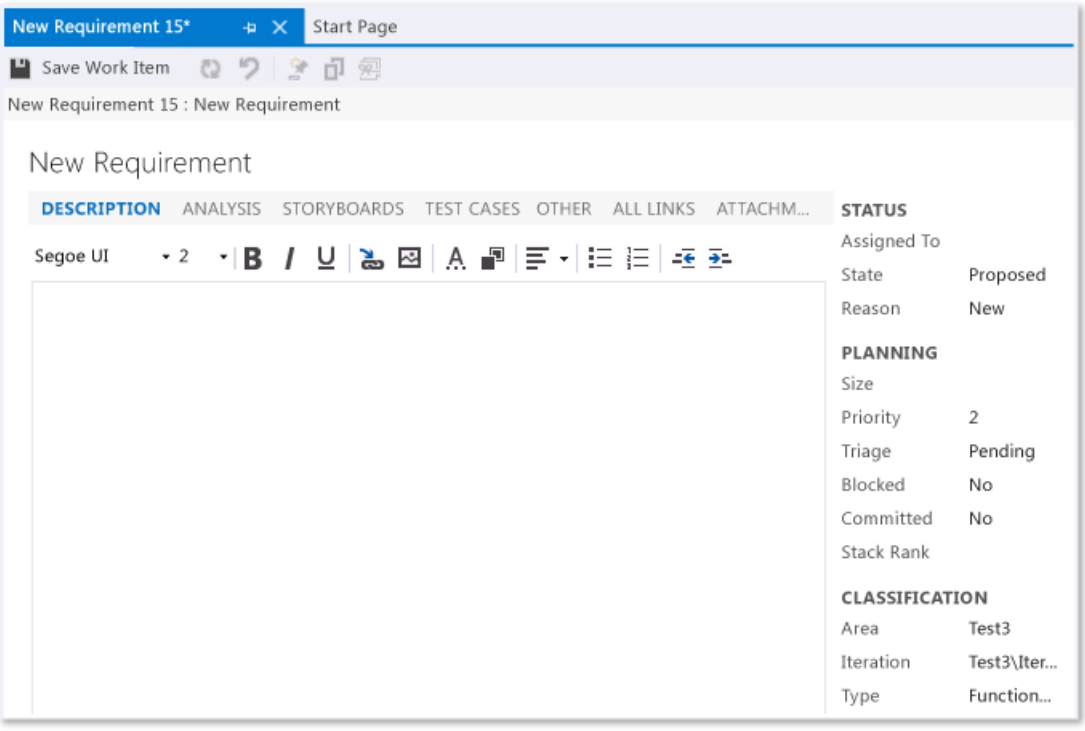
Assign product requirements to iterations

Representatives of the business stakeholders and the development team should work together to assign product requirements to iterations. Typically, you assign the requirements in a meeting, where you share or project the Office Excel view of the Product Requirements query.

The assignment is completed by using the following pieces of information:

- The priority of the requirement. See the notes in the following subsection.
- The estimated cost. Given the number of team members and the length of the iteration, each iteration has only a fixed number of hours that are available for development. Furthermore, a significant number of those hours will be used for iteration planning and other tasks that don't directly involve development.
- Dependencies among the product requirements. In an incremental series of requirements, the simplest requirements must be tackled before enhancements in the same area.

  You can define the requirement in a work item by specifying various information, as the following illustrations show:



Some guidelines on prioritization

Many detailed schemes exist for prioritization. We'll examine some of these schemes when we consider iteration planning. For now, at the project level, we include some guidelines that may be useful to help manage risk and optimize added value.

1. Prioritize minimal end-to-end scenarios.

   Aim to achieve an end-to-end scenario as early in the project as possible. Later, add more features to the different parts of the scenario. This practice ensures that the principal functions of the platform and the principal ideas in the requirements are tried early.

   By contrast, don't divide the schedule according to the architecture. A schedule that completes the database, then the business logic, and then the user interface will probably require a great deal of rework to integrate the parts at the end. In the same manner, a horizontal split such as {sales component, warehouse component, payment component} isn't recommended. It would probably produce a wonderful system for selling on the Web but run out of time before the business has a means of getting money from its customers. Complete components can be scheduled for later iterations only if they're truly optional add-ons.

2. Prioritize technical risk.

   Develop technically risky elements early in the schedule. Take a "fail early" approach to risk. If something can't be accomplished, you want to know this risk early in the project so that it can be canceled or replaced with an alternative approach. So prioritize technically risky requirements into early iterations.

3. Prioritize reduction of uncertainty.

   The business stakeholders won't be sure about some requirements. It's difficult to predict what product behavior will work best in the business context. Prioritize work that is likely to reduce the uncertainties. This risk reduction can often be achieved by developing a simpler version of the scenario with which users can experiment. Defer the full scenario to a later iteration, in which the results of these experiments can be considered.

4. Prioritize highly valuable requirements.

   If possible, try to establish an opportunity-cost-of-delay function for each scenario. Use these functions to determine the requirements that can potentially bring more value to the customers earlier. Prioritize these requirements into earlier iterations. This prioritization may buy you the option of releasing a partial product early

5. Group scenarios that are common to multiple personas.

   If you have scenarios that have utility for two or more personas, group these scenarios together. Rank them by the number of personas that require the scenario. Prioritize the scenarios that apply to a larger number of personas into early iterations.

6. Rank personas.

   Personas represent market segments or user groups. Marketing people or business owners can express the priority of such segments or groups based on utility to be delivered or the value of the segment. If segments or user groups can be ranked in priority, show this rank by listing the personas for each segment by rank. Identify the scenarios for the highest ranked personas, and prioritize these into earlier iterations in the schedule.

   In general, prioritize the reduction of risk because of the possibility of failure. Prioritize common functionality because it's likely to be required and unlikely to change. We want to prioritize more valuable requirements. We want to enable the option for early release of the product to a subset of personas by prioritizing all scenarios that are required to satisfy the needs of any one persona.

Plan tests

The work estimate for each requirement must include the effort that is required to test the requirement, either manually or by creating an automated test.

Before it's considered completed, each product requirement must be linked to a set of test case work items that together demonstrate whether the requirement has been met, and the tests must pass.

When you create or revise product requirements, the corresponding test plan must be updated.

Revise the product requirements

Revisit this activity before each iteration to consider revised and new requirements, revised priorities, and revised estimates. There will be more revisions in the first few iterations.

After the first few iterations, members of the development team will be more confident about the estimates. They should go through the estimates for the next one or two iterations and revise the Original Estimates fields of the requirements that are assigned to those iterations.

Related articles

- Background to CMMI
- CMMI process

Manage change with the CMMI process in Azure Boards

10/4/2022 • 3 minutes to read • Edit Online

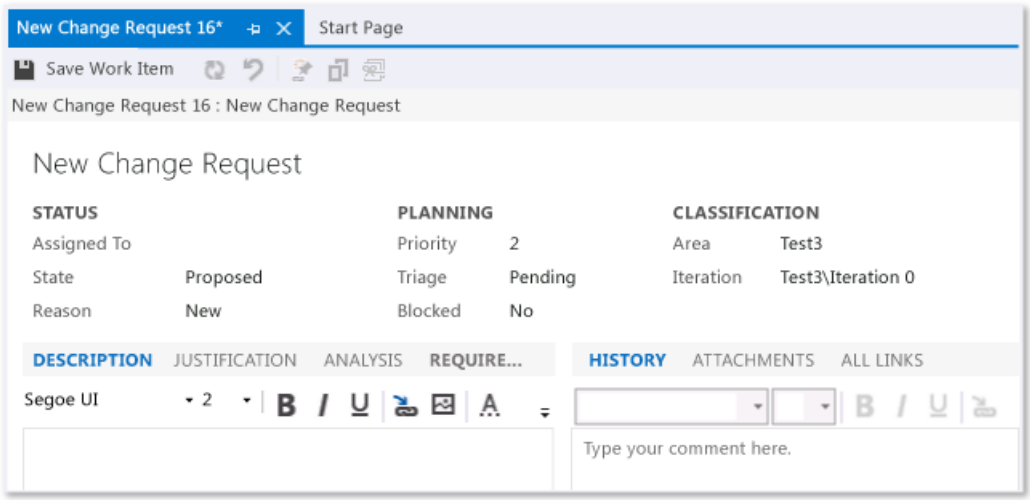
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can use change request work items to track and control all changes to the product and supporting systems. All change requests are started as the result of a deviation from the baseline, which consists of the original requirements that were identified for the project. For example, if a meeting with a user uncovers new requirements, a change request should be created to propose updating the requirements baseline. For more information about CMMI, see Background to CMMI.

Create a change request

When you realize that an original requirement must change, you create a change request work item and link it to the old requirement work item by using an Affects link type. A requirement work item that has details of what is new or has changed should also be created and linked to the change request. All change requests are extensively analyzed for impact on the user, product, and teams. During this analysis, tasks may be broken out for estimation. These new task work items should be linked to the new requirement work item to provide traceability. This result is accomplished by adding the tasks on the Implementation tab of the work item form.

The change request and resultant new work items must contain details of all new work that is required and all existing work that is to be removed, modified, or obviated. You can create a change request from the home page of the web portal or from the Work page in Team Explorer as described in Get started using work items. You can specify the change that you're requesting in the Title field, the team member who owns the change, and other information about the request.



For more information about how to complete the work item, see CMMI work items and workflow.

Analyze a change request

Before a change request is analyzed, it should be triaged by a configuration control board. A configuration control board is a group of people who are responsible for approving and denying change requests and who ensure that changes are implemented correctly. You can indicate that a request must be triaged by setting the Triage field in the work item to Pending. For more information, see Change request fields (CMMI). Analysis of

change requests can be a drain on resources, and it's important that the change request queue doesn't put undue demands on the team and affect the project timeline.

A change request should be analyzed to determine the scope of its impact on existing and planned work. The effect must be known so that it can be used to estimate the cost in person-hours to implement the change.

Analyze the risk of accepting the change. Are external teams dependent upon the code or feature that would be changed, and could their schedules be adversely affected? Could assignment of resources to this change adversely affect other important feature areas or requirements of the product?

As part of your analysis, request input from stakeholders and add input to the change request work item. If the change requires changes to other planning documents, add the changes to the change request. Then, change those documents as appropriate. These changes will maintain the revision history and enable everyone to see the details. The revision history mitigates risk from bad communication and provides important evidence for a Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal.

If a change request is accepted, change the State from Proposed (the default for new change requests) to Active.

Monitor change requests

While a change request is active, you can monitor it by viewing the Open Change Requests With Requirements query. Also, you can create an email alert for when a Change Request is created. Change requests should be processed in a reasonable amount of time.

If a change request doesn't receive the attention that it requires, escalate the matter by creating an issue work item. Link the new issue to the change request, and escalate the issue to get the change request impact assessment on track.

Manage risks with the Capability Maturity Model Integration (CMMI) process in Azure Boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Risk implies that actual outcomes may vary, sometimes significantly, from expected outcomes. Both the probability of this variance and the degree of variance between actual and expected outcomes is encapsulated in the term "risk." When you manage risk, you strategically minimize the variance between the outcome that you want and the actual outcome.

The ability to identify, classify, analyze, and manage risks is an organizational capability that is required to achieve a Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal at level 3. For more information about CMMI, see Background to CMMI.

By managing these event-driven risks, a significant contribution is made to the overall goal of managing risk at the project, portfolio, and organizational levels. Good event-driven risk management contributes to an outcome that is satisfactory to all stakeholders and that deviates little from the initially expected outcome. It contributes to an expectation of "no surprises!"

Define risks

Thinking of risks in this manner is sometimes referred to as the event-driven risk model. This term implies that a list of risks is a list of potential future events. Each risk describes some event that could occur in the future. The risk might include some information about the probability of occurrence. It should include a description of the impact that such an occurrence would have on the project plan. It may also include a description of ways to reduce the probability of occurrence and ways to mitigate the impact of occurrence. It may also include suggested forms of recovery after an occurrence.

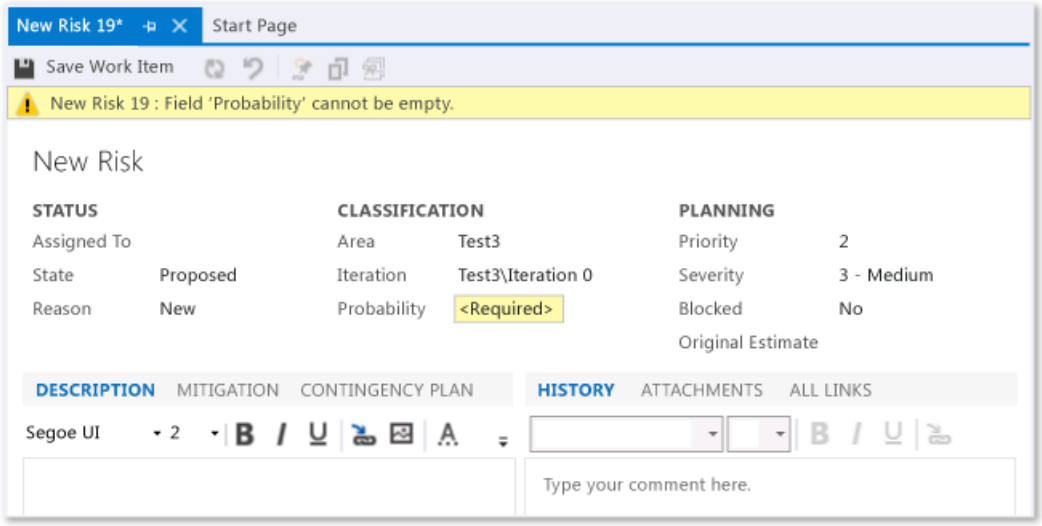
For each risk that is identified, create a risk work item in the project.

Create the risk work item

The Risk Management (RSKM) process area in the CMMI focuses on the management of these event-related risks. MSF for CMMI Process Improvement and Visual Studio Team Foundation Server make this process easier by providing the risk work item type. By using the risk work item type, you can define and track a list of risks. It provides fields to describe the risk, the probability of occurrence. It also provides fields for actions that can be taken to reduce the probability of occurrence, mitigate the impact, and implement contingency plans for recovery.

The initial project risks should be identified during project planning. The risk list should be revisited during iteration planning at the start of each project iteration.

The work item form for a risk stores data in the fields that the following illustration shows:



Select actions to be taken

After you create a list of risks and they've been sufficiently analyzed, it's time to decide what, if any, actions will be taken to manage these risks. Are there any actions that will reduce the probability of occurrence that you want to take now or describe in an iteration plan? Are there any actions that would mitigate the impact of occurrence that you want to take now or describe in an iteration plan? Taking actions to reduce or mitigate risks costs time and resources. This cost must be traded against using those resources and the available time to move the project work forward and turn the scope into working software. Document the risk reduction and mitigation actions that you plan on the Mitigation tab of the risk.

The overall risk profile for the project must be considered when you decide when to take action to reduce the probability or mitigate the impact of risks. If the risk profile says "any loss of life is unacceptable," any risk that might cause a loss of life must be managed, and reduction and mitigation actions must be planned and taken.

Ensure that risks are managed in line with project governance constraints and in an appropriate balance with the need to achieve delivery of all value-added work within the available time and budget.

If a risk is selected for reduction of probability or mitigation of its impact, plan this activity by breaking it into task work items and link each to a risk work item.

Monitor risks

Project risks should be monitored regularly.

Use the risks query to monitor risks. Scan each active risk on the list, and consider whether the probability of occurrence has increased, whether the potential impact has changed, and whether any mitigation trigger events have occurred. If there's any material change in the information that is captured for a risk, update the work item. Also, consider whether further action needs to be taken to reduce the risk or to mitigate its impact.

The current risk status of the project should be communicated. Reports should include information about any risks that were recently uncovered, any reduction or mitigation actions that are in progress, and any change in status that would cause a change in the earlier assessment of the risk.

Make contingency plans

For risks where a recovery action was defined, a plan should be made to implement the contingency if the event of occurs. For example, if there's a risk that a server might fail and the contingency is to borrow hardware from another department, you should have a plan to enact this contingency if the server fails. Making plans in

advance reduces the coordination challenge if the event occurs. A higher maturity organization with a greater capability for risk management makes contingency plans and knows how to enact them without significant impact to other project activities. Lower maturity organizations suffer panic and chaos while trying to recover from unexpected events. An organization that seeks a SCAMPI appraisal at level 3 should have documented evidence to show that contingency plans were made and, when appropriate, followed.

Break out the contingency plan into a series of tasks or actions to be taken. Estimate each task. Create a schedule and a recommended list of assigned personnel. Describe all the resources that will be required to execute the contingency plan.

Add the contingency plan to the risk work item on the Contingency Plan tab, or add the plan as an attachment.

Manage iteration activities in Azure Boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

In MSF for CMMI Process Improvement, you plan a project as a series of iterations. Each iteration is typically four to six weeks long, during which the development team implements a specified set of requirements.

Prepare for an iteration

Iteration planning takes place at or before the start of each iteration. It includes the following tasks:

- Review the requirements that are assigned to the iteration, and define them in more detail.
- Create task work items for the work that must be performed to implement and test each requirement. Link the tasks to the requirement work item by using the parent link type.
- Set the Original Estimate field of each task. Divide tasks that have estimates that are longer than a few days.
- Compare the estimates with the time that is available for the iteration. If the estimate total is too long, simplify some of the requirements, or defer them to later iterations.

  For more information, see Plan an iteration.

During an iteration

Task execution

Team members start and complete tasks, recording these events in work items. Completion of a task might include checking in program code and other artifacts. Each task should last no more than a few days; larger tasks are split during iteration planning. For more information, see and Day in the life of a Developer: Suspend work, fix a bug, and conduct a code review.

If a team member runs into any obstacle to their work that cannot be resolved immediately, they should log an issue work item.

Tests

Manual or automatic tests should be developed, and test cases should be linked to the product requirements. A product requirement cannot be considered completed until the work item is linked to test cases that pass and that demonstrate that it's working.

Development work for the tests should be included in the tasks that are linked to the product requirement.

Rolling and nightly builds

The build system builds the product from recently checked-in updates and runs automated tests. You can set principal tests to run on a continuous basis, and you can set a full suite to run every night. This practice helps to ensure that multiple increments don't create an accumulation of bugs. For more information, see Continuous integration & delivery.

Stand-up meeting

The whole team conducts a brief daily review of progress on the tasks of the iteration. Team members can use the taskboard or project the Progress Dashboard on the wall, share it by using Office Live Meeting, or both.

- Each team member briefly reports recent progress, work in hand for the day, and any blocking issues.
- The project manager or team leader reports on progress toward resolving issues. For more information, see Manage issues.
- The bug count is reviewed. Bugs should be given priority over new development. Aim to keep the bug count low throughout the project. If the number of bugs increases, discuss the causes and the possible impact on development work.
- The burndown rate is reviewed.

Scope adjustments

The Burndown Chart might indicate that the tasks won't be completed by the end of the iteration. The project manager or team leader then starts a discussion about how requirements can be simplified so that tasks can be cut. For more information, see Burndown and Burn Rate.

The requirements and corresponding tests are adjusted. A new requirement feature is put on the project plan for the missing functionality. In the project plan review that is held toward the end of the iteration, the feature might be assigned to a future iteration or cut.

Change requests and risks aren't considered during an iteration.

Triage

Some team members (not the whole team) meet regularly to review bugs. Every team member must log a bug when they discover a defect. A logged bug starts in the Proposed state, and the purpose of the triage meeting is to decide whether to fix it, postpone it to a later iteration, or reject it.

For more information, see Track bugs.

At the end of an iteration

Verification

The requirements are considered completed only if the associated tests pass.

Retrospective

Process improvement is an important CMMI goal.

An iteration retrospective reflects on what went well or badly in the iteration and considers improvements to the process and tools that are used by the team. A significant volume of material about retrospectives is available on the Web.

Team members should avoid any assignment of blame. Try to improve the process so that mistakes that are made by individuals are less likely to have an effect.

When you introduce a change in your process, make sure that the team agrees on the following decisions:

- How you'll know whether it was an improvement.
- When you'll make that assessment.
- What you will do as a result.

Integration

If this project is part of a larger program, each team does its work in a branch of the version control system. The Main branch is reserved for integrating the work of the teams. At the end of an iteration, the team might run an integration with the main branch. For more information, see Use branches.

The integration consists of two steps:

- A forward integration, to merge the newer code from the main branch into the local project branch. After doing the merge, automatic and manual tests are run. This merge will create some defects. The defects are fixed at high priority.
- A reverse integration. The local branch code is merged into the main branch, and the build and full test suite on the main branch runs. The changes are reversed if any errors occur. Introducing errors to the main branch is frowned upon. If no errors occur, the integration is declared completed.

  We recommend that you do an integration at the end of each iteration. If you delay it, the list of bugs to be fixed after forward integration is longer. If it takes a long time to fix the bugs, the main branch will have new material, and you'll have to do another forward integration.

Prepare for the next iteration

Toward or at the end of an iteration, several project management activities are done. These include reviewing risks and reviewing the plan as it relates to change requests and changed development estimates.

For more information, see Project activities.

Plan an iteration with the Capability Maturity Model Integration (CMMI) process in Azure Boards

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Developing software in iterations means you divide your work into incremental stages such that you have software with progressively more working features at the end of each iteration. Ideally, you have something to show the customer after the first iteration. Iterations let you receive feedback early so you can make course corrections early.

The matter of planning iterations comes down to deciding how long you want your iterations to be, determining how much work your team can get done in that time, and planning what work should be included in each iteration.

The MSF for CMMI Process Improvement template supplies an Iteration Path field in each work item to help you track your work by iteration. You can customize that path to reflect the iterations that you plan to do. For more information about CMMI, see Background to CMMI.

Create tasks to implement and test each requirement

The iteration plan is represented by the list of tasks that are scheduled for the iteration. Each task is linked to the product requirement that it implements.

The task list is visible in the taskboard, Work Breakdown query, and on the Progress Dashboard.

At the start of the iteration, the team reviews the requirements that are scheduled for this iteration and creates task work items. The task work items describe the work (such as design, development, and testing) that is required to complete the requirement.

During the iteration, team members update the completion status and work remaining fields of the tasks. If the team keeps this information current, the Progress dashboard and other reports indicate how much work remains and the slope of the burndown chart indicates whether the work is likely to be completed on time.

Estimate the appropriate work load for the iteration

It's most likely that, during project planning, it was agreed that product increments should be developed in a series of time-bound iterations. Typically, these iterations vary from one week to four weeks.

The template provides the following reports, which are useful when estimating how much work to plan for an iteration.

- **Status on All Iterations** This report helps you track team performance over successive iterations. Use the report to see how many requirements and how many hours were completed in an iteration.
- **Requirements Overview** This report lists all requirements, filtered by area and iteration and sorted in order of importance. This report can show you how much work was completed by the team in an iteration.
- **Burndown and Burn Rate** Burndown shows the trend of completed and remaining work over a specified time period. The burn rate shows the completed and required rate of work based on the length of the iteration.

Schedule an iteration demonstration and handoff

Plan time to demonstrate the incremental functionality to stakeholders, to gather the team for a retrospective, and to hand off the completed work for validation tests.

Typically, you should give time on the last day of the iteration to demonstrate the working functionality to stakeholders.

Record the feedback, and save it on the project portal. If the demonstration brings new tasks or requirements to light, create work items as necessary. These tasks are planned for future iteration plans.

Launch an iteration

Kick off the iteration with a mini-version of the project launch. Bring the team together. Outline the goals and the scope of the iteration. Discuss and present the plan and any targets. Ensure that all team members have enough context to continue with the work in a self-organizing manner. Make time and space for questions from team members, and record any issues or risks that are brought up during the meeting. Store these items as minutes in the project portal. As a project manager, follow up by creating risk and issue work items, as appropriate.

Track an iteration

Throughout the iteration, you can monitor its progress daily by using the burndown chart shown on the taskboard, or the reports that are provided with the template. You'll want to pay extra attention to the Remaining Work, Unplanned Work, and Requirements Overview to make sure that the iteration is tracking against expectations.

Other resources

For more information, see the following Web resources:

Project Retrospectives: A Handbook for Team Reviews, Norman Kerth; Dorset House, 2001.

Agile Retrospectives: Making Good Teams Great, Esther Derby and Diana Larsen; Pragmatic Bookshelf, 2006.

Manage issue work items in Azure Boards

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

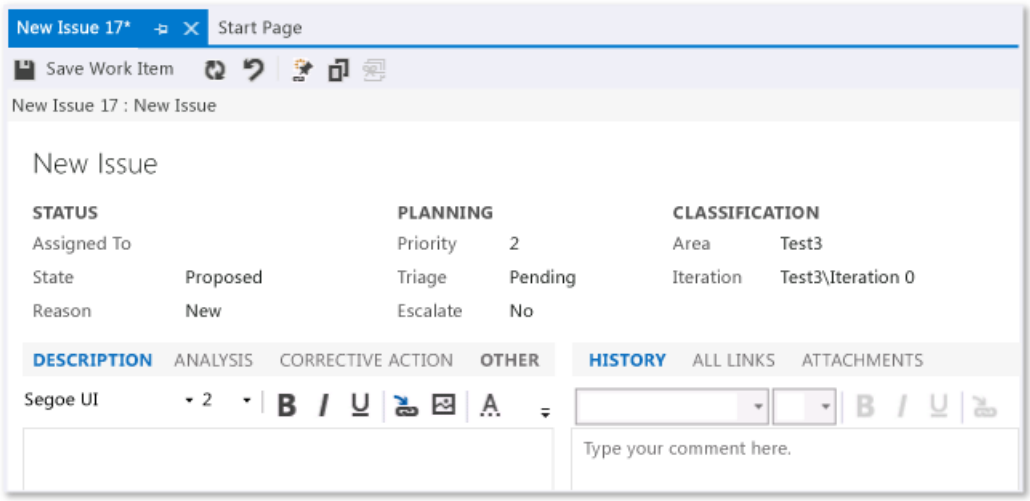
You can use the issue work item to help you track problems with the project plan and its activities and tasks. Issues aren't to be confused with bugs. The bug work item type is provided to track problems with the code and specific failing tests. The issue work item type is provided to help you track all other problems with the project. Some examples are ambiguity in the requirements, unavailability of personnel or other resources, problems with environments, other project risks that are occurring, and, in general, anything that puts successful delivery of the project at risk.

What makes issues different is that they represent unplanned activities. Issue resolution isn't normal project work. So, it must be tracked and given special attention. Use issue work items to track these project problems. Then, use the reporting and queries helps to develop a core capability to manage and resolve issues quickly and effectively.

Create an issue work item

When an issue occurs, create an issue work item, describe the problem, and describe suggested resolutions, if any are known. The issue work items for the project create a significant body of evidence for a Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal. For more information about CMMI, see Background to Background to CMMI.

The work item form for an issue stores data in the fields and tabs that appear in the following illustrations:



Review the issues

The open issues on the project should be reviewed regularly.

To view the issues, run the Open Issues query that is provided with the template. Sort the issues by status using "proposed" to triage any new issues.

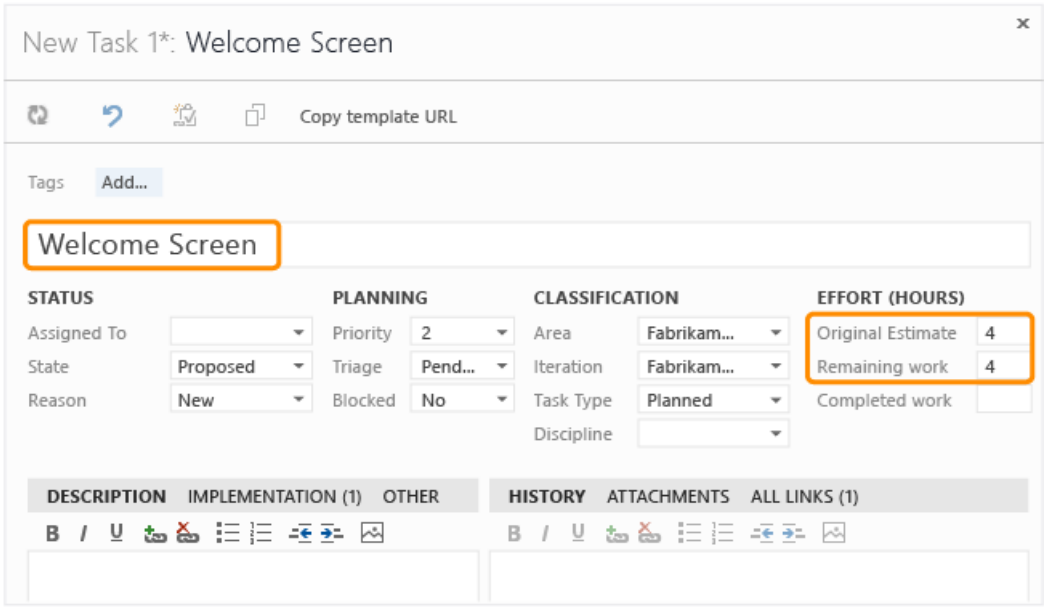
Analyze issues

Each new issue should be analyzed for both its symptoms and root cause. A plan of corrective action should be made to address the symptoms or (preferably) the root cause. Record the plan on the Corrective Action tab of

the issue. The decision to work around the issue or try to fix the root cause should reflect the project risks. These decisions should be documented in the issue work item. They'll provide evidence for a SCAMPI appraisal and demonstrate a level of capability at risk management that is important for level 3 appraisal.

Working around symptoms is a lower-maturity behavior that shows a level of capability that is adequate for a CMMI model level 2 or 3 appraisal. Root cause analysis and resolution imply that the organization intends that the issue shouldn't recur. This expectation is a higher-maturity behavior. It demonstrates a level of capability in issue resolution and process improvement that is typical of an organization that achieves a level 4 or 5 appraisal.

Record the action plan, and then break the work into task work items, linked to the issue work item as children. Tasks should be assigned to individual team members for resolution. Each task should be created together with a task type of "corrective action."



Verify Resolved Issues

During your regular review of open issues, it's a good idea to review issues that have been marked as resolved. If a consensus accepts the documented resolution, mark the issue as "closed," and set its reason to "resolved." Use the open issues query, and filter for state equals "resolved."

Review issues for resolution

After all tasks under an issue have been completed, the stakeholders should decide whether the issue has been resolved.

Open the issue work item and any blocked work items. You can refer to the Corrective Actions tab to view the original plan for action and what action was taken. You can also see the task work items that are associated with the issue by displaying the All Links tab for the Child links. Has the corrective action successfully unblocked the work items and resolved the issue? If not, rework the corrective actions, and reassign them to team members. Was the corrective action completed promptly? Was an unexpected external (or special cause) event and impact on the schedule critical path avoided? Are the project commitments safe, or must they be renegotiated? Record all this detail in the work item. This detail generates valuable evidence for a CMMI appraisal.

If the stakeholders are satisfied that the issue has been resolved successfully, mark the issue as "resolved." It can then be formally closed.

If the issue hasn't been successfully resolved, rework the corrective action tasks, and assign them to suitable personnel for resolution. Reconsider the priority of the issue, and consider raising it to speed up resolution and to avoid delay.

More resources

For more information about SCAMPI appraisals, see the following Web page: Software Engineering Institute.

Related articles

Fields that track bugs, issues, and risks (CMMI)

Engineering processing areas in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Engineering section of the MSF for CMMI Process Improvement guidance covers the value-added activities. These activities are for discovering the information that is required to design and build software products. The Engineering grouping of process areas in the CMMI includes:

- Requirements Development
- Requirements Management
- Technical Solution
- Product Integration
- Verification
- Validation

All of these areas are model level 2 or 3 process areas.

- **Vision and Requirements:** Capture the product vision and requirements
  - Develop requirements
  - Arrange requirements into a product plan
- **Architecture:** Define or capture the architecture of your solution.
  - Create a solution architecture
- **Develop:** Implement and build your solution.
  - Implement development tasks
  - Building a product
- **Verify and Validate:** Verify your solution.
  - Track bugs

For more information about CMMI, see Background to CMMI.

Develop requirements in Azure Boards

10/4/2022 • 16 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Requirements describe what the stakeholders expect from the product. Express your requirements in terms that allow them to be easily discussed with the business stakeholders, using the vocabulary and concepts of the business domain. Requirements shouldn't discuss or depend on the implementation. Requirements include not only the behavioral and quality of service expectations of the users but also statutory constraints and commercial standards.

By creating requirements in TFS, you gain the following benefits:

- Verify that requirements have been satisfied by linking them to test cases.
- Monitor progress toward implementing the requirements by linking them to task work items.
- Structure the requirements into overall and more detailed requirements so that you can manage them more easily and so that progress reports can summarize the information.
- Model the requirements in Visual Studio Ultimate, linking model elements to requirements in Team Foundation Server.

  This article doesn't attempt to replicate the large body of literature that is available on the subject of determining requirements. Instead, it focuses on the aspects that are important for using the Visual Studio tools in a manner that conforms to CMMI. For more information about CMMI, see Background to CMMI.

  The activities that are described in this article, like any development activities, shouldn't be performed in strict order. Develop a domain model while you're writing scenarios because one activity will help you improve the other activity. Develop the scenarios as the time for coding them approaches. Feed the experience with code that has been written and demonstrated back to the scenarios that have yet to be implemented.

Decide when to develop requirements

TFS supports iterative working, and this practice is most effective when the early iterations are used to gain feedback from prospective users and other stakeholders. This feedback can be used to improve the requirements that have been stated for future iterations. This feedback results in a product that is much more effective in its ultimate installation than a product that is developed over the same period without any user trial. If your project is one component among many in a larger program, early integration with other components allows the program architects to improve the overall product.

This flexibility must be balanced against the need to give firm commitments to your customer or to partners in parallel projects.

To a controlled extent, requirements are developed and refined throughout the project. Because the detailed requirements are likely to change during the project, determining them in full before the appropriate implementation is likely to result in wasted effort.

- In Iteration 0, develop a set of requirements that describe of the main features, with enough detail to form a product plan. The product plan assigns requirements to iterations and states what requirement will be fulfilled at the end of each iteration. Create a domain model of the major concepts and activities.

Define the vocabulary that will be used for those concepts both in discussion with the users and in the implementation. Determine broad requirements that pervade every feature, such as security and other quality of service requirements.

- At or near the start of each iteration, develop the requirements for those features in more detail. Determine the steps that the users will follow, defining them with the help of activity or sequence diagrams. Define what happens in exceptional cases.
- Verify as often as possible all the requirements that have been implemented. Pervasive requirements, such as security, must be verified with tests that are extended for each new feature. If possible, automate the tests because automatic tests can be performed continuously.

Managing requirements changes

The following guidelines let you operate an incremental process while monitoring it to satisfy CMMI requirements.

- Don't change the requirements that are set for an iteration. In the rare case of an abrupt change in circumstances, you might have to cancel an iteration. If so, review the product plan and start a new iteration.
- Look for uncertainties in the requirements. Try to arrange the plan so that user experience with early iterations yields information that reduces the uncertainties.
- Use change request work items to record requests to change behavior that has already been implemented, unless the requested improvement is already part of the plan. Link each change request to the appropriate requirement work items.
- Review change requests when you review the product before each iteration. Examine the impact the request has on dependent projects and users, and estimate the cost related to changes in your code. If a change request is accepted, update the requirement.
- Update the tests to conform to every requirements change.
- Choose a cut-off date (for example, after iteration 2 or 3) after which requirements changes must be much more strongly justified. If your project is for a paying customer, this date is the date to have the customer approve a baseline set of requirements and switch from hourly payment to fixed price.
- Use bug work items to record implemented behavior that doesn't behave according to the explicit or implicit requirements. Where practical, create a new test that would have caught the bug.

Write a vision statement

Discuss a vision statement with the team, and display it on the project's Web portal for Team Foundation Server.

A vision statement is a short summary of what benefit the product will bring. What can the users that they couldn't do before? The vision statement helps clarify the scope of the product.

If the product already exists, write a vision statement for this version. What can the product's users do that they couldn't do before?

Write scenarios

Work with your customer and other stakeholders to create scenarios, and enter them as requirement work items, with the Requirement Type field set to Scenario.

A scenario or use case is a narrative that describes a sequence of events, shows how a particular goal is achieved, and usually involves interaction between people or organizations and computers.

Give it a descriptive title that clearly distinguishes it from others when viewed in a list. Make sure that the principal actor or actors are stated and that their goal is clear. This title is a good example:

Customer buys a meal.

You can write a scenario in the following forms. Sometimes it can help to use more than one form:

- One or two sentences in the work item description:

  A customer orders a meal on a Web site and pays with a credit card. The order is passed to a restaurant, which prepares and delivers the meal.

- Numbered steps in the work item description:

  1. A customer visits the Web site and creates an order for a meal.
  2. The Web site redirects the customer to a payment site to make payment.
  3. The order is added to the restaurant's work list.
  4. The restaurant prepares and delivers the meal.

- Storyboard. A storyboard is essentially a cartoon strip that tells the story. You can draw it in PowerPoint. Attach the storyboard file to the requirement work item, or upload the file to the team portal, and add a hyperlink to the work item.

  A storyboard is especially useful for showing user interactions. But for a business scenario, it's recommended to use a sketch style that makes it clear that the storyboard isn't the final design for the user interface.

- Requirement documents. Requirement documents give you the freedom to provide the appropriate level of detail for each requirement. If you decide to use documents, create a Word document for each requirement, and attach the document to the requirement work item, or upload the file to the team portal and add a hyperlink to the work item.

- Unified Markup Language (UML) sequence diagram. A sequence diagram is especially useful where several parties interact. For example, ordering the meal requires the customer, the DinnerNow Web site, the payment system, and the restaurant to interact in a specific sequence. Draw the sequence diagram in a UML model, check it into Team Foundation Server, and enter a link in the requirement work item. For more information, see UML Sequence Diagrams: Guidelines.

Write specific scenarios

Start by writing specific scenarios, which follow a particular set of actors through a specific sequence. For example, "Carlos orders a pizza and garlic bread at the DinnerNow Web site. The Web site redirects Carlos to Woodgrove Bank's payment service. Fourth Coffee prepares the pizza and delivers it."

Specific scenarios help you envisage the system in use and are most useful when you first explore a feature.

It can also be useful to create named personas that describe the backgrounds and other activities of people and organizations. Carlos sleeps rough and uses an Internet café; Wendy lives in a gated community; Sanjay orders meals for his wife at work; Contoso runs a chain of 2,000 restaurants worldwide; Fourth Coffee is run by a couple who deliver by bicycle.

More generic scenarios that are written for "a customer," "a menu item," and so on, can be more convenient but are less likely to lead to the discovery of useful features.

Levels of detail

In Iteration 0, write a few important scenarios in some detail, but write most scenarios in outline. When an iteration approaches in which a particular scenario is to be fully or partly implemented, add more detail.

When you first consider a scenario, it can be useful to describe the business context, even aspects in which the product takes no part. For example, describe the DinnerNow method of delivery: Does each restaurant organize

its own deliveries, or does DinnerNow run a delivery service? The answers to such questions provide useful context for the development team.

The more detailed scenarios that you develop at the start of an iteration can describe user interface interactions, and storyboards can show user interface layout.

Organize the scenarios

You can organize scenarios by using the following methods:

- Draw use case diagrams that show each scenario as a use case. This method is recommended because it makes the scenarios easy to present and discuss. For more information, see UML Use Case Diagrams: Guidelines.
    - Link each use case to the work item that defines the scenario. For more information, see Link model elements and work items.
    - Draw Extends relationships to show that one scenario is a variation of another. For example, "Customer specifies separate payment and delivery addresses" is an extension of the basic "Customer makes an order" use case. Extensions are useful to separate out scenarios that will be implemented in a later iteration.
    - Draw Includes relationships to separate a procedure such as "Customer logs on," which is common to several use cases.
    - Draw generalization relationships between general scenarios such as "Customer pays" and specific variants such as "Customer pays by card."
- Create parent-child links between scenario work items. You can view the hierarchy in Team Explorer. For more information, see Arrange requirements into a product plan.

Model the business domain

Create a UML model that describes the principal activities and concepts that are involved in the use of your product. Use the terms that are defined in this model as a "ubiquitous language," in the scenarios, in discussions with the stakeholders, in the user interface and any user manuals, and in the code itself.

Many requirements are not explicitly stated by your customer, and comprehending the implied requirements depends on an understanding of the business domain, that is, the context in which the product will work. Some of the work of requirements gathering in an unfamiliar domain is about gaining knowledge of that context. After this kind of knowledge has been established, it can be used on more than one project.

Save the model in version control.

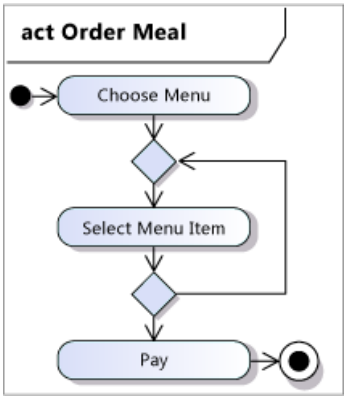
For more information, see Model user requirements.

Model behaviors

Draw activity diagrams of the scenarios. Use swimlanes to group the actions that are performed by different actors. For more information, see UML Activity Diagrams: Guidelines.

Although a scenario usually describes a specific sequence of events, an activity diagram shows all the possibilities. Drawing an activity diagram can prompt you to think about alternative sequences and to ask your business clients what should happen in those cases.

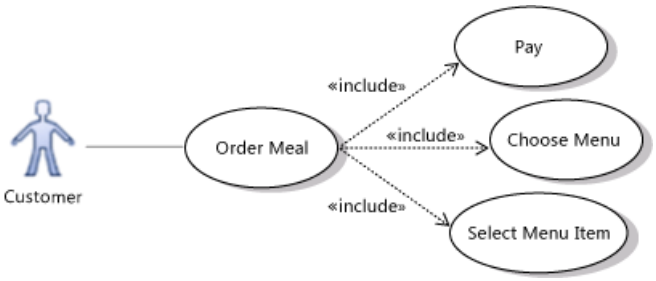
The following illustration shows a simple example of an activity diagram.



Where the interchange of messages is important, it might be more effective to use a sequence diagram that includes a lifeline for each actor and major product component.

Use case diagrams let you summarize the different flows of activity that your product supports. Each node on the diagram represents a series of interactions between the users and the application in pursuit of a particular user goal. You can also factor common sequences and optional extensions into separate use case nodes. For more information, see UML Use Case Diagrams: Guidelines.

The following illustration shows a simple example of a use case diagram.



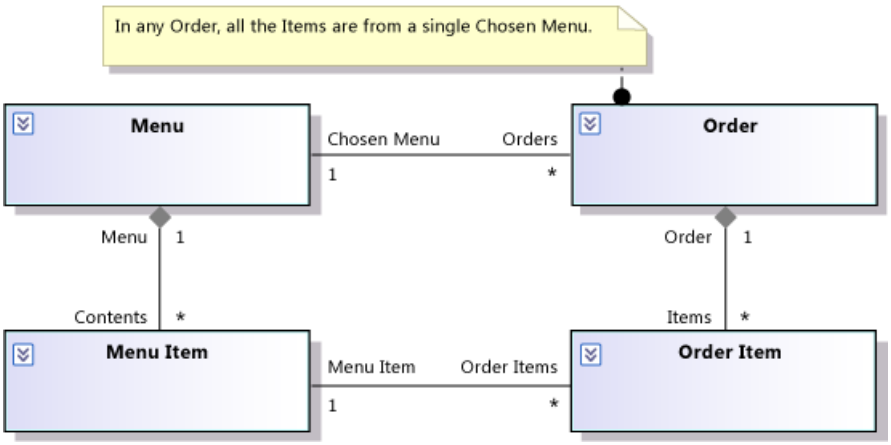
Model concepts

Draw domain class diagrams to describe the important entities and their relationships that are mentioned in the scenarios. The DinnerNow model, for example, shows Restaurant, Menu, Order, Menu Item, and so on. For more information, see UML Class Diagrams: Guidelines.

Label the roles (ends) of the relationships with names and cardinalities.

In a domain class diagram, you don't typically attach operations to the classes. In the domain model, the activity diagrams describe the behavior. Assigning responsibilities to program classes is part of the development work.

The following illustration shows a simple example of a class diagram.



Static constraints

Add to the class diagrams constraints that govern the attributes and relationships. For example, the items on an order must all come from the same restaurant. These types of rules are important for the design of the product.

Model consistency

Ensure that the model and scenarios are consistent. One of the most powerful uses for a model is to resolve ambiguities.

- The scenario descriptions use the terms that are defined in the model and are consistent with the relations that it defines. If the model defines menu items, the scenarios shouldn't refer to the same thing as products. If the class diagram shows that a menu item belongs to exactly one menu, the scenarios shouldn't talk of sharing an item between menus.
- Every scenario describes a series of steps that are allowed by the activity diagrams.
- Scenarios or activities describe how each class and relationship in the class diagram is created and destroyed. For example, what scenario creates a menu item? When is an order destroyed?

Develop Quality of Service Requirements

Create work items that specify quality of service requirements. Set the Requirement Type field to Quality of Service.

Quality of service or non-functional requirements include performance, usability, reliability, availability, data integrity, security, afford-ability, service-ability and upgrade-ability, deliver-ability, maintainability, design, and fit and finish.

Consider each of these categories for each scenario.

The title of each quality of service requirement should capture its definition by presenting a context, an action, and a measurement. For example, you might create the following requirement: "During a catalog search, return the search results in less than three seconds."

It's useful to capture more detail that explains why the requirement is necessary. Describe why the persona would value the requirement and why this level of service is required. Provide context and justification. This explanation may include useful risk management information such as data from a market survey, or a customer focus group. Or, it may include a usability study, help desk reports/tickets, or other anecdotal evidence.

Link the quality of service requirement to any scenario (requirement work item) that is affected by the quality of service. Linking related work items allows users of Team Foundation Server to keep track of dependent requirements. Queries and reports can show how quality of service requirements affect the functional requirements that are captured as scenarios.

Review requirements

When the requirements have been written or updated, they should be reviewed by the appropriate stakeholders to ensure that they adequately describe all user interactions with the product. Common stakeholders might include a subject matter expert, a business analyst, and a user experience architect. The scenarios are also reviewed to ensure that they can be implemented in the project without any confusion or problems. If any problems are spotted, the scenarios must be fixed so that they're valid at the conclusion of this activity.

Create a review work item to track the review. This item provides important evidence for a Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal and may provide a good source of information for root cause analysis in the future.

Review each scenario for the following characteristics:

- The scenario is written in the context of what task users must complete, what they already know, and how they expect to interact with the product.
- The scenario outlines a problem and is not obscured by proposed solutions to the problem.

- All relevant user interactions with the product are identified.
- The subject matter expert, the business analyst, and the user experience architect review each scenario in the context of the project to validate that all scenarios can be implemented successfully. If a scenario isn't valid, it's revised so that it's valid.
- The scenario can be implemented with the available techniques, tools, and resources and within budget and schedule.
- The scenario has a single interpretation that is easily understood.
- The scenario doesn't conflict with another scenario.
- The scenario is testable.

Validation

Plan to deploy beta versions of the product into its working environment before its final release. Plan to update the requirements, based on stakeholder feedback from that deployment.

Validation means ensuring that the product fulfills its intended use in its operating environment. In MSF for CMMI, validation is achieved by demonstrating working software to the stakeholders at the end of every iteration throughout the project. The schedule is arranged in such a way that concerns that are fed back to the developers from early demonstrations can be dealt with in the plan for the remaining iterations.

To achieve true validation, the product must not only be run in a demonstration or simulated context. As far is as usability, it should be tested in real conditions.

Inspect and edit the requirements

The scenario and quality of service requirements, which lead to most development tasks, can be inspected by using the customer requirements query. If you prefer to display all requirements, you can write a query that returns all work items of the requirement work item type. Set the columns of the result to show the iteration path.

Your team should create most requirements in the early iterations of the project. New requirements will be added and others adjusted as feedback is gained from early versions.

More resources

For more information, see the following Web resources:

- Modern Requirements Suite 4TFS, a requirement management software platform that bi-directionally connects Microsoft Office technologies with Azure Boards and TFS to increase stakeholder engagement and provide end-to-end traceability.
- A Practical Guide to Feature Driven Development, Stephen R. Palmer and Malcolm J. Felsing; Prentice-Hall PTR, 2002.
- Streamlined Object Modeling: Patterns, Rules and Implementation, Jill Nicola, Mark Mayfield, and Mike Abney; Prentice Hall PTR, 2001.
- Agile Modeling: Effective Practices for Extreme Programming and the Unified Process, Scott Ambler; Wiley, 2002.
- Domain Driven Design: Tackling Complexity in the Heart of Software, Eric Evans; Addison Wesley Professional, 2003.
- Object Design: Roles, Responsibilities and Collaborations, Rebecca Wirfs-Brock and Alan McKean; Addison

Wesley Professional, 2002.

Arrange requirements into a product plan in Azure Boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

After you analyze your customer requirements sufficiently to understand what the product should do, you must work out a plan to implement the product. Or, for an existing product, you must work out what functionality is missing and work out a plan for making the changes. But the requirements don't automatically tell you the plan.

This article outlines a method of obtaining a plan, starting from a set of requirements. This method is one among a variety that will work on Visual Studio, and you should adapt it so that it suits your needs.

You can use the backlog and portfolio backlogs to define and map requirements and features.

Review requirements and features

There are two kinds of requirement in this method: customer requirements and features. Customer requirements are what you get by analyzing what the customer wants from the product. Features are items in the product plan, which correspond to small subsets of the customer requirements. Each feature may include pieces of the customer requirements that come from different parts of the user experience and different functional areas.

Customer requirements

- Customer requirements are determined by discussion with the prospective users and other stakeholders.
- To help analyze these requirements, you'll typically create storyboards and models, and you decompose the scenarios into smaller steps, forming a tree. You can link modeling elements such as use cases and activities to scenario work items.
- There are two kinds of customer requirement:
  - Scenarios, also known as use cases, represent sequences of interactions between the users and product, in pursuit of specific goals. An example scenario might have the title "User buys a book."
  - Quality of Service requirements include performance, security, usability, and other criteria.
- You can represent these requirements as work items of type requirement, with the Requirement Type field set to Scenario or Quality of Service. For more information, see Develop requirements.
- These requirement work items should be linked to system tests so that you can ensure that all the requirements are tested. See Create a test plan.
- View the backlog or open the Customer Requirement query to list these requirement work items.
- Use the Requirements Progress report to monitor which requirements have been satisfied.

Features

- A feature is an item in a product plan that represents a group of tasks. In product planning, representatives of the development team and stakeholders assign features to iterations. For more information, see Plan a project.
- Enter features as requirement work items with the Requirements Type field set to Feature.

- The feature's title states, in users' terms, what the users can do with the product, that they couldn't do in previous iterations. There are no items, or few items, on the plan that don't deliver new user value.

  For example, this sequence of features could form an implementation plan:

  - "A buyer can pick a book from a list and add it to a wish list."
  - "The book list displays prices. In the wish list, the total price is displayed."
  - "Vendors can attach tags to books. Buyers can filter the book list by tag."

    Notice that no feature touches just one step in the user experience, and no feature involves just one part of the product architecture. Instead, as the features are implemented, several functions are revisited and augmented with new user value.

- A feature is assigned to an iteration during product planning. All the tasks under a feature must be assigned to the same iteration.
- A feature describes a partial realization of the customer requirements. It's a subset of the customer requirements, and it might implement each customer requirement to a limited extent.
- Every feature can be linked to one or more test cases that test the part of the requirements that the feature represents. These test cases are a subset of the system tests that are linked to the customer requirements.
- The feature's state must not be marked complete until its tests are fully defined and pass.
- Every feature is a group of development and test tasks. It's the root of a tree of tasks. The development tasks implement the partial requirements that the feature describes. The test tasks design and execute the appropriate test cases.
- You use the Product Requirements query to list features.

Finding features

Separating requirements into incremental features is a creative task that must involve developers, analysts, and stakeholders. A feature defines a piece of the product's functionality that can sensibly be implemented separately from the surrounding functions. As such, a workable set of feature definitions and an ordering into a plan depends partly on the architecture of the system.

For this reason, planning and the initial design of the product must work in parallel, particularly in Iteration 0 where the bulk of the plan is being sketched.

Decompose scenarios into smaller steps

To help you arrange the requirements into features, it helps to decompose the scenarios into smaller steps.

Storyboards often help with this activity. A storyboard is a sequence of pictures that illustrate the scenario. UML activity diagrams are useful for showing alternative paths, and UML sequence diagrams can help you discuss interactions between several actors. After you use these tools to analyze a scenario, you can enter the decomposed scenarios into Team Explorer. This action lets you link test cases to the scenarios and ensure that the requirements have been satisfied. For more information, see UML Activity Diagrams: Guidelines and UML Sequence Diagrams: Guidelines.

Features - requirements fulfilled in each iteration

A feature is a requirement that summarizes what the users can do at the completion of each iteration. You can create more than one feature for each iteration. Enter them as requirement work items, setting the Requirement Type to Feature.

Use your assignments of scenarios to work items to help you define the features. The following example feature plan is derived from the assignments of scenarios to iterations in the previous section:

- Iteration 1
  - Customer chooses items from a menu, adds them to an order, and adds a delivery address.
- Iteration 2
  - Customers start by displaying a list of restaurants and then choose one.
  - When the customer completes an order, the order appears on the chosen restaurant's screen.
  - The prices of items and the total price are displayed on the order.
- Iteration 3
  - Restaurant marks the order as "Done" when the prepared meal has been dispatched. The meal is logged against the restaurant.
  - Each restaurant can enter and update its menu.
  - Customer can browse the menu of every restaurant before selecting one.
- Iteration 4
  - Customer enters payment details on completing an order. Customer's card is charged when the restaurant marks the order as Done.
  - Restaurant is paid for orders that are marked as Done.
- Iteration 5
  - Restaurants can set their delivery area. Customer enters postal code at start of session. The Web site displays only restaurants that can deliver to the local area.

Partly implemented scenarios

Decomposing the scenarios into small steps helps you to separate some steps that can be implemented earlier from others that can be implemented later.

But sometimes you can separate out other aspects of the scenarios. In this example, the team might implement a basic version of the user experience in early iterations and then improve it later. So you might add the following feature:

- Iteration 6 - Restaurant can choose the color scheme and font of its menu and upload its own logo and pictures of meals.

  This type of feature doesn't emerge directly from the decomposition into steps, but it usually emerges in discussion of storyboards. User experience features are good candidates for later iterations.

Entering and inspecting features

Create work items with work item type of requirement, and set the Requirement Type field to Feature. Set the feature title to the short description.

Tracing features to requirements

You can link features to requirements in the following ways:

- Link feature work items to the leaf scenario requirements of their iterations. Link them by using Related Item links because the leaf scenarios already have parents.
- Link test case work items to the scenarios and quality of service requirements that they test. Link features to the subset of test cases that should pass when the feature has been developed. In this manner, the test

cases act as the link between features and customer requirements.

Create a work item for each quality of service feature

Quality of service requirements are pervasive in software design. For example, security requirements are not related to a particular development task.

Nevertheless, for each quality of service requirement, you should create a feature work item whose children are mainly testing tasks that ensure that a quality of service criterion is met. These work items are called quality of service features.

Some quality of service features can have development tasks. In an early iteration, you might implement a version of the system that can handle only a few users, as a proof of concept. For a later iteration, you might add a feature that specifies the target capacity as stated in the customer requirements.

Review the product plan

Before the start of every iteration, hold a meeting to review the product plan. The first product planning meeting creates the plan, and later meetings review it based on earlier iterations. For more information, see Plan a project.

In a product plan review, discuss the features with business stakeholders, and be prepared to reprioritize them and arrange them into different iterations. The meeting should include business stakeholders and representatives of the development team.

The meeting discusses the sequence in which features will be developed. This can be done by projecting or screen-sharing the Office Excel view of the Product Requirements query and ordering the features by iteration.

An alternative technique is to place the features in a specific sequence and then consider how much can be done in each iteration. For example, the developers might discuss whether "Customer can display the prices" should be moved from Iteration 2 to Iteration 3, without moving it in the sequence. To place the items in a sequence, add an extra column that is named Rank to the spreadsheet, and insert integers that denote the sequence. Order the spreadsheet by this column. The ranks won't be stored in Team Foundation Server, but you can save the spreadsheet. When you open the spreadsheet again, select any cell in the work item table, and then select Refresh on the Team tab.

Product planning considers the priorities of the features and the development costs. Priorities come from the business stakeholders, with some guidance about risk from the developers. Cost estimates come from the developers. To get an accurate idea of the costs, the development team must have already done some work on the architecture of the product and might need some experience from the early iterations. For this reason, the cost estimates should be refined at every product plan review.

Plan the iteration

After the product plan review, plan the iteration. The product plan determines the features that will be delivered by the end of the iteration. The iteration plan determines what work the team will do to implement and test the features.

The following activities are part of iteration planning:

- Create tasks for development and testing, and link them as children to the feature requirements.
- Create test cases for the aspects of the customer requirements that are to be developed in each feature. The test cases should be linked to the customer requirements so that you can monitor how complete the requirements are.

  You can also link test cases to the features so that you can track the correspondence between features

and requirements. Don't mark a feature complete until the linked test cases pass.

For more information, see Plan an iteration.

Investigate architectural strategies and create a solution architecture

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Part of creating a good architecture is investigating alternative architectural strategies. Alternative strategies have different benefits that are based on platform selection, technologies that are used, and code reuse. Each strategy is designed and proofs of concept are built to further investigate the costs and benefits of each strategy. The strategies are assessed against product and quality requirements, and ultimately a strategy is chosen to implement the product. Finally, security and performance are architectural concerns for which work must be done over the entire product.

Create alternative architecture partitioning designs

The problem is analyzed and different approaches are considered. A group of requirements are selected that represent key business and technological challenges. Examine the characteristics of these challenges, such as integration of legacy systems, and predict future needs based on current needs, reusability of code, and maintenance costs.

Create an application diagram

Using the domain model and requirements as input, create an application diagram that represents the core logical elements of the system. This diagram will later be partitioned into system diagrams. Alternative partitioning schemes will be considered and evaluated.

One way to represent an application diagram is as a Unified Modeling Language (UML) use case diagram. This type of diagram can show the major subsystems and their dependencies. Also, you can place use cases in each subsystem to show which subsystem manages each user scenario.

Establish evaluation criteria

Determine which criteria to use to identify requirements and scenarios that represent significant architectural challenges. Consult the existing enterprise architecture documents for criteria. Review any business requirements, technical requirements, and enterprise standards that must be applied to new applications. Capture other criteria that are known to be architecturally significant, such as integration with legacy systems, reusability of code, reusing existing vendor libraries and platforms, and controlling maintenance costs. Capture other criteria that represent risks and cost when implementing a technical solution.

Select a candidate group of requirements

Evaluate each quality of service requirement and product requirement against the evaluation criteria. If a requirement represents an architectural challenge, consider it a candidate for modeling. For example, a requirement that the new product must support older customer databases meets the criteria of integrating with legacy systems. Such a requirement is a candidate for modeling how the integration would work.

Select a candidate group of scenarios

Evaluate each scenario against the evaluation criteria. If a scenario represents an architectural challenge, consider it a candidate for modeling. For example, a scenario in which the user downloads a client update meets the criteria that concerns maintenance costs. Such a scenario is a candidate for modeling how best to handle client updates.

Reduce the candidate group

Review the candidate scenarios and requirements. Remove scenarios and requirements that duplicate the evaluation criteria or are better represented by other scenarios and requirements. Trim the candidate group to a core group that represents the key architectural challenges, risks, and costs of the new application. Keep the scenarios and requirements that best represent the evaluation criteria, that present the most risk, and that present the most potential cost when architecting a technical solution. Keep the scenarios and the requirements that are the most comprehensive or key parts of the application.

Create partitioning criteria

Using the requirements as motivation, analyze established architectural patterns (such as façade or model-view-controller), and identify potential candidates for implementation. Identify candidate patterns through their motivation, and consider their design tradeoffs related to coupling, cohesion, extensibility, adaptability, and flexibility. Select a set of candidates for implementation as alternatives for assessment.

Design system architecture and deployment

The system architecture defines the groupings and configurations of elements that are identified in the application diagram. System diagrams are created that capture the system architecture for each possible architecture approach. Deployment diagrams show the deployment steps that are based on dependencies and core functionality. An infrastructure architect creates a logical datacenter diagram that describes the logical structure of the datacenter where the application will be deployed. The deployment diagrams are validated against the logical datacenter diagram to ensure that the systems can be deployed.

Create a system model

The architect and the lead developer create system diagrams from the application diagram. Through system diagrams, you can design reusable application systems as units of deployment by composing them from elements on the application diagram. You can also design larger and more complex systems that contain other systems so that you can use them in distributed system scenarios and abstract the details of applications in those systems. Check in each new diagram file to version control.

You can represent system diagrams in Visual Studio in the following ways:

- Use case diagrams. The main user scenarios are represented as use cases, and the major components of the system are shown as subsystems. Each use case can be placed inside the subsystem that deals with it. For more information, see UML Use Case Diagrams: Guidelines.
- UML component diagrams. These diagrams let you show communications channels between the components, in addition to dependencies. You might also want to create class diagrams to describe the types that are visible at the interfaces to the components, and you can create sequence diagrams to show their interactions. For more information, see UML Component Diagrams: Guidelines, UML Class Diagrams: Guidelines, and UML Sequence Diagrams: Guidelines.
- Layer diagrams. A layer diagram describes the block structure of the application. It shows only components and the dependencies between them. It has the benefit that, after the code is written, you can validate the code and the dependencies against the diagram. For more information, see Layer Diagrams: Guidelines.

  For each subsystem, you can create a package that describes its types and behavior in more detail. For more information, see Define packages and namespaces.

Create proofs of concept

Significant risks to the project can be mitigated by creating an architectural proof of concept. It's important to address risk as early as possible in the project so that key strategic and architectural decisions can be made while it's still easy to modify fundamental pieces of the architecture. Creating early proofs of concept reduces

overall project risk and unknowns. Lower project risk and fewer unknowns make planning and estimating in later iterations more accurate. Proofs of concept can be temporary and discarded after the issues have been addressed, or they can be built as the foundation of the core architecture.

Examine risk

Understand the elements that lead to the identification of the risk or architectural decisions. Examine related scenarios and quality of service requirements. Check for any target environment implications.

Plan the approach

Determine the form of the proof of concept that is needed. Use the application and system diagrams to help plan. Solve only the architectural problem that is identified by the risk. Look for the simplest resolution.

Build and run the proof of concepts

Build the proof of concept. You can implement the proof of concept from the application diagram. Maintain focus on the problem to be solved. Deploy the proof of concept to a physical environment that is congruent to the logical datacenter diagram. The physical environment should match the settings of the logical datacenter diagram as closely as possible. Test the proof of concept against the high-risk issues.

Assess alternatives

The Lightweight Architecture Alternative Analysis Method (LAAAM) is used to help decide between different architectural strategies for building an application. The LAAAM typically takes one day to complete. Start by building a utility tree that describes key quality and functional drivers of the application that are based on requirements. Each driver is written as a scenario that takes the form of a statement that is written as context, stimulus, and response. Use an assessment matrix to evaluate how well each strategy addresses each scenario.

Create a utility tree

Examine quality of service requirements and product requirements to determine the key drivers of quality and function in the application. Construct a utility tree that represents the overall quality of the application. The root node in the tree is labeled Utility. Subsequent nodes are typically labeled in standard quality terms such as modifiability, availability, and security. The tree should represent the hierarchical nature of the qualities and provide a basis for prioritization. Each level in the tree is a further refinement of the qualities. Ultimately, these qualities become scenarios.

Construct an assessment matrix

For each leaf in the utility tree, write a scenario. The scenario is in the form of context, stimulus, and response (for example, "Under typical operation, perform a database transaction in fewer than 100 milliseconds").

Create a spreadsheet or a table, and enter each scenario as a row in this assessment matrix. Enter each architectural strategy as a column. At each intersection of strategies and scenarios, enter a rating on a scale between 1 and 4.

The rating should take into account the following factors:

- **Development cost** Is this solution easy or difficult to implement? What is its impact on other areas?
- **Operational cost** At run time, will this solution work easily, or will it adversely affect usability, performance, and so on?
- **Risk** Is this solution certain to address the scenario well, or are there unknown costs? Could this solution have an adverse impact on the team's ability to accommodate future enhancements in the requirements?

  If a proof of concept has been built for a strategy, use information from that proof of concept to help determine the values.

  At the bottom of the table, sum the values from the scenarios. Use these figures as an input to the

discussion that leads to decisions on the alternative architectures.

Upload the completed assessment matrix to the project portal.

Select the architecture

After the assessment matrix is created, a review meeting is held to determine which architecture to use in the next iteration. The assessment matrix and information that is discovered from creating the proofs of concept is used to help make a decision. After the architecture is selected, diagrams for the architecture are checked in as the reference solution, and a justification document is created that captures the reasons behind the selection.

Prepare for review

The architect and the lead developer identify the appropriate reviewers for reviewing the proposed architectures and circulate documentation for the architectures to each participant.

Review system architecture and deployment architecture

During the review meeting, the system diagrams, the deployment report, and the logical datacenter diagram are reviewed. The goal is to choose an architecture to implement in the next iteration.

Consider the assessment matrix rankings for each architecture to help evaluate the suitability of each architecture. Consider any information that is discovered from the proofs of concept such as cost or complexity that is involved with implementing the different architectures. If the logical datacenter diagram represents an existing datacenter that cannot be modified, don't review it. If a datacenter is being created, review the diagram for deployment considerations. Select the architecture to be used. Review the architectural concept against the scenarios to validate that the solution meets the customer needs and is complete.

Create a reference solution

Create a justification document that captures the decisions of the meeting. Upload it to the project portal. For the selected architecture, check in any application, system, or logical datacenter diagrams as the reference solution to use for implementing features in the next iteration. Communicate to the entire team and any dependent teams the decision on what architecture is selected for the next iteration.

Develop a performance model

Performance modeling is used to identify and address potential performance issues in the application. A performance model is developed from a quality of service requirement, which is then broken into development tasks. Each development task is assigned a performance budget for implementation.

Identify the scenarios that are linked to the performance quality of service requirement. Map the development tasks to the scenarios. From the quality of service requirements list, determine the workload for the application. Using the workload estimates and the quality of service requirements list, identify the performance goals for each key scenario. These include goals such as response time, throughput, and resource use. Identify the performance-related resources that have been budgeted to meet the performance goals. Some examples of performance-related resources are execution time and network bandwidth. Determine the maximum allowable allocation of each resource.

Spread the budgeted resources across the processing steps for each scenario. When not sure of how to assign the budget, make best guesses, or divide the resources evenly among the steps. Budgeting is refined during validation. Attach or write the allocation on the appropriate development task.

Find budget allocations that pose a risk to meeting performance goals. Consider tradeoffs that help meet performance goals such as design and deployment alternatives. Reevaluate quality of service requirements if necessary.

Identify the scenarios that don't meet budget allocations. Measure the performance of the scenarios. Use prototyping in early iterations if code isn't available. Repeat the budgeting, evaluation, and validation steps as

necessary by using data that is acquired during validation.

Develop a threat model

For more information, see the following page on the Microsoft Web site: Security Developer Center.

Implement development tasks in Azure Boards

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

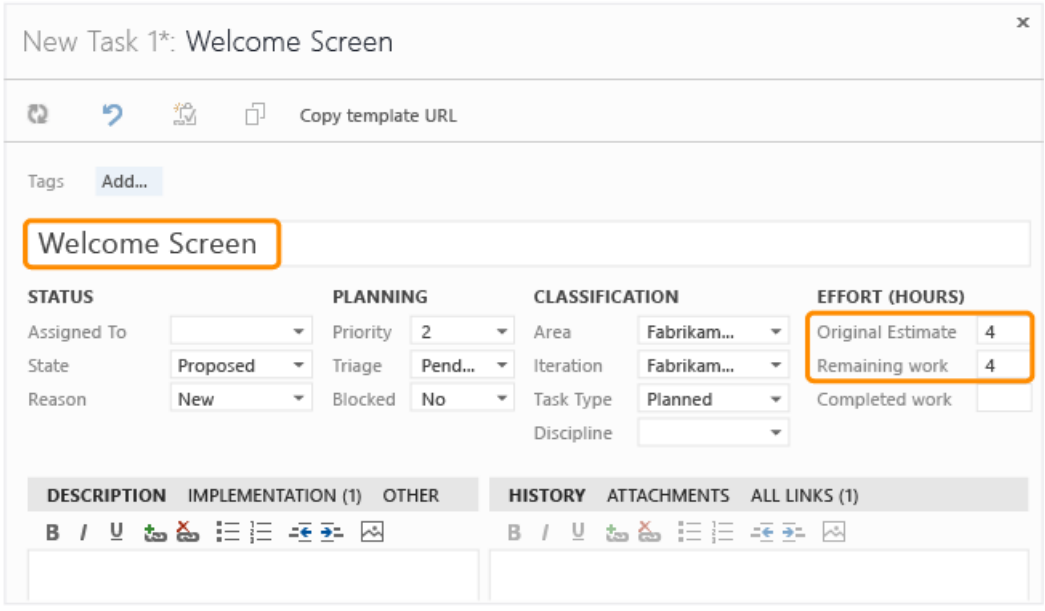
A development task is a small piece of development work that stems from a requirement. Implementing a development task involves adding the appropriate new functionality to your software. After you complete a development task, it should be unit tested, reviewed, code analyzed, and integrated into the existing code base.

Estimate the cost of development tasks

Estimating the cost of development tasks helps control the scope of features and schedule development work. Cost estimates for all development tasks should be completed and any issues should be resolved before the iteration planning meeting. If the total cost of the development tasks is more than can be done in an iteration, a task must be deferred or reassigned. After a development task is chosen, it's the responsibility of the developer to cost the task.

Create a task work item for each development task that is chosen, and link it to the requirement from which it was created. This action is accomplished from the Implementation tab on either the task or the requirement work item. Base your estimates on the time that was required to complete similar tasks, and be sure to factor in the cost of writing unit tests. For each task, enter the estimate into the Original Estimate field of the task work item.

The form for task work items stores data in the fields and tabs that the following illustrations show:



After tasks have been created and estimated, use the Work Breakdown query to view the breakdown of all your requirements and tasks. For more information, see CMMi process, List work items.

Create design documents

Your design documents should include enough information to describe to a developer how to write code to implement the requirement in the product.

Design documents can be a collection of specifications, requirement work items, and other documents,

depending on your team process.

Consider using design patterns, object-oriented design, structural models, modeling languages, entity relationship models, and other techniques in the guidelines for the design that is determined for your team. It's also a good idea to document the rationale for key decisions that were made. For example, if there's a significant effect on cost, schedule, or technical performance, document the reason for the decisions behind these effects, and include that information in your design.

After you create the necessary design documents, store them where your team members can share them. For more information, see Manage documents and document libraries.

Conduct a design review

A design review is used to ensure that the new or revised design is technically correct, complete, testable, and of high quality and that it implements the requirement correctly. Design reviews are a key method of guaranteeing quality early by identifying problems before they appear in the code. Design reviews also provide more insight about the design from other developers.

The developer who is responsible for creating the design should organize the design review by identifying reviewers, scheduling the review, and distributing the design to all the reviewers.

Any stakeholders who are involved or affected by the design should participate in the review. Typically this review might include the project manager, the lead developer, and the tester for the design area. All developers who are on the same team as the developer whose code is being reviewed should also participate in the review.

Schedule the review meeting, and distribute the design documents early enough to give each reviewer sufficient time to read them. Plan the length of the review meeting to correspond to how many technical details must be reviewed.

Verify design quality

Ensure that the design is testable. Does it build code that cannot be verified or validated in a reasonable manner? If so, you can't ensure the quality of the code, and the design must be reworked. Examine the design documents for problems that will lead to code errors. Look for incorrect interface descriptions, design mistakes, or naming confusion. Compare the design documents against existing criteria, such as operator interface standards, safety standards, production constraints, design tolerances, or parts standards. Create bug work items that describe any flaws that are found in the design documents, and assign them to the responsible developer.

Create a Review work item for the design

A review work item is created to document the results of the design review. The review team must decide the next steps for the design, which depend on the number of the changes necessary. If no changes are necessary, set the State of the work item to Closed, set the Reason to Accepted (as is), and note that coding can start on the design. If minor changes are necessary, set the State of the work item to Resolved, and set the Reason to Accepted with Minor Changes. This setting indicates that coding can start after the minor changes have been implemented in the design. If major changes are necessary, set the State of the work item to Resolved, and set the Reason to Accepted with Major Changes. The design must be reworked and another design review must be performed before coding can start on the design.

Unit Tests

Unit tests verify the correct implementation of a unit of code. Writing and completing unit tests identifies bugs before testing starts and helps reduce the cost of quality control. Developers must write unit tests for all code that will be written as part of implementing a development task or fixing a bug. For more information, see Unit test your code.

Code Analysis

Code analysis checks code against a set of rules that help enforce development guidelines. The goal of code analysis is to have no code analysis violations or warnings. Code analysis can inspect your code for more than 200 potential issues in naming conventions, library design, localization, security, and performance.

If you start to run code analysis early in your development cycle, you can minimize violations and warnings on an ongoing basis.

However, if you run code analysis on existing code that has not been checked before, you may have many rule violations. If so, you might want to create a baseline set of critical rules that the code must pass and then expand the rule set as the more critical issues are resolved. That way, a team can move forward on new functionality as it improves its existing code base.

For more information, see Overview of code analysis for managed code in Visual Studio and Create or Update Standard Code Analysis Check-in Policies.

Code review process

The lead developer should organize the code review by identifying the reviewers, scheduling the code review, and sending the code for review to all reviewers. To prepare for the code review, take the following steps:

1. Create a review work item to track the decisions that are made in the review. If no changes are necessary, set the State of the work item to Closed, set the Reason to Accepted (as is), and note that coding can start on the design. If minor changes are necessary, set the State of the work item to Resolved, and set the Reason to Accepted with Minor Changes, which indicates that coding can start after the minor changes have been implemented. If major changes are necessary, set the State of the work item to Resolved, and set the Reason to Accepted with Major Changes. The design must be reworked and another design review must be performed before coding can start on the design.
2. Determine who will participate in the code review. Typically, at least the lead developer and the architect who is responsible for the code area should participate in the review.
3. Schedule a review meeting with the reviewers, and allow sufficient time for each reviewer to read and understand the code before the meeting. Plan the length of the review meeting to correspond to how much code must be reviewed.

Code review

A code review is used to ensure that new or changed code meets an established quality bar before it's integrated into the daily build. Quality considerations are coding standards, conformance to architecture and design, performance, readability, and security. Code reviews also provide more insight from other developers about how code should be written.

TASK	DESCRIPTION
Verify Code Relevance	The code that is being reviewed is relevant to the task for which the code is written. No code changes should be allowed that don't address the functionality that is implemented or corrected.
Verify Extensibility	The code is written so that it can be extended, if that is the intent, or reused in other areas of the system. String constants that are used in the code are correctly put in resources that can be internationalized.

TASK	DESCRIPTION
Verify Minimal Code Complexity	Repeated code can be simplified into common functions. Similar functionality is put in a common procedure or function.
Verify Algorithmic Complexity	The number of execution paths in the code that is reviewed is minimized.
Verify Code Security	The code is checked for the protection of assets, privilege levels, and the use of data at entry points.

Refactor code

Code is refactored after a code review has determined that changes must be made to address code quality, performance, or architecture.

Read the code review work item notes to determine how you'll refactor the code.

Apply the refactoring incrementally, one change at a time. Change the code and all references to the modified area as necessary.

Complete unit tests so that the area remains semantically equivalent after the refactoring. Fix any unit tests that don't work. Run code analysis, and fix any warnings. Run unit tests again if code changes are made because of code analysis.

Integrate changes

The final step is to integrate the changes by checking them in to version control. Before code is checked in, any tests that are required by your process should be performed. For more information about how to check code for problems before it's checked in, see Enhancing Code Quality with Team Project Check-in Policies.

If the work item that is associated with the changes is a scenario or a quality of service requirement of which you aren't the owner, notify the owner that the changes are complete. Set the task work item to Resolved, and assign it to one of the testers who created the test cases for the work item.

If the work item that is associated with the changes is a bug, set the bug work item to Resolved, and assign it to the original person who created it.

Building a product in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To integrate development work into a component, a subsystem, a system, or a finished product, you must build the integrated code. Visual Studio Team Foundation Server provides visibility into the code base and the tools to control and manage builds and code integration. The Capability Maturity Model Integration (CMMI) model requires a basic understanding of product integration for configuration management at model level 2. However, the Product Integration process area at model level 3 requires a full product-integration strategy and management system. Team Foundation Server creates the evidence that you need for a Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal.

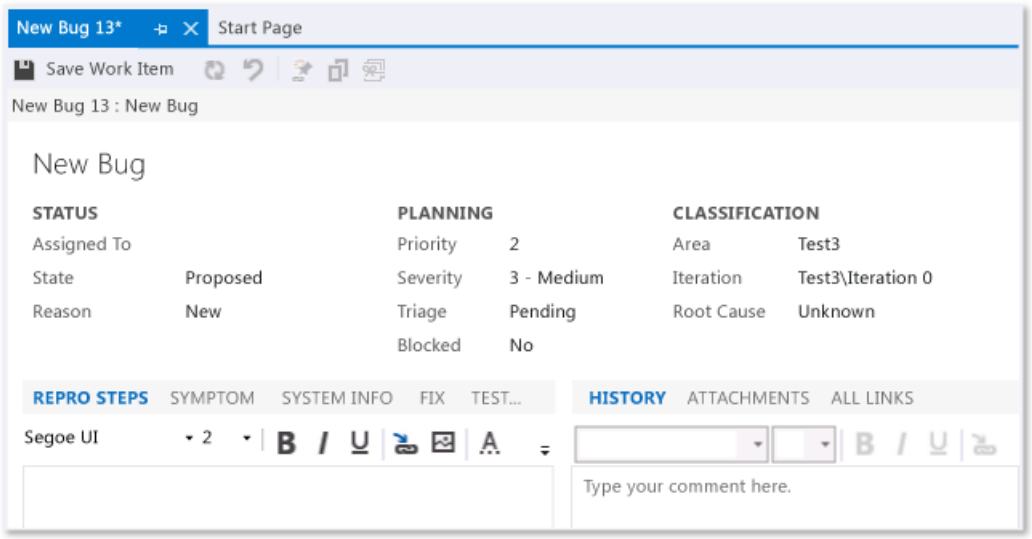
Continuous integration is a good approach to product integration that will work for many project risk profiles. For more information, see Set up CI. If this strategy isn't appropriate for your project, you should hold a meeting of technical stakeholders to discuss an appropriate plan for product integration. In the meeting, include the architect, lead developers, testers, configuration management engineers, process engineers, and a change or release manager. The team should create tasks to create appropriate build scripts.

Track bugs

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As you run tests to verify your requirements, you are bound to find bugs. Use the bug work item to describe and track progress for each bug that you find.



For more information about how to create bug work items, see Run manual tests. As bugs are found, follow the process in this article to prioritize them, to make sure that they get fixed, and to make sure that you have a record of the work and the decisions that were involved.

Triage bugs

Triage meetings should be held at set intervals after the development work and testing have started on the project. How often you hold the meetings and how long they should be depends on your situation.

Typically, bug triage meetings are run by a project manager and attended by team leads and perhaps business analysts and stakeholders who can speak about specific project risks. The project manager can use the Active Bugs query for new and reopened bugs to generate a list of bugs to be triaged.

Before triage starts, devise a set of criteria to determine which bugs should be fixed and in what priority. The criteria will typically identify the severity of bugs, bugs that are associated with features of significant value (or significant opportunity cost of delay), or other project risks.

The triage criteria should be stored in the Documents folder of your project. Over time, the document will be updated. It is assumed that your project has version control turned on and that the specific criteria that are being used at any given time on the project can be retrieved for audit and appraisal evidence purposes.

Early in the project, you will likely decide to fix most of the bugs that you triage. However, as the project proceeds, the triage criteria (or bar) may be raised to reduce the number of bugs that are fixed.

Raising the triage criteria bar and allowing reported bugs to remain unfixed is a trade-off. It is a trade-off that says that fixing the bug is less important than meeting the project scope, budget, and schedule.

Use your criteria to determine which bugs to fix and how to set their State, Priority, Severity, and other fields. By default, the template provides four priorities: 1 for "must fix" through 4 for "unimportant." If you change the definitions in the process template, you should update the guidance, the help text, and any criteria documents

accordingly.

Fix bugs

After a bug has passed triage and has been prioritized, it should be assigned to a developer for additional investigation.

The bug work item has a tab for repro steps, which should provide what you need to reproduce the bug. However, you may also be able to use IntelliTrace help you reproduce difficult bugs. For more information about IntelliTrace, see Tracking test results.

After the developer decides on a course of action, they should note their decisions in the bug work item.

Decide on the fix

Working with other team members, the developer may recommend a fix that has implications for other sections of the code and that may require significant regression testing. Any conversations that relate to assessing the risk of such a fix should be recorded in the bug work item history because it documents useful decision analysis and risk management evidence for an audit or Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal. For more information about CMMI see Background to CMMI.

Update and run unit tests for fixing the bug

Unit tests verify the correct implementation of a unit of code. Writing and performing unit tests identifies bugs before testing starts and, therefore, helps reduce the cost of quality control. Developers must write unit tests for all code that will be written as part of implementing a development task or implementing a fix for the bug. For more information, see Unit Test Your Code.

You may prefer to write the unit test before making any code fixes in a test-first development strategy. This type of strategy is preferred by agile software developers. The CMMI does not require that unit tests are written in a particular sequence, only that effective verification of functionality is achieved.

Evidence that unit tests were both written and run is required for a CMMI appraisal. Be sure to link the tests to the task work items for the code fix, and link those tasks to the bug work item. This provides the traceability of evidence that a SCAMPI lead appraiser will look for.

Review and refactor the code for fixing the bug

A code review is used to ensure that new or changed code meets an established quality bar before the code is integrated into the daily build. Quality considerations are coding standards, conformance to architecture and design, performance, readability, and security. Code reviews also provide additional insight from other developers about how code should be written. For more information about how to review and refactor code, see Implement development tasks.

Close bugs

After a bug is fixed, it should be assigned to a tester to verify that the problem is solved before the bug work item is closed.

Verifying a fix

To verify a fix, the tester should attempt to reproduce bug, look for additional unexpected behavior, and, if necessary, reactivate the bug.

When verifying a bug resolution, you may find that the bug was not completely fixed or you may disagree with the resolution. In this case, you discuss the bug with the person who resolved it, come to an agreement, and possibly reactivate the bug. If you do reactivate a bug, make sure that you include the reasons for reactivating the bug in the bug description so that information is preserved.

Closing the bug work item

A bug is closed for many reasons. In a simple case, the bug was verified as fixed and can also be closed. However, a bug can be deferred to the next version of the product, proven not to be reproducible, or confirmed as a duplicate. Each of these reasons must be documented, and the bug must be closed correctly to ensure that there is no confusion about why it is closed.

Changes made to process templates

10/4/2022 • 11 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To support the addition of new features, changes are introduced periodically to the core system processes or process template—Agile, Scrum, or CMMI. A process—used by the Inheritance process model—determines the building blocks used to track work. A process template—used by the Hosted XML and On-premises XML process models—specifies an interdependent-related set of XML definition files that provide the building blocks and initial configuration for tracking work and other functional areas. For an overview of process models and customization options, see Customize your work tracking experience.

> **NOTE**
>
> This article describes changes made to the core system processes with updates made to Azure DevOps Services and on-premises Azure DevOps Server, formerly named Team Foundation Server (TFS). These processes are available for both cloud and on-premises versions of Azure Boards. Projects hosted on Azure Boards update automatically with each service upgrade. Whereas, updates to projects defined on-premises may require running the Configure Feature Wizard after upgrading to a later version.
>
> The Configure Features Wizard has been deprecated for Azure DevOps Server 2019. You can only run the wizard on TFS 2018 and earlier versions.
> If you've customized your project and haven't upgraded your on-premises deployment for a while, you may need to manually apply some changes to gain access to new features. Review the following table to determine which changes may apply to your situation. See New features added when you upgrade for a description of each feature added with the updates.

Azure DevOps Server 2020.1

The following changes were introduced as indicated.

- Removed system rule **Assign work item to the person who created it when the bug is moved to the Resolved state** from the Bug work item type in the Agile process.
  - Azure DevOps Services (October 2020)
  - Azure DevOps Server 2020.1

  For those customers who want to continue using this rule, you can add the following custom workflow rules rule to the Bug in your process:

  When `A work item state changes to` *Resolved* **Then** `Copy the value from` *Specify field*.

- The logic managing the **Activated By**, **Activated Date**, **Resolved By**, and **Resolved Date** fields has changed. The values assigned to these fields are now governed based on the state categories of *Proposed* and *Resolved*. To learn more, see Query by assignment or workflow changes.
  - Azure DevOps Services (August 2020)
  - Azure DevOps Server 2020.1

Azure DevOps Server 2019

The following changes were introduced with Azure DevOps Server 2019.

- Predefined Shared Queries are no longer created upon project creation. The definitions for Shared Queries have been removed from the process template. Shared Queries defined in custom process templates are created. You can also add Shared Queries to a process template as described in Add work item queries to a process template.

TFS 2017

- Added the **WebLayout** section within the **FORM** section of all work item type definitions. This section supports the new work item tracking experience in the web portal. It includes the **SystemControls** section and the new **LinksControlOptions** for managing link relationships. To learn more, see New work item experience, WebLayout and Control elements, and LinksControlOptions XML elements (Web form).

> **NOTE**
> When you upgrade an on-premises Azure DevOps to TFS 2017, the new web form is automatically available when you add projects to a collection. For existing projects, an administrator is required to enable the new form. The reason the new form isn't automatically enabled for existing projects is to prevent overwriting customizations made to existing work item type definitions.

TFS 2015

- Added the Bugs Behavior feature, and enhancements to the Planning Tools and Portfolio Backlogs features. Several enhancements were made to support the Scaled Agile Framework (SAFe).

The changes introduced support the following feature additions or enhancements:

- Process template names have been changed to Agile, CMMI, and Scrum and have been repurposed as locked, system templates. You can export these templates for customization, but you can no longer overwrite these templates with your changes.
- Second-level portfolio backlog, Epic, plus configurable option for teams to activate portfolio backlogs.
- Team configurable option to choose which backlogs and portfolio backlogs are active.
- Tracking **Time Criticality** of portfolio backlog items. The **Time Criticality** field captures how the business value reduces over time for a Feature or Epic. Higher values indicate that the item is inherently more time critical than those items with lower values.
- Tracking the **Value Area** for portfolio backlog and backlog items. The **Value Area** field differentiates items based on work done to support *Architectural* requirements or *Business* needs.
- Support any-to-any workflow transitions on Agile boards.
- Team configurable option to choose to track bugs on backlogs and boards either as requirements or as tasks. This required adding fields to the bug work item type definition and adding a process configuration behavior.

The following changes were made to the default process templates:

Work item types added: Epic

Miscellaneous work item type changes:

- Feature: Added **Effort**, **Time Criticality**, and **Value Area** fields; added workflow transition from *Active* to *Removed*
- Bug: Added fields and workflow states to support the show bugs on backlog and boards team-configurable option
- Minor layout changes to work item type forms to reflect additions of new fields; added **ID** field to all forms
- Added work item type `refname` attribute to all work item type definitions.

Categories: Added Epic Category.

Process configuration changes:

- Added Epic portfolio backlog
- Feature: Added **Effort** and **Value Area** fields to the default columns of the backlog
- Requirement Category backlog: Added **Value Area** to the default columns of the backlog
  - Increased the default work item count limits on all boards to 1000
  - Added new properties to specify the default behavior for new teams.

ProcessTemplate changes: Process template names no longer specify the version or year; Agile, CMMI, Scrum.

Changes made to Agile work item type definitions:

User Story:

- Added **Acceptance Criteria**, **Priority**, and **Value Area** fields
- Added transitions from *Active* to *Removed* and *Resolved* to *Removed*
- Removed rules that populated **Activated By** and **Activated Date** fields when State=*Resolved*

Bug:

- Added **Activity**, **Story Points**, **Original Work**, **Completed Work**, **Remaining Work**, **Severity**, and **Value Area** fields
- Added *New* state and corresponding workflow transitions
- Added several field rules to copy or set defaults during state transitions
- Added *Fixed and verified* as a **Resolved Reason**.

Task: Added rule to empty **Remaining Work** field to support zeroing out the field when the State is set to *Closed*.

Changes made to CMMI work item type definitions:

Requirement:

- Added **Acceptance Criteria**, **Priority**, and **Value Area** fields
- Added transitions from *Active* to *Removed* and *Resolved* to *Removed*
- Removed rules that populated **Activated By** and **Activated Date** fields when state=*Resolved*.

Bug: Added **Size**, **Discipline**, **Original Work**, **Completed Work**, and **Value Area** fields.

Changes made to Scrum work item type definitions:

- **Product backlog item:** Added **Priority** and **Value Area** fields; added workflow transition from *Committed* to *Removed* workflow states.
- **Bug:** Added **Activity**, **Remaining Work**, **Priority**, and **Value Area** fields; added rule to zero out **Remaining Work** when **State=Done**.
- **Task:** Added rule to require **Remaining Work** when **State=In Progress**; removed **Backlog Priority** field from work item form.

TFS 2013.4

The following changes were made to the work item type definitions:

Scrum: Bug and Product backlog item: Removed the **Backlog Priority** field from the form

Agile:

- Bug: Added the **Story Points** field.
- User Story: Removed the **Stack Rank** field from the form.

CMMI:

- Added the **Size** field to the Bug definition.
- Removed the **Stack Rank** field from the Requirement form.

TFS 2013.3

Added support for the Test Plan and Test Suite feature to support customization and tracking of these items similar to other work item types.

The following changes were made to the default process templates:

- Work item types added: Test Plan and Test Suite.
- Categories added: Test Plan Category and Test Suite Category.
- Category updates: Added Test Plan and Test Suite to the Hidden Types Category.

TFS 2013.2

Added support for the Shared Parameters feature, which allows you to run tests with different data.

The following changes were made to the default process templates:

- Work item types added: Shared Parameter.
- Categories added: Shared Parameter Category.
- Category updates: Added Shared Parameter to the Hidden Types Category.

TFS 2013 RTM

Added the Portfolio Backlog feature and changes to simplify process configuration management.

The following changes were made to the default process templates:

- Work item types added: Feature.
- Categories added: Feature Category.
- Process configuration changes: Replaced `AgileConfiguration` and `CommonConfiguration` with a single file, ProcessConfiguration, which supports these other features: portfolio backlogs, color assignment to Work item types, tags added to the default columns on all backlog pages.
- Build changes: Removed the build templates from the build plug-in. You now access build templates through the user interface [Use the Default Template for your build process](/previous-versions/visualstudio/visual-studio-2013/dd647547(v=vs.120).
- Reporting Services updates: To update your project with the latest reports, see Upload reports to a project.

TFS 2012.1

Added the Portfolio Backlog feature and introduced changes to support Planning Tools.

Changes to work item type definitions to support status updates in the Kanban and taskboards include the following items:

- Each of the default process templates were updated to support other regressive transitions. These transitions, shown in red in the following illustration, support moving items back to the backlog when they were incorrectly set to done or resolved. Now when you inadvertently drag a work item on the Kanban board or the taskboard to a resolved or closed state, you can drag it back to an earlier workflow state.

The following work item types now support any-to-any workflow transitions:

- Visual Studio Scrum 2.1: Bug, Product Backlog Item, Task
- MSF Agile 6.1: Bug, Task, User Story
- MSF Scrum 6.1: Bug, Task, Requirement



To apply the changes to your existing projects, you need to replace the **WORKFLOW** sections defined for each of the updated work item types with the new definitions. You can do this by modifying the work item type definition. See Design the Workflowand Import, export, and manage Work Item Types.

TFS 2012 RTM

Added the Code Review, My Work, Feedback, Storyboarding, and Planning Tools features.

The following changes were added to all default process templates:

- **Work item types added:** Code Review Request, Code Review Response, Feedback Request, and Feedback Response.
- **Work item form changes:**
  - Many work item forms have been updated to display the larger font for the **Title** field and a two-column layout of tab groups.
  - Maintained mnemonic support only for the following fields: **Area Path**, **Assigned To**, **History**, **Iteration Path**, **State**, and **Title**. Mnemonics have been removed from all other fields within the type definitions.
- Categories added: Code Review Request Category, Code Review Response Category, Feedback Request Category, Feedback Response Category, and Hidden Types Category.
- **Process configuration:** Added `CommonConfiguration` and `AgileConfiguration` definition files to support Agile planning tools, code review, feedback, and My Work.
- **Build plug-in:** Added a new build process template. When you upgrade from earlier versions of Team Foundation Server, you can continue to use [legacy build processes](/previous-versions/dd647548(v=vs.140).
- **Process template plug-in:** Added `version` element to support updating projects when configuring new features.

- Updated support files that contain forward links to process guidance content to point to the latest content.

The following changes were made to the Agile process:

- The *New* state was added as the initial state for User Stories and Tasks. This change was introduced to support transitioning User Stories and Tasks from a *New* state to an *Active* state. Without the *New* state, User Stories assigned to an iteration remain on the product backlog until they are resolved, which is not a desired or expected behavior. To learn more, see [Update the Workflow for Agile Team Projects](/previous-versions/visualstudio/visual-studio-2012/hh500412(v=vs.110).
- To support removing cut work items from the backlog, the *Removed* state was added to the workflow state for the following work item types: User Story and Task.
- Added the **Storyboards** tab to the User Story work item type.
- The Product Planning and Iteration Backlog Excel workbooks have been removed. The Agile planning tools replace this functionality.

The following changes were made to the CMMI process:

- Requirement: Added the **Size** field used in calculating team velocity and forecast; added the **Storyboards** tab
- Shared queries: Added **Corrective Action** and **Mitigation Action**.

The following changes were made to the Scrum process:

- Removed the **Sprint** type definition and **All Sprints** query, whose functionality has been replaced with the Agile planning tools.
- Removed the **Backlog Priority** field from the work item form for the Product Backlog type definition. This field is used in the background to track the relative priority of items displayed on the backlog pages.
- The *Removed* state was added to the workflow state for the Bug, Product Backlog Item, and Task type definitions, which supports removing cut items from the backlog.
- Added the **Storyboards** tab to the Product Backlog Item type definition.
- Added the Backlog Overview report to the set of SQL Server Reporting Services reports, similar to the AgileStories Overview report.

Related upgrade notes

To update your existing projects hosted on an on-premises Azure DevOps to access new features, run the Configure Features wizard. When you're upgrading from a much earlier version of TFS or you've customized your project, you'll need to make some manual updates.

See the following resources as they relate to updating your project:

- Update a customized process template to access new features
- Add features using a manual update process
- Before you upgrade TFS 2008 or TFS 2010
- More configuration options
- Upload reports to a project

Default permissions and access levels for Azure Boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As a member of an Azure Boards project, you can use most features to track work. Limitations to select features are based on the *access level* and *security group* to which a user is assigned. The **Basic** access level and higher supports full access to all Azure Boards features. **Stakeholder** access level provides partial support to select features, allowing users to view and modify work items, but not use all features. The built-in security groups —**Readers**, **Contributors**, and **Project Administrators**—and team administrator role grant permissions to specific features.

In the tables provided in this article, a ✔ indicates that the corresponding access level or security group has access to a feature by default.

> **NOTE**
> Team administrators can configure settings for their team's tools. Organization owners and members of the Project Administrators group can configure settings for all teams.

For a comparison chart of Stakeholder versus Basic access, see the Feature matrix. To assign or change an access level, see Add users and assign licenses. If you need to grant specific users select permissions, you can do so.

Work item feature access

You can use work items to track anything you need to track. To learn more, see Understand how work items are used to track issues, tasks, and epics.

> **NOTE**
> You can change the work item type or move work items to another project within a project collection. These features require that the data warehouse is disabled. With the data warehouse disabled, you can use the Analytics Service to support your reporting needs. To learn more about disabling the data warehouse, see Disable the data warehouse and cube.

Task or permission

Readers

Contributors

Project admins

View work items in this node (Area Path permission)

✔

✔

✔

Edit work items in this node (Area Path permission)

✔

✔

Create tag definition

✔

✔

Change work item type (Project-level permission)

✔

✔

Move work items out of this project (Project-level permission)

✔

✔

Email work items

✔

✔

✔

Apply a work item template

✔

✔

Delete and restore work items (Project-level permission) (able to restore from the Recycle bin)

✔

✔

Permanently delete work items (Project-level permission)

✔

Provide feedback (through the Microsoft Feedback client)

✔

✔

Request feedback

✔

✔

NOTE

Work items are subject to rules applied to them. Conditional rules based on user or group membership are cached for your web browser. If you find yourself restricted to update a work item, you may have encountered one of these rules. If you believe you've encountered an issue that doesn't apply to you, see Work item form IndexDB caching issues. To learn more about conditional rules, see Rules and rule evaluation.

Boards feature access

Boards present work items as cards and support quick status updates through drag-and-drop.

Task	Readers	Contributors	Team admins Project admins
View boards and open work items	✔	✔	✔
Add work items to a board; update status through drag-and-drop		✔	✔
Reorder work items or reparent child items through drag-and-drop; update a field on a card		✔	✔
Add work items to a board; update status, reorder, or reparent child items through drag-and-drop; update a field on a card		✔	✔
Add child items to a checklist		✔	✔
Assign to a sprint (from card field)		✔	✔
Configure board settings			✔

Backlogs features access

Backlogs display work items as lists. A product backlog represents your project plan and a repository of all the information you need to track and share with your team. Portfolio backlogs allow you to group and organize your backlog into a hierarchy.

Task

Readers

Contributors

Team admins
Project admins

View backlogs and open work items

✔

✔

✔

Add work items to a backlog

✔

✔

Use bulk edit features

✔

✔

Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using the Planning pane

✔

✔

Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using drag-and-drop

✔

✔

Configure team settings, backlog levels, show bugs, work days off

✔

Sprints feature access

Sprints provide a filtered view of work items that a team has assigned to specific iteration paths or sprints.

Task

Readers

Contributors

Team admins Project admins

View sprint backlogs, taskboards, and open work items

✔

✔

✔

Add work items to a sprint backlog or taskboard

✔

✔

Prioritize/reorder a sprint backlog or taskboard; add child items to a backlog item; reassign items to a sprint using the Planning pane

✔

✔

View team capacity and work details

✔

✔

Set team capacity

✔

✔

Use bulk edit features

✔

✔

Define team sprints

✔

Queries and semantic search

Queries are filtered lists of work items based on criteria that you define by using a query editor.

> **TIP**
> By default, Contributors can't create and save shared queries. We recommend that Project Administrators create a query folder for each team and give the team administrators or the team group query permissions to manage their folder. You need **Delete** permissions to rename or move a shared query or folder, and **Contribute** permissions for the folder where you move the query to. To learn more, see Set permissions on queries and query folders.

Task

Readers

Contributors

Project admins

View and run managed queries, view query charts

✔

✔

✔

Create and save managed My queries, query charts

✔

✔

Create, delete, and save Shared queries, charts, folders

✔

Stakeholder access

Stakeholder access supports business owners. It also supports analysts and other team members who don't manage the work of a project, but need to view and add ideas to the backlog, add context and information to work items, and review status and progress. All members of an organization who don't use Visual Studio but want to contribute to work item tracking and monitor progress can be assigned as a stakeholder. Note, even if you change the permission level for a user assigned **Stakeholder** access, the user will be blocked from accessing the feature.

> **NOTE**
> For public projects, Stakeholder access gives users full access to all work-tracking features. To learn more, see Stakeholder access quick reference.

Related articles

- Get started as a stakeholder
- Add another team
- Add a team administrator
- Manage teams and configure team tools
- Grant or restrict access to select features and functions
- Set permissions and access for work tracking

Permissions and access for work tracking

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As a member of an Azure DevOps project, you can use most of the features to track work. Limitations to select features are based on the *access level* and *security group* to which a user is assigned. The **Basic** access level and higher supports full access to all Azure Boards features. **Stakeholder** access level provides partial support to select features, allowing users to view and modify work items, but not use all features. The built-in security groups—**Readers**, **Contributors**, and **Project Administrators**— and team administrator role grant permissions to specific features.

As a member of an Azure DevOps project, you can use most of the features to track work. Limitations to select features are based on the *access level* and *security group* to which a user is assigned. The **Basic** access level and higher supports full access to all features under the **Work** hub. **Stakeholder** access level provides partial support to select features, allowing users to view and modify work items, but not use all features. The built-in security groups—**Readers**, **Contributors**, and **Project Administrators**— and team administrator role grant permissions to specific features.

In the tables provided in this article, a ✔ indicates that the corresponding access level or security group has access to a feature by default.

> **NOTE**
> Team administrators can configure settings for their team's tools. Organization owners and members of the **Project Administrators** group can configure settings for all teams. To be added as an administrator, see Add team administrators or Change project-level permissions.

For a comparison chart of Stakeholder versus Basic access, see the Feature matrix. To assign or change an access level, see Add users and assign licenses. If you need to grant specific users select permissions, you can do so.

Work items

You can use work items to track anything you need to track. To learn more, see Understand how work items are used to track issues, tasks, and epics.

> **NOTE**
> You can change the work item type or move work items to another project within a project collection. These features require that the data warehouse is disabled. With the data warehouse disabled, you can use the Analytics Service to support your reporting needs. To learn more about disabling the data warehouse, see Disable the data warehouse and cube.

Task or permission

Readers

Contributors

Project admins

View work items in this node (Area Path permission)

✔

✔

✔

Edit work items in this node (Area Path permission)

✔

✔

Create tag definition

✔

✔

Change work item type (Project-level permission)

✔

✔

Move work items out of this project (Project-level permission)

✔

✔

Email work items

✔

✔

✔

Apply a work item template

✔

✔

Delete and restore work items (Project-level permission) (able to restore from the Recycle bin)

✔

✔

Permanently delete work items (Project-level permission)

✔

Provide feedback (through the Microsoft Feedback client)

✔

✔

Request feedback

✔

✔

NOTE

Work items are subject to rules applied to them. Conditional rules based on user or group membership are cached for your web browser. If you find yourself restricted to update a work item, you may have encountered one of these rules. If you believe you've encountered an issue that doesn't apply to you, see Work item form IndexDB caching issues. To learn more about conditional rules, see Rules and rule evaluation.

Boards

You use **Boards** to implement Kanban methods. Boards present work items as cards and support quick status updates through drag-and-drop.

Task	Readers	Contributors	Team admins Project admins
View boards and open work items	✔	✔	✔
Add work items to a board; update status through drag-and-drop		✔	✔
Reorder work items or reparent child items through drag-and-drop; update a field on a card		✔	✔
Add work items to a board; update status, reorder, or reparent child items through drag-and-drop; update a field on a card		✔	✔
Add child items to a checklist		✔	✔
Assign to a sprint (from card field)		✔	✔
Configure board settings			

✔

Backlogs

Backlogs display work items as lists. A product backlog represents your project plan and a repository of all the information you need to track and share with your team. Portfolio backlogs allow you to group and organize your backlog into a hierarchy.

Task

Readers

Contributors

Team admins
Project admins

View backlogs and open work items

✔

✔

✔

Add work items to a backlog

✔

✔

Use bulk edit features

✔

✔

Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using the Planning pane

✔

✔

Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using drag-and-drop

✔

✔

Configure team settings, backlog levels, show bugs, work days off

✔

Sprints

You use sprint tools to implement Scrum methods. The Sprints set of tools provide filtered views of work items that a team has assigned to specific iteration paths or sprints.

Task

Readers

Contributors

Team admins Project admins

View sprint backlogs, taskboards, and open work items

✔

✔

✔

Add work items to a sprint backlog or taskboard

✔

✔

Prioritize/reorder a sprint backlog or taskboard; add child items to a backlog item; reassign items to a sprint using the Planning pane

✔

✔

View team capacity and work details

✔

✔

Set team capacity

✔

✔

Use bulk edit features

✔

✔

Define team sprints

✔

Queries and semantic search

Queries are filtered lists of work items based on criteria that you define by using a query editor. Adhoc searches are powered by a semantic search engine.

> **TIP**
> By default, Contributors can't create and save shared queries. We recommend that Project Administrators create a query folder for each team and give the team administrators or the team group query permissions to manage their folder. You need **Delete** permissions to rename or move a shared query or folder, and **Contribute** permissions for the folder where you move the query to. To learn more, see Set permissions on queries and query folders.

Task

Readers

Contributors

Project admins

View and run managed queries, view query charts

✔

✔

✔

Create and save managed My queries, query charts

✔

✔

Create, delete, and save Shared queries, charts, folders

✔

Delivery plans

Delivery plans display work items as cards against a calendar view. This format can be an effective communication tool with managers, partners, and stakeholders for a team. Users granted **Stakeholder** access for private projects have no access to delivery plans, while users granted **Stakeholder** access for public projects has the same access as regular Contributors granted **Basic** access.

You can manage permissions for individual plans. To learn more, see Edit or manage Delivery Plan permissions.

Task

Readers

Contributors

Team admins
Project admins

View delivery plans

✔

✔

✔

Create, edit, or delete a delivery plan, Contributors can only edit or delete plans that they create

✔

✔

Manage permissions for a delivery plan, Contributors can only manage permissions for plans that they create

✔

✔

Test management

Test plans, test suites, test cases and other test artifacts are specific work item types that support manual and

exploratory testing. You set test permissions at the project level from the **Project settings>Security** page.

Permission

Level

Readers

Contributors

Project Admins

View test runs

Project-level

✔

✔

✔

Create test runs
Delete test runs

Project-level

✔

✔

Manage test configurations
Manage test environments

Project-level

✔

✔

Create tag definition
Delete and restore work items

Project-level

✔

✔

Permanently delete work items

Project-level

✔

View work items in this node

Area Path

✔

✔

✔

Edit work items in this node

Manage test plans
Manage test suites

Area Path

✔

✔

> **NOTE**
> The **Change work item type** permission doesn't apply to test-specific work items. Even if you choose this feature from the work item form, changing the work item type is disallowed.

Area permissions for web-based test case management and test execution control access to the following actions.

The **Manage test suites** permission enables users to:

- Create and modify test suites
- Add or remove test cases to/from test suites
- Change test configurations associated with test suites
- Modify the suite hierarchy by moving a test suite

The **Manage test plans** permission enables users to:

- Create and modify test plans
- Add or remove test suites to or from test plans
- Change test plan properties such as build and test settings

Project-level resources

You set project-level information permissions from **Project settings** > **Permissions**. You set permissions for area and iteration paths under **Project settings** > **Project configuration**. These resources are defined for a project which all valid users of the project can view.

Task

Stakeholder

Readers

Contributors

Team Admins

Project Admins

View project-level information

✔

✔

✔

✔

✔

Edit work items under the node

✔

✔

✔

Area nodes and Iteration nodes: Create, delete, edit child nodes

✔

✔

✔

✔

✔

Edit project-level information

✔

Area nodes and Iteration nodes: Create, delete, edit child nodes

✔

✔

✔

✔

✔

The **Edit project-level information** permission includes the ability to perform these tasks for the project:

- Create and modify areas and iterations
- Edit check-in policies
- Edit shared work item queries
- Edit project level permission ACLs
- Create and modify global lists
- Edit event subscriptions (email or SOAP) on project level events.

Team administrator role and permissions

The following table summarizes a subset of the default permissions assigned to the project Readers, Contributors and Project Administrators groups and the Team Administrator role. Team admin permissions extend only to the team for which they're an administrator. Project Administrator permissions extend across all teams defined for the project.

Permission

Readers

Contributors

Team Administrators

Project Administrators

Add a team administrator

✔

✔

Add team members

✔

✔

View shared work item queries

✔

✔

✔

✔

Manage shared query and query folder permissions
(Contribute, Delete, Manage Permissions)

✔

Add and edit dashboards

✔

✔

Stakeholder access

Stakeholder access supports business owners and analysts and other team members who don't contribute to code, build, and test activities. They contribute by adding ideas to the backlog, adding context and information to work items, and reviewing status and progress. All members of an organization who don't use Visual Studio but want to contribute to work item tracking and monitor progress can be assigned as a stakeholder. To learn more about Stakeholder access, see Work as a stakeholder.

For a comparison chart of Stakeholder versus basic access, see the Feature Matrix.

For information about each access levels, see About access levels. To assign access levels, see:

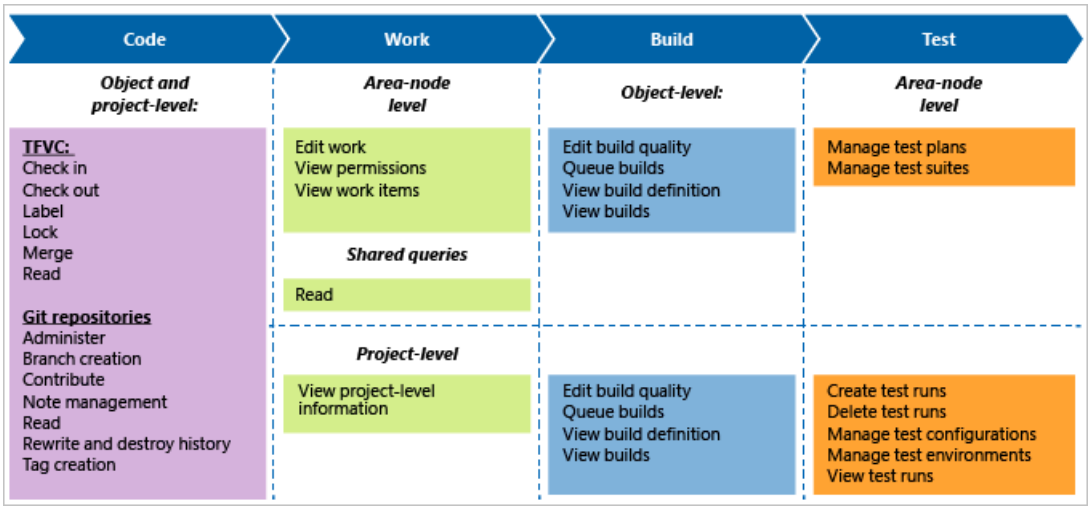
- **Azure DevOps Services**: Add users and assign licenses in Azure DevOps
- **Azure DevOps Server, TFS**: Change access levels

Grant team members additional permissions

For teams to work autonomously, you may want to provide them with permissions that they don't have by default. Suggested tasks include providing team administrators or team leads permissions to:

- Create and edit child nodes under their default area path
- Create and edit child nodes under an existing iteration node
- Create shared queries and folders under the Shared Queries folder.

By default, team members inherit the permissions afforded to members of the project Contributors group. Members of this group can add and modify source code, create and delete test runs, and create and modify work items. They can collaborate on a Git project or collaborate with other team members and check in work to the team's code base (TFVC).



If your on-premises deployment includes reporting, add users to those resources. See Grant permissions to view or create SQL Server reports in TFS.

Related notes

- Set permissions and access for work tracking
- Get started as a Stakeholder
- Add another team
- Manage teams and configure team tools
- Work item form IndexDB caching issues

Set work tracking permissions

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You grant or restrict access to various work tracking features by granting users or groups specific permissions for an object, project, or collection. Or, you can specify custom rules for a process or project that apply to users or groups which may restrict or require users to perform a select action. In general, you'll want to add users to a project's Contributors group to provide access to most features as listed in Permissions and access for work tracking.

> **NOTE**
> For public projects, Stakeholder access gives users greater access to work tracking features and full access to Azure Pipelines. To learn more, see Stakeholder access quick reference.

Prerequisites

- To set work tracking permissions, you must be a member of the **Project Administrators** group, or have explicit permissions to manage the work tracking area as described in this article.
- To set process permissions, you must be a member of the **Project Collection Administrators** group, or have explicit permissions to edit a collection process.

Support business workflows through custom rules

Custom rules don't set permissions, but do impact the effective permissions of a user at run-time to modify a work item or set the value of a work item field. Azure Boards supports the following work tracking customizations that support business workflows.

- Apply select rules upon work item creation, state change, specified state.
- Apply select rules when a field value is empty, set to a specific value, or was changed or not changed to a value.
- Restrict the transition to a specific state when moving from a specified state.
- Apply select rules based on user or group membership of the user modifying a work item.

Common actions set by rules include:

- Make a field required or read-only
- Clear or set the value of a field, or copy a field value to another field
- Hide a field.

For example, you can specify rules that effectively restrict a group of users from performing the following tasks:

- Creating a work item
- Transitioning a work item to a closed or completed state
- Changing the value of a field.

Or, perform the following automations:

- Reassign a work item based on state change

- State transitions of parent work items based on state changes made to their child work items.

Some restrictions are placed on applying custom rules to system fields. For example, you can't specify rules that set or clear the value for **Area Path** or **Iteration Path** as these are system fields. To learn more about custom rules you can define and system field restrictions on custom rules, see Rules and rule evaluation. For sample scenarios that define custom rules, see Sample rule scenarios.

Work tracking roles and permission levels

The following table summarizes the different permissions you can set at the object, project, or collection level. The team administrator role provides access to add and modify team resources.

Role or permission level	Functional areas set
Team administrator role To add a user to the team administrator role, see Add a team administrator.	• Manage teams and configure team tools • Define and edit team dashboards • Add and manage team-level work item templates • Add team administrators
Object-level permissions	• Modify work items under an area path • Create and edit nodes under an area path or iteration path • Define and edit queries or query folders • Define and edit Delivery Plans
Project-level permissions	• Create work item tags • Move work items out of a project • Permanently delete work items • Edit shared work item queries • Add teams and team administrators) • Edit project-level permissions • Create work item tags • Permanently delete work items • Edit shared work item queries • Add teams and team administrators) • Edit project-level permissions
Project collection-level permissions	• Create, delete, or edit a process (Inheritance process model) • Delete field from account (Inheritance process model) • Manage process permissions (Inheritance process model) • Edit collection level permissions Project collection-level permissions include all permissions you can set at the collection-level.

- Edit collection level permissions
  Project collection-level permissions include all permissions you can set at the collection-level.

> **NOTE**
> If you haven't defined the group yet whose permissions you want to set, then first create the group, usually at the project level. To learn how, see Add or remove users or groups, manage security groups.

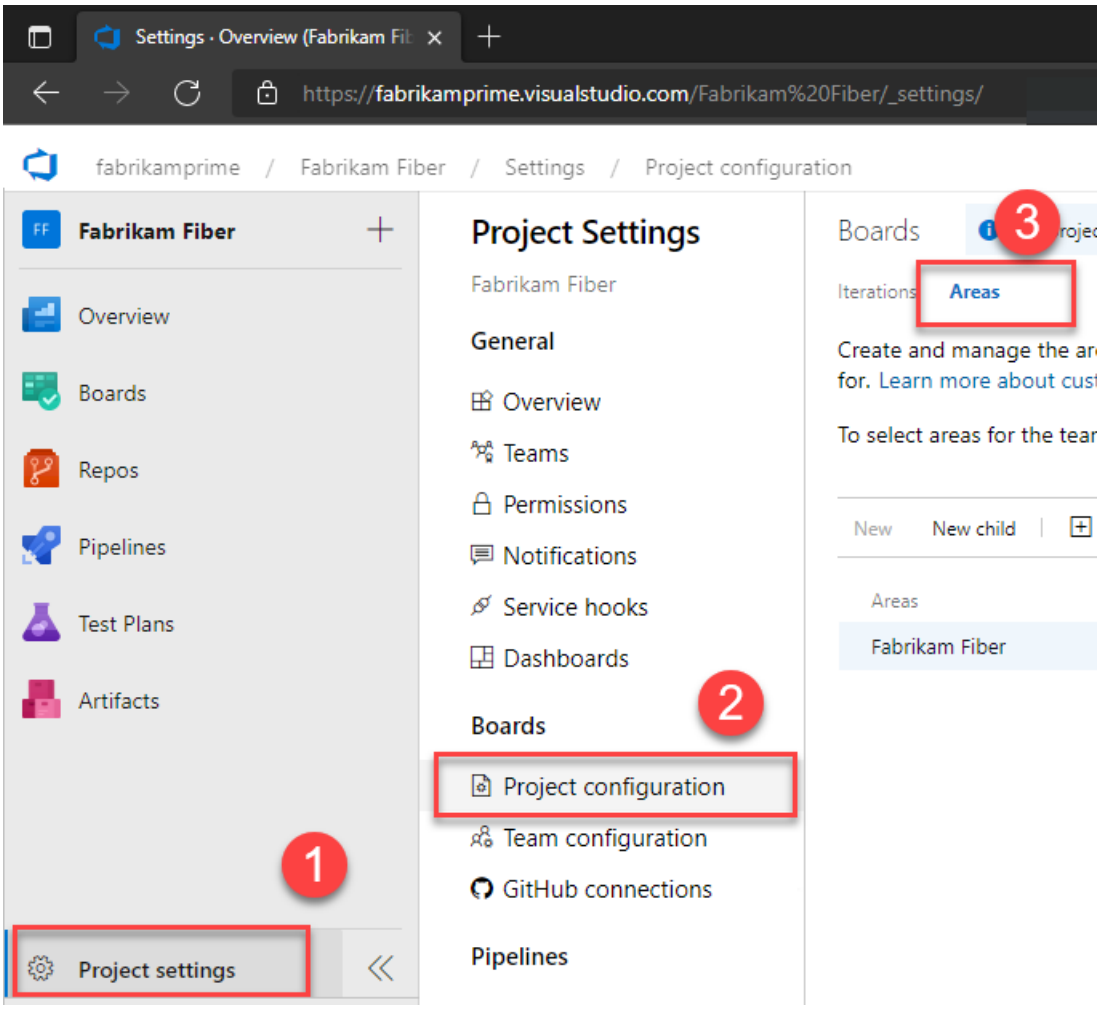
Create child nodes, modify work items under an area or iteration path

Area path permissions let you grant or restrict access to edit or modify work items, test cases, or test plans assigned to those areas. You can restrict access to users or groups. You can also set permissions for who can add or modify areas or iterations for the project.

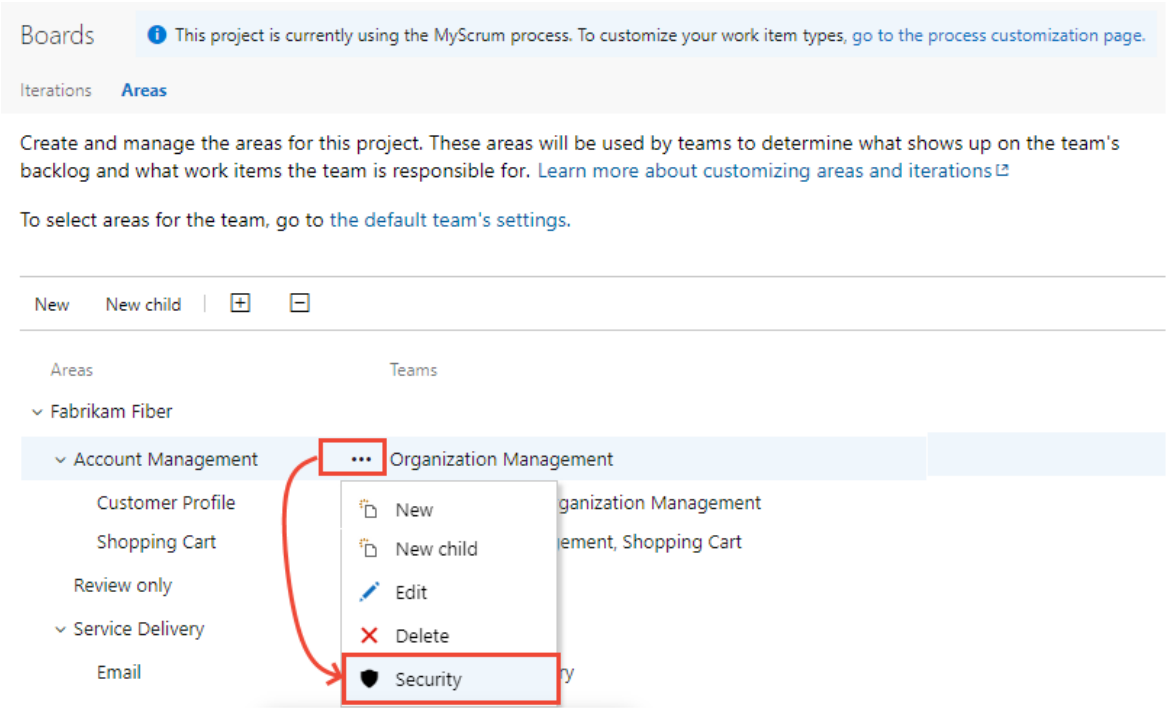
> **NOTE**
> Project members granted permissions to create or edit **Area Paths** or **Iteration Paths** is separate from the permissions that control configuring team **Area Paths** and **Iteration Paths**. To configure team settings, you must be added to the team administrator role, or be a member of the **Project Administrators** group. *

You define both areas and iterations for a project from the **Project Settings** > **Project configuration**.

1. Choose (1) **Project Settings**, then choose (2) **Project configuration** under **Boards**, and then choose (3) **Areas** or **Iterations** to modify Area Paths or Iteration Paths.

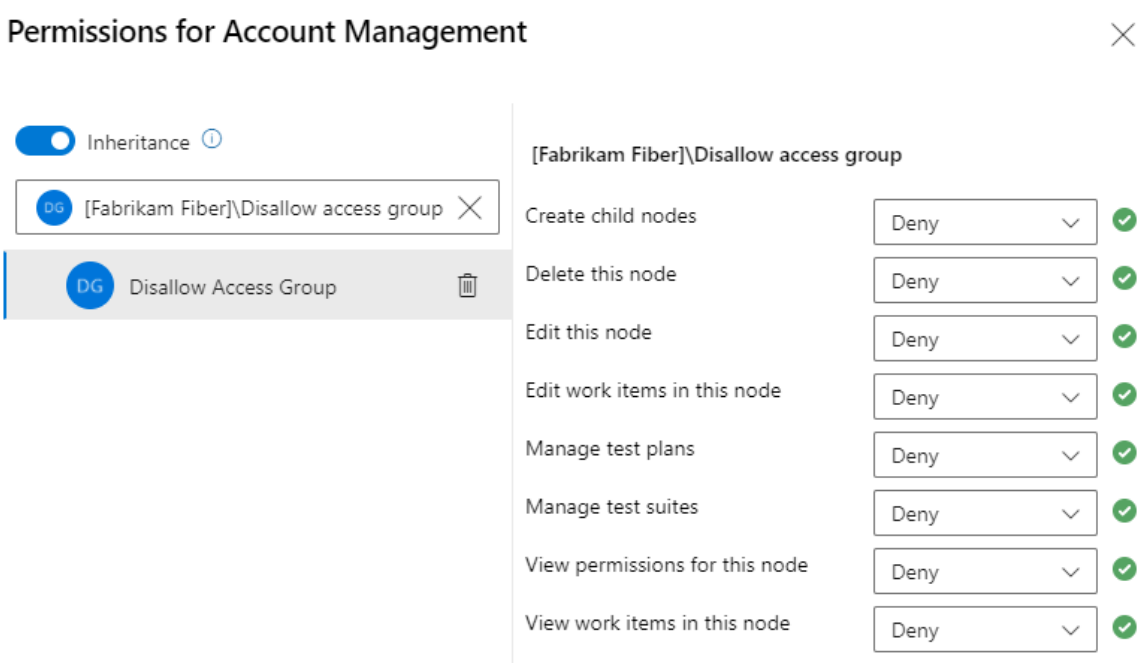


2. Choose the ... context menu for the node you want to manage and select **Security**.



3. Select the group or project member, and then change the permission settings. To add a user or group, enter their name in the search box.

   For example, here we added the *Disallow Access Group*, and disallowed members of this group the ability to view, modify, or edit work items in the **Account Management** area path.



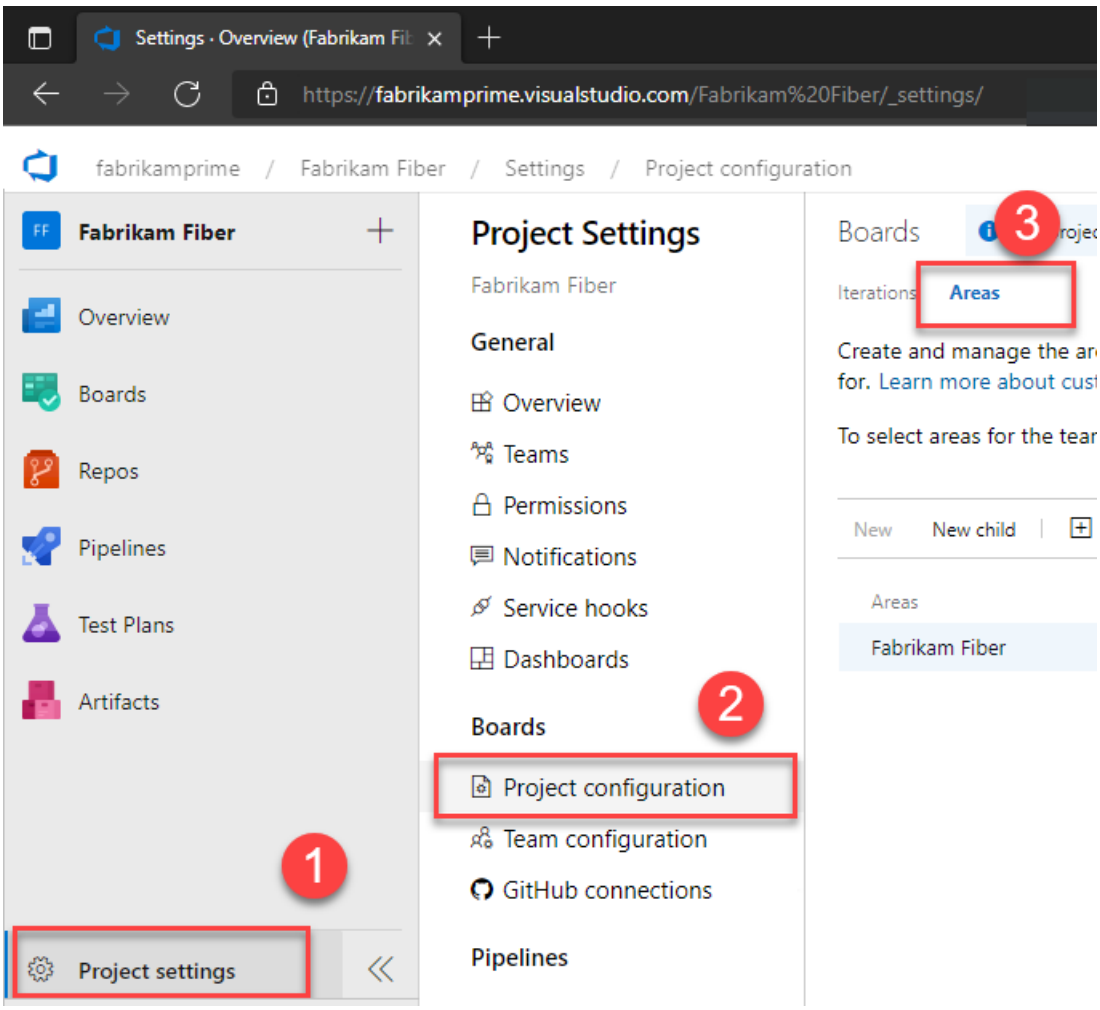
   You can specify two explicit authorization states for permissions: **Deny** and **Allow**. In addition, permissions can exist in one of three additional states. To learn more, see Get started with permissions, access, and security groups.

4. (Optional) Choose the **Inheritance** slider to disable inheritance. Disabling **Inheritance** persists all inherited permissions as explicit Access Control Entries (ACEs).

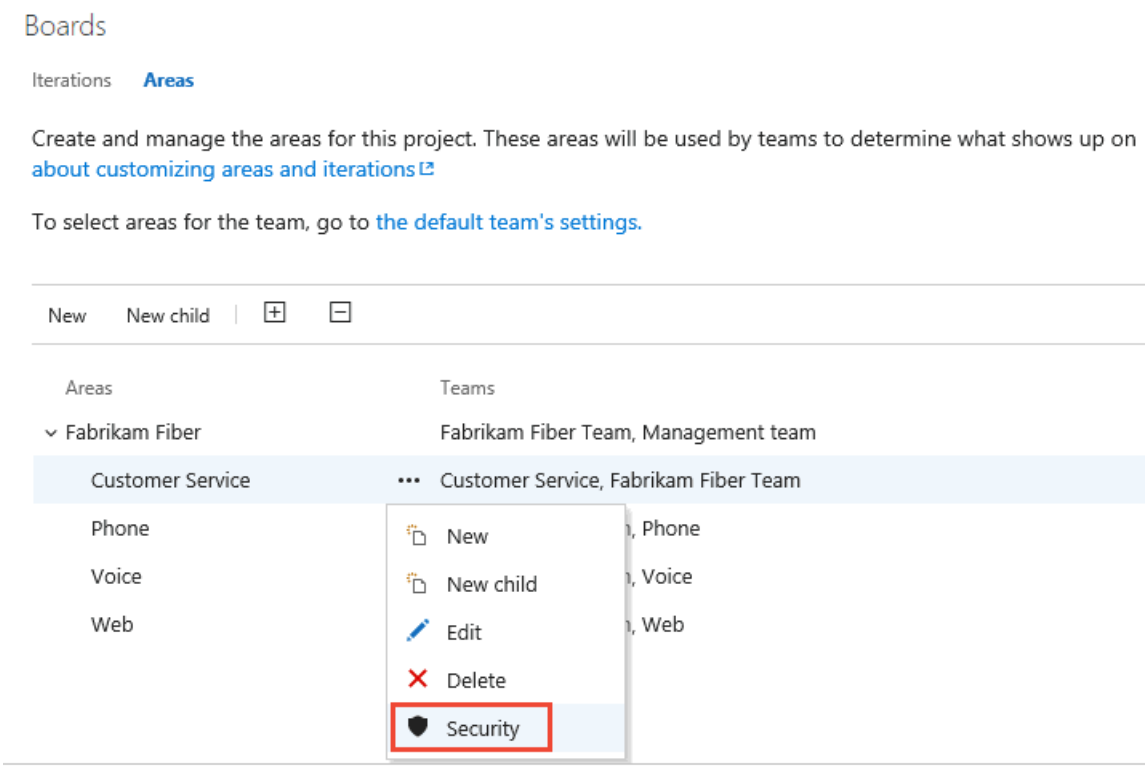
5. When done, simply close the dialog. Your changes are automatically saved.

You define both areas and iterations for a project from the **Project Settings>Project configuration**.

1. Choose (1) **Project Settings**, then choose (2) **Project configuration** under **Boards**, and then choose (3) **Areas**.

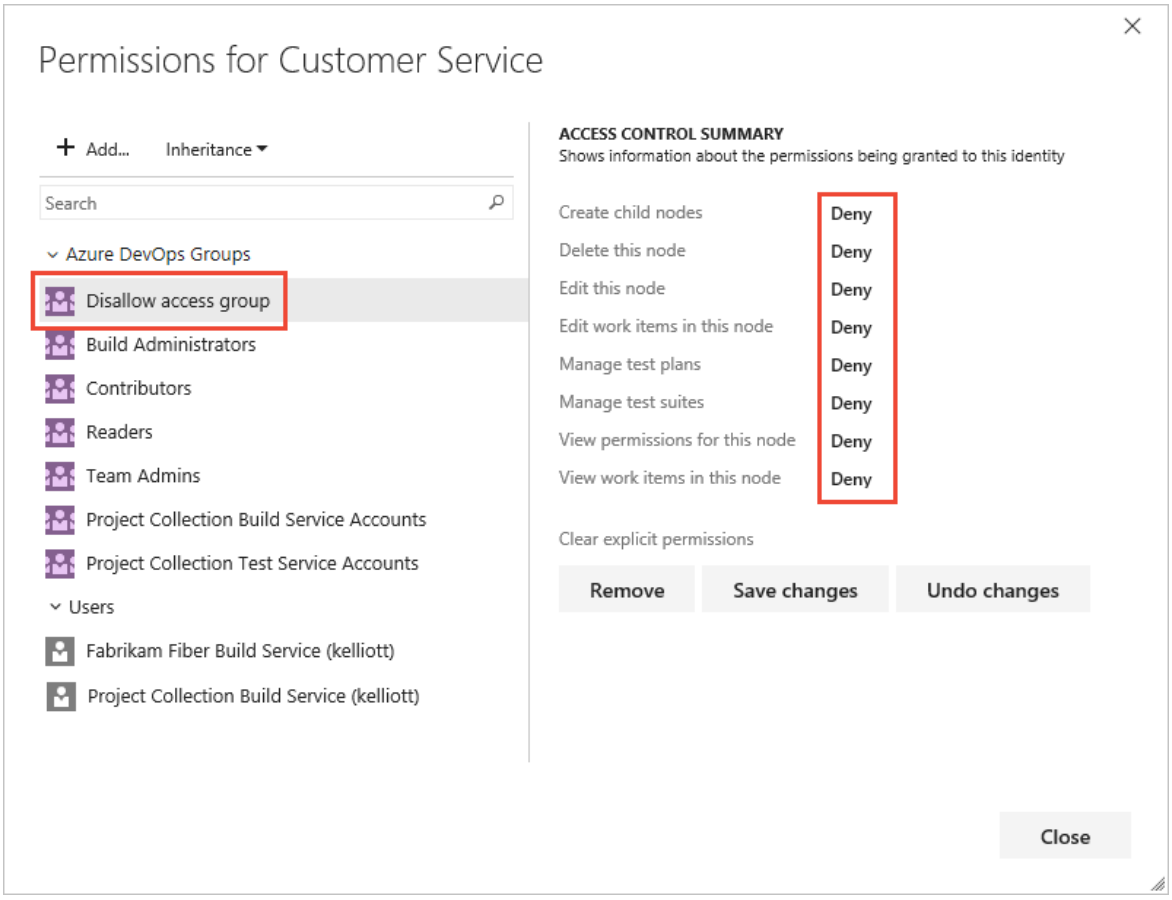


2. Choose the **...** context menu for the node you want to manage and select **Security**.



3. Select the group or team member, and then change the permission settings. To add a user or group, enter their name in the search box.

For example, here we've added the *Disallow Access Group*, and disallowed members of this group the ability to view, modify, or edit work items in the Customer Service area path.

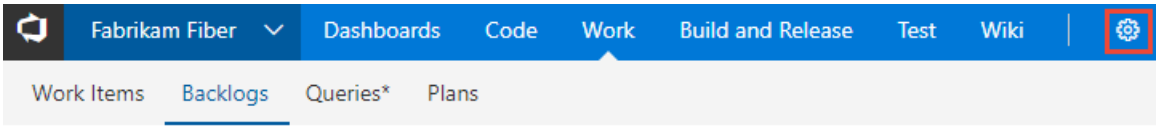


You can specify two explicit authorization states for permissions: **Deny** and **Allow**. In addition, permissions can exist in one of three additional states. To learn more, see Get started with permissions, access, and security groups.

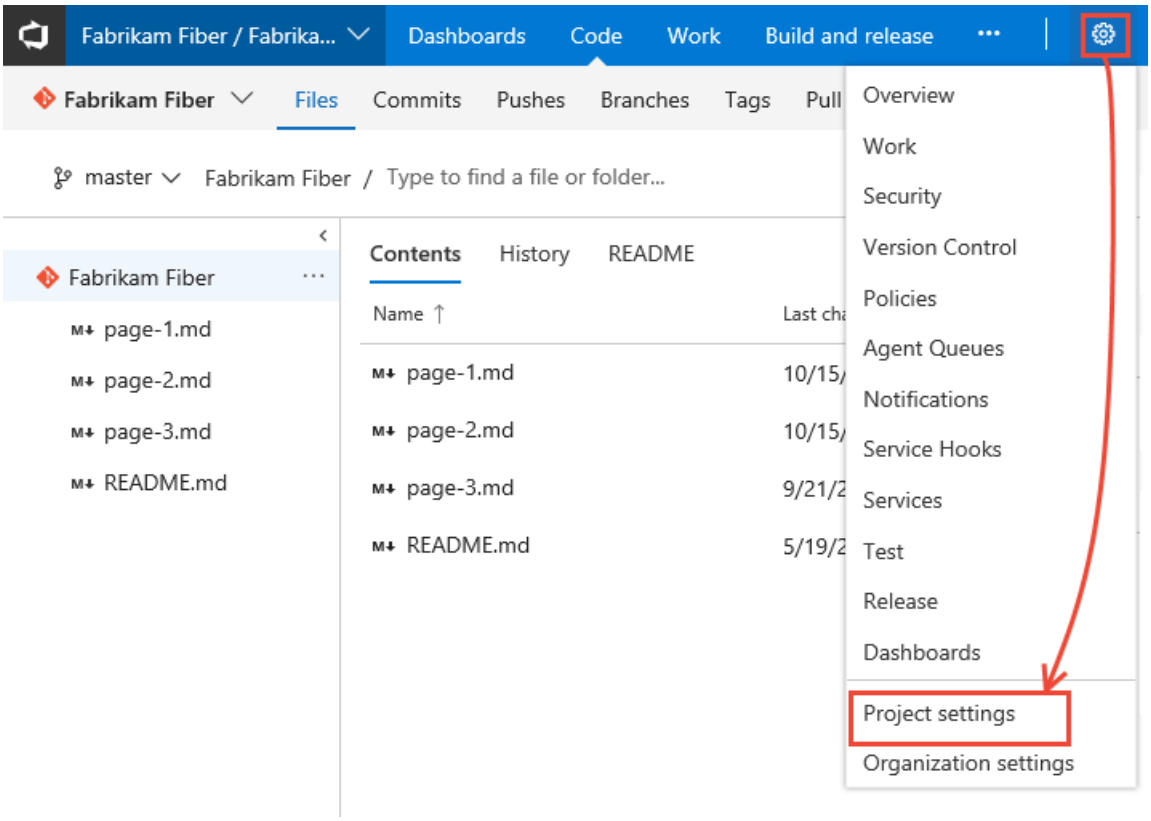
4. (Optional) Toggle **Inheritance** to **Off** to disable inheritance. Disabling **Inheritance** persists all inherited permissions as explicit Access Control Entries (ACEs).

5. When done, choose **Save changes**.

1. From the web portal for the project, choose the gear icon.

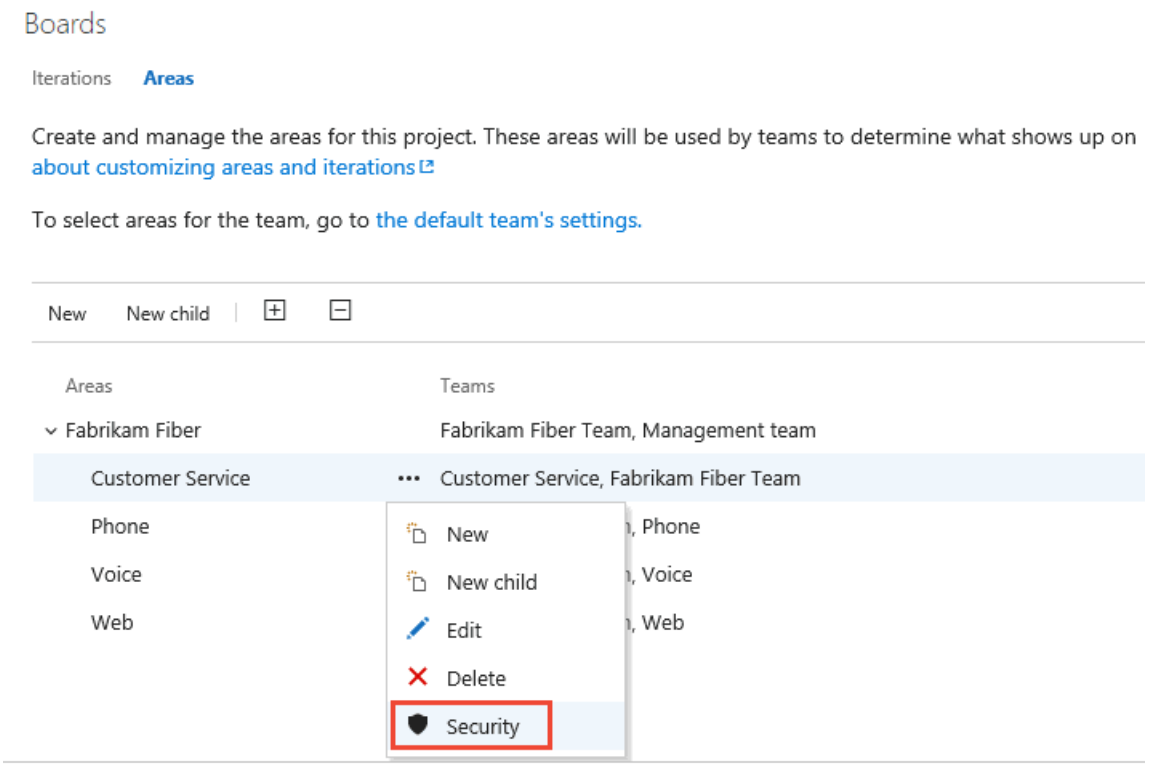


If you're currently working from a team context, then hover over the and choose **Project settings**.



2. Choose **Work** and then **Areas**.

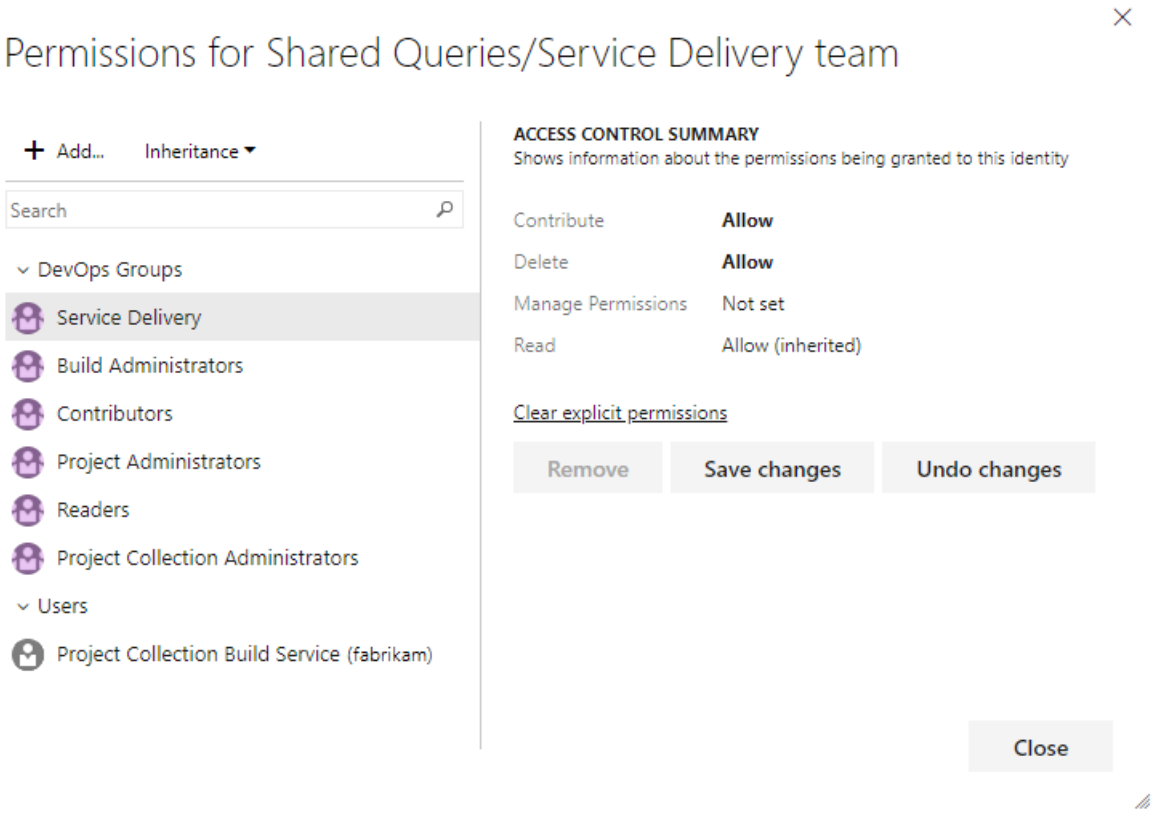
3. Choose the ... context menu for the node you want to manage and select **Security**.



Set permissions on queries or query folders

You can specify who can add or edit query folders or queries at the object-level. To manage permissions for a query or query folder, you must be the creator of the query or folder, a member of the Project Administrators or Project Collection Administrators group, or granted explicit access through the object's Security dialog.

Query folder Permissions dialog



For details, see Set permissions on a shared query or query folder. To learn more about queries, see Create managed queries to list, update, or chart work items.

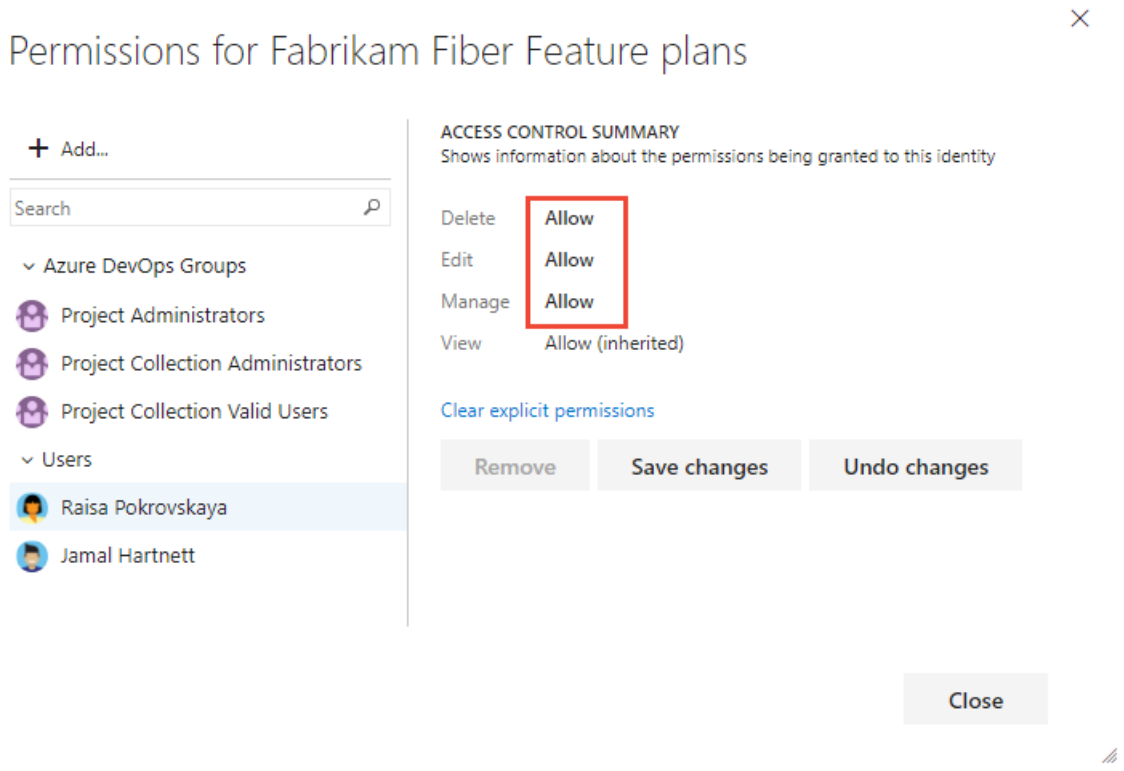
Set permissions on work item tags

By default, all users of the Contributors group can create and add tags to work items. To set permissions for a group or user to restrict this ability, you can set the **Create tag definition** to **Deny** at the project-level. To learn how, see Change the permission level for a project-level group.

Edit or manage permissions for Delivery Plans

Delivery Plans are an object within a project. You manage plan permissions for each plan similar to the way you manage permissions for shared queries or query folders. The creator of a Delivery Plan as well as all members of the Project Collection Administrators and Project Administrators groups have permissions to edit, manage, and delete plans.

Delivery Plan Permissions dialog



To learn more, see Edit or manage Delivery Plan permissions. To learn more about Delivery Plans, see Review team plans.

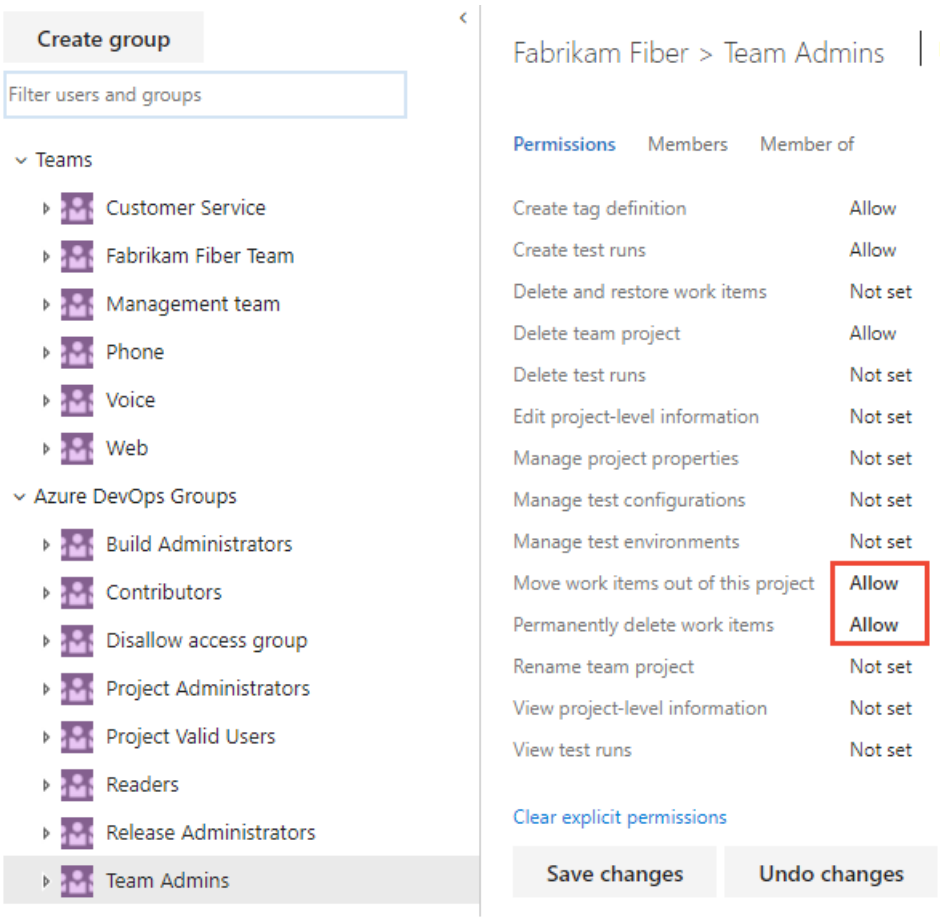
Move or permanently delete work items

By default, Project Administrators and Contributors can change the work item type and delete work items by moving them to the **Recycle Bin**. Only Project Administrators can permanently delete work items and test artifacts. Project admins can grant permissions to other team members as needed.

For example, as a project admin you can grant a user, team group, or other group you've created to have these permissions. Open the Security page for the project and choose the user or group you want to grant permissions. To learn how to access project-level **Security**, see Change project-level permissions.

> **NOTE**
> The **Move work items out of this project** permission requires the project uses the Inherited process model.

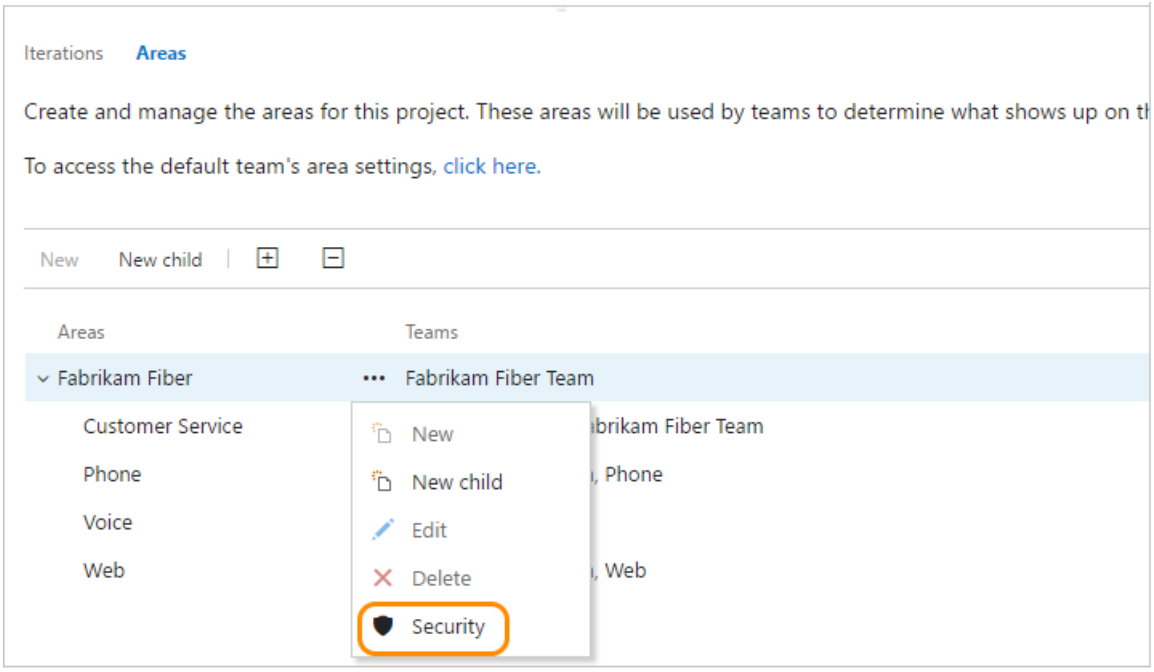
In this example, we grant members assigned to the team administrator role, who belong to the Team Admin groups, permissions to move work items to another project and to permanently delete work items.



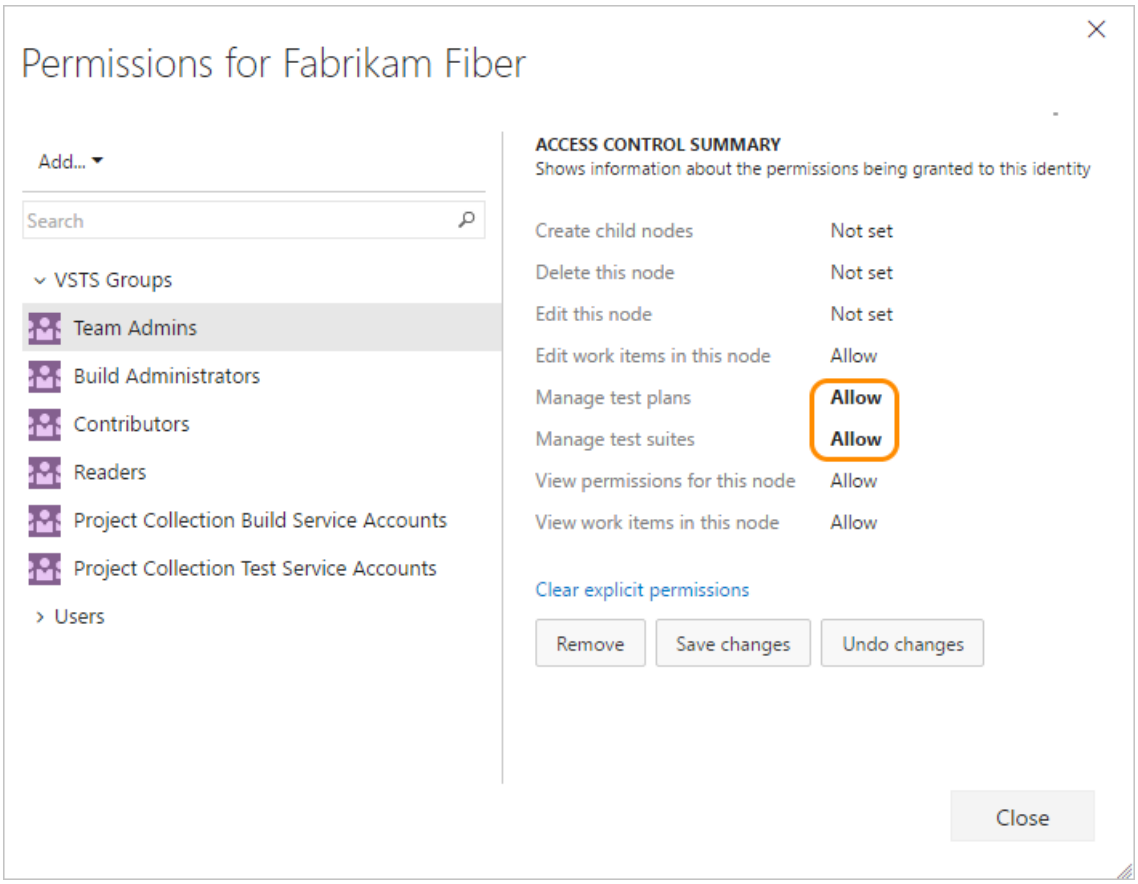
Manage test plans and test suites

In addition to the project-level permissions set in the previous section, team members need permissions to manage test artifacts which are set for an area path.

Open the **Security** page for area paths and choose the user or group you want to grant permissions.



Set the permissions for **Manage test plans** and **Manage test suites** to **Allow**.



To have full access to the Test feature set, your access level must be set to Basic + Test Plans. Users with Basic access and with permissions to permanently delete work items and manage test artifacts can only delete orphaned test cases.

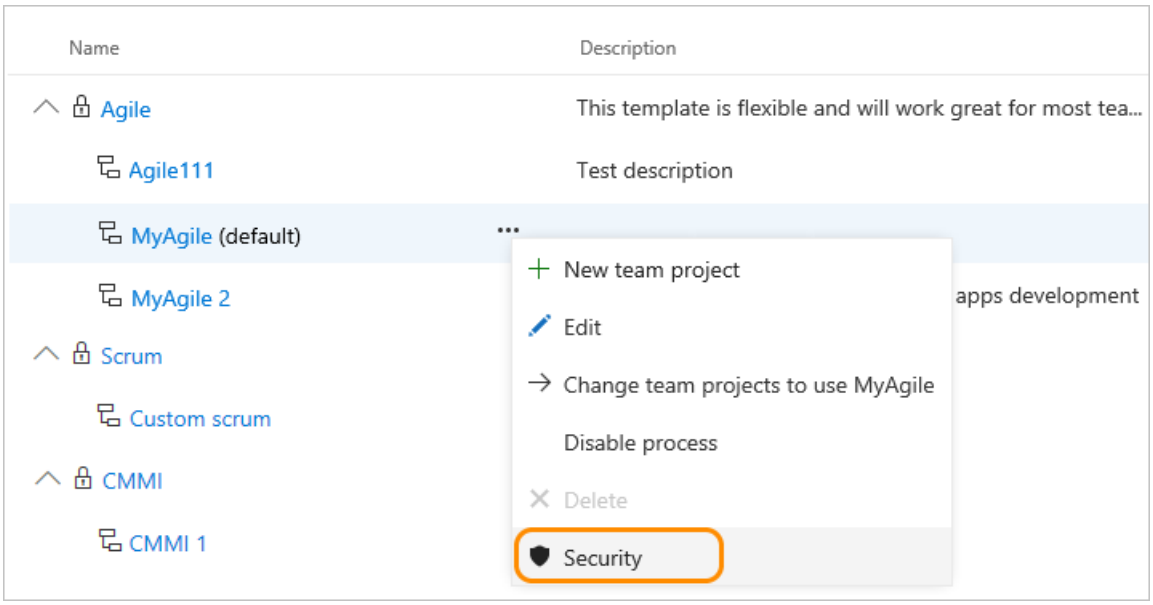
Customize an inherited process

By default, only Project Collection Administrators can create and edit processes. However, these admins can grant permissions to other team members by explicitly setting the **Create process**, **Delete process**, or **Edit process** permissions at the collection level for a specific user.

To customize a process, you need to grant **Edit process** permissions to a user account for the specific process.

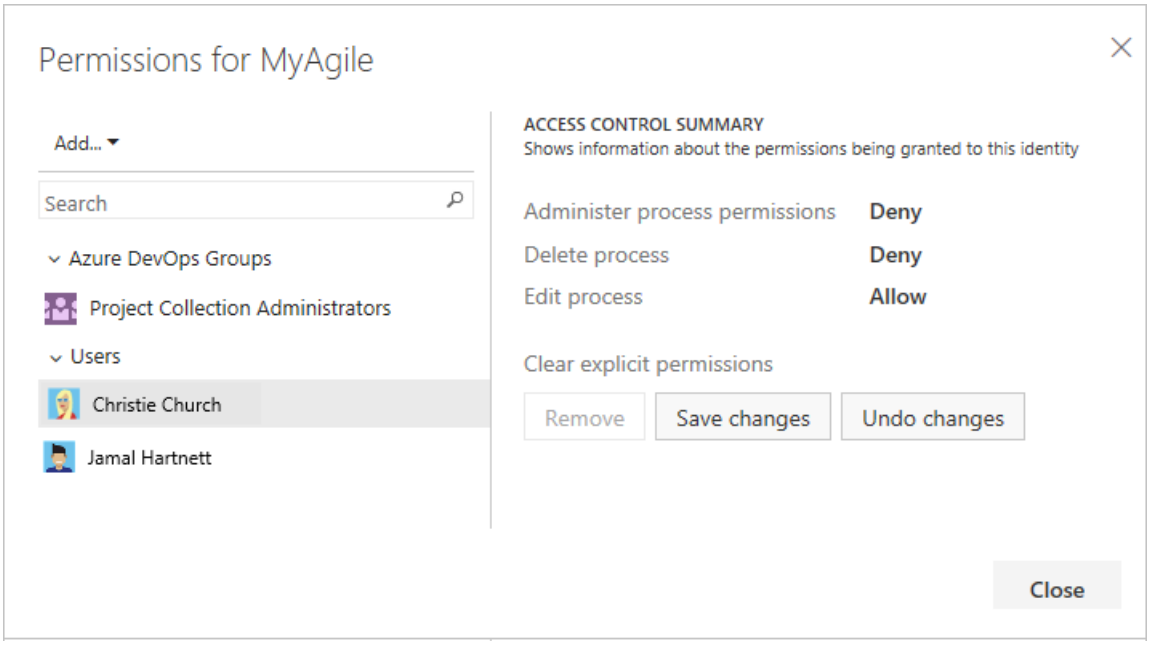
> **NOTE**
> Users added to the **Project-Scoped Users** group won't be able to access Process settings if the **Limit user visibility and collaboration to specific projects** preview feature is enabled for the organization. To learn more, see Manage your organization, Limit user visibility for projects and more.

1. Open the ... context menu for the inherited process and choose **Security**. To open this page, see Customize a project using an inherited process.



2. Add the account name of the person you want to grant permissions to, set the permissions to **Allow** that you want them to have, and then choose **Save changes**.

   Here we add Christie Church and allow her to edit the process.



NOTE

Each process is a securable unit and has individual access control lists (ACLs) that govern creating, editing, and deleting inherited processes. At the collection level, project collection administrators can choose which processes can be inherited from and by whom. When you create a new inherited process, the process creator as well as project collection administrators have full control of the process and can also set individual ACLs for other users and groups to edit and delete the process.

Additional options for restricting access to work items

See Restrict access, Restrict modification of work items based on a user or group for additional options for customizing work item types to support restrictions.

Related articles

- Grant or restrict access
- Rules and rule evaluation
- Set permissions on queries and query folders
- Permissions and access for work tracking
- Permissions and groups reference
- Delete and restore work items
- Delete test artifacts
- Work item form IndexDB caching issues

Grant or restrict access using permissions

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can grant or restrict access to resources that you manage in Azure DevOps. You may want to open up or close down access to a select set of features and for a select set of users. While the built-in security groups provide a standard set of permission assignments, you may need additional security requirements not met by these assignments.

If you're new to administrating permissions and groups, review Get started with permissions, access, and security groupsto learn about permission states and inheritance.

In this article you learn how to do the following tasks:

- Recommended method for granting and restricting permissions
- Delegate tasks by assigning select permissions to specific roles
- Limit visibility to organization information
- Limit people picker to project users and groups
- Restrict access to view or modify objects
- Restrict modification of work items based on a user or group

- Recommended method for granting and restricting permissions
- Delegate tasks by assigning select permissions to specific roles
- Restrict access to view or modify objects
- Restrict modification of work items based on a user or group

> **TIP**
> Because you set many permissions at an object-level, such as repositories and area paths, how you structure your project determines the areas you can open up or close down.

Recommended method for granting and restricting permissions

For maintenance purposes, we recommend you use either the built-in security groups or custom security groups to manage permissions.

You can't change the permission settings for the Project Administrators group or the Project Collection Administrators group, which is by design. However, for all other groups, you can change the permissions.

If you manage a small number of users, then you may find changing individual permissions a valid option. However, custom security groups allow you to better track roles and permissions assigned to those roles.

Delegate tasks to specific roles

As an administrator or account owner, it's a good idea to delegate administrative tasks to those team members who lead or manage an area. Several of the main built-in roles that come with default permissions and role assignments are:

- Readers
- Contributors

- Team Administrator (role)
- Project Administrators
- Project Collection Administrators

For a summary of permissions for the above roles, see Default permissions and access, or for the Project Collection Administrators, see Change project collection-level permissions.

To delegate tasks to other members within your organization, consider creating a custom security group and then granting permissions as indicated in the following table.

Role	Tasks to perform	Permissions to set to Allow
Development lead (Git)	Manage branch policies	Edit policies, Force push, and Manage permissions See Set branch permissions.
Development lead (TFVC)	Manage repository and branches	Administer labels, Manage branch, and Manage permissions See Set TFVC repository permissions.
Software architect (Git)	Manage repositories	Create repositories, Force push, and Manage permissions See Set Git repository permissions
Team administrators	Add area paths for their team Add shared queries for their team	Create child nodes, Delete this node, Edit this node See Create child nodes, modify work items under an area path Contribute, Delete, Manage permissions (for a query folder), See Set query permissions.
Contributors	Add shared queries under a query folder, Contribute to dashboards	Contribute, Delete (for a query folder), See Set query permissions View, Edit, and Manage dashboards, See Set dashboard permissions.
Project or product manager	Add area paths, iteration paths, and shared queries Delete and restore work items, Move work items out of this project, Permanently delete work items	Edit project-level information, See Change project-level permissions.
Process template manager (Inheritance process model)	Work tracking customization	

Administer process permissions, Create new projects, Create process, Delete field from account, Delete process, Delete project, Edit process
See Change project collection-level permissions.

Process template manager (Hosted XML process model)

Work tracking customization

Edit collection-level information, See Change project collection-level permissions.

Project management (On-premises XML process model)

Work tracking customization

Edit project-level information, See Change project-level permissions.

Permissions manager

Manage permissions for a project, account, or collection

For a project, Edit project-level information
For an account or collection, Edit instance-level (or collection-level) information
To understand the scope of these permissions, see Permission lookup guide. To request a change in permissions, See Request an increase in permission levels.

You can also grant permissions to manage permissions for the following objects:

- Set Git repository permissions
- Manage Git branch permissions
- Set TFVC repository permissions
- Administer build and release permissions
- Manage Wiki permissions.

Limit visibility to organization and project information

By default, users added to an organization can view all organization and project information and settings. To restrict access to only those projects that you add users to, you can enable the **Limit user visibility and collaboration to specific projects** preview feature for the organization. To enable this feature, see Manage or enable features.

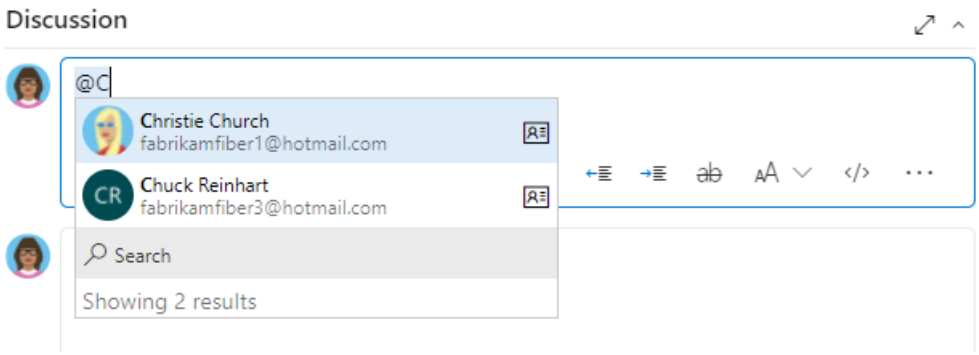
With this feature enabled, users added to the **Project-Scoped Users** group can't view most **Organization settings** and can only connect to those projects to which they've been added.

Limit people picker to project users and groups

For organizations that manage their users and groups using Azure Active Directory (Azure AD), people pickers provide support for searching all users and groups added to Azure AD, not just those added to a project. people pickers support the following Azure DevOps functions:

- Selection of a user identity from a work tracking identity field such as **Assigned To**
- Selection of a user or group using **@mention** in a work item discussion or rich-text field, a pull request discussion, commit comments, or changeset or shelveset comments
- Selection of a user or group using **@mention** from a wiki page

As shown in the following image, you simply start typing into a people picker box until you find a match to a user name or security group.



Users and groups who are added to the **Project-Scoped Users** group can only see and select users and groups in the project they are connected to from a people picker. To scope people pickers for all project members, see Manage your organization, Limit identity search and selection.

Restrict access to view or modify objects

Azure DevOps is designed to enable all valid users to view all objects defined in the system. You can restrict access to resources by setting the permission state to **Deny**. You can set permissions for members that belong to a custom security group or for an individual user. To learn more about how to set these types of permissions, see Request an increase in permission levels.

Area to restrict

Permissions to set to Deny

View or contribute to a repository

View, Contribute
See Set Git repository permissions or Set TFVC repository permissions.

View, create, or modify work items within an area path

Edit work items in this node, View work items in this node
See Set permissions and access for work tracking, Modify work items under an area path.

View or update select build and release pipelines

Edit build pipeline, View build pipeline
Edit release pipeline, View release pipeline
You set these permissions at the object level. See Set build and release permissions.

Edit a dashboard

View dashboards
See Set dashboard permissions.

Restrict modification of work items or select fields

For examples that illustrate how to restrict modification of work items or select fields, see Sample rule scenarios.

Next steps

Remove user accounts

Related articles

- Troubleshoot permissions
- Rules and rule evaluation
- Default permissions and access
- Permission lookup guide
- Get started with permissions, access, and security groups
- Permissions and groups reference
- Change project-level permissions
- Change project collection-level permissions

Configure and customize Azure Boards

10/4/2022 • 18 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This article provides guidance to configure and customize Azure Boards. You should read this article if you are tasked with administrating a project for several teams and supporting the following business objectives:

- Support portfolio management views
- View calendar views to update status and progress
- Track dependencies across teams or projects
- Track time estimates or actual work completed

> **NOTE**
> This article applies to Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features included in this article, such as Rollup, Analytics, and some portfolio planning tools, are only available for the cloud at this time.

If you're just getting started as a Project Administrator, see also Get started as an administrator.

What to consider?

When configuring or customizing work tracking tools, you'll want to consider the tools your teams use and how they will use them. Whether your teams follow Scrum, Kanban, or some combination of Scrumban, you can gain the most advantage of the Azure Boards tools by understanding the dependencies they have on configurations and customizations.

The primary items to consider as you structure your project are:

At a project level:

- How many teams you want to define
- How to structure area paths to support portfolio management views
- Field customizations
- Work item type customizations or custom work item types
- Portfolio backlog customizations
- Workflow customizations

At a team level:

- How you'll use your product backlog to plan and prioritize your work
- Whether you'll track bugs as requirements or as tasks, or not use bugs at all
- Whether or not you'll use tasks to track time and capacity
- How you'll use portfolio backlog levels
- How you'll inform upper management of progress, status, and risks

Once you determine how you'll use the work tracking building blocks and tools, you'll want to make any necessary configurations and customizations to support your business and communicate to your teams how they should use the tools.

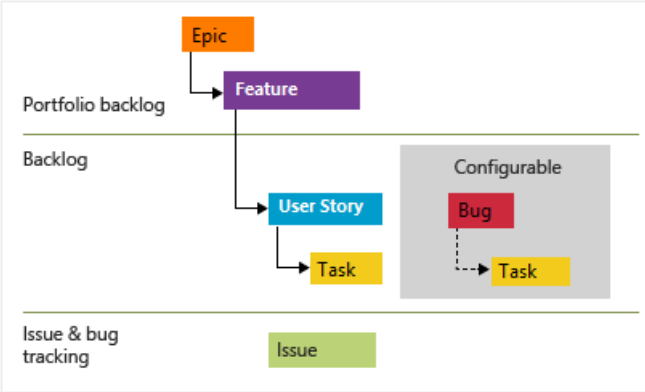
Work item types and portfolio backlogs

The first choice made in work tracking is the process selected when creating a project. For a comparison of each process, see Choose a process. Each process—Agile, Basic, Scrum, and CMMI—supports a set hierarchy of work item types. This hierarchy supports a product backlog and portfolio backlog(s).

The default work item types for each supported process are shown in the following tabs. Backlog work item types correspond to the Requirements category. Tasks correspond to the Task category.

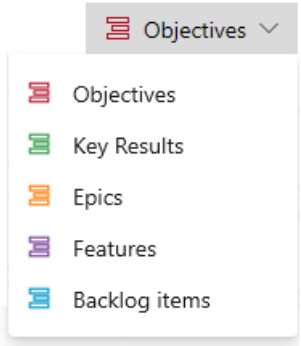
- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

You can add custom work item types at each level, and even add custom portfolio backlogs. Here, for example, is a project that added Objectives and Key Results as custom work item types and corresponding portfolio backlogs to the Scrum process.



Work tracking options and recommended usage

One of the main choices teams have is choosing the work item types they use to track their work. The following table summarizes the main options, recommended usage, and supported tasks and tools.

Work tracking options	Tasks and tools supported
Tasks only	

Not recommended

There is no way to quickly enter new tasks in a backlog nor prioritize a backlog of tasks. In addition, there is no support for calendar views, cross-team views, or portfolio planning

Requirements with child-dependent tasks

Supports Scrum methods

Recommended for teams that follow Scrum methods and want to track time associated with work.

- Quickly define and prioritize backlog items: Product backlog
- Forecast sprints using team velocity: Forecast
- Plan sprints: Backlog Planning tool
- Plan and track capacity: Sprint capacity tool
- Track estimated and remaining work: Taskboard
- Monitor sprint burndown based on remaining work such as hours or days: Sprint burndown
- Conduct daily scrums, update and monitor task status: Sprint Taskboard
- Estimate work: Define Story Points, Effort, or Size
- View progress bars, counts, or sums of rollup on tasks: Rollup
- Track dependencies across teams and projects: Delivery Plans

Many teams start out using Scrum methods to track and plan their work using the tools available through the Sprints hub. The Sprints tools support estimating and tracking remaining work and use of capacity planning. If you don't plan on using these tools, then adding child-dependent tasks is optional. Developers might add them simply as a checklist of items they need to complete a user story or backlog requirement.

Requirements only, such as user stories (Agile), issues (Basic), product backlog items (Scrum), requirements (CMMI)

Supports Kanban and Scrumban methods

Recommended for teams that follow Kanban or Scrumban methods, estimate work using Story Points, Effort, or Size, and don't track time associated with work.

- Quickly define and prioritize backlog items: Product backlog
- Plan sprints: Backlog Planning tool
- Estimate work: Define Story Points, Effort, or Size
- Forecast sprints using team velocity: Forecast
- Monitor sprint burndown based on requirement estimates: Sprint burndown
- Update requirement status: Kanban board
- Track dependencies across teams and projects: Delivery Plans

Requirements grouped under portfolio work item types, such as epics and features

Supports calendar views, cross-team views, and portfolio planning

Recommended for organizations with several teams that want to view rollups and calendar views associated with multiple teams, and take advantage of all portfolio planning tools.

- Quickly define and prioritize portfolio items: Portfolio backlogs
- Quickly define child user stories of portfolio items: Portfolio checklists
- Map work items to features and epics: Mapping tool
- View cross-team progress calendar view: Delivery plans
- View calendar view of all team features: Feature Timeline
- View calendar view of a specific epic: Epic Roadmap
- View progress bars, counts, or sums of rollup on child items: Rollup

- Track dependencies across teams and projects: Delivery Plans

Configure and customization options

The following table indicates the areas you can configure and customize and the tools impacted by those customizations. Each area is customized either at the Organization, Project, or Team level as noted, or a combination of two. For a description of the Standard tools, Analytics tools, and Portfolio planning tools, see What is Azure Boards, In-context reports: Work tracking, and Plans (Agile at scale).

Configure or customize

Standard tools

Analytics

Portfolio planning tools

Area paths, project configuration, and team subscriptions (Project, Team)

- Boards>All tools
- Backlogs>All tools
- Sprints>All tools

- Cumulative flow diagram
- Velocity
- Burndown trend

- Delivery plans
- Feature timeline
- Epic Roadmap
- Portfolio plans (Beta)

Iteration paths, project configuration, and team subscription (Project, Team)

- Backlogs>Sprint planning
- Sprints>Sprint backlogs
- Sprints>Sprint capacity
- Sprints>Taskboard

- Velocity
- Burndown trend

- Delivery plans
- Feature timeline
- Epic Roadmap
- Portfolio plans (Beta)

Show bugs on backlogs & boards (Team)
Custom work item types, Product backlog (Process)
Custom work item types, Taskboard (Process)

- Boards>Product board
- Backlogs>Product backlog
- Backlogs> Mapping tool
- Sprints>Sprint backlogs
- Sprints>Taskboard

- Velocity

Custom work item types, Portfolio backlog (Process)
Additional portfolio backlogs (Process)

- Boards>Portfolio boards
- Backlogs>Portfolio backlogs
- Backlogs> Mapping tool

- Velocity

Custom workflow (Process)

- Boards>Product board
- Boards>Portfolio boards
- Sprints>Taskboard

- Cumulative flow diagram

Custom field (Process)

- Boards>Product board
- Boards>Portfolio boards

- Rollup progress bars, sum, or count

Area paths, product teams, and portfolio management

Area paths are used to group work items by product, feature, or business areas and to support teams responsible for work assigned to those areas. You can define a hierarchical set of area paths or a flat set. Typically, you define a hierarchical set of area paths when you want to support a business hierarchy that wants to track progress of several teams.

Area paths and hierarchical grouping

The two main ways to group work items are by area path and by parenting them under a portfolio work item type as described early in this article. The two are not mutually exclusive. Note the distinctions between the two usages:

- Area paths assigned to a team determine what work items appear in a team view: product backlog, portfolio backlog, delivery plans, or other portfolio planning tool
- Grouping work items under a parent feature or epic determine what rollup views are supported and how work appears in a portfolio planning tool

You can also assign tags to work items to group them for query and filter purposes. So when you structure your teams and projects, you want to make sure you understand how you'll use these grouping tools to support your business needs. Your choices impact the use of portfolio planning tools.

Area path-dependent tools

To perform the following tasks, you must define area paths:

- Query and chart work items based on Area Path
- Assign work to more than one team
- Work with management and feature teams
- Filter a backlog, query, board, or plan using Area Paths

TIP

You can define your area path structure and assign area paths to teams. Or, you can add a team and create the area path with the team name at that time. If teams are fully independent, create a flat set of area paths. However, if you want to create a hierarchy of teams, then you'll want to create a tree-hierarchy of area paths. To learn more, see Configure a hierarchy of teams.

To use the following tools, teams must subscribe to area paths:

- Boards> all tools
- Backlogs> all tools
- Sprints> all tools
- Delivery plans, calendar view of team deliverables
- Rollup
- Analytics> all

- Boards> all tools
- Backlogs> all tools
- Sprints> all tools
- Delivery plans, calendar view of team deliverables

Area paths and team assignments

A default team and default area path are defined for each project. For small teams, a single team is sufficient to begin planning and tracking work. As organizations grow, however, it's useful to add teams to support their ability to manage their backlog and sprints.

Here is an example of area paths and their assignment to teams, which support portfolio management views for the Account Management and Service Delivery teams.

Areas	Teams
Fabrikam Fiber	
Account Management	Account Management
Customer Profile	Account Management, Customer Profile
Shopping Cart	Account Management, Shopping Cart
Service Delivery	Service Delivery
Email	Email, Service Delivery
Internet	Internet, Service Delivery
Service Status	Service Delivery, Service Status
TV	Service Delivery, TV
Voice	Service Delivery, Voice

- You create hierarchical area paths to support sub categories of features and product areas
- To provide portfolio views, you assign two or more area paths and include sub-areas to a portfolio management team
- Area paths assigned to a team determine what work items are filtered in a team view: product backlog, portfolio backlog, delivery plans, or other portfolio planning tool
- Grouping work items under a parent feature or epic determine what rollup views are supported and how work appears in a calendar view such as Feature Timeline and Epic Roadmap

Prior to adding teams, we recommend you read the following articles:

- Portfolio management
- About area paths
- About teams and Agile tools
- Agile culture.

Recommendations:

- Consider what views upper management may want to view and how to best support them
- Consider how you want to use rollup both for a team and portfolio management
- Define epics and scenarios for large initiatives that will take two or more sprints to complete
- Define requirements for work that can be accomplished in a single sprint and can be assigned to a single individual
- Define tasks to track more granular details or when you want to track time spent working

> **TIP**
> - Work items can only be assigned to a single individual. So when defining work items, consider how many work items are needed to assign the work to those individuals who will be tasked to complete the work.
> - Choose the Node Name field as a column option to view the leaf area node in a backlog list or board card.
> - Don't create parent-child links among work items of the same type, such as story-story, bug-bug, task-task.

Most Azure Boards tools support a filtered view of work items based on area path and/or iteration path. Additional filters can also be applied based on keyword, assignment, work item type, and more.

Treat bugs as requirements or tasks

Each team can choose how they want to manage bugs. Some teams like to track bugs along with requirements on the backlog. Other teams like to track bugs as tasks performed in support of a requirement. The bugs then appear on their taskboard.

If you use the Scrum process, your default setup is to track bugs along with product backlog items (PBIs). If you work in a project based on the Agile or CMMI processes, bugs don't automatically appear on your backlog.

Talk with your team to determine how they want to manage bugs. Then change your team settings accordingly.

> **TIP**
> After you refresh a backlog or board and you don't see bugs where you expect them, review How backlogs and boards display hierarchical (nested) items. Only leaf nodes of nested items appear on sprint taskboards.

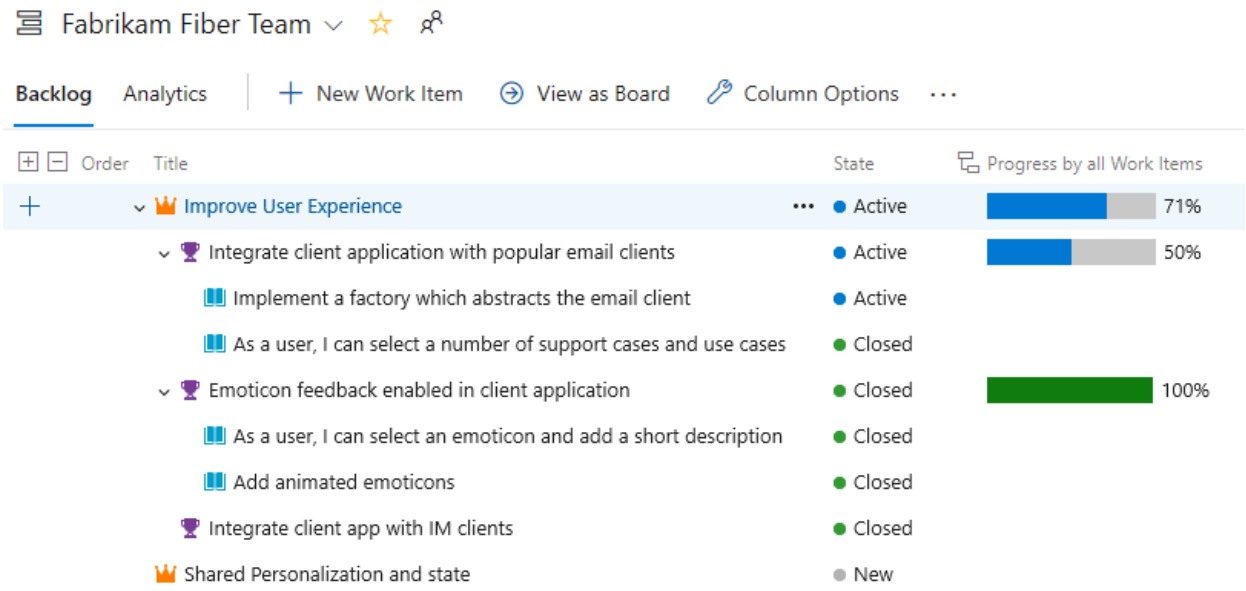
Add system work item types to a backlog

If you want to track issues or impediments along with your requirements or in a portfolio backlog, you can by adding them to your custom Inherited process. For details, see Customize your backlogs or boards (Inheritance process).

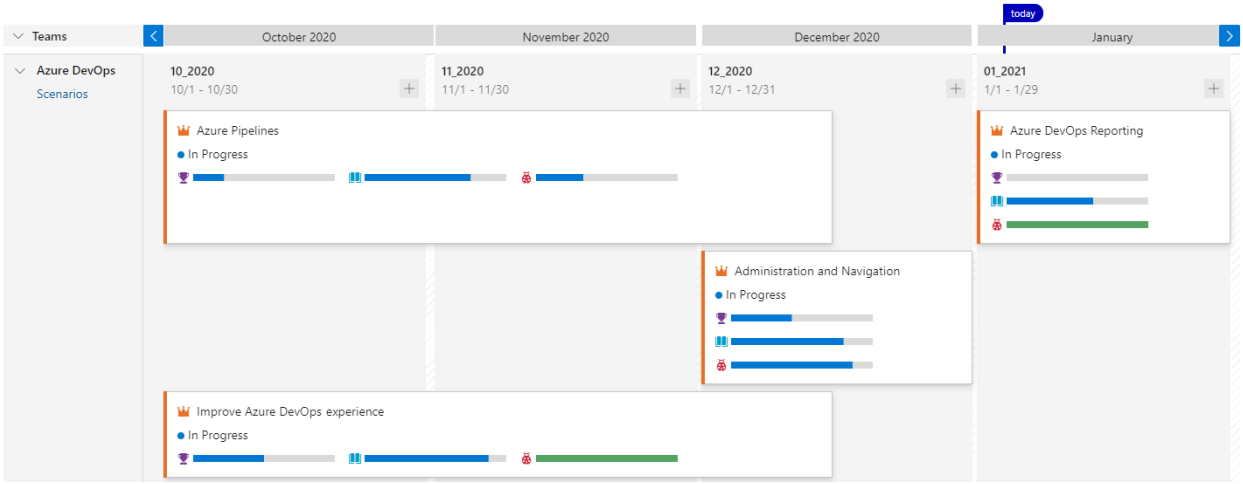
Rollup, hierarchy, and portfolio management

Rollup columns allow you to view progress bars or totals of numeric fields or descendant items within a hierarchy. Descendant items correspond to all child items within the hierarchy. You can add one or more rollup columns to a product or portfolio backlog.

Here we show **Progress by all Work Items** which displays progress bars for ascendant work items based on the percentage of descendant items that have been closed.



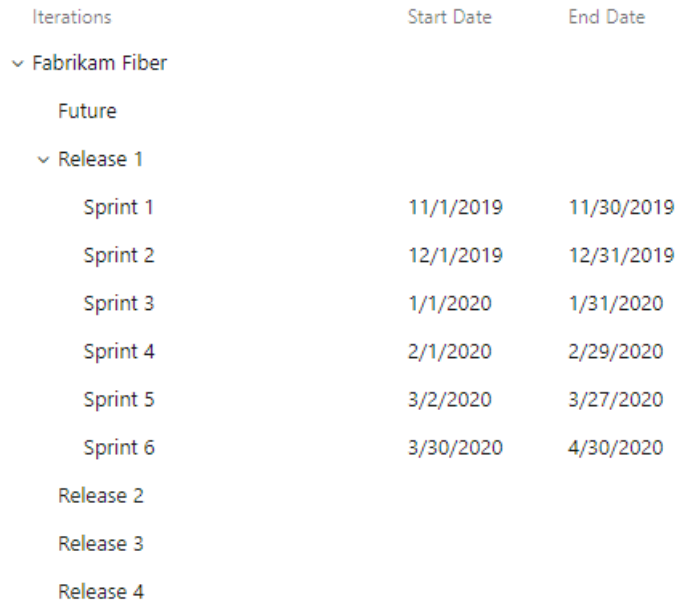
In addition, the new Delivery Plans supports rollup views of epics, features, and other custom portfolio backlogs.



Iteration paths, sprints, releases, and versioning

Iteration paths support Scrum and Scrumban processes where work is assigned to a set time period. Iteration paths allow you to group work into sprints, milestones, or other event-specific or time-related period. Each iteration or sprint corresponds to a regular time interval referred to as a sprint cadence. Typical sprint cadences are two weeks, three weeks, or a month long. To learn more about Iteration paths, see About area and iteration paths.

Iteration paths can be a simple flat list, or grouped under release milestones as shown in the following image.



Iterations	Start Date	End Date
Fabrikam Fiber		
Future		
Release 1		
Sprint 1	11/1/2019	11/30/2019
Sprint 2	12/1/2019	12/31/2019
Sprint 3	1/1/2020	1/31/2020
Sprint 4	2/1/2020	2/29/2020
Sprint 5	3/2/2020	3/27/2020
Sprint 6	3/30/2020	4/30/2020
Release 2		
Release 3		
Release 4		

> **NOTE**
> While Iteration Paths don't impact Kanban board tools, you can use Iteration Paths as a filter on boards. To learn more, see Filter your Kanban board.

Define iteration paths and assign them to teams when you want to use the following tools:

- Assign work items to sprints using the Planning pane
- Query and chart work items based on Iteration Path
- Forecast your product backlog
- Sprints> all tools
- Delivery Plans, a calendar view of team deliverables
- Velocity chart and Sprint burndown chart

- Assign work to sprints using the Planning pane
- Query and chart work items based on Iteration Path
- Forecast your product backlog
- Sprints> all tools
- Delivery Plans, a calendar view of team deliverables
- Velocity chart and Sprint burndown chart

- Assign work to sprints using the Planning pane
- Query and chart work items based on Iteration Path
- Forecast your product backlog
- Sprints> all tools
- Delivery Plans, a calendar view of team deliverables
- Velocity chart

> **TIP**
> If a team hasn't subscribed or selected an iteration path, that iteration path won't appear in a team view or tool.

Time tracking

Most organizations following Scrum processes use time estimates for Sprint capacity planning. Azure Boards tools fully support tracking time for this purpose. The main field used is the task Remaining Work field, which typically zeros out at the end of the sprint.

However, some organizations require time tracking to support other purposes, such as for billing or maintaining time allocation records. Time values for estimated work and completed work are of interest. The Agile and CMMI processes provide these fields—Original Estimate, Completed Work, Remaining Work—for use in tracking time. You can use them for that purpose. However, Azure Boards provides limited native support for time tracking. Instead, you may want to consider using a Marketplace extension to support your additional time tracking requirements.

> **NOTE**
> The Original Estimate, Completed Work, Remaining Work fields were designed to support integration with Microsoft Project. Integration support with Microsoft Project is deprecated for Azure DevOps Server 2019 and later versions, including the cloud service.

Process changes that impact all teams

Any change made to a process applied to a project impacts all teams in that project. Many changes won't cause much disruption to the teams they support. However a few do, and those are described in this section.

Custom fields

Adding custom fields to a work item type doesn't impact any specific tool. The fields simply appear in the corresponding work items. If you add a custom numeric field, however, you can use it to support rollup on backlogs as well as the following reporting tools:

- In-context Velocity report and dashboard widget
- In-context Sprint Burndown report and dashboard widget
- Dashboard Burndown and Burnup widget

> **NOTE**
> All default and custom fields are shared across all projects in a collection or organization. There is a limit of 1024 fields that you can define for a process.

Custom work item types

When you make one or more of the following customizations, you impact the team tools as indicated.

- Add a custom work item type to the Requirement category:
- Add a custom work item type to the Task category:
- Add a custom work item type to the Epic or Feature category:
- Add a custom portfolio backlog level

When you add a custom work item type (WIT) to one of the following work tracking categories, you impact team tools in the following ways:

- Task category:
  - Child work items of the new WIT appear on the product backlog
  - Work items based on the new WIT appear on the sprint backlogs and Taskboards
- Requirement category:
  - Work items based on the new WIT appear on the product backlog and Kanban board
  - Each team must configure the Kanban board to support the new WIT

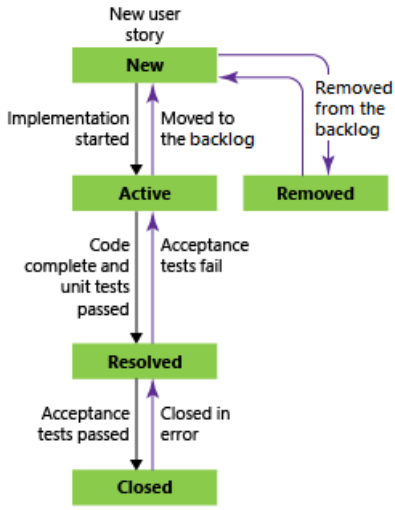
- Epic or Feature category:
  - Work items based on the new WIT appear on the corresponding portfolio backlogs and Kanban boards
  - Each team must configure the Kanban boards to support the new WIT
  - The new WITs may not appear on one or more of the portfolio planning tools

Custom workflow

Each process supports a default workflow. This workflow defines the default columns that appear on the Kanban boards and sprint Taskboards.

- Agile process
- Basic process
- Scrum process
- CMMI process

Workflow states: User Story, Agile process



Sometimes, teams want to track the status of their work that go beyond these default states. Support is provided for this in one of two ways:

- Add custom workflow states to the work item type\
  - This option impacts all teams and requires that they update their Kanban board configuration
- Add columns to a Kanban board
  - This option only impacts the team that adds the columns

Both workflow states and Kanban columns appear in the Cumulative Flow diagram for a team. Individuals can choose which columns appear in the chart.



To learn more, see Cumulative flow diagram.

Who can make changes?

Because process-level, project-level and team-level settings can have a wide impact, changes are restricted to the following people who have the required permissions.

Process-level changes

To create, edit, or manage Inherited processes and apply them to projects, you must be a member of the **Project Collection Administrators** group. Or, you must have the corresponding permissions **Create process**, **Delete process**, **Edit process**, or **Delete a field from organization** set to **Allow**. See Set permissions and access for work tracking, Customize an inherited process.

For additional information, see the following articles:

- About the inheritance process
- Customize a project
- Create and manage a process

Project-level changes

To add Area Paths or Iteration Paths, you must be a member of the **Project Administrators** group.

Or, to add, edit, and manage Area Paths or Iteration Paths under a specific node, you must have been granted one or more of the following permissions set to **Allow**:

- **Create child nodes**
- **Delete this node**
- **Edit this node**
- **View permissions in this node**

For additional information, see the following articles:

- Define area paths & assign to a team
- Define iteration paths (sprints) & assign team iterations

Team-level changes

All team tools can be configured by a team administrator or a member of the **Project Administrators** group.

Team administrators are tasked with performing the following operations:

- Add team members
- Subscribe to area and iteration paths
- Configure backlogs and other common team settings
- Configure Kanban boards
- Manage team notifications

For details on configuring backlogs and boards, see Manage and configure team tools.

Try this next

Get started as an administrator

Related articles

- Azure Boards Configuration and Customization FAQs
- Set up your Backlogs and Boards
- Inherited process model
- Manage and configure team tools

Define area paths and assign to a team

10/4/2022 • 14 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Add area paths to support teams and group work items based on product, feature, or business areas. Once you define area paths at the project level, you assign them to a team under the team configuration. You can also create a hierarchy of area paths to support sub-areas, up to 14 levels deep.

To perform the following tasks, you must define area paths:

- Query and chart work items based on Area Path
- Assign work to more than one team
- Work with management and feature teams
- Filter a backlog, query, board, or plan using Area Paths

> **TIP**
> You can define your area path structure and assign area paths to teams. Or, you can add a team and create the area path with the team name at that time. If teams are fully independent, create a flat set of area paths. However, if you want to create a hierarchy of teams, then you'll want to create a tree-hierarchy of area paths. To learn more, see Configure a hierarchy of teams.

To use the following tools, teams must subscribe to area paths:

- Boards> all tools
- Backlogs> all tools
- Sprints> all tools
- Delivery plans, calendar view of team deliverables
- Rollup
- Analytics> all

- Boards> all tools
- Backlogs> all tools
- Sprints> all tools
- Delivery plans, calendar view of team deliverables

To understand how the system uses area paths, see About area and iteration paths.

Prerequisites

- If you don't have a project yet, create one now.
- Ensure you're a member of the **Project Administrators** group to add an area path under the root node or edit or delete any child node. To acquire these permissions, see Change project-level permissions.
- Have one or more of the following permissions set to **Allow**, to add, edit, and manage area paths under a node:
  - **Create child nodes**

- **Delete this node**
- **Edit this node**
- **View permissions in this node**

  By default, the user who created the project has these permissions already set. For more information, see Set permissions and access for work tracking.

- Ensure you're added as a team administrator or are a member of the **Project Administrators** group to set team area paths.

For naming restrictions on area paths, see About areas and iterations, Naming restrictions.

Get started

Each team has access to a number of Agile tools as described in About teams and Agile tools. Each tool references the team's default area path(s). Most teams choose one area path and several iteration paths to support their work tracking activities. However, to support other scenarios, it's possible for teams to choose several area paths to appear on their backlogs and boards.

New projects contain a single, root area that corresponds to the project name. A team is created with the same project name and the root area path is assigned to that team.

If you're new to managing projects and teams, the most straight forward sequence for configuring your project and teams is as follows:

1. Determine the number and names of area paths that you want to support to categorize your work. At a minimum, add one area path for each team you define. For more information, review About areas and iterations.
2. Determine the number and names of teams you want to support. For more information, review About teams and Agile tools.
3. Open **Project settings > Project configuration** and define the area paths to support steps 1 and 2 at the project level. Follow the steps provided later in this article: Open Project Settings, Project configuration and Add area paths.
4. Define the teams you need to support step 2. For more information, see Add a team, move from one default team to several teams.
5. Open the team configuration and assign the default and additional area path(s) to each team. Follow the steps provided later in this article: Open team settings and Set team default area path(s).
6. Assign the area path of work items to an area path you defined. Use bulk modify to modify several work items at once.

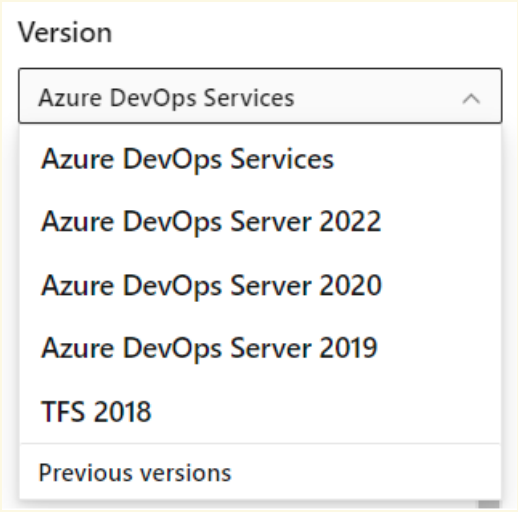
> **NOTE**
> While you can assign the same area path to more than one team, doing so can cause problems if two teams claim ownership over the same set of work items. To learn more, see About boards and Kanban, Limitations of multi-team Kanban board views.

As needed, you can do the following actions at any time:

- Add additional child nodes
- Rename an area path (except the root area path)
- Move a child node under another node
- Delete a child node
- Rename a team

- Change the area path assignments made to a team

IMPORTANT



To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

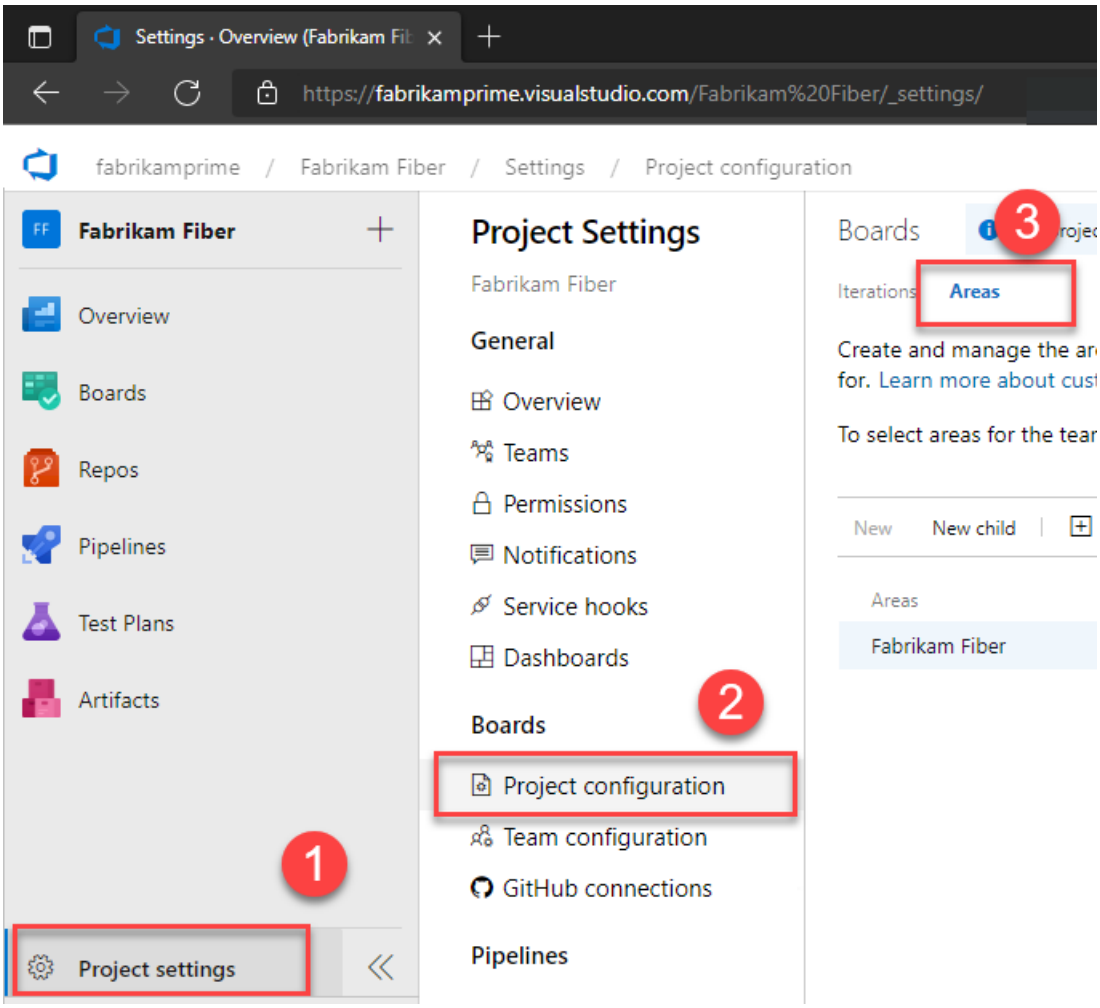
Open Project Settings, list project areas

Define both areas and iterations for a project from the **Project Settings** > **Boards** > **Project configuration**.

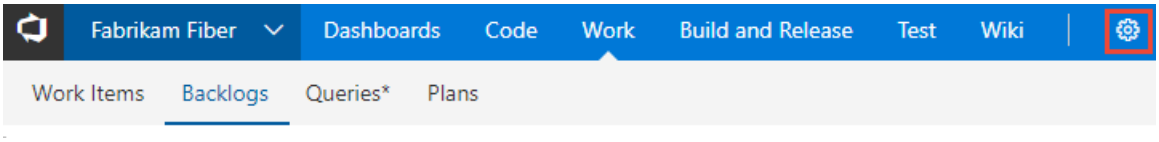
Define both areas and iterations from the **Work** pages of the **Project Settings** context. From the user context, open the admin context by choosing **Settings** .

- Browser
- Azure DevOps CLI

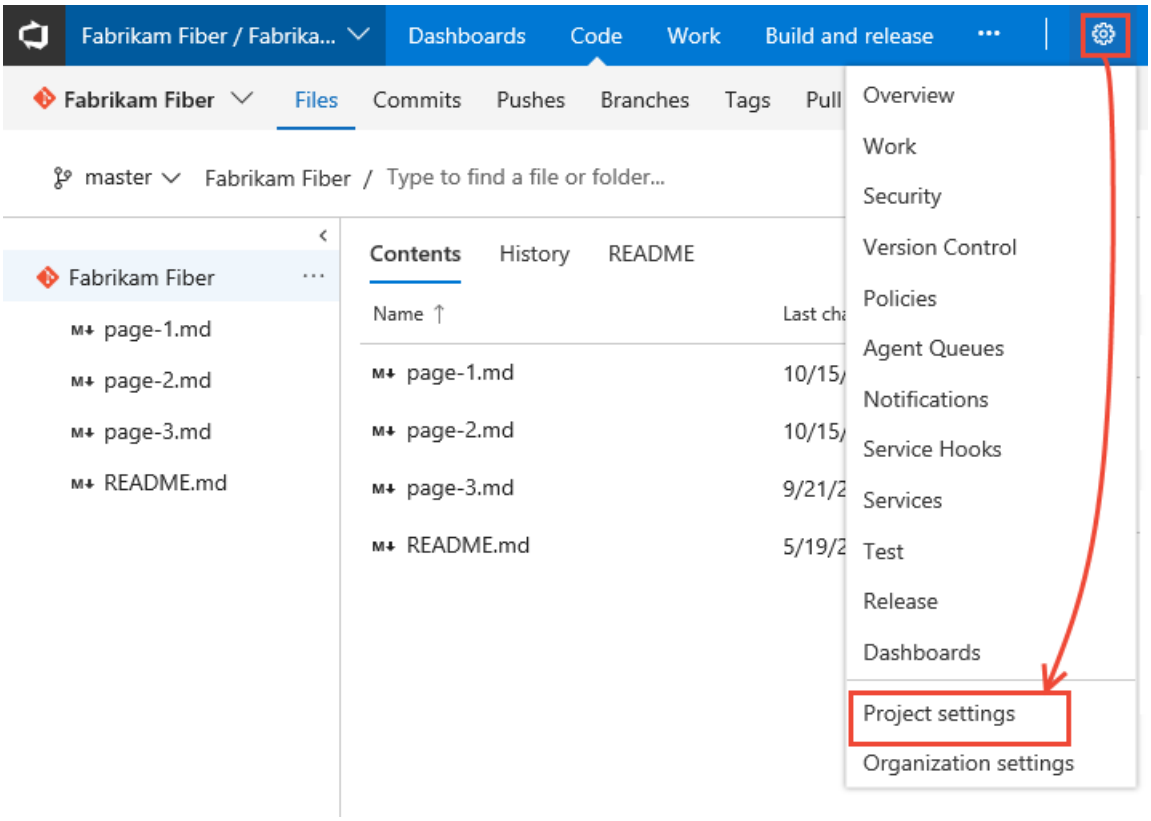
- From your web portal, choose (1) **Project settings**, choose (2) **Project configuration** and then (3) **Areas**.



1. From the web portal for the project, choose **Project settings** .



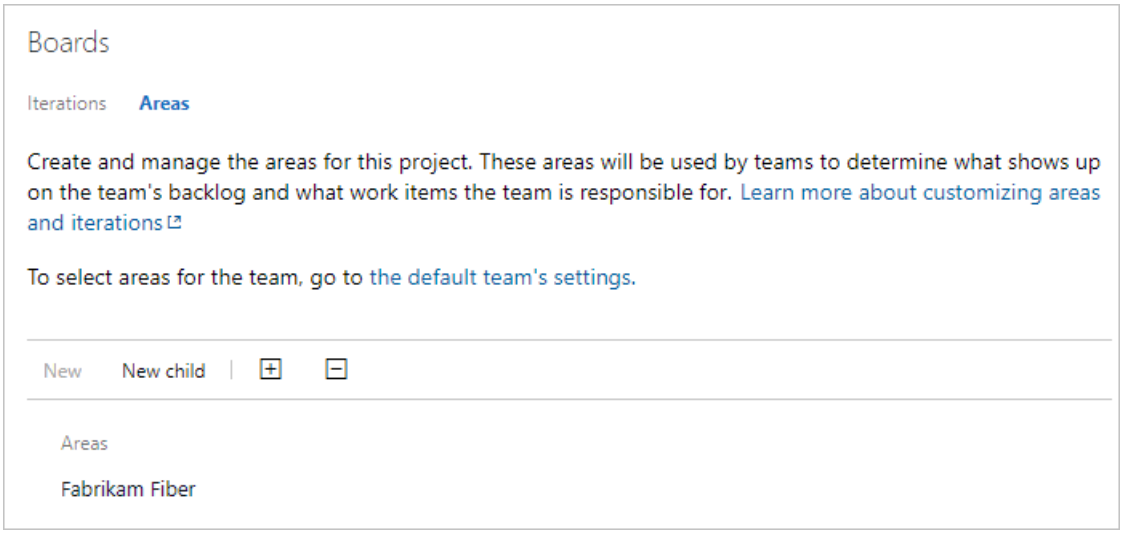
If you're currently working from a team context, then hover over **Settings** and choose **Project settings**.



2. Choose **Work**.

Add an area path

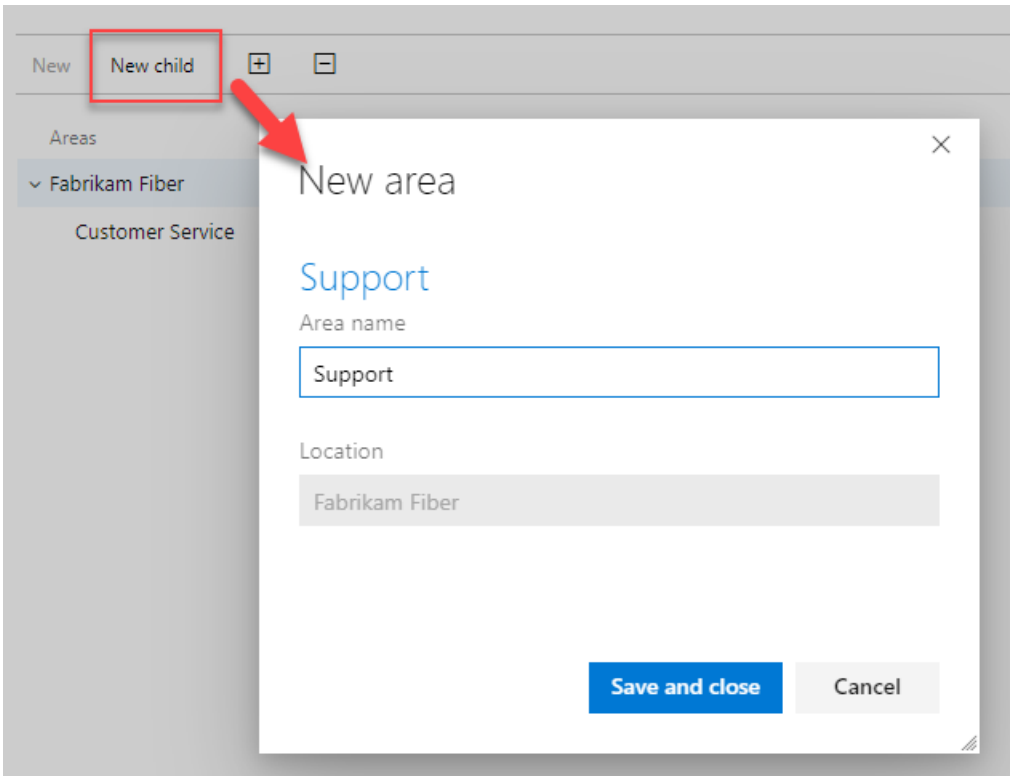
Only one area is defined, by default. Add area paths under the root area path for the project, or as a child to another area path.



- Browser
- Azure DevOps CLI

- To add a child node, highlight the area path and then choose **New child**. Optionally, you can select ··· for the area path and choose **New child**.

  Enter a name (255 characters or less) for the node. For additional name restrictions, see About areas and iterations, Naming restrictions.

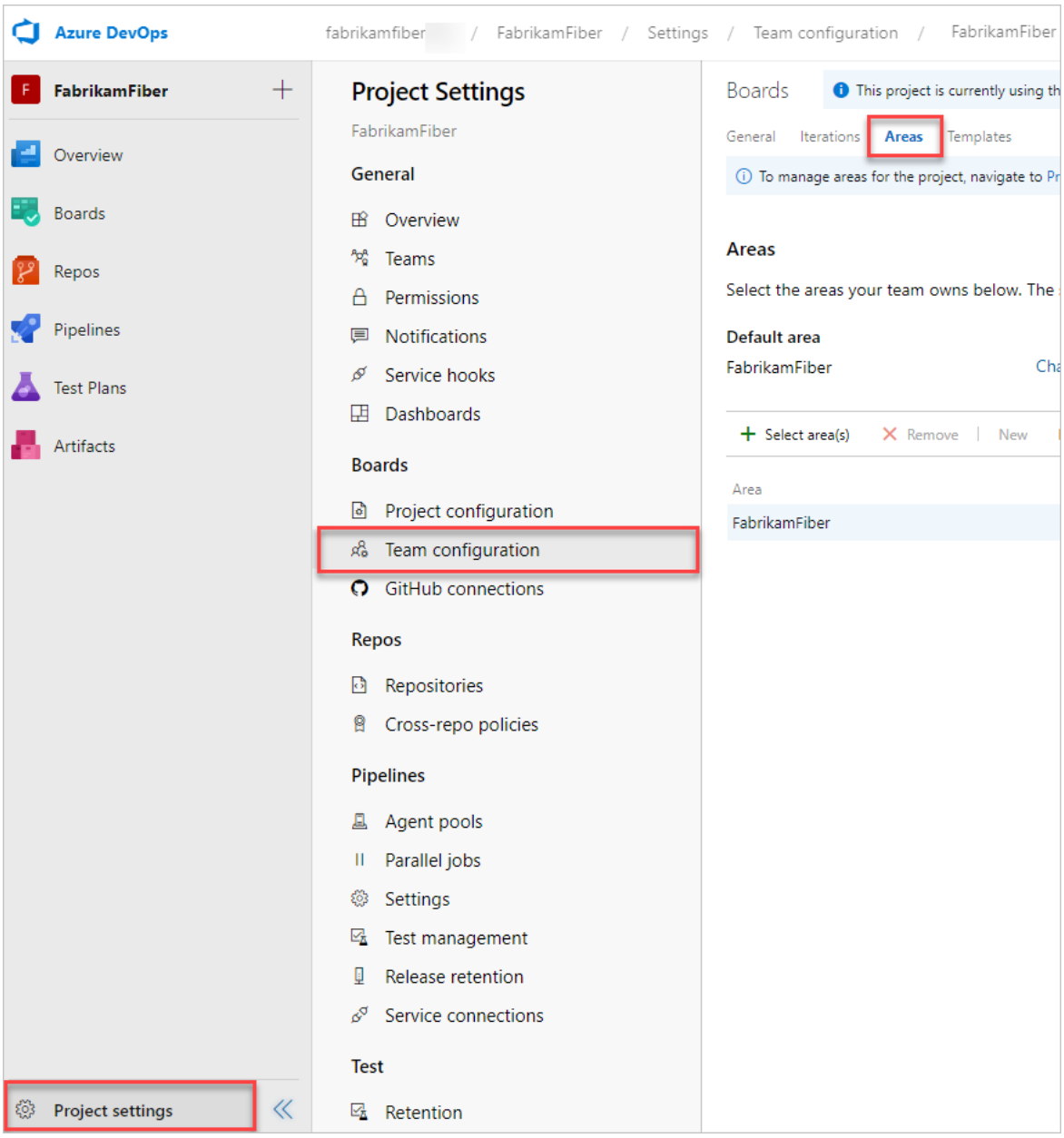


Open team settings, list team area paths

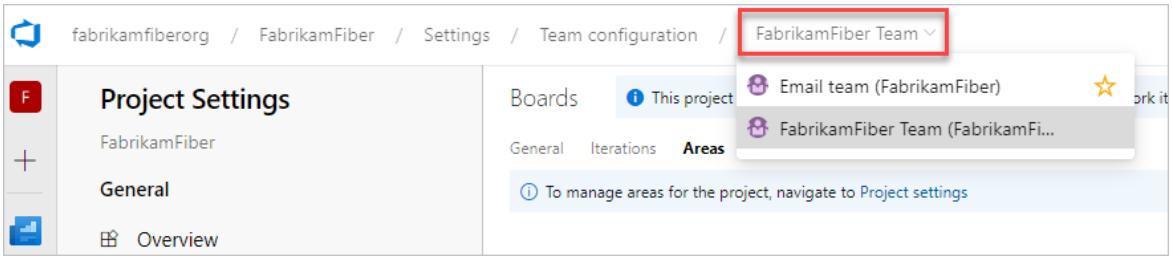
You set team defaults from team settings. If you're not a team administrator, get added as one. Only team or project administrators can change team settings.

- Browser
- Azure DevOps CLI

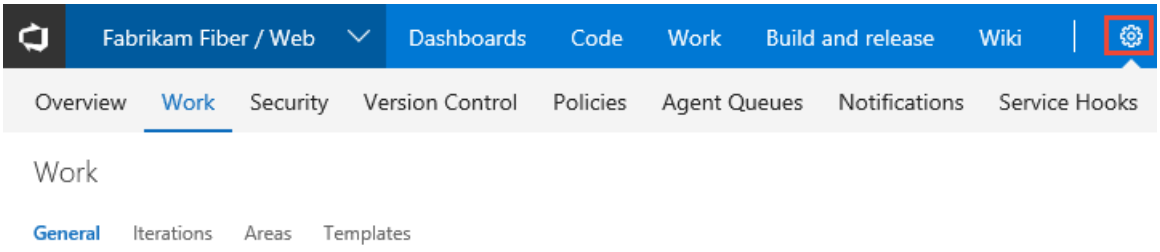
1. Open your project, and then select **Project settings** > **Team configuration** > **Areas**.



2. If you need to switch the team context, use the team selector within the breadcrumbs.



You open team settings from the upper navigation bar. Select the team you want and then choose **Team settings** . For more information about switching your team focus, see Switch project, repository, team

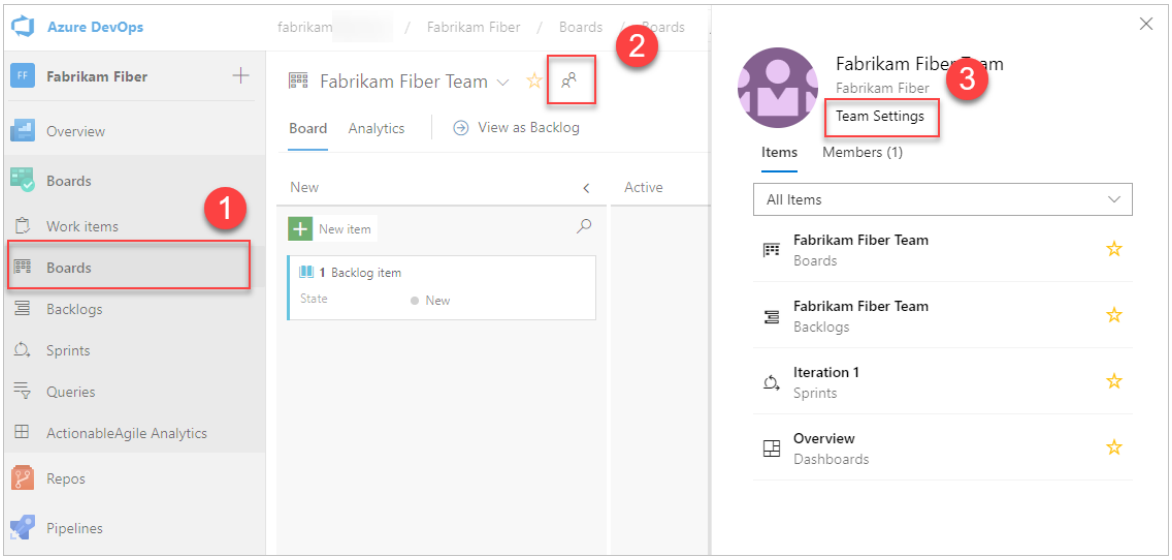


Open team settings from the team profile

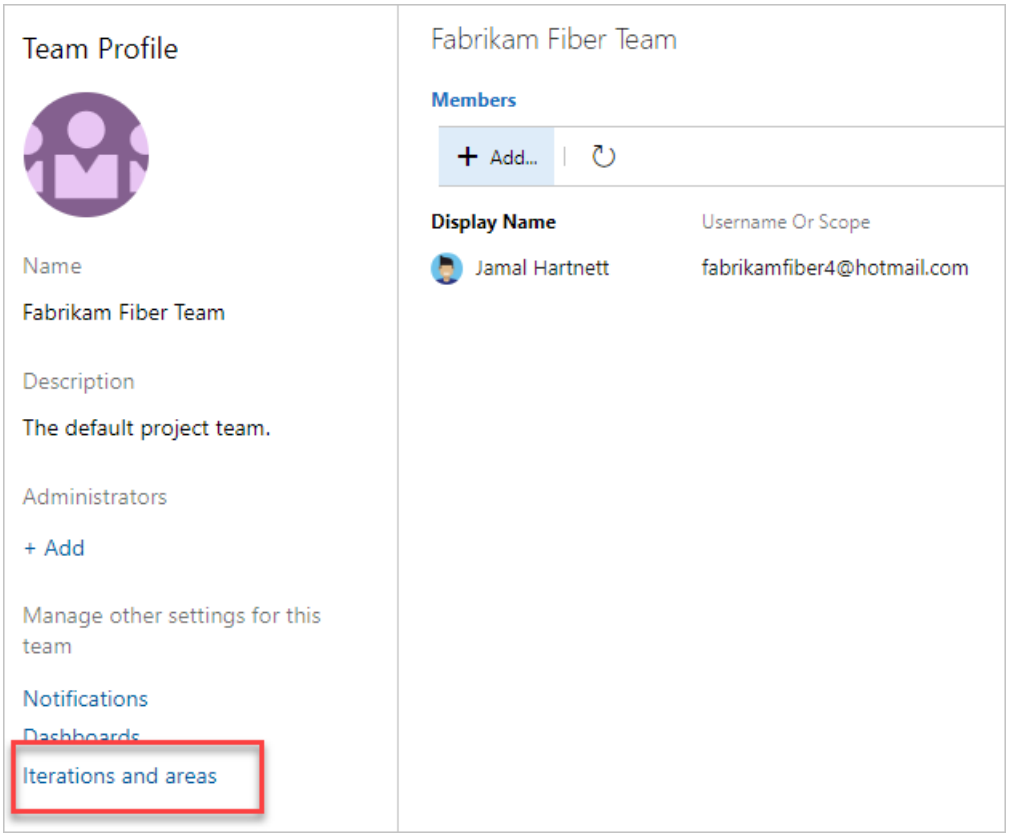
You define both areas and iterations from **Project Settings** > **Team configuration**. You can quickly navigate to it from a team work tracking backlog, board, or dashboard.

1. Open a backlog or board for a team and choose **Team profile** > **Team Settings**.

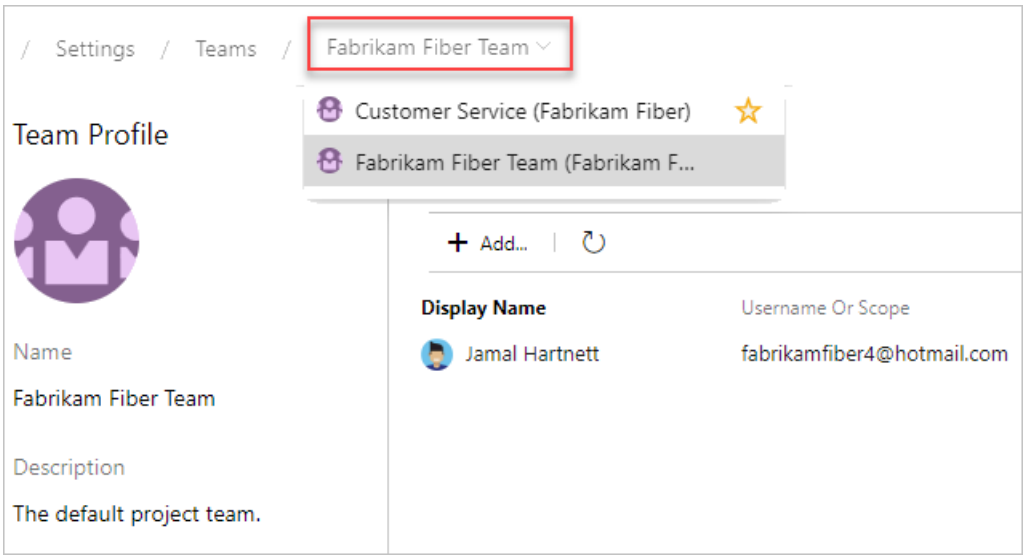
   Here we open the Board for the Fabrikam Fiber team and from there the team profile.



2. Choose **Iterations and areas**.



3. If you need to switch the team context, use the team selector within the breadcrumbs.



Set team area path(s)

All work items that are assigned to a team area path appear on the backlogs and boards for that team. You can select one or more area paths and optionally include their subarea paths. Choose to include subarea paths when you want to support rollup views of work done across several teams or areas.

> **NOTE**
> Teams can be assigned a maximum of 300 **Area Paths**. To learn more, see Work tracking, process, and project limits.

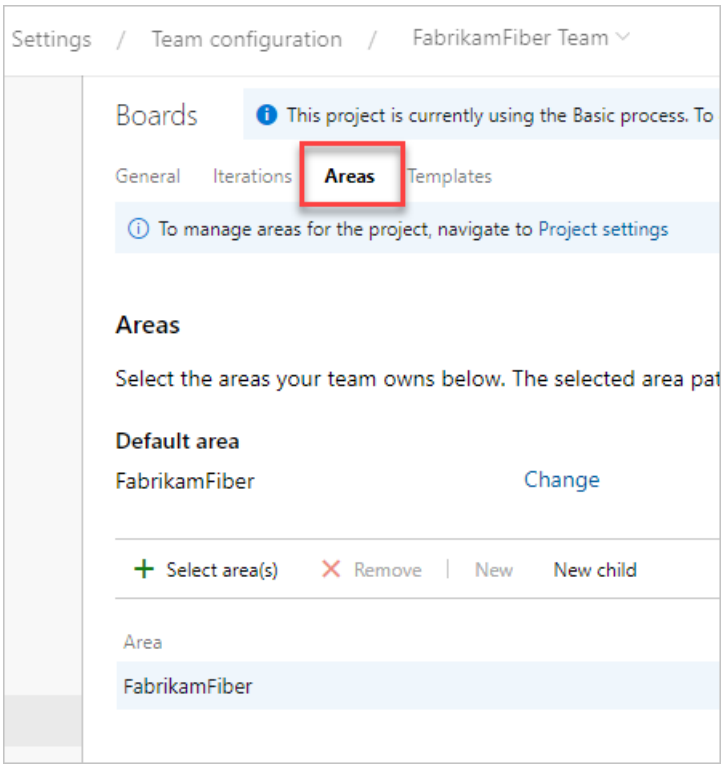
The default area path determines the default area path assigned to work items that are created from the team context.

> **IMPORTANT**
> Work items that appear on more than one team's Kanban board can yield query results that don't meet your expectations. Because each team can customize the Kanban board columns and swimlanes, the values assigned to work items which appear on different boards may not be the same. The primary work around for this issue is to maintain single ownership of work items by team area path.

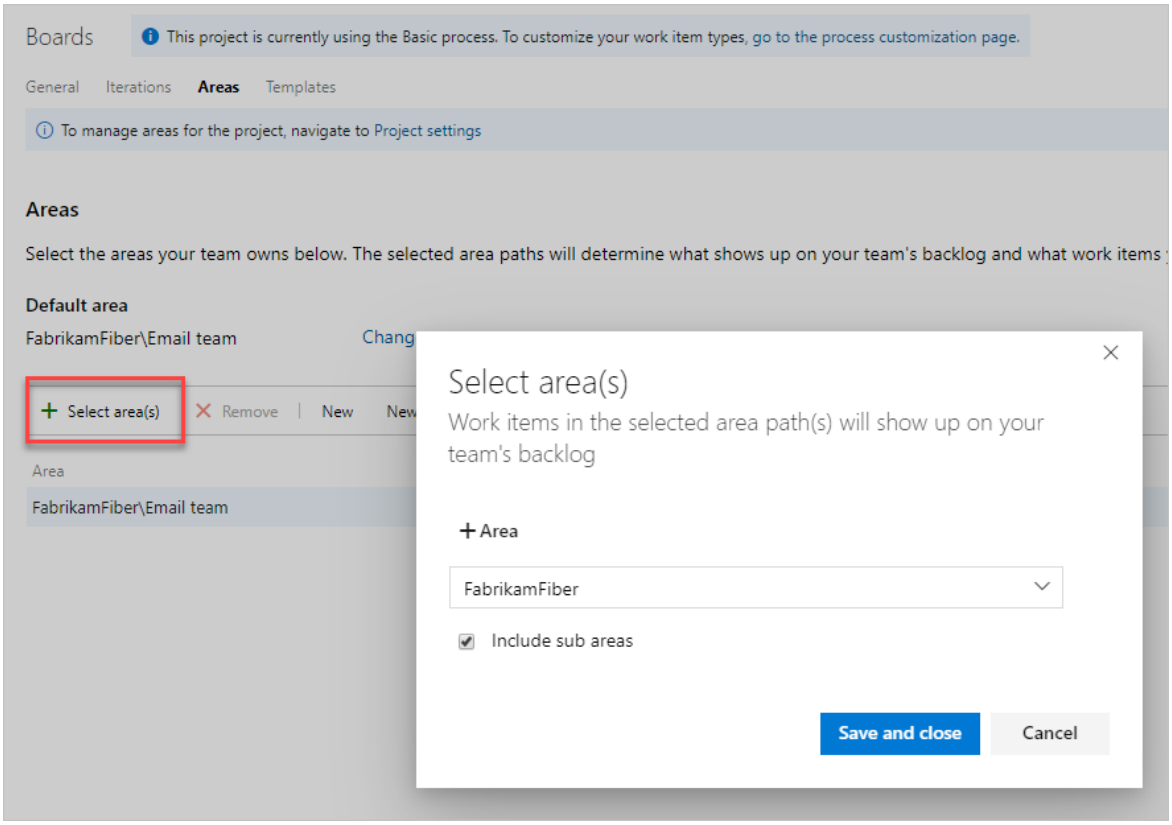
- Browser
- Azure DevOps CLI

1. Open **Areas** for the team context.

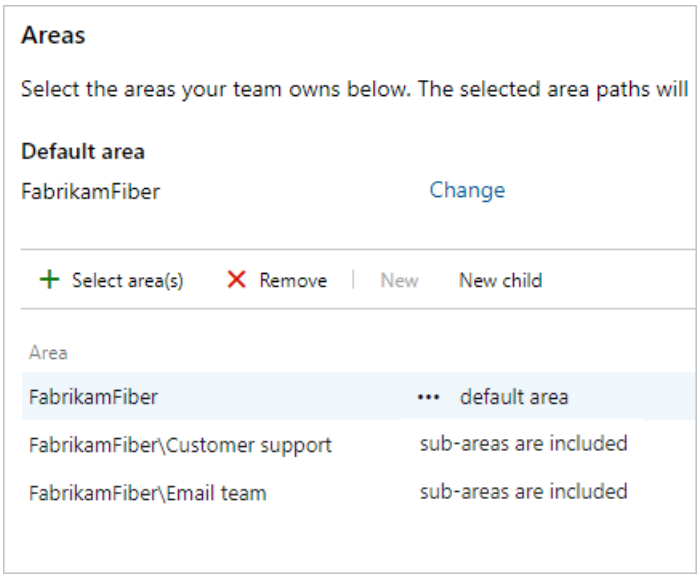
   Here, we show the Areas for the Fabrikam Fiber team.



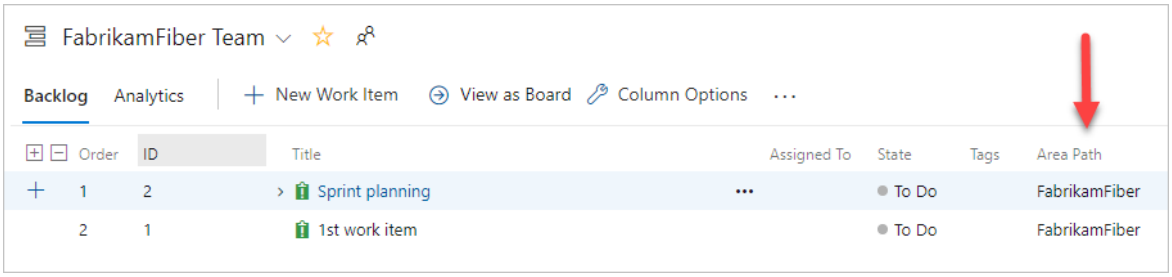
2. Choose the area path(s) to be active for each team.

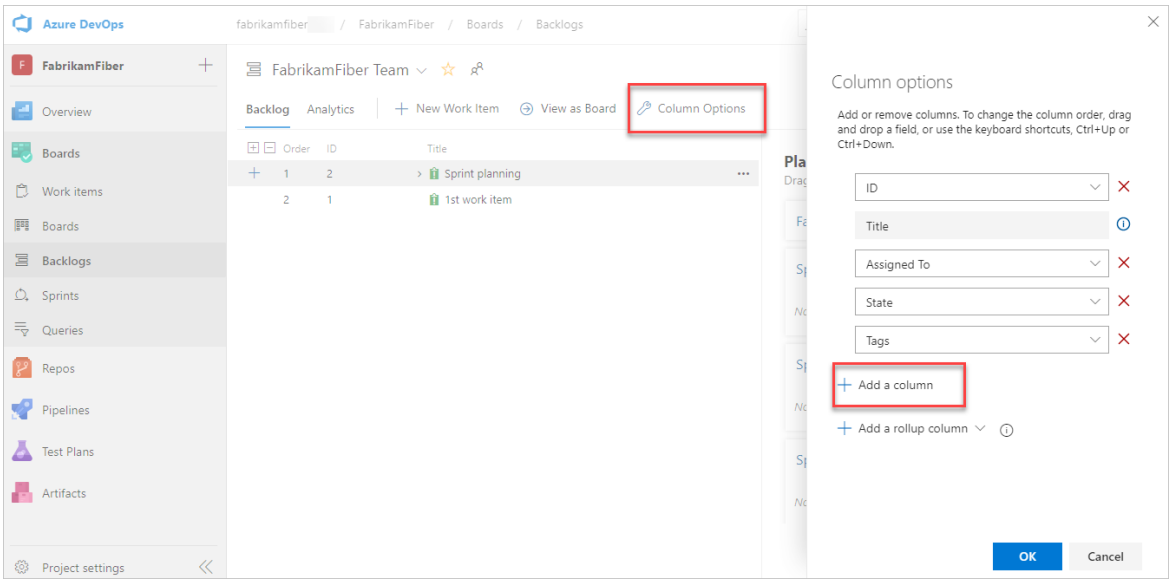


In this instance, we choose to activate the subarea paths for the project. The management team can now track progress across all teams.



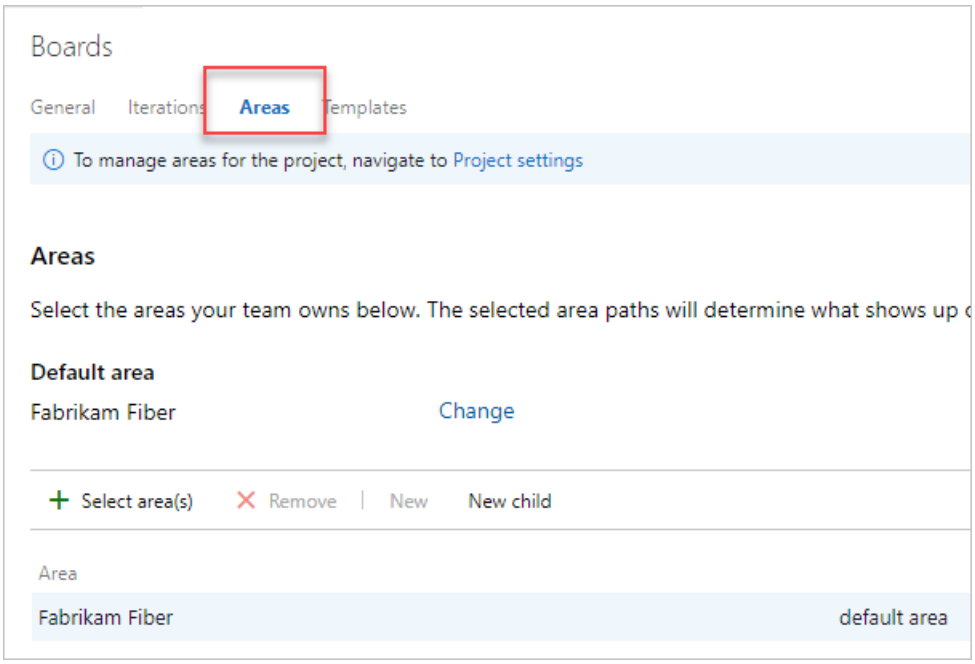
3. When you've finished, refresh the product backlog page for the team, and you'll see those work items assigned to the team. Add area path to the columns shown to see the assignments made to work items.



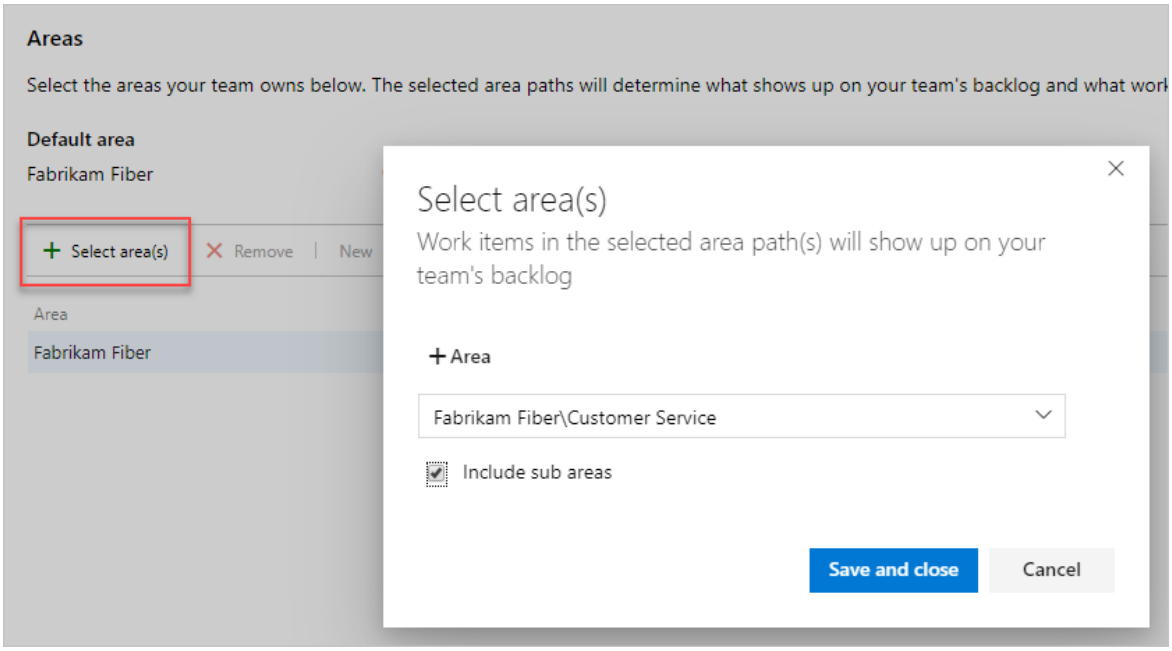


1. Open **Areas** for the team context.

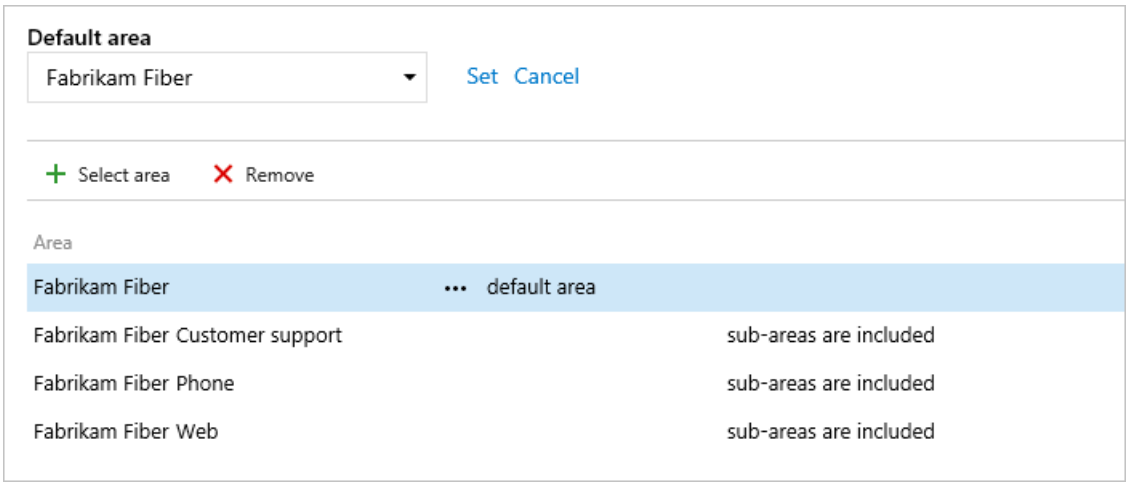
   Here, we show the Areas for the Fabrikam Fiber team.



2. Choose the area path(s) to be active for each team.

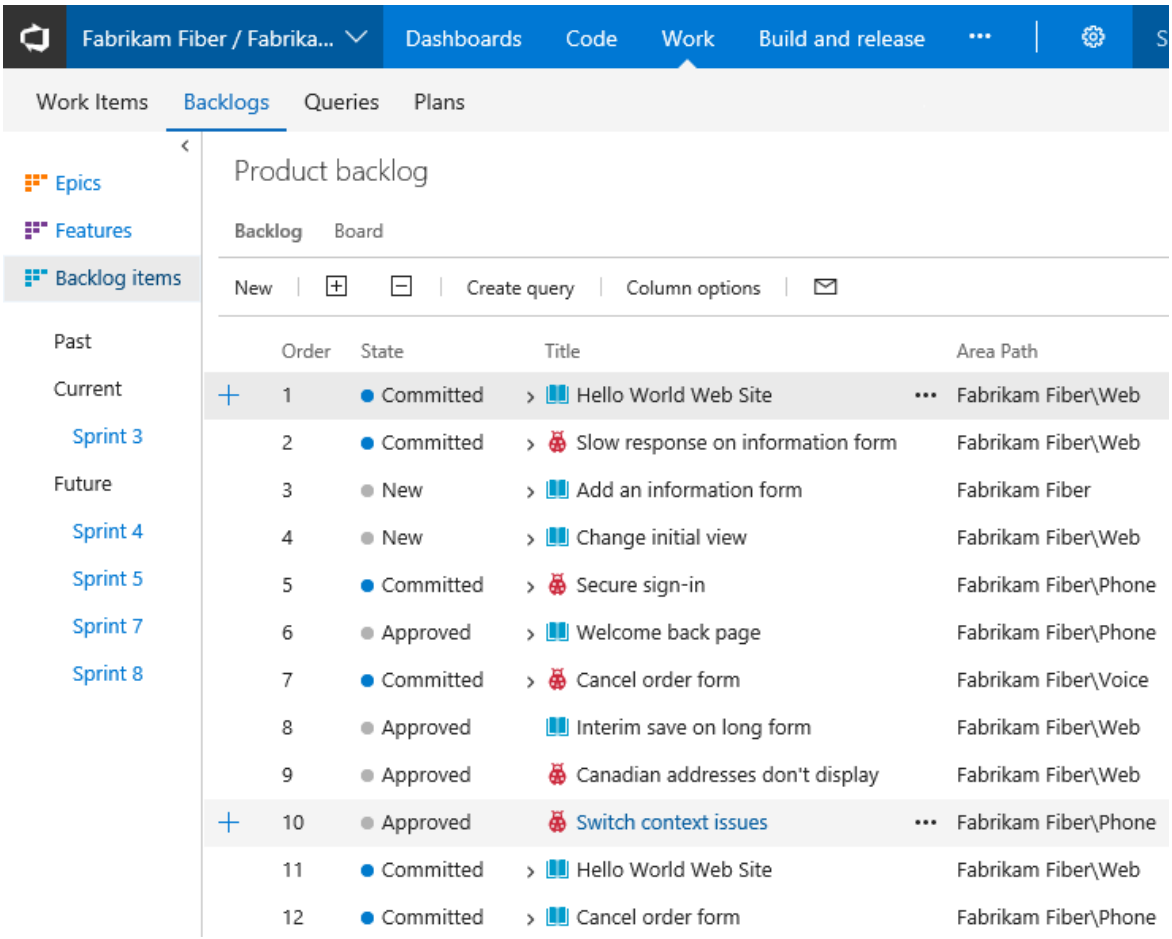


In this instance, we choose to activate all three subarea paths for the project. The management team can now track progress across all three teams.



3. When you've finished, refresh the product backlog page for the team, and you'll see those work items

assigned to the team. Add area path to the columns shown to see the assignments made to work items.

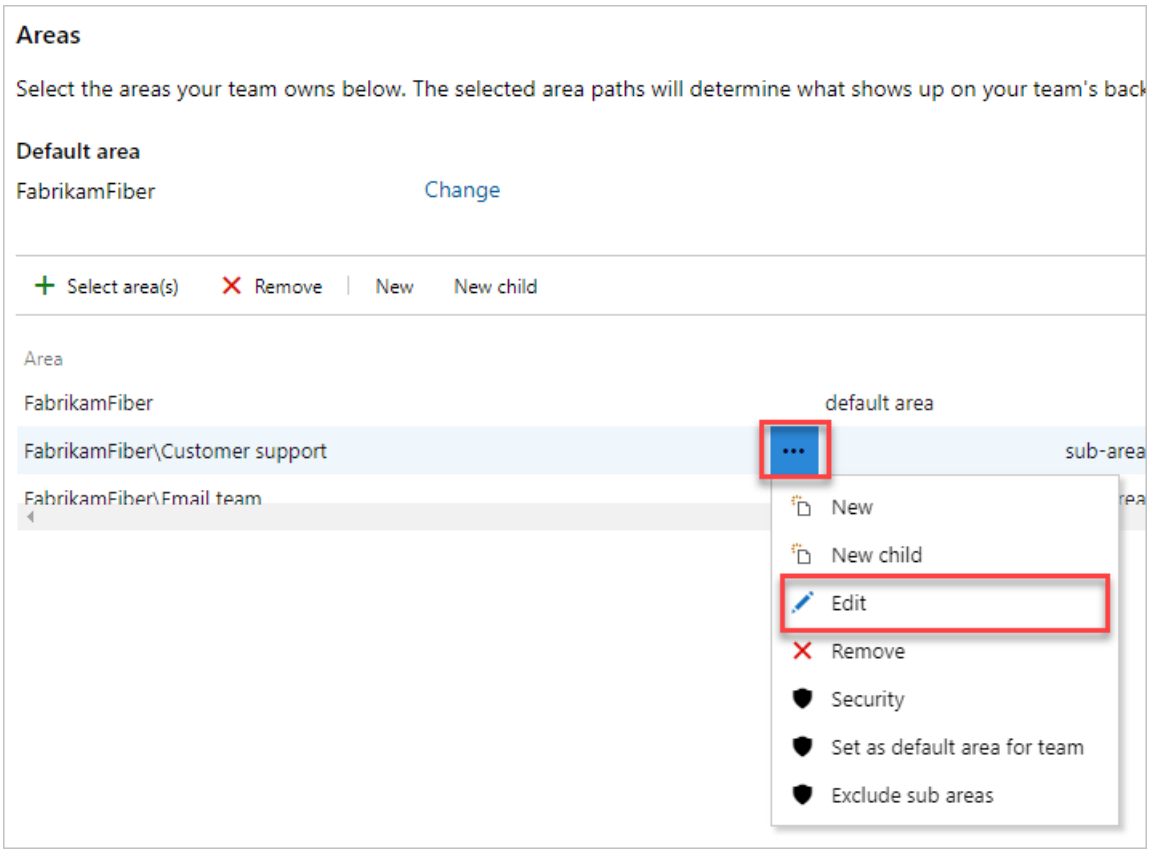


Rename, move, or delete an area path

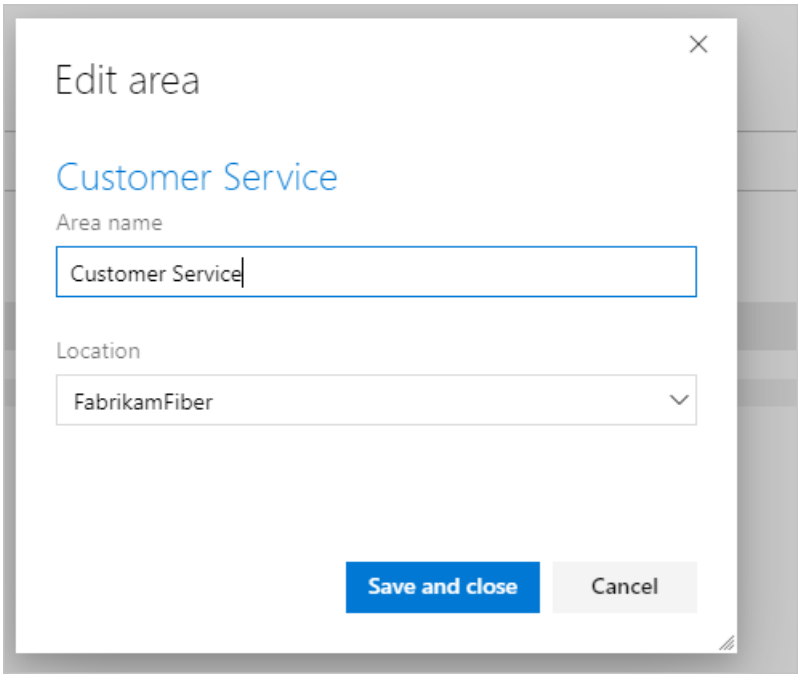
The system automatically updates work items and queries that reference your updated area path.

- Browser
- Azure DevOps CLI

1. To rename an area or iteration path, choose **Actions** ··· for the node, and then select **Edit**.



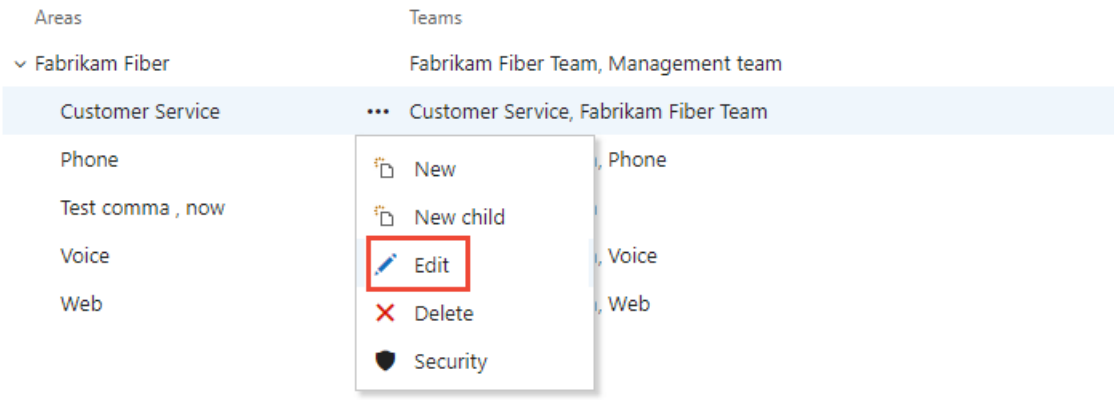
2. In the dialog that opens, enter the new name.



3. To move the node within the hierarchy, change the Location field.
4. To delete a node, choose the **Delete** option from the actions menu.

> **NOTE**
> When you delete an area node or change the Location field for a node, the system automatically updates the existing work items with the node that you enter at the deletion prompt.

1. To rename an area or iteration path, choose **Actions** ··· for the node, and then select **Edit**.



2. In the dialog that opens, enter the new name.

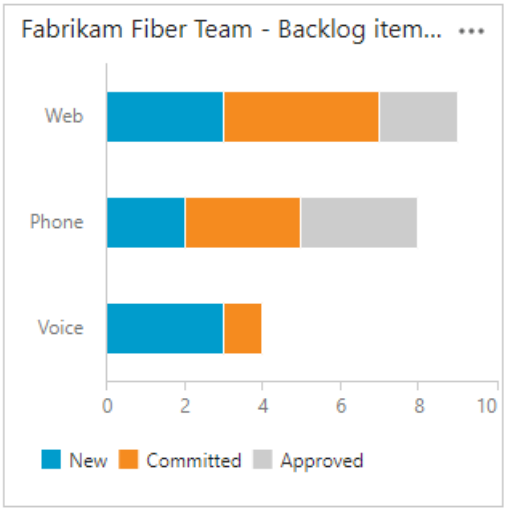
   !Screenshot of Edit area dialog, TFS 2018 and earlier on-premises versions.

3. To move the node within the hierarchy, change the Location field.

4. To delete a node, choose the **Delete** option from the actions menu.

Chart progress by area

You can quickly generate queries to view the progress based on an area path. For example, visualize progress of work items that are assigned to each team's area path, as shown in the following stacked bar chart. Choose Node Name to get the leaf node of the area path.



Q & A

Q: Do I have to assign an area path to a team?

A: No. You assign area paths to teams so that the work items assigned to that area path appear on the team's backlogs and boards. By default, all work items get assigned to the root area path. These work items appear in the default team that's defined for the project.

Next steps

Set iteration paths or sprints

Related articles

As you can see, area paths play a major role in supporting Agile tools, teams, and managing work items. Learn

more about working with these fields from the following articles:

- About areas and iterations
- Add another team
- Configure team settings and add team administrators
- Agile tools that rely on areas or iterations
- Query by area or iteration path
- Set permissions and access for work tracking

Programmatic resources

Area paths and iteration paths are also referred to as *Classification Nodes*.

- az boards area (Azure DevOps CLI)
- Teams (REST API)
- Classification Nodes (REST API)

- Teams (REST API)
- Classification Nodes (REST API)
- Define the classification plug-in (Process Template)

Define iteration paths (sprints) and configure team iterations

10/4/2022 • 18 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Iteration Paths, also referred to as *sprints*, support assignment of work items to time-box intervals. You define iteration paths at the project level, and then each team selects the paths that they want to use. Iteration paths are a shared resource used by all teams that select them. You can create a flat set of iteration paths or a hierarchy of paths to support releases, sub-releases, and sprints.

Define iteration paths and assign them to teams when you want to use the following tools:

- Assign work items to sprints using the Planning pane
- Query and chart work items based on Iteration Path
- Forecast your product backlog
- Sprints> all tools
- Delivery Plans, a calendar view of team deliverables
- Velocity chart and Sprint burndown chart

- Assign work to sprints using the Planning pane
- Query and chart work items based on Iteration Path
- Forecast your product backlog
- Sprints> all tools
- Delivery Plans, a calendar view of team deliverables
- Velocity chart and Sprint burndown chart

- Assign work to sprints using the Planning pane
- Query and chart work items based on Iteration Path
- Forecast your product backlog
- Sprints> all tools
- Delivery Plans, a calendar view of team deliverables
- Velocity chart

> **TIP**
> If a team hasn't subscribed or selected an iteration path, that iteration path won't appear in a team view or tool.

For information about naming restrictions and limits placed on addition of Iteration Paths, see About areas and iterations, Naming restrictions.

> **TIP**
> If all you need to do is change the iteration dates, you can do that quickly as shown in Change sprint dates. However, if you need to define the iteration paths and tree structure, then follow the guidance provided in this article.

Prerequisites

- To add an iteration path to a project, you must be a member of the **Project Administrators** group. If you don't have a project yet, create one now. By default, the user who created the project has these permissions set.
- To add, edit, and manage iteration paths under a node, you must have one or more of the following permissions set to **Allow** for the node that you want to manage: **Create child nodes**, **Delete this node**, **Edit this node**, and **View permissions for this node**.
- To set team iteration paths, you must be added as the team administrator or be a member of the **Project Administrators** group.

For more information about acquiring permissions, see Change project-level permissions or Set permissions and access for work tracking.

Get started

Newly created projects contain a single, root area path that corresponds to the project name. You add area paths under this root. Also, each project typically specifies a predefined set of iteration paths to help you get started tracking your work. You only need to specify the dates.

If you're new to managing projects and teams, complete the following steps.

1. Review Configure and customize Azure Boards.
2. Define the area paths and teams following the guidance provided in Define area paths and assign to a team.
3. Determine the length of the iteration you want to support. Recommended practice is to have all teams use the same sprint cadence. For guidance, review About areas and iterations.
4. Determine if you want a flat structure or hierarchy of sprints and releases.
5. Open **Project settings > Project configuration** and define the iteration paths to support steps 2 and 3 at the project level. Follow the steps provided later in this article: Open Project Settings, Project configuration and Add iterations and set iteration dates.
6. Open the team configuration and assign the default and additional area path(s) to each team. Follow the steps provided later in this article: Open team settings and Set team default iteration path(s).
7. Each team should assign the default iteration path they selected to their work items. Do so for those work items to show up on their product backlogs and boards. Use bulk modify to modify several work items at once. See also Assign backlog items to a sprint.

As needed, you can do the following tasks at any time:

- Add additional child iteration nodes
- Rename an iteration path (except the root path)
- Move a child iteration path under another node
- Delete a child iteration path
- Change the default and selected iteration paths assigned to a team

Team backlog iteration versus default iteration

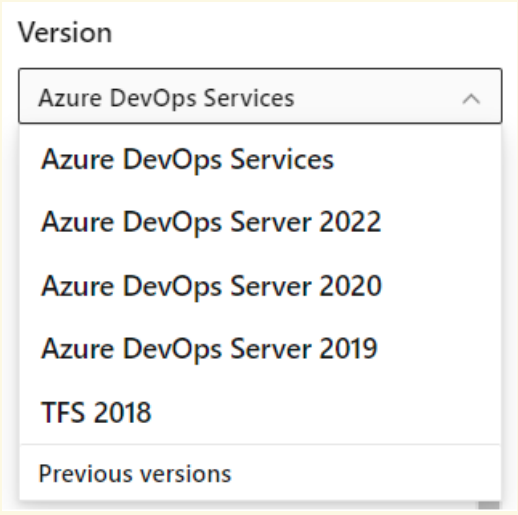
Each team has access to many Agile tools, as described in About teams and Agile tools. Each tool references the team's default area path(s). Several tools reference the team's default and selected iteration paths or sprints. Most teams choose one area path and several iteration paths to support their work tracking activities. However, to support other scenarios, it's possible for teams to choose several area paths to appear on their backlogs and boards.

Teams can set a default iteration different from the backlog iteration. The backlog iteration determines which items appear on the team's backlogs and boards. And, the default iteration determines what value is assigned to

work items created from the team context.

All work items that you create from your team context are automatically assigned both the team's default area path and default iteration path.

IMPORTANT



To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

Open Project Settings, list project iterations

You define both areas and iterations for a project from the **Project Settings > Boards > Project configuration**.

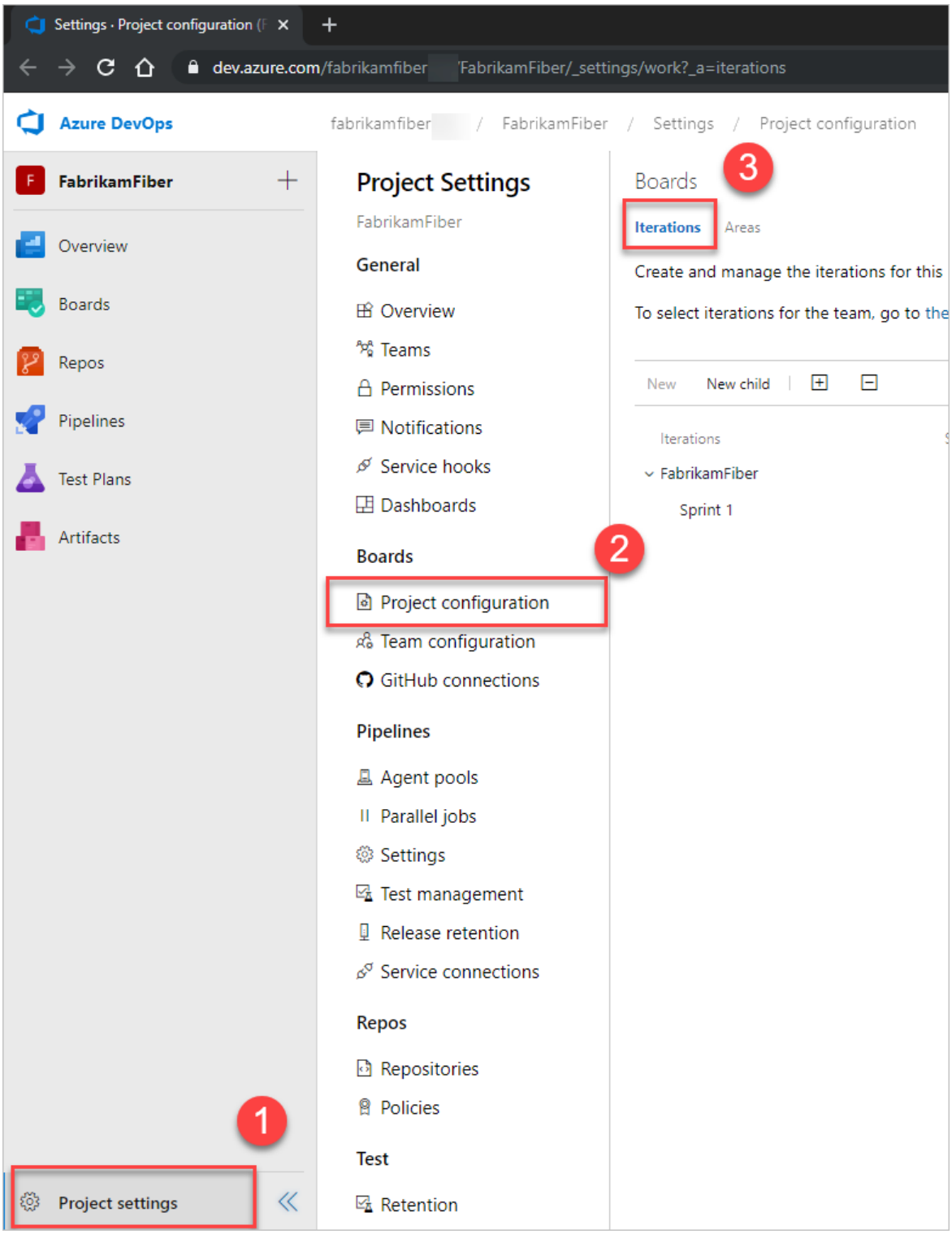
You define both areas and iterations from the **Work** pages of the **Project Settings** context. From the user context, you open the admin context by choosing **Settings** .

- Browser
- Azure DevOps CLI

From the web portal, open **Project Settings**.

Define both areas and iterations for a project from the **Project settings > Boards > Project configuration**.

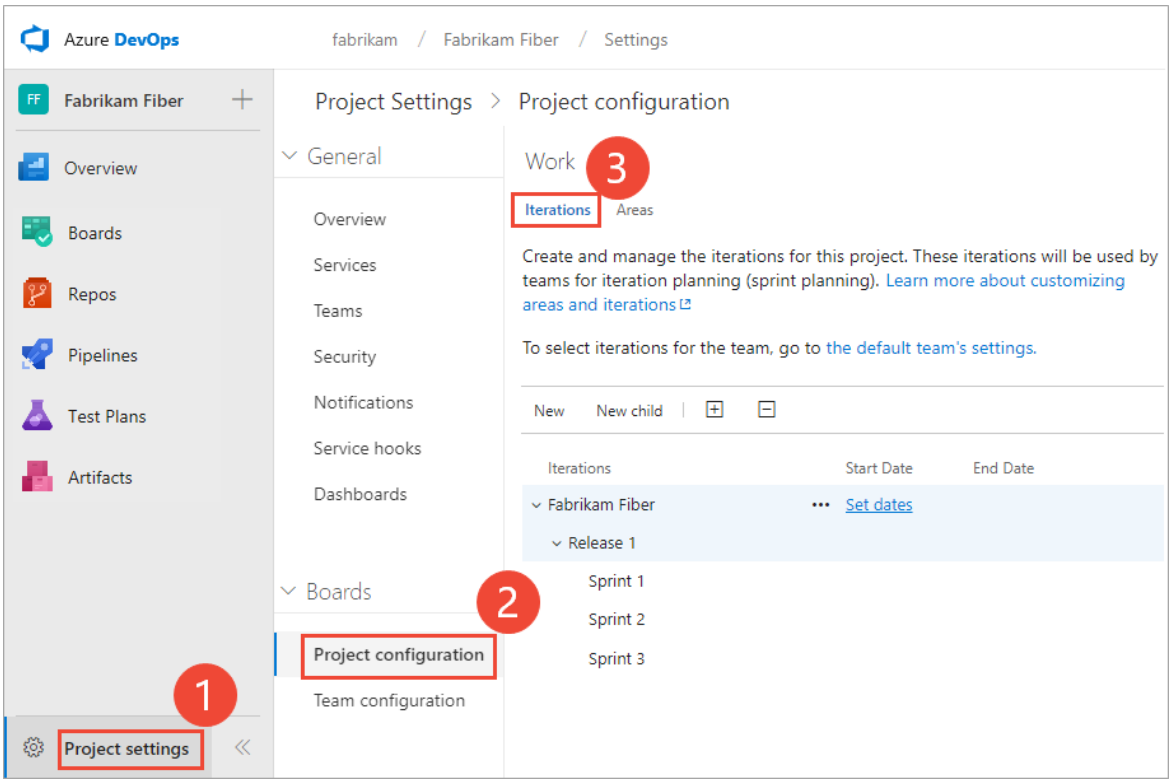
1. Choose (1) **Project Settings**, expand **Boards** if needed, and then choose (2) **Project configuration** and (3) **Iterations**.



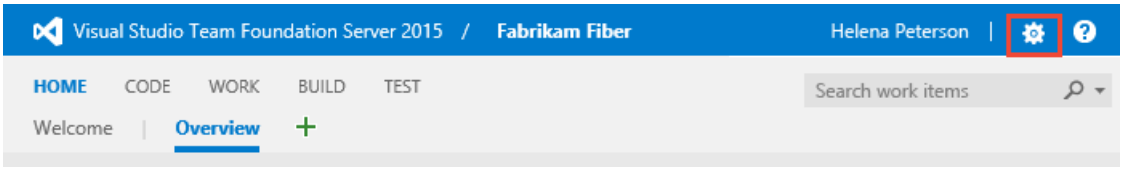
From the web portal, open **Project Settings**.

Define both areas and iterations for a project from the **Project configuration > Iterations**.

1. From the web portal, choose **Project settings**(1), select **Project configuration**(2), and then **Iterations**(3).



From the web portal, choose **Project settings** to open project administration pages. Then, choose **Iterations**.



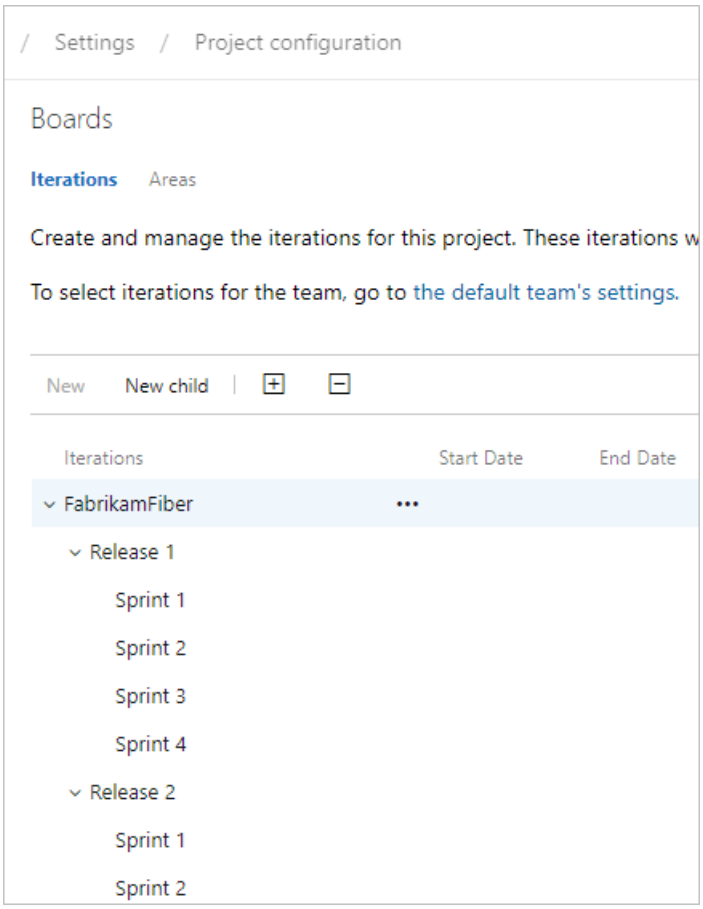
Add iterations and set iteration dates

You can add iterations in the same way that you add areas. For more information about working within a sprint cadence, see Scrum and sprint planning tools.

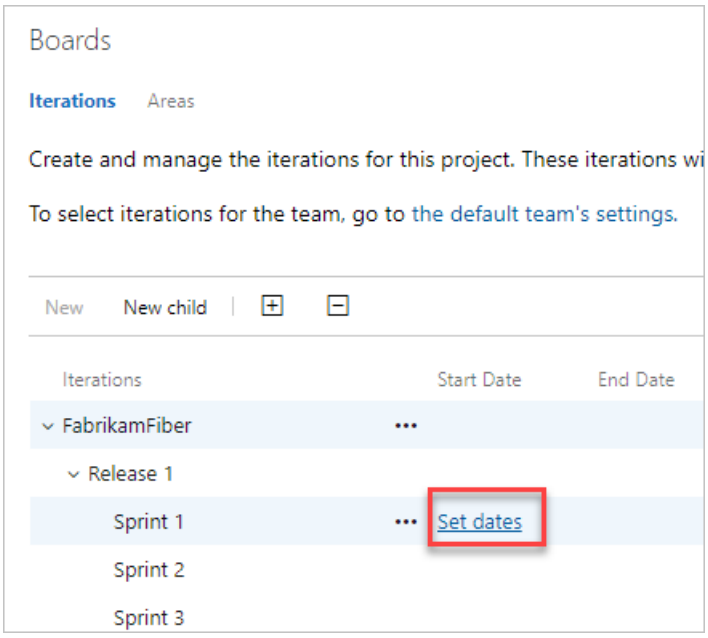
- Browser
- Azure DevOps CLI

1. Add and modify area paths from **Project settings** > **Project configuration** > **Iterations**.

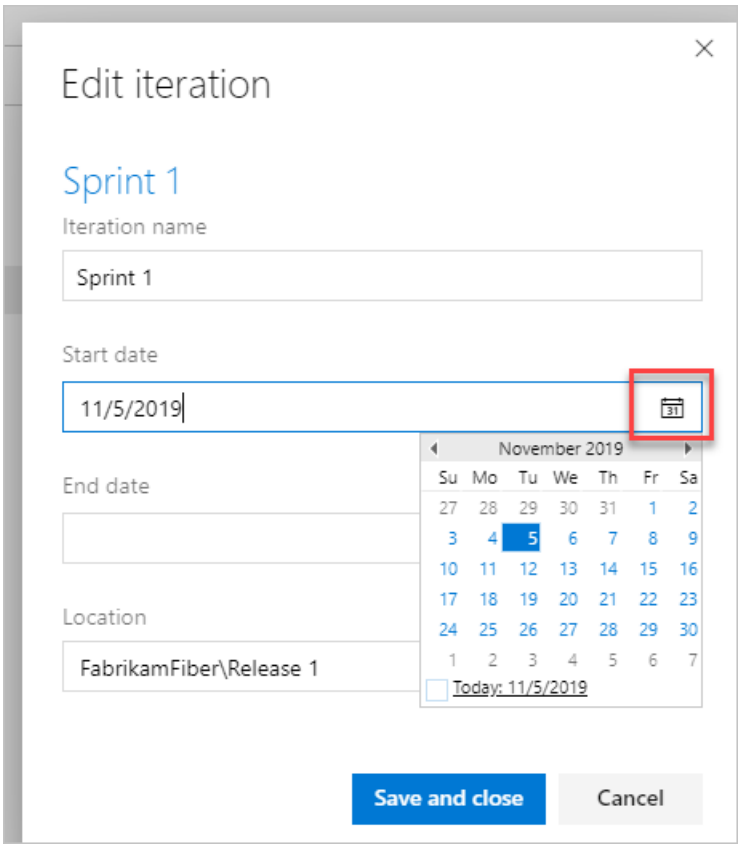
   For Scrum-based projects, you see the following set of sprints.



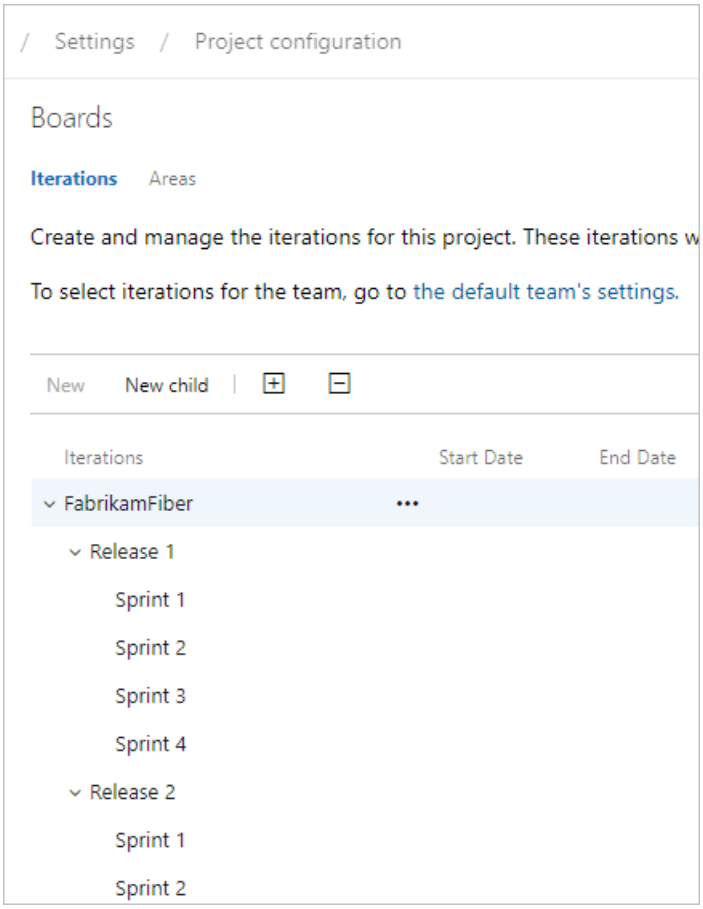
2. To schedule the start and end dates for each sprint that your teams use, highlight the sprint and choose **Set dates**. Or, you can select **Actions** ··· for the iteration path and choose **Edit**.



3. Choose the calendar icon to select new dates.



4. When you're finished, you have a set of sprints scheduled - similar to the following image:

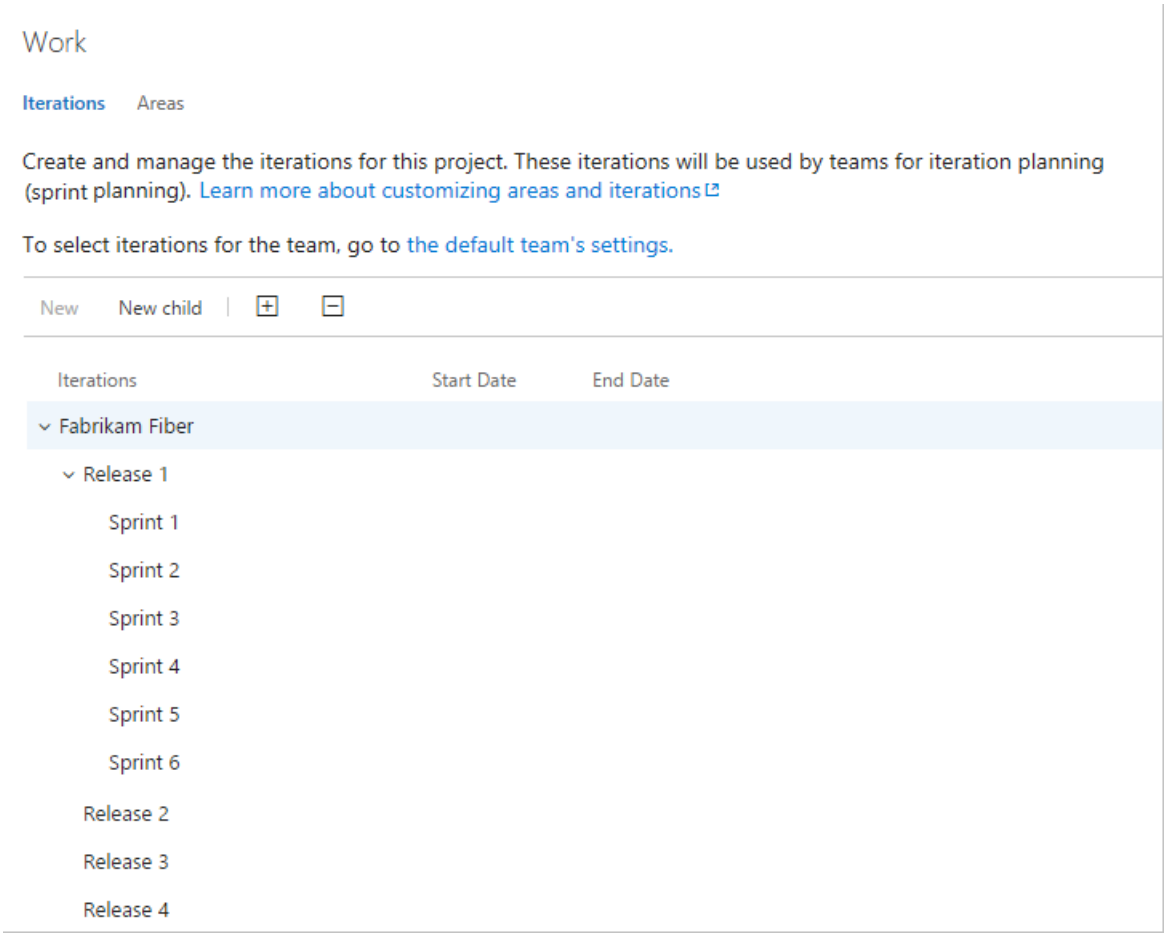


Your next step is to choose the sprints each team uses.

From **Iterations**, you can add iterations that teams can then select for their use.

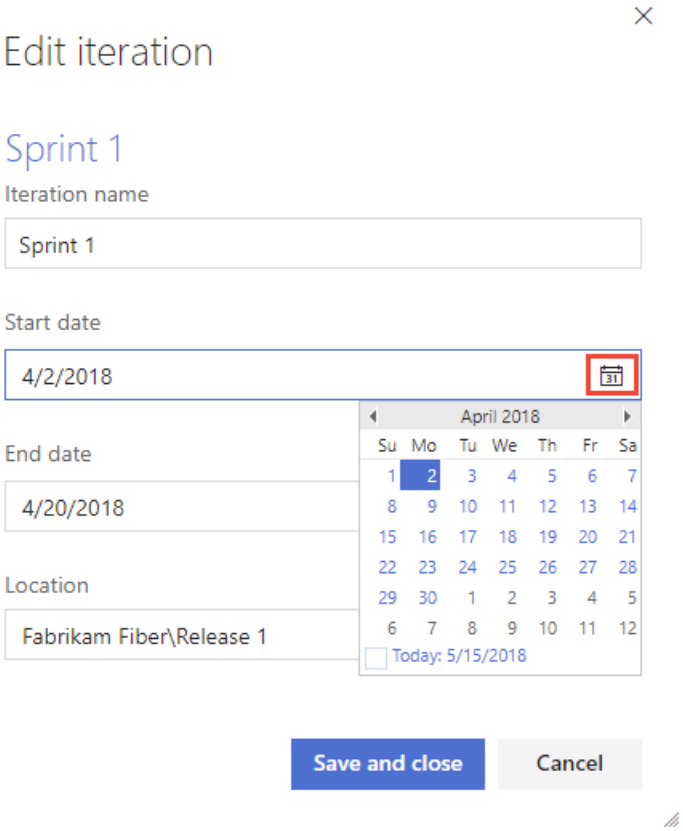
1. Add and modify area paths from the **Work** > **Iterations** page from the project admin or settings context.

For Scrum-based projects, you see the following set of sprints.

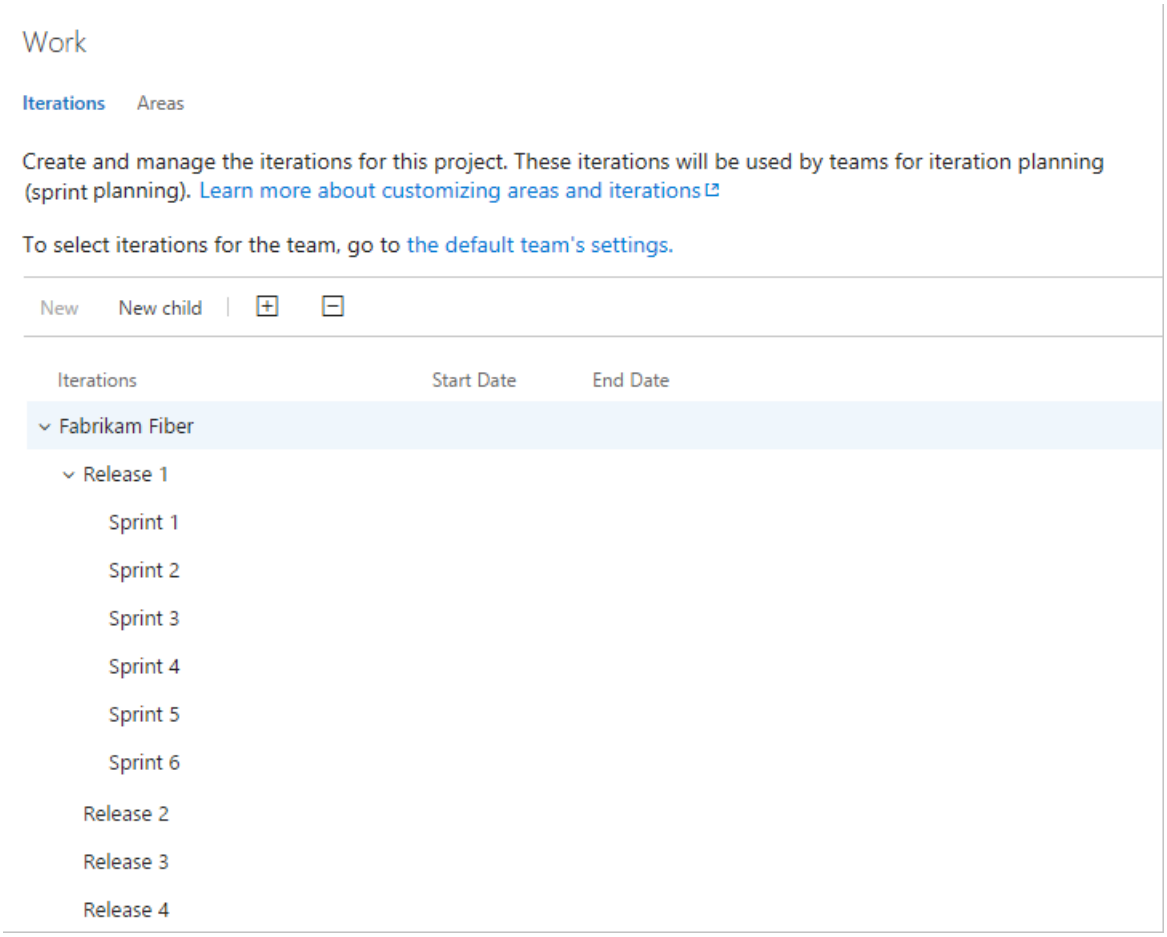


2. To schedule the start and end dates for each sprint your teams use, Highlight the sprint and choose **Set dates**. Or, you can select **Actions** ··· context menu for the iteration path and choose **Edit**.

   Choose the calendar icon to choose new dates.



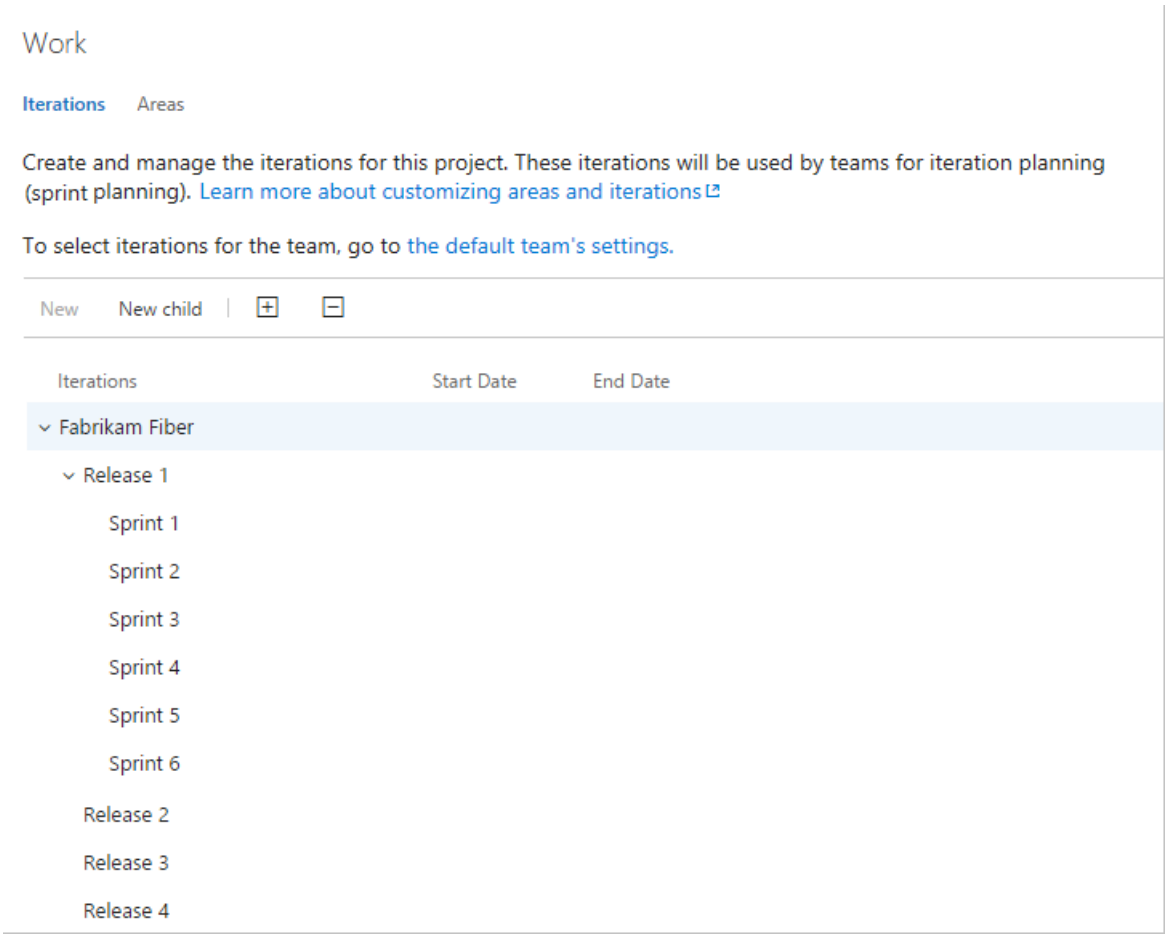
3. When you're finished, you have a set of sprints scheduled - like this:



Your next step is to choose the sprints each team uses.

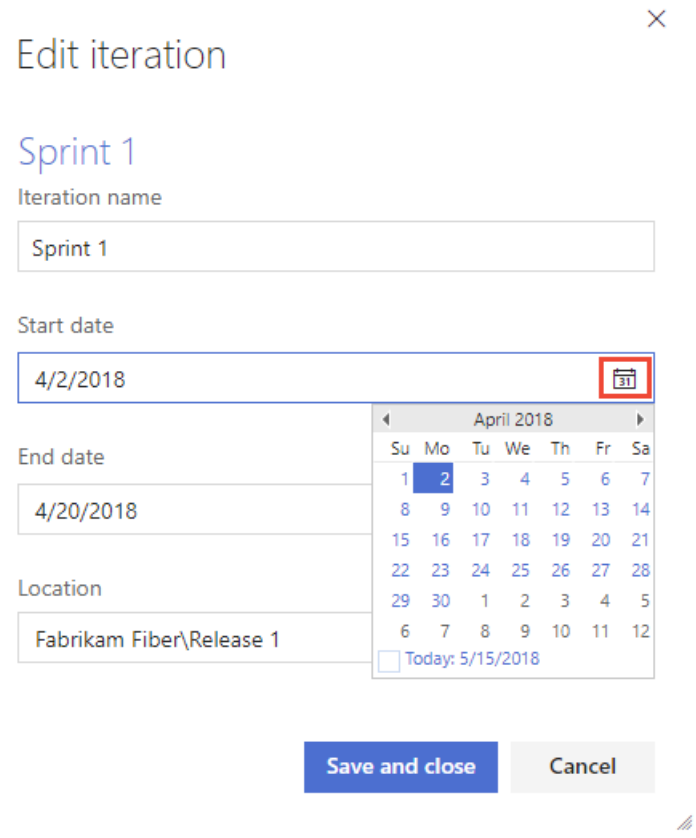
Add and modify area paths from the **Work** > **Iterations** page from the project admin or settings context.

For Scrum-based projects, you see the following set of sprints.

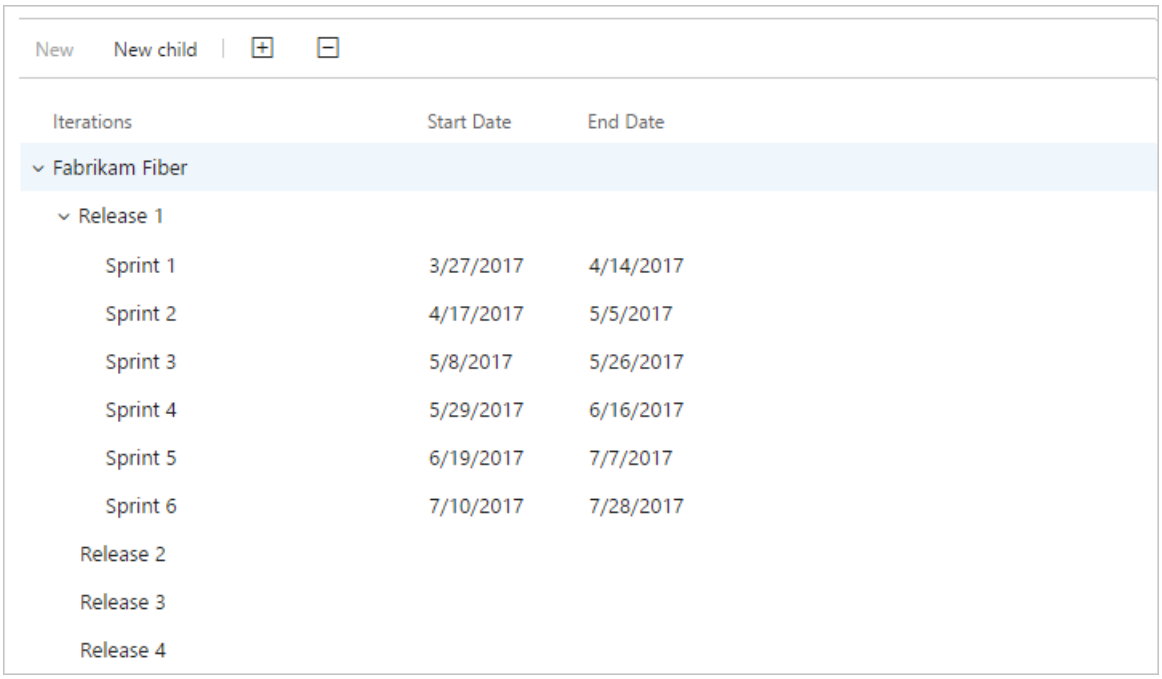


1. To schedule the start and end dates for each sprint your teams use, Highlight the sprint and choose **Set dates**. Or, you can select **Actions** ··· for the iteration path and choose **Edit**.

   Choose the calendar icon to choose new dates.



2. When you're finished, you have a set of sprints scheduled - like this:



Your next step is to choose the sprints each team uses.

Open team settings, list team iterations

You can set team defaults from team settings. If you're not a team administrator, get added as one. Only team or project administrators can change team settings.

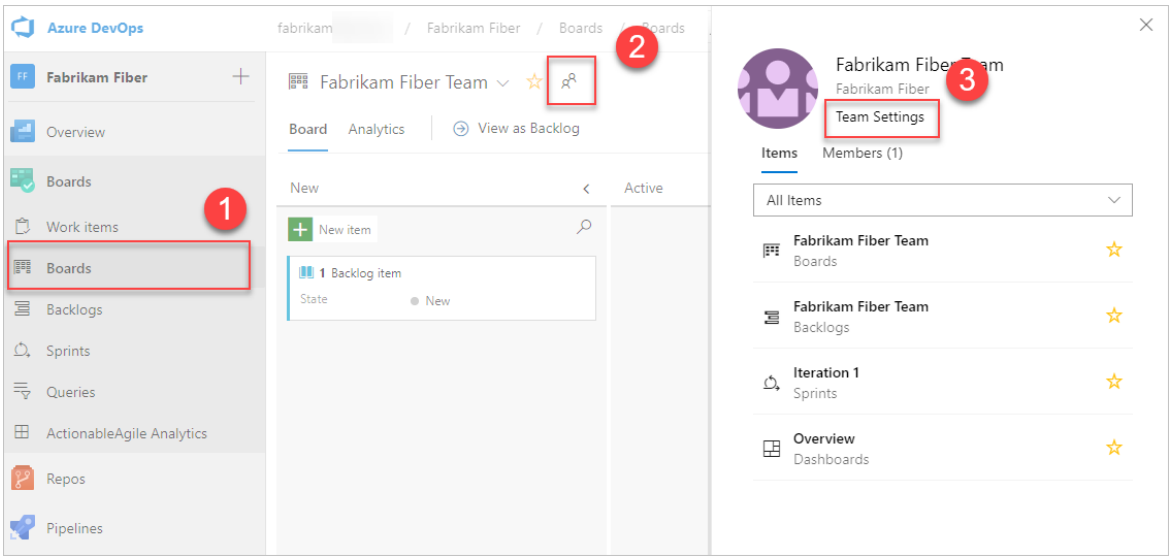
> **NOTE**
> Teams can be assigned a maximum of 300 **Iteration Paths**. To learn more, see Work tracking, process, and project limits.

- Browser
- Azure DevOps CLI

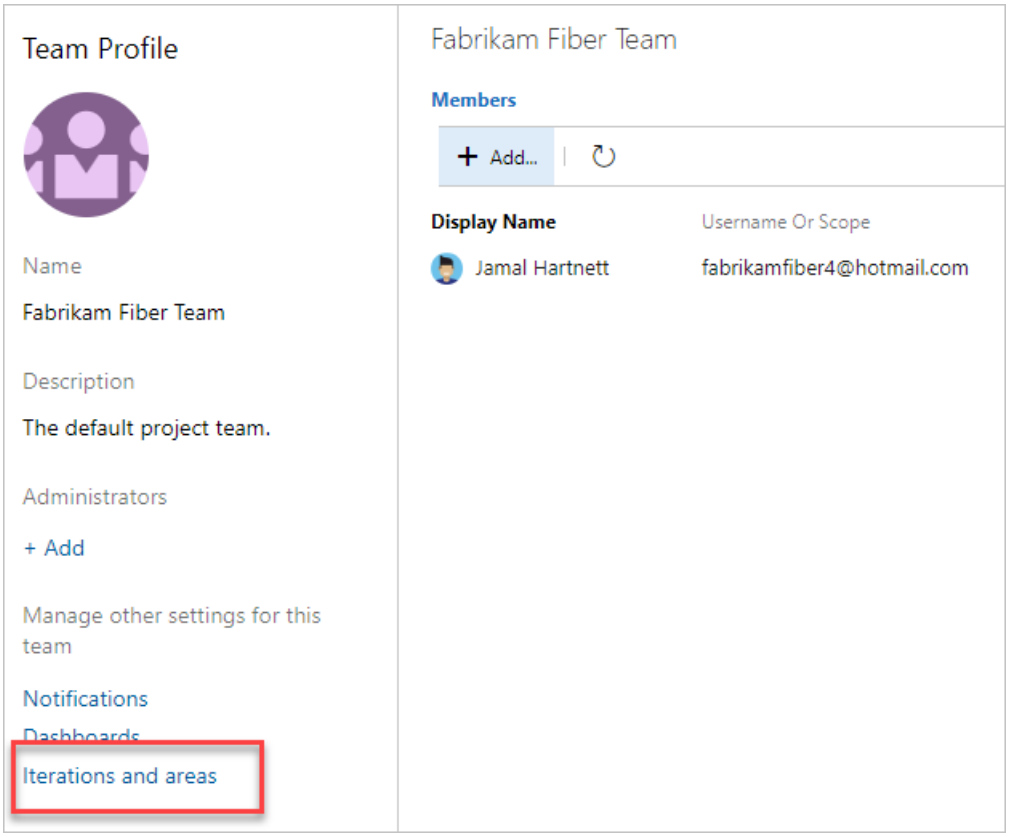
You define both areas and iterations from **Project settings > Boards > Team configuration**. You can quickly navigate to it from a team work tracking backlog, board, or dashboard.

1. Open a backlog or board for a team and choose **Team profile** > **Team Settings**.

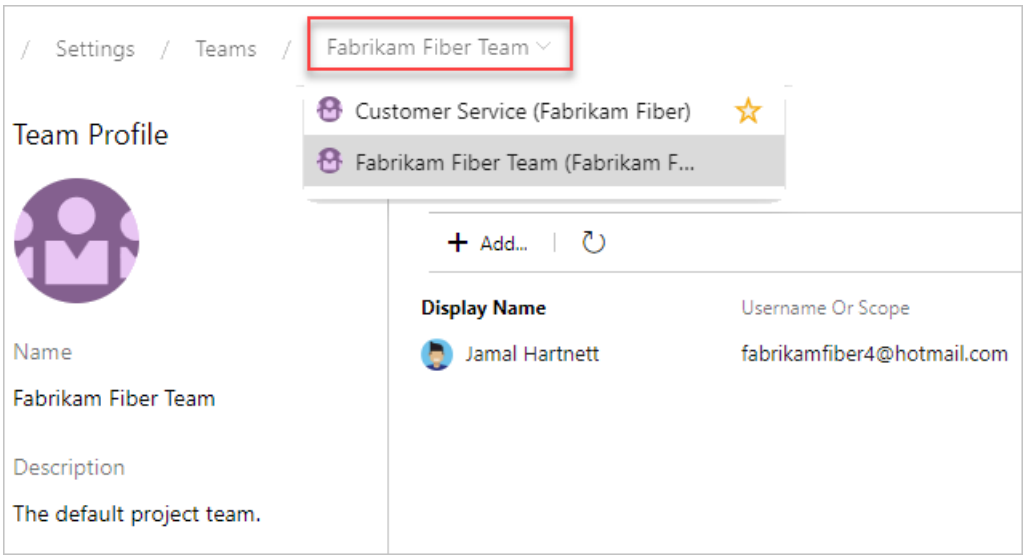
   Here we open the Board for the Web team and from there the team profile.



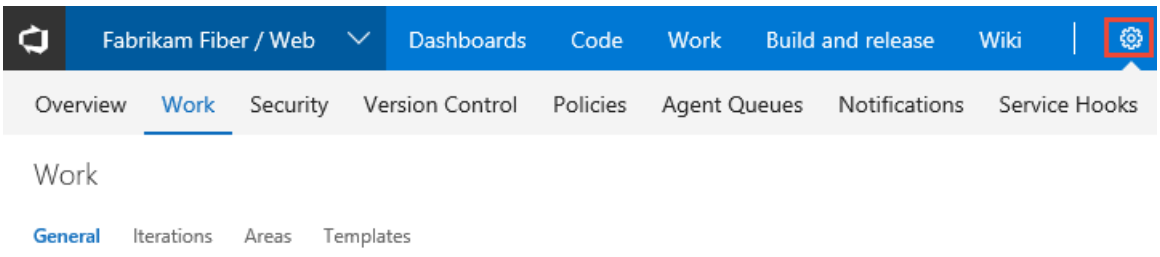
2. Choose **Iterations and areas**.



3. If you need to switch the team context, use the team selector within the breadcrumbs.



You open team settings from the upper navigation bar. Select the team you want and then choose **Team settings** . For more information about switching your team focus, see Switch project, repository, team



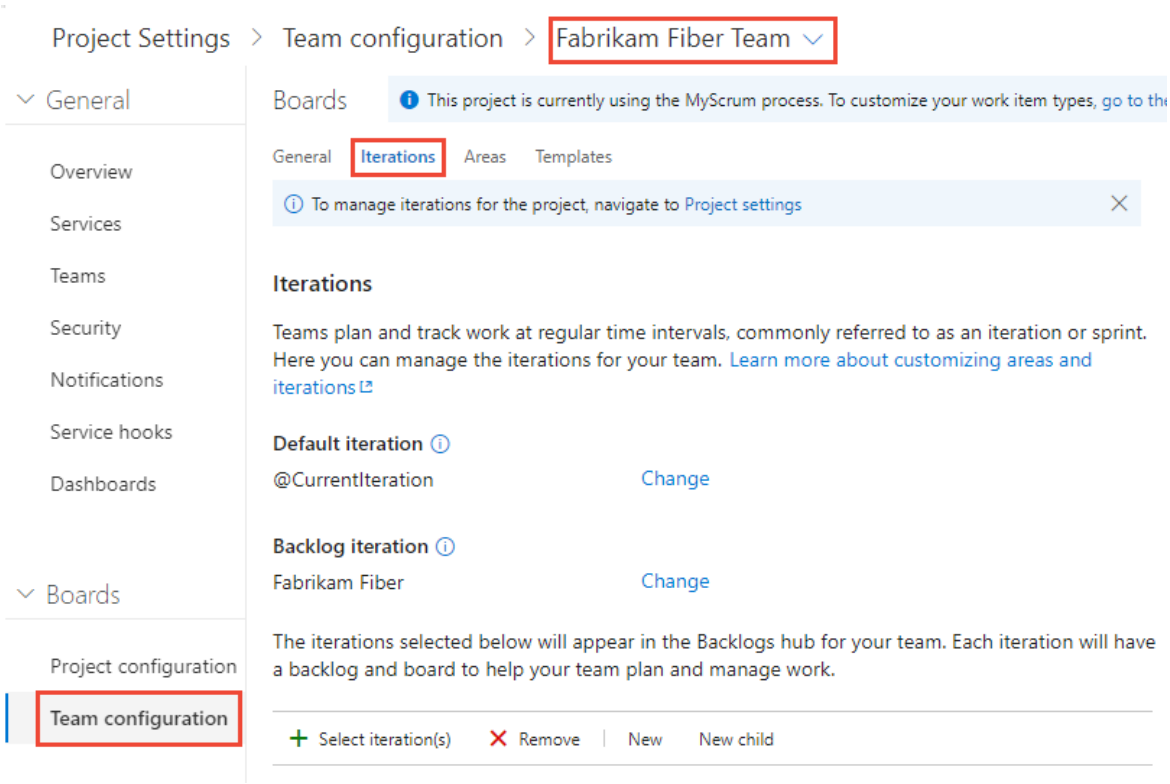
Select team sprints and set the default iteration path

You define sprints for the project and then select them to be active for each team. You assign the default iteration to use when creating new work items.

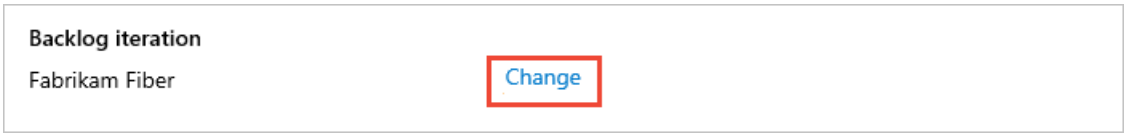
- Browser
- Azure DevOps CLI

1. Open **Project settings** > **Boards** > **Team Configuration** > **Iterations** for a team.

   Here, we navigate to the Fabrikam Fiber Team.



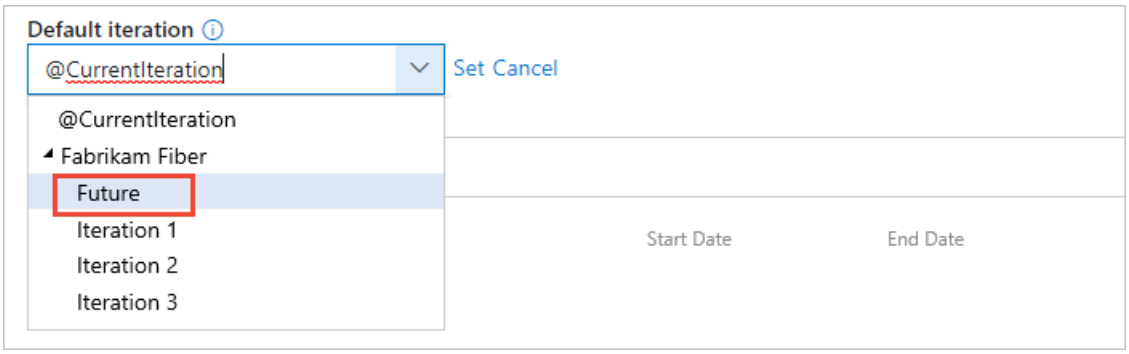
2. **Backlog iteration**. Only work items assigned to an iteration equal to or under this backlog iteration appear in the team's backlogs and boards.



   Also, all work items added through a team's backlog or board are assigned the backlog iteration.

3. **Default iteration**. The default iteration defines the iteration that's used when you create a work item from the team backlog or Kanban board. You can specify any iteration defined under the **Backlog iteration** path. To assign new work items to the current iteration, specify **@CurrentIteration**. The same macro used in queries to list work items assigned to the currently active iteration assigned to the team is used.

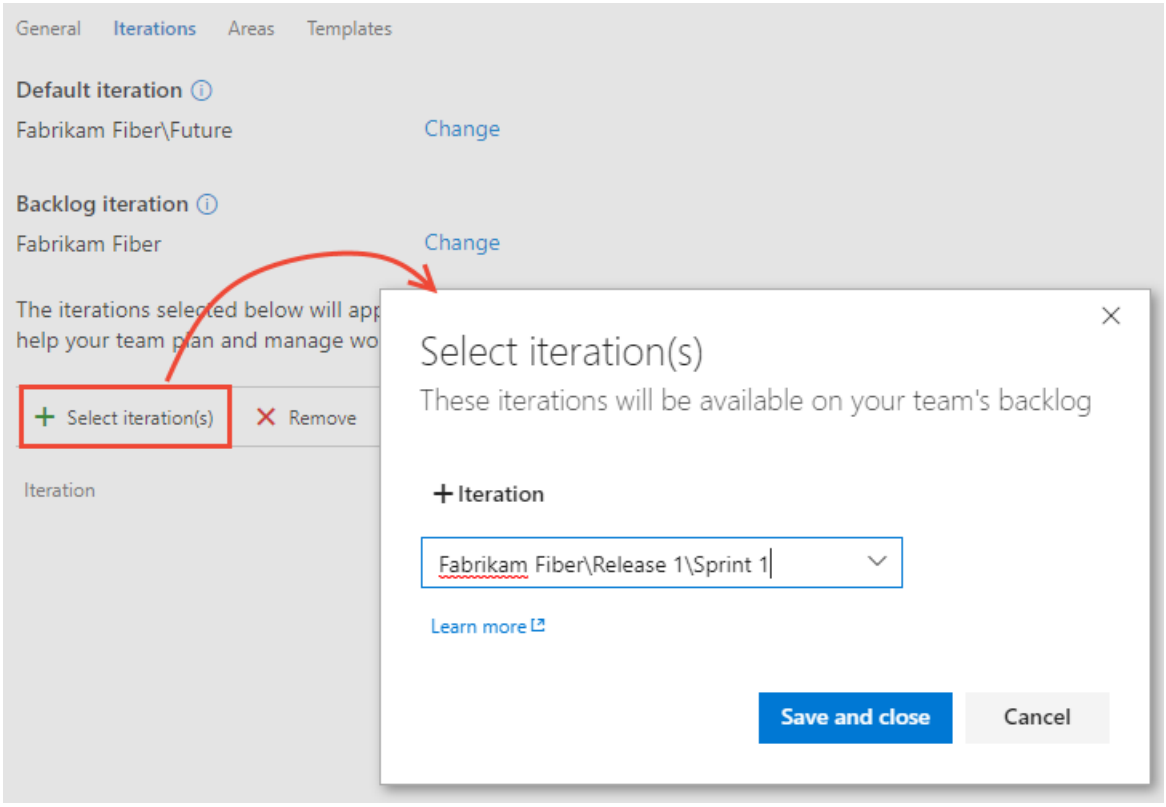
   For example, you might want all new work items added to a future iteration path, which you use to triage and assign to specific sprints at periodic intervals.



NOTE

New work items added through the **Work Items** page or the **New Work Items** widget on a team dashboard don't reference the **Default Iteration Path** assigned to the team. Instead, new work items are assigned the last **Iteration Path** selected by the user. New work items added through a team's **Sprints** backlog or taskboard are always assigned the **Iteration Path** associated with the selected sprint.

4. **Active sprints**. Add an iteration for each sprint backlog you want active for the team. Add each sprint, one by one, by selecting it from the menu.



When you're done, you should see a list of sprints, similar to the following.



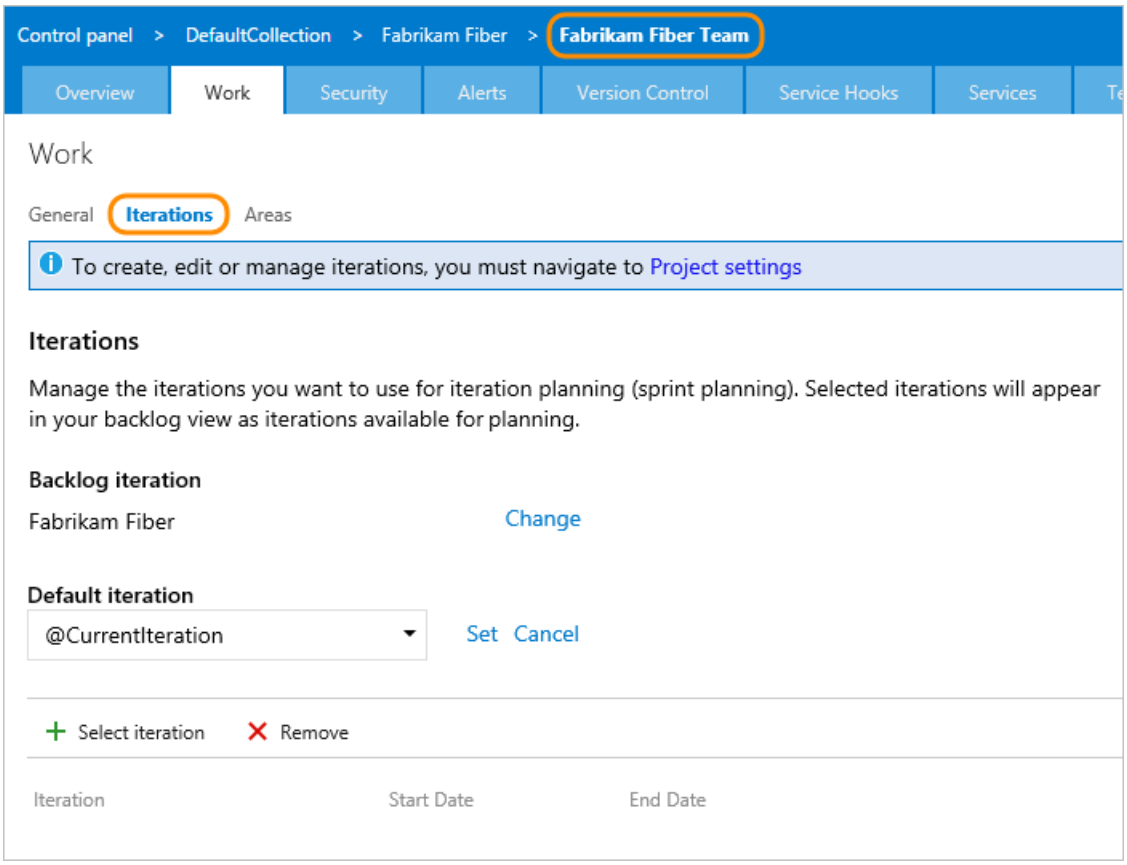
If you don't see the sprints or dates that you need, you can add or edit iterations for the project, provided

you have the required permissions. For more information, see Define iteration (sprint) paths.

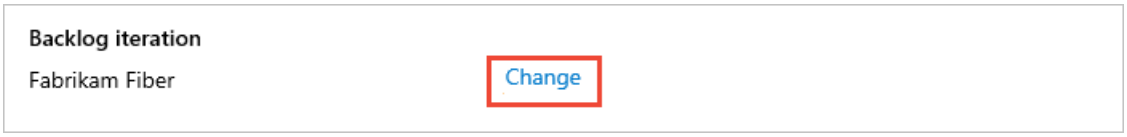
5. To see the newly activated sprint backlogs, refresh your team's product backlog page.

1. Open **Work** > **Iterations** for a team.

   Here, we navigate to the Fabrikam Fiber Team.



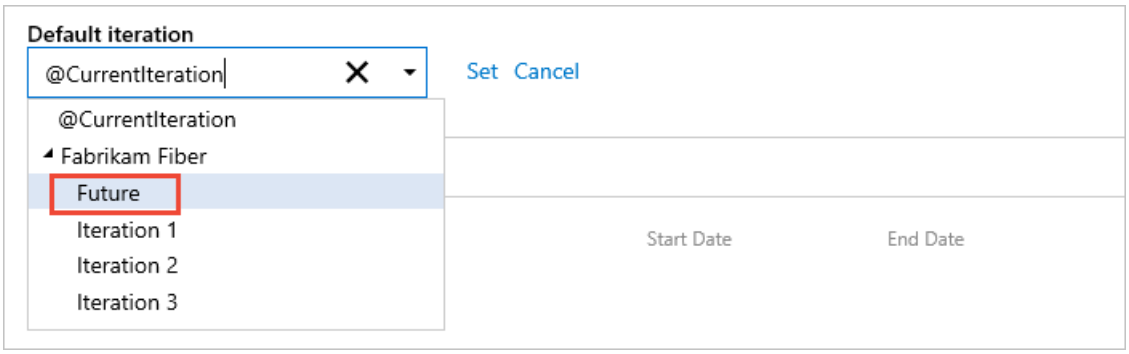
2. **Backlog iteration**. Only work items assigned to an iteration equal to or under this backlog iteration appear in the team's backlogs and boards.



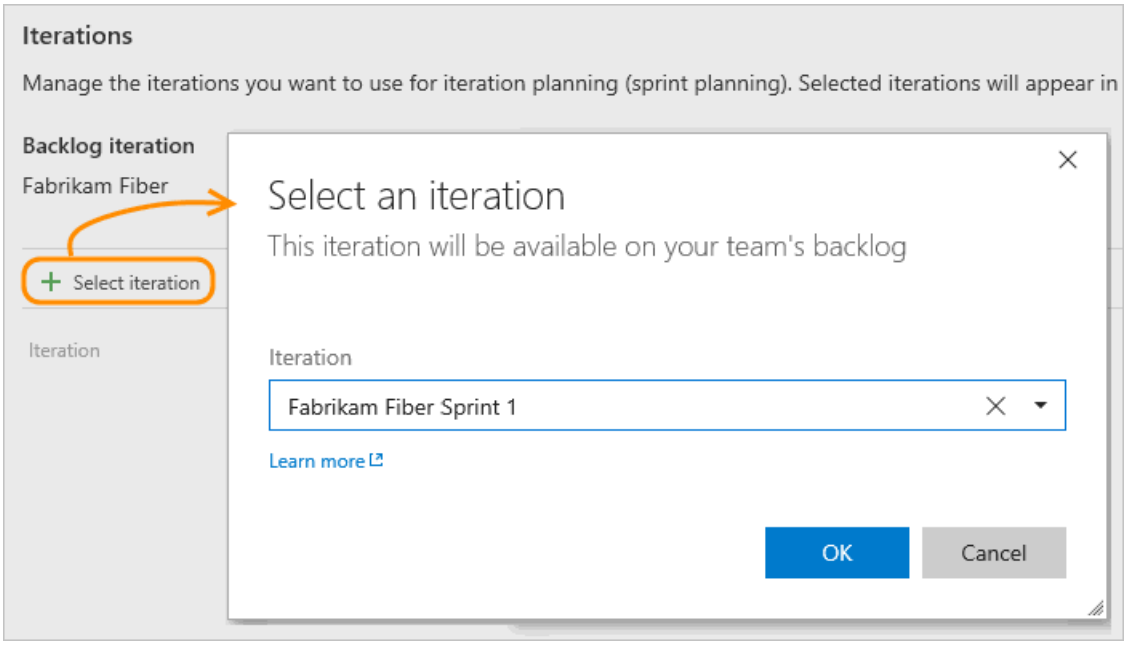
   Also, all work items added through a team's backlog or board are assigned the backlog iteration.

3. **Default iteration**. The default iteration defines the iteration that's used when you create a work item from the team dashboard and queries page. You can use an explicit value or use **@CurrentIteration** to assign new work items to the team's current iteration. The same macro used in queries to list work items assigned to the currently active iteration assigned to the team is used.

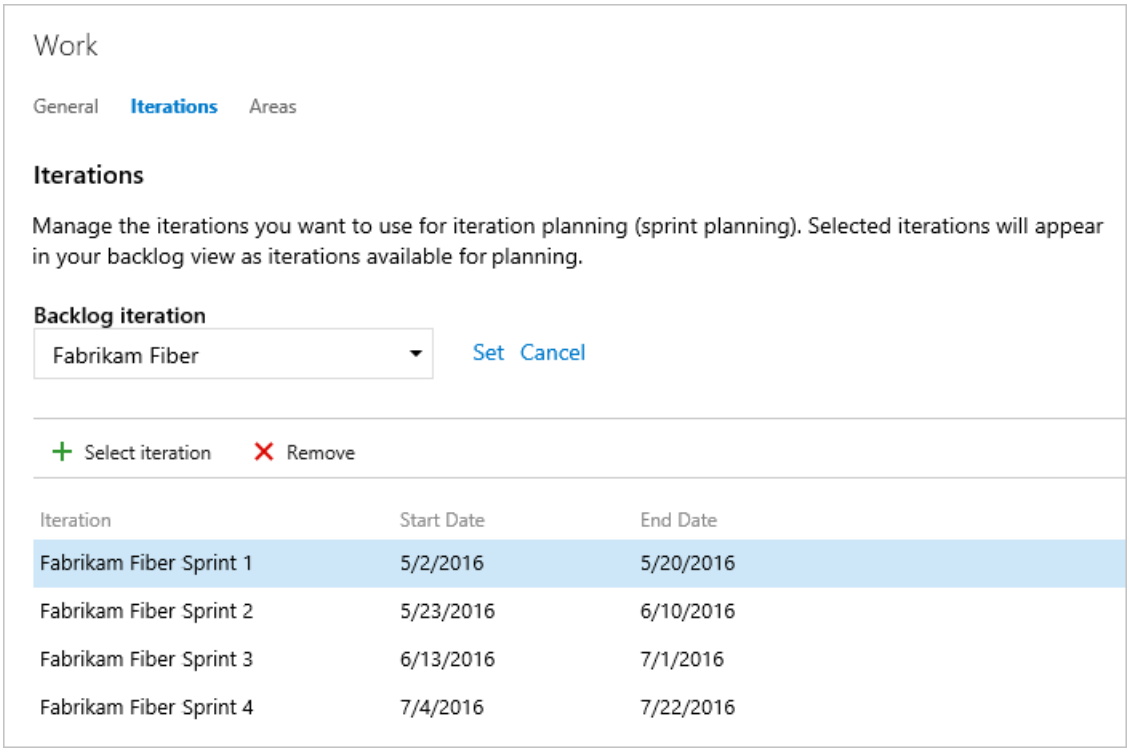
   For example, you might want all new work items added to a future iteration path, which you use to triage and assign to specific sprints at periodic intervals.



4. **Active sprints**. Add an iteration for each sprint backlog you want active for the team. Add each sprint, one by one, by selecting it from the menu.



When you're done, you should see a list of sprints, similar to the following.



If you don't see the sprints or dates that you need, then return to the project administration context and define them there.

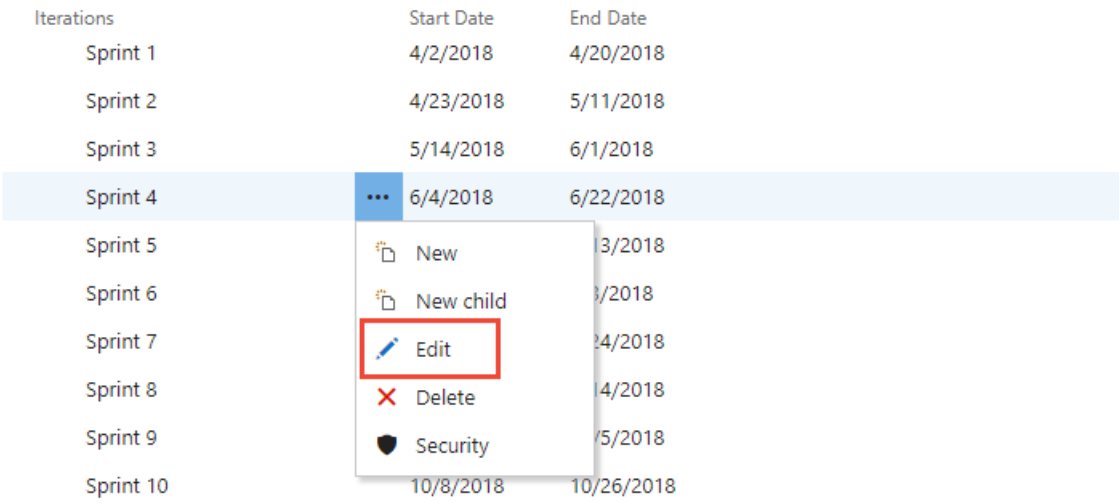
5. To see the newly activated sprint backlogs, refresh your team's product backlog page.

Rename, move, or delete an iteration

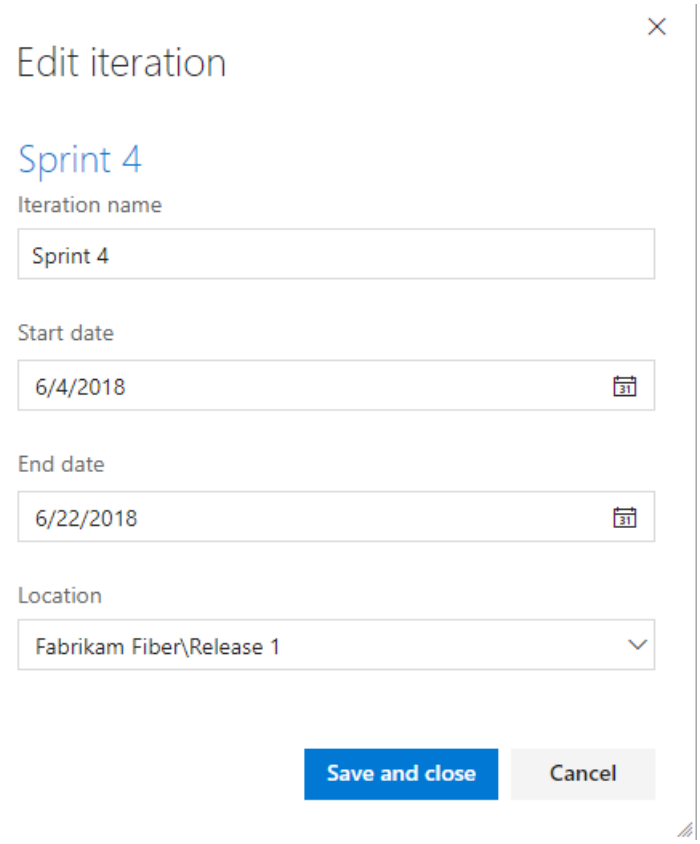
When you rename an iteration, or move the node within the tree hierarchy, the system automatically updates the work items and queries that reference the existing path(s).

- Browser
- Azure DevOps CLI

1. To rename an iteration path, choose **Actions** ··· for the node, and then select **Edit**.



2. In the dialog that opens, enter the new name.



3. To move the node within the hierarchy, change the Location field.

4. To delete a node, choose the **Delete** option from the actions menu.

> **NOTE**
> When you delete an iteration node, the system automatically updates the existing work items with the node that you enter at the deletion prompt.

-

Export/import iteration paths

You can't export or import the structure of tree paths for one project to use with another project.

However, using the `az boards iteration` commands you can list the iterations defined for one project and then add them to another project. Once they are added, you can then use the `az boards iteration team` commands to add them to a team and set the default and backlog iteration path for the team. Using the Classification Nodes (REST API) and Teams (REST API), you can perform similar actions.

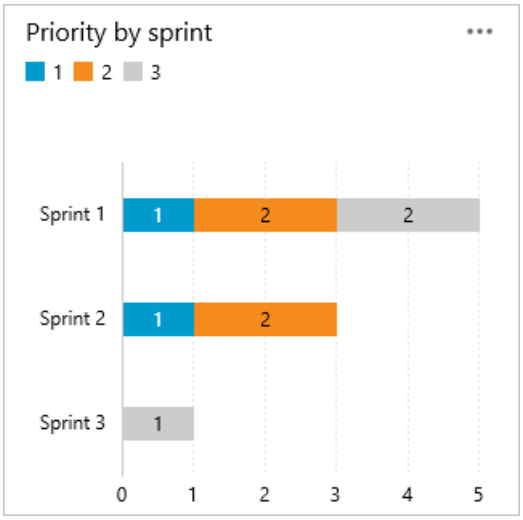
Archive iteration paths

After a while, you may want to archive iteration paths that were used for sprints that are a year or more out of date. You can do that by moving the iteration path under a node that you label "Archive". All work items are updated with the moved iteration path. Also, teams can de-select those sprints that have past. All data is maintained in the data store with the new iteration path assignments.

Before you archive the iterations, consider if you've captured all the reports that you may want.

Chart progress by iteration

You can quickly generate queries to view the progress for those areas. For example, you can visualize progress of work items that are assigned to sprints, as shown in the following stacked bar chart.



Q & A

Q: Do I have to assign iteration paths to a team?

A: If your team doesn't use sprints to plan and track work, then no. You can leave the defaults assigned to the team as they are. You can then use the product and portfolio backlogs and boards, however you can't gain much use of sprint planning tools.

Related articles

- About areas and iterations

- Add another team
- Configure team settings and add team administrators
- Assign backlog items to a sprint
- Agile tools that rely on areas or iterations

Programmatic resources

Area paths and iteration paths are also referred to as *Classification Nodes*.

- az boards iteration (Azure DevOps CLI)
- Teams (REST API)
- Classification Nodes (REST API)

- Teams (REST API)
- Classification Nodes (REST API)
- Define the classification plug-in (Process Template)

Set up your project's backlogs and boards in Azure Boards

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

In most cases, you can start using your product and portfolio backlogs once your project is created. A default team is created along with associated backlogs and boards. You can start adding work items to your product backlog using the Backlog or Board.

However, you may need to ensure you've configured your backlogs and boards correctly. Ensure the configuration if you've added a team and want to start using the team backlogs and boards. Changes may be made to a project or team configuration over time. These changes can influence the work items that appear on your backlog and boards.

For an overview of the tools associated with your team, see Manage and configure team tools.

Default backlog and board work items

The first thing you need to understand is that your product **Backlog** and **Board** display work items that meet the following criteria:

- Work item type belongs to the Requirements category. The types differ depending on the process selected for your project:
  - Basic : Issue, Backlog name=**Issues**
  - Agile: User Story, Backlog name=**Stories**
  - Scrum: Product Backlog Item, Backlog name=**Backlog items**
  - CMMI: Requirement, Backlog name=**Requirements**
- Work item **Area Path** matches one of the selected team's Area Paths
- Work item **Iteration Path** is under the team's Default Iteration Path

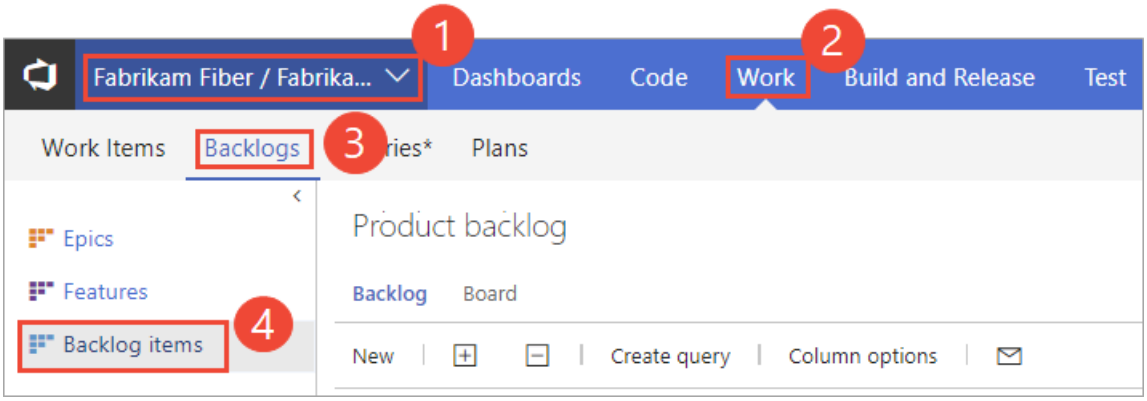
> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

- Work item type belongs to the Requirements category. The types differ depending on the process selected for your project:
  - Agile: User Story, Backlog name=**Stories**
  - Scrum: Product Backlog Item, Backlog name=**Backlog items**
  - CMMI: Requirement, Backlog name=**Requirements**
- Work item **Area Path** matches one of the selected team's Area Paths
- Work item **Iteration Path** is under the team's Default Iteration Path

You can determine the work item types that belong to your Requirements category. Determine the items by opening your product Backlog and checking the product backlog name.



As shown in the following image, (1) choose the team, (2) **Work**, (3) **Backlogs**, and then the product backlog.



Look up your team's Area Path(s) and Iteration Paths. For more information, see Define area paths and assign to a team and Define sprint paths and configure team iterations.

Default sprint backlog and Taskboard work items

Your sprint backlog and Taskboard apply the filters associated with your team's default backlog and board work items along with the Iteration Path you select.

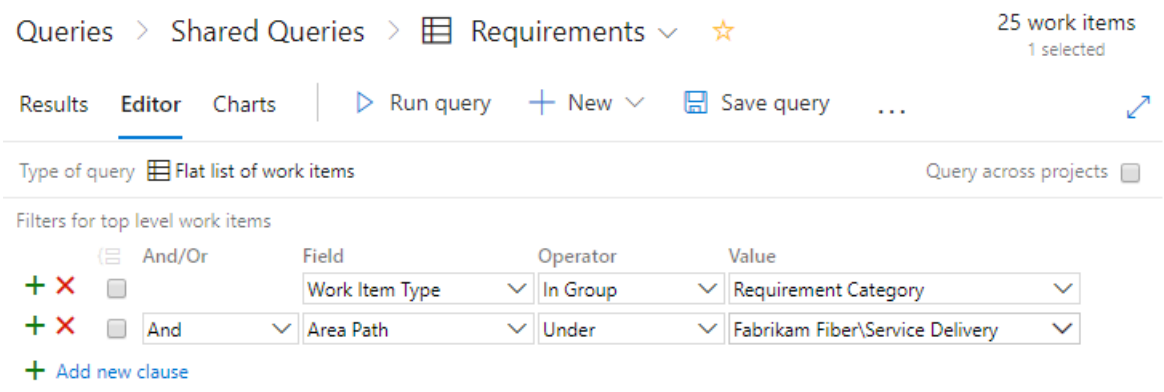
You can only select Iteration Paths that have been preselected by your team.

Your sprint backlog displays only those work items assigned to the selected sprint. Child tasks assigned to other sprints aren't displayed.

Review checklist for work items, backlogs, and boards

If you don't see the work items you expect on your product Backlog or Kanban board, complete the following checks:

1. Make sure you've selected the team backlog or board of interest. To learn how, see Use breadcrumbs and selectors to navigate and open artifacts.
2. Create a query of your backlog items, specifying the work item types that belong to your Requirements category and the Area Path associated with your team, for example:



3. Add the **State**, **Area Path**, and **Iteration Path** fields to the column options.

4. Check the query results and that the values of the work items you expect to show up on your backlog meet these criteria:

   - **Area Path** belongs to your team's area path(s)
   - **Iteration Path** belongs under your team's default iteration path
   - **State** isn't Closed, Completed, Done, or Removed.

> **NOTE**
> You can also filter your product backlog to show or hide work items that are in an **In Progress** state category, corresponding to an Active, Resolved, Committed, Doing workflow state.

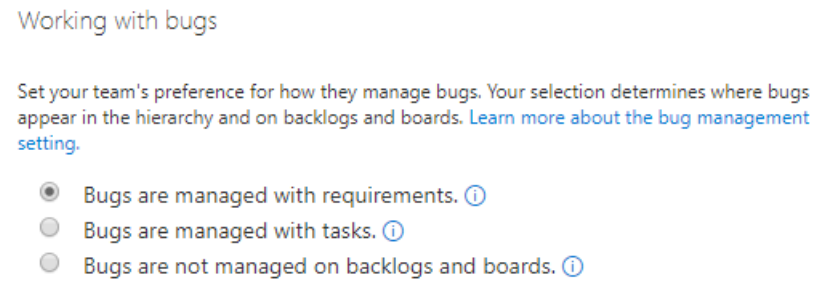
Add bugs to your backlogs and boards

For all processes except the Basic process, each team manages the way bugs are tracked. Track bugs in the Requirements category because they show up on the Backlog and Kanban board or the Tasks category. They can also show up on the Taskboard or the Bugs category where they don't appear on either backlogs or boards.

> **NOTE**
> Bug work item types aren't available with the Basic process. The Basic process tracks bugs as Issues and is available when you create a new project from Azure DevOps Services or Azure DevOps Server 2019.1 or later versions.

Each team can manage the way bugs are tracked. You can track bugs as belonging to the Requirements category. Those bugs show up on the Backlog and Kanban board or the Tasks category. They can show up on the Taskboard or the Bugs category where they don't appear on either backlogs or boards.

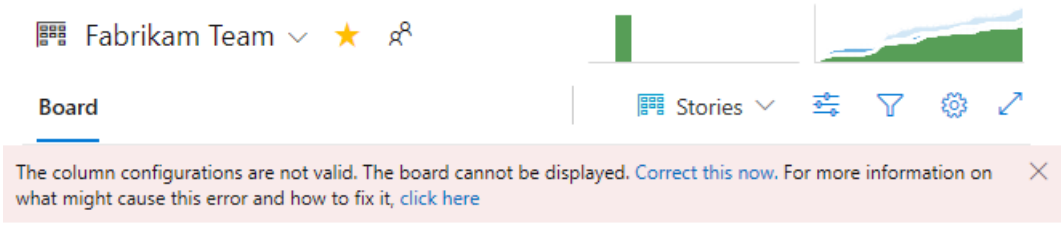
If you want bugs to show up on your Backlog and Board, choose **Bugs are managed with requirements**.



For details, see Show bugs on backlogs and boards.

Correct your Kanban board configuration

If you see the following error when you open your Kanban board, you need to correct the configuration. The main reason for this error is that the workflow states of work item types that have been added to the Requirements category aren't mapped to the column.

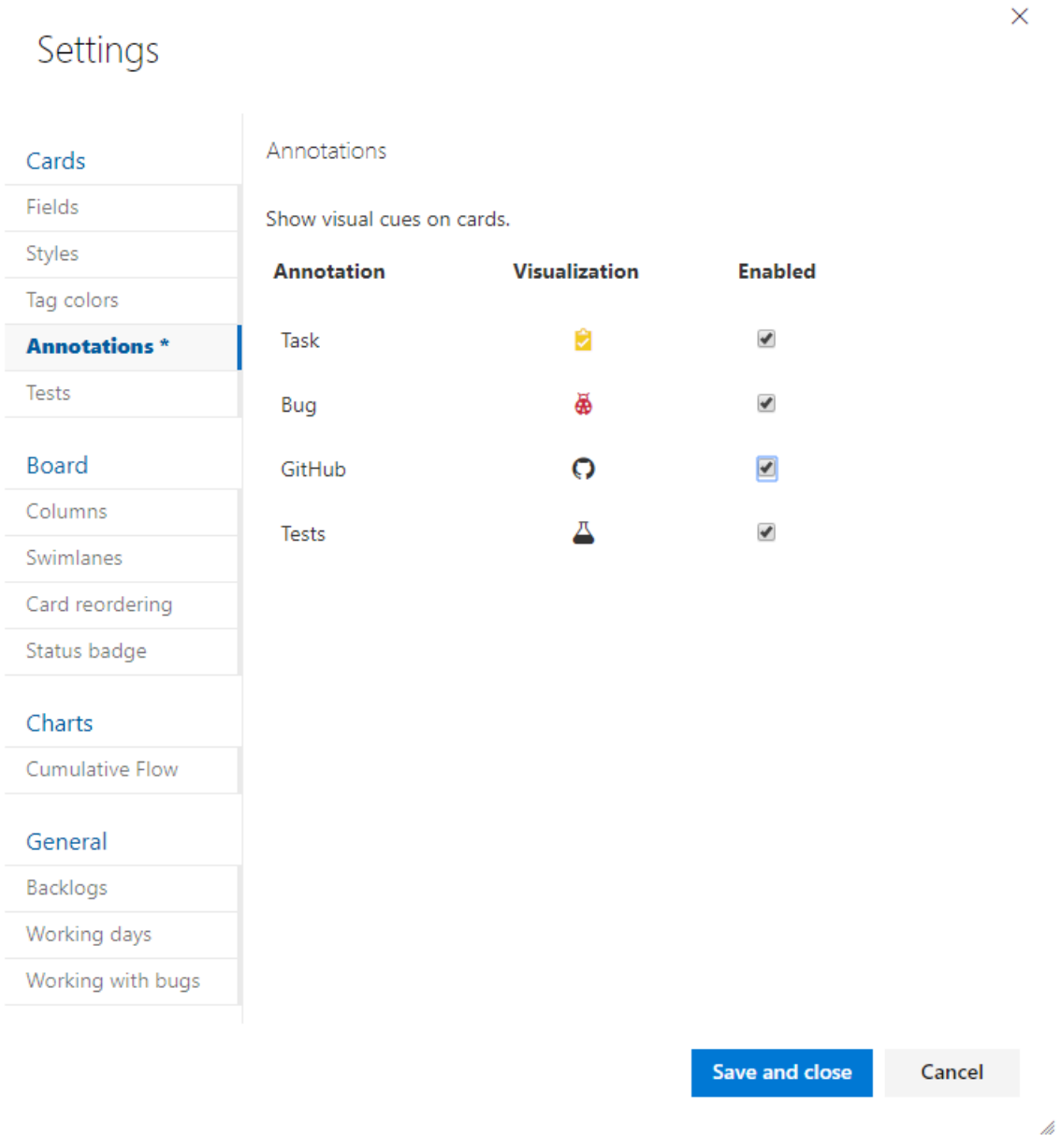


Choose **Correct this now** to open the Settings dialog. To map the workflow states, refer to Add columns to your Kanban board, Update Kanban column-to-State mappings.

Customize your Kanban board checklist items

Checklists are a great way to create work items that are automatically linked with a parent-child link to another work item on a Kanban board. You can customize the work item types that you can add as a checklist by opening the Board Settings, choose **Annotations**, and enable the work item types you want to appear on the board. For details, see Customize cards.

For example, here we've chosen to track bugs along with tasks, and enable Task, Bug, GitHub objects, and Tests to appear within checklists.



NOTE

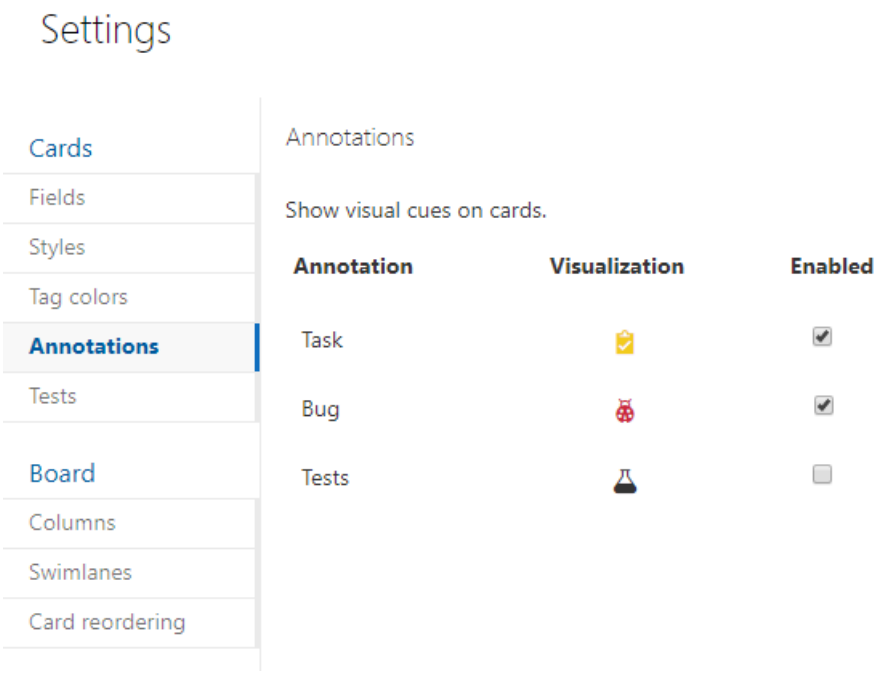
The GitHub annotations require Azure DevOps Server 2019 update 1 or later version.

To learn more about checklists, see the following articles:

- Add tasks or child items as checklists
- Add, run, and update inline tests

- Link GitHub commits, pull requests, and issues to work items

For example, here we've chosen to track bugs along with tasks, and enable Task and Bug to appear within checklists.



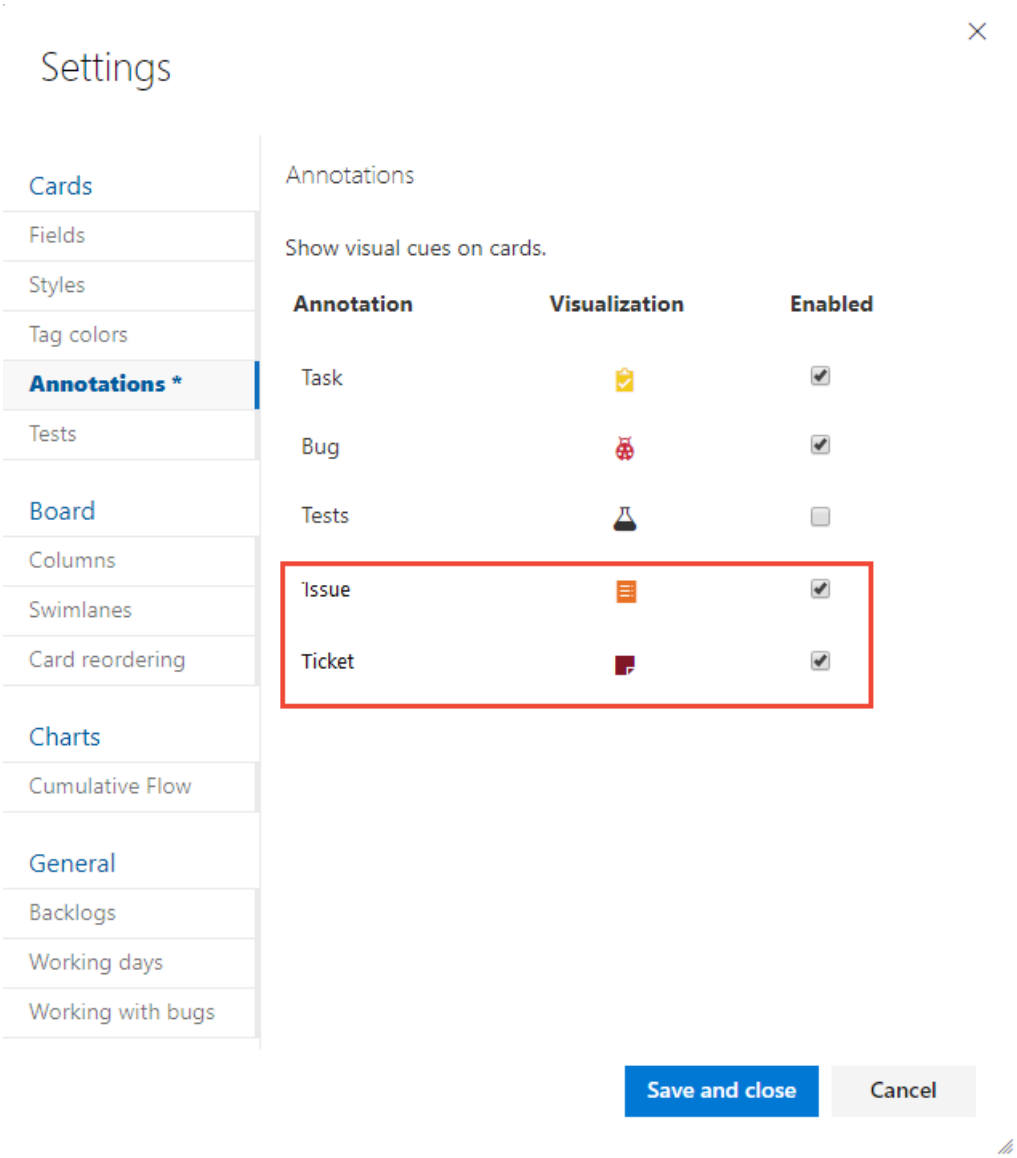
To learn more about checklists, see the following articles:

- Add tasks or child items as checklists
- Add, run, and update inline tests

Add other work item types to your Kanban board checklist

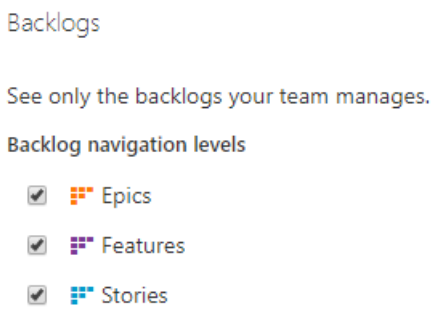
If you added work item types to the Task Category as described in Add custom work item types to your Taskboard later in this article, you can choose if these types appear within a checklist on your product Kanban board. You make this choice by opening **Board Settings**, choose **Annotations**, and enable the work item types you want to appear on the board. You can enable up to five annotations. For details see Customize cards.

For example, here we've chosen to track bugs along with tasks, and we enable Issue and Ticket and Task and Bug. To learn more about checklists, see Add tasks or child items as checklists and Add, run, and update inline tests.



Hide or show backlog levels

Your team can also choose to hide or show one or more backlog levels. Feature teams often manage backlog items, while management teams manage features and epics. In this situation, you can enable or disable a backlog level.



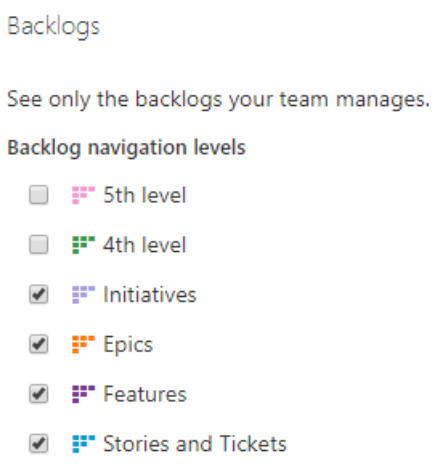
For details, see Select backlog navigation levels for your team.

Add custom work item types to your backlogs and portfolio backlog levels

If you want to track different work item types on your product backlog, you can do that by adding custom work item types and adding them to a specific backlog level.

You can also add custom work item types and add them to portfolio backlogs. You can add up to five portfolio backlogs.

For example, here we've added Initiatives, fourth level, and fifth level work item types to support five levels of portfolio backlogs. We've also added a custom work item type named Ticket and added that to the product backlog.



For details, see the following resources:

- Add and manage work item types (Inherited process)
- Customize your backlogs or boards (Inherited process)
- Customize an inheritance process

Inheritance process model:

- Add and manage work item types (Inherited process)
- Customize your backlogs or boards (Inherited process)
- Customize an inheritance process

On-premises XML process model:

- Add a work item type to a backlog and board
- Add portfolio backlogs
- Customize the On-premises XML process model

- Add a work item type to a backlog and board
- Add portfolio backlogs
- Customize the On-premises XML process model
- Add portfolio backlogs

Add custom work item types to your Taskboard

To add custom work item types to appear on your sprint Taskboard, follow the steps outlined next based on the process model your project uses.

> **NOTE**
> You can enable work item types that you add to your Iteration backlog to appear as a checklist on your product Kanban board. To learn how, see Customize your Kanban board checklist items provided earlier in this article.

Track custom work items with the Inherited process model

For example, if you want to track a custom work item type, Tickets, along with Tasks and Bugs, you do the following tasks:

1. Define the Ticket custom work item type. See Add and manage work item types.
2. Add the Ticket work item types to the Iteration backlog. For details, see Customize your backlogs or boards for a process.

Add existing and custom work item types with the On-premises XML process model

For on-premises deployments that use the On-premises XML process model to customize work tracking, you can add existing and custom work item types to the sprint Taskboards. For example, if you want to track Issues (or Impediments for the Scrum process) and a custom work item type, Tickets, along with Tasks and Bugs, you would do the following tasks:

1. Define the Ticket custom work item type. See Add or modify a work item type.
2. Add Issue and Ticket work item types to the Task Category by modifying the Categories XML file. For details, see Categories XML element reference.

   For example, here we add Issue and Ticket to the Task Category.

   ```
   <CATEGORY name="Task Category" refname="Microsoft.TaskCategory">
     <DEFAULTWORKITEMTYPE name="Task" />
   <WORKITEMTYPE name="Issue" /
   <WORKITEMTYPE name="Ticket" /
   </CATEGORY>
   ```

3. Make sure that the Issue and Ticket workflow states are mapped to category states. As needed, modify the ProcessConfiguration XML file to add Issues and Tickets to the `TaskBacklog` section.

   For example, here the New, Active, and Closed states are mapped for the Task Category.

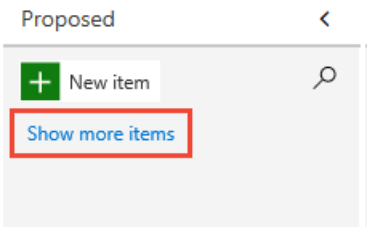
   ```
     <TaskBacklog category="Microsoft.TaskCategory" pluralName="Tasks" singularName="Task"
   workItemCountLimit="1000">
       <States>
         <State value="New" type="Proposed" />
         <State value="Active" type="InProgress" />
         <State value="Closed" type="Complete" />
       </States>
   . . .
     </TaskBacklog>
   ```

4. To verify your changes, open a Sprint backlog and make sure you can add an Issue or Ticket in the same way you add a Task. See Add tasks.

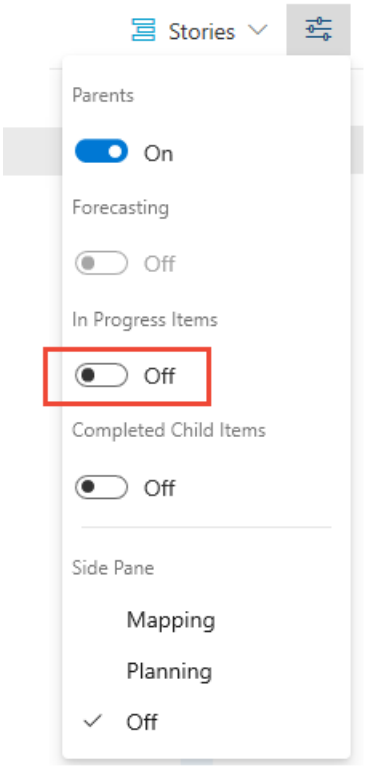
Other factors that can affect work items in your backlogs and boards

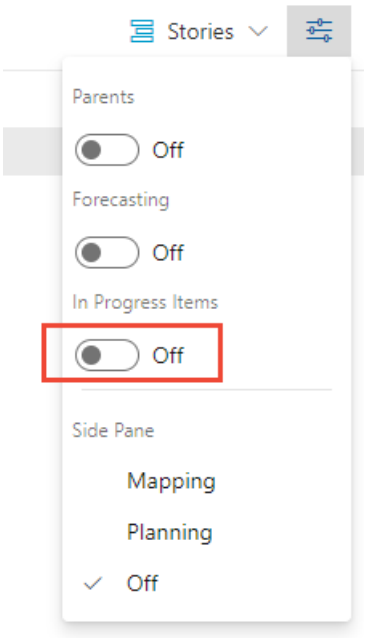
The following settings can influence the type and number of work items that will appear in your backlogs and boards.

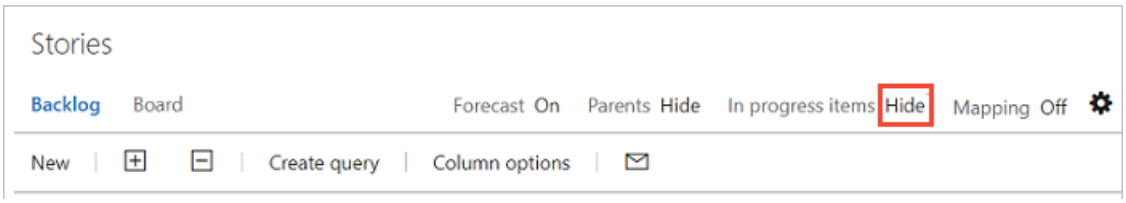
- In your Kanban board, newly added work items may not appear if they're stack ranked lower within the product backlog. By choosing **Show more items**, you can cause the board to refresh and display more work items.



- If you have nested work items that belong to the same category, only leaf nodes may appear on the Kanban board (for TFS 2018.1 and earlier versions). For this reason, we recommend that you don't nest work items of the same work item type or belonging to the same category. To learn more, see Fix reordering and nesting issues, How backlogs and boards display hierarchical (nested) items.
- If you've turned off the **In Progress** view, then those work items where work has started won't appear in the backlog list.







- Work items appear in the priority order in which they're added or moved to. This order or sequence is managed by the **Stack Rank** (Basic, Agile, and CMMI processes) or **Backlog Priority** (Scrum) field. To learn more, see the Stack rank section in Backlogs, portfolios, and Agile project management.
- Each backlog can display up to 999 work items. If your backlog exceeds this limit, then you may want to consider adding a team and moving some of the work items to the other team's backlog.
- Sprint backlogs show only those work items that meet the team's area path and the **Iteration Path** defined for the sprint.
- Inheritance process model: If an administrator disables or deletes a work item type, it will no longer appear on backlogs and boards.
- On-premises XML process model: If an administrator deletes or destroys a work item type, it will no longer appear on backlogs and boards.

Related articles

- Add a team, move from one default team to several teams
- Refine your backlog
- Create your backlog
- Backlog priority or stack rank order
- Use categories to group work item types
- Workflow states & state categories

Configure your backlog view in Azure Boards

10/4/2022 • 11 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Your backlogs are designed to support many project management tasks. Chief among them are:

- Define work to be done
- Prioritize work
- Group work into a hierarchical view
- Assign work to iterations
- Forecast work

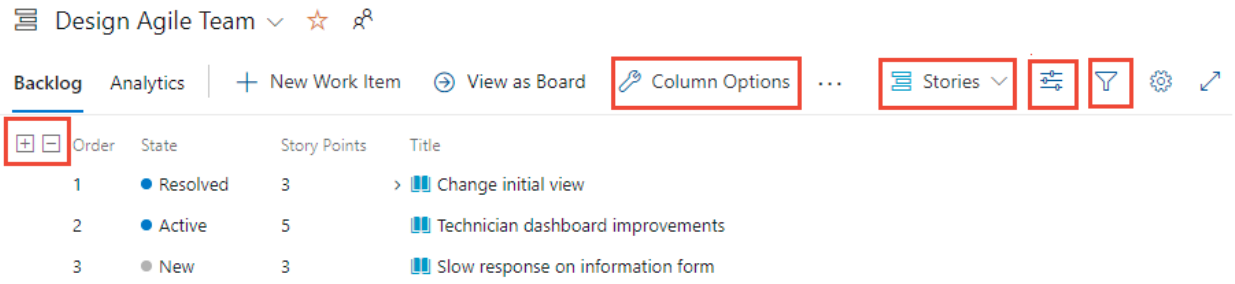
Each backlog—product or portfolio—is a tool you share with your team members. When you add backlog items, prioritize items, or link work items using parent-child links, team members see the changes when they refresh their backlog.

To effectively perform select tasks, it's good to know how to set your view options to support those tasks.

> **TIP**
> You can't sort your backlog on a column. To view a sorted listed, select **Create query** from your backlog. Save and open the query. Modify the query if needed to be a flat list query. You can then sort the query results. To learn more about queries, see Use the query editor to list and manage queries.

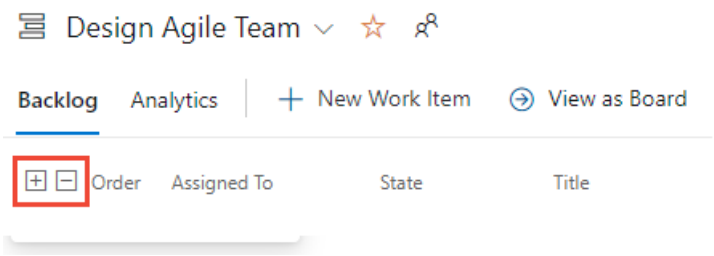
Backlog configuration options

You have the following tools to configure your backlog view: **Expand/Collapse one level**, **Column Options**, **Backlog level selector**, **View options**, and **Filter** toolbar. The options you set for each backlog level—stories, features, epics (Agile process) or product backlog items, features, epics (Scrum process)—are distinct. The options you set persist until you change them.



Expand and collapse the hierarchy

The default view when you select a backlog level is to show the collapsed view. Only those items associated with the backlog level selected are shown. You can expand and collapse the hierarchy by using the ⊞ and collapse ⊟ icons to expand or collapse one level of the hierarchy. Your selection doesn't persist once you switch to a different page or view.



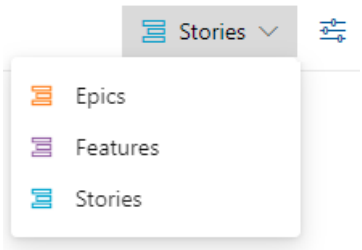
Backlog level selector

The number of backlog levels available to you are set by your team administrator and may have been customized to add custom work item types or backlog levels. Each backlog automatically applies the filters associated with the **Area Paths** and **Iteration Paths** selected for the team.

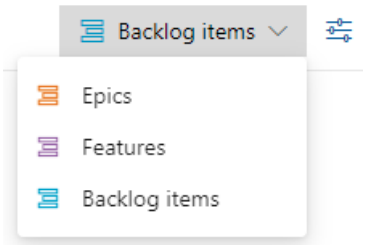
> **NOTE**
> Prior to using the tools described in this article, we recommend that you review Set up your project's backlogs and boards to ensure you have configured your backlog to support your team's needs.

From the **Backlogs** page, you can select the product backlog or a portfolio backlog. You select a backlog from the backlog level selector next to the **View options** icon. The labels within this selector vary depending on the process selected for your project, customization made to that process, and configurations made by your team administrator as shown in the following images.

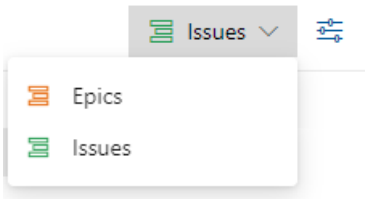
Agile process



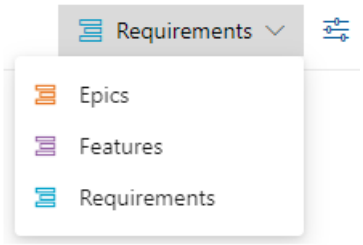
Scrum process



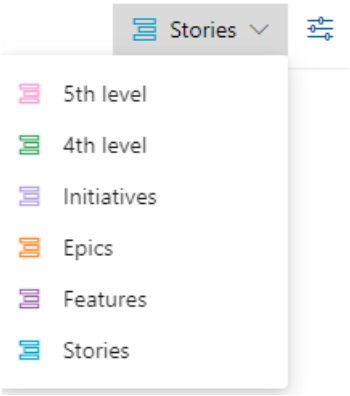
Basic process



CMMI process



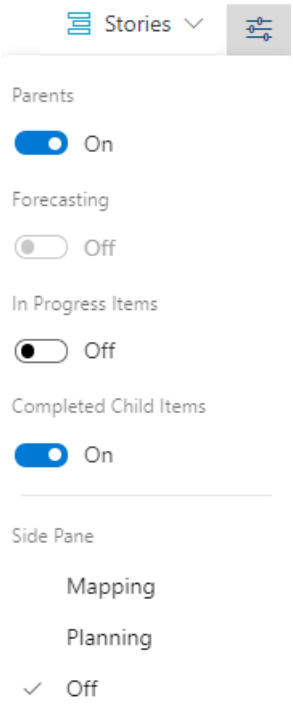
Customized process



For information on team configuration of backlog levels, see Select backlog navigation levels for your team.

View options menu

The **View options** menu controls the following options.



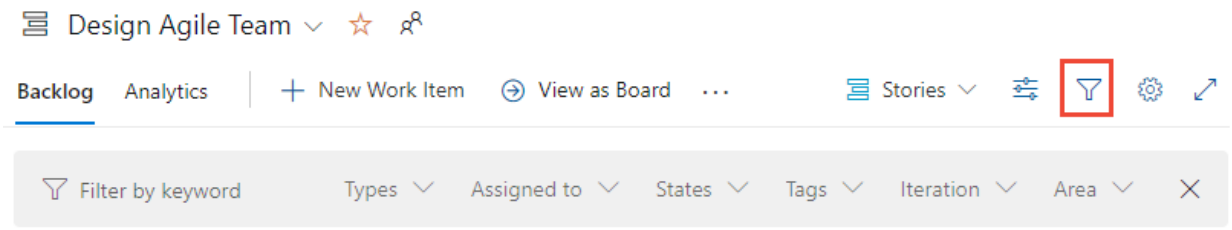
- **Parents**: Show the hierarchical grouping of parent-child work items. Useful when adding child work items, reparenting a work item, or displaying rollup columns.
- **Forecasting**: Show the **Forecast** tool and forecast lines. The **Forecast** option only appears for the first-level backlog and depends on the assignment of **Story Points**, **Effort**, or **Size**.
- **In Progress Items**: Show items whose workflow **State** corresponds to an *In Progress* workflow state category. If you turn the **In Progress** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or a custom workflow state defined in the In Progress state category won't appear in the backlog. To learn more about category workflow states, see How to use workflow states and state

categories.

- **Completed Child Items**: Show child items that have been completed. Typically you turn this **On** when reviewing reviewing a rollup column.
- **Mapping**: Shows the **Mapping** pane to support drag-and-drop linking of work items to parent items. The **Mapping** option doesn't appear when you've selected the highest backlog level configured for your team.
- **Planning**: Shows the **Planning** pane to support drag-and-drop of work items to **Iteration Paths**.

Filter bar

Turn on filtering when you want to find one or more work items based on a keyword, tag, assignment, or other field you display using **Column Options**. You enable the filter feature by choosing **Filter**.



Filtering displays a flat list of all items in the hierarchy when you have selected to show **Parents**. The hierarchical grouping is restored once you dismiss the filter toolbar. The filter toolbar persists until you dismiss it.

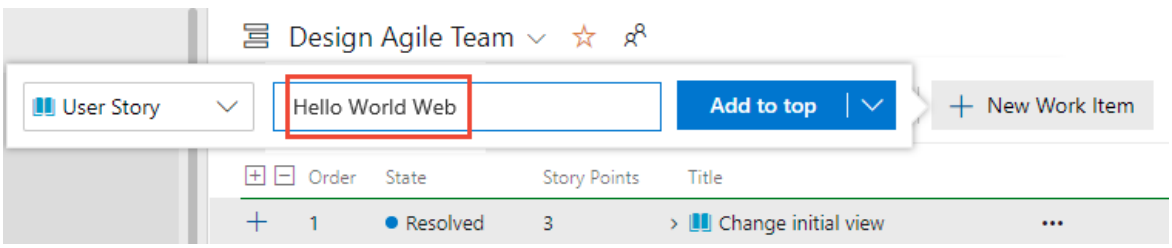
To learn more, see Filter backlogs, boards, and plans.

Define backlog items

It's useful to be able to add work items quickly and refine details later when more information becomes available. Often you use a query to help triage work added to the backlog to review, refine, and add details to work items that you entered through your backlog.

To quickly define many items to a backlog, perform the following steps.

1. Select the backlog level that you want to add items to.
2. From the **View options** menu, turn the slider for **Parents** and **Forecasting** to **Off**.
3. (Optional) turn **In Progress Items** on or off.
4. (Optional) Minimize the number of columns displayed on your backlog, or select those fields you want to view.
5. Choose **New Work Item**, enter a title, choose **Add to top** or **Add to bottom**, and then press **Enter**. We recommend you add items to the bottom of the backlog if your team has a process for prioritizing backlog items.



Work items are automatically assigned the default **Area Path** and **Iteration Path** selected for the team.

> **NOTE**
> If you have **Stakeholder** access , you can only add work items to the bottom of the backlog. For details, see Stakeholder access quick reference.

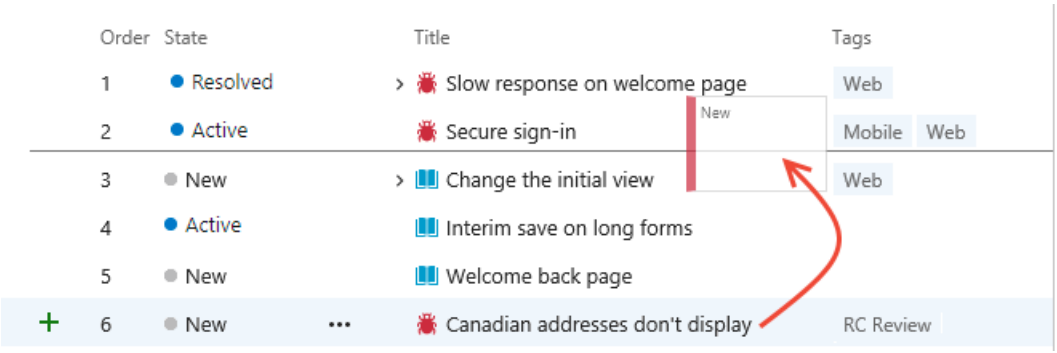
6. Continue entering a **Title** and pressing the **Enter** key to define more backlog work items.

To learn more, see Create your product backlog and Define features and epics.

Prioritize your product backlog

If your team follows Agile or Scrum methods, they'll want to prioritize the backlog to make sure that the most important work to complete is situated at the top of the backlog. To prioritize a backlog, follow these steps.

1. Select the backlog level you want to prioritize.
2. Turn the **Parents** view option **Off**.
3. Drag the work items up or down within the backlog. Or, if you prefer to use the keyboard, hold down the Alt key and use the up and down arrows.



To reorder a backlog, you must have **Basic** or higher level access. If you have Stakeholder access, you can't reorder backlog items. For more information, see Stakeholder access quick reference.

> **NOTE**
> Changes you make to the priority impact all backlog items. When other team members refresh their backlog, they'll see the new priorities. A background process updates the **Stack Rank** (Agile, Basic, and CMMI processes) or **Backlog Priority** (Scrum process) fields. These fields are used by the system to track the relative ranking of items on the product, feature, epic, or other portfolio backlog. By default, these fields don't appear on the work item form. The priority ranking is assigned separately to each backlog level, which you can check by adding the field to a backlog and viewing a hiearchical list.

Backlogs that participate in portfolio management or that contain nested same-type child items might not allow you to reorder the items. For more information, see these articles:

- Backlogs, portfolios, and Agile project management, Work with multi-team ownership of backlog items
- Fix reordering and nesting issues

Prioritize a portfolio backlog

The method for prioritizing a portfolio backlog is similar to that described for a product backlog. The main difference is that you prioritize child items within each portfolio item. Each backlog level—**Stories**, **Features**, **Epics**—supports priority ordering distinct from every other level.

Prioritize the portfolio items:

1. Select the portfolio backlog level you want to prioritize.
2. Turn the **Parents** view option **Off**.
3. Drag the work items up or down within the backlog.
4. Within each item, you can expand to see child items and drag these items into priority order.

Prioritize child items:

1. Expand each portfolio item.
2. Drag each child item up or down within the expanded item.

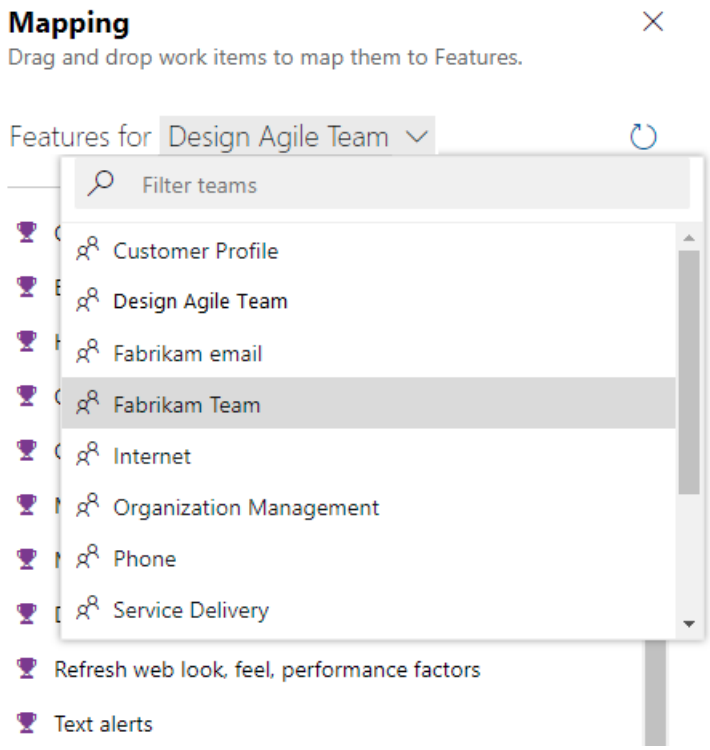
> **NOTE**
> To reorder a backlog, you must have **Basic** or higher level access. If you have Stakeholder access, you can't reorder backlog items. For more information, see Stakeholder access quick reference.

Link work items to a parent (mapping)

You can drag items to quickly link one or several work items to a parent portfolio item.

> **TIP**
> Prior to opening mapping work items, add the portfolio backlog items you want to link work items to and prioritize them. The **Mapping** pane lists the portfolio backlog items in priority order.

1. Select the backlog level you want to link to parent items. For example, choose **Stories** to link to **Features**.
2. Open **View options** and choose **Mapping**. The Mapping pane opens. By default, the pane lists the next-level up portfolio items for the current team.
3. (Optional) To map items to parent items owned by a different team, choose it from the team selector in the **Mapping** pane as shown in the following image.



4. Drag work items from the backlog to the portfolio item listed in the **Mapping** pane. The system creates a parent-child link in the background. The backlog item turns bold and then unbold as the system saves the changes.

   Note, you can select multiple backlog items and drag them to a portfolio item. To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the **Ctrl** key. Then, you can drag the selected items.

5. (Optional) You can also drag a backlog item within the expanded hierarchical view to reparent a work item.

> **TIP**
> To view the work items that are unparented, you can add the **Parent** field as a column. The **Title** of the parent item is listed for those items that have been linked to a parent.

To learn more, see Organize your backlog and map child work items to parents.

Add child items to a portfolio backlog item

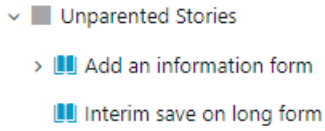
2. Select the portfolio backlog level, such as **Features**, that you want to add items to.
3. Choose + **Add User Story, Bug** for the feature you want to add the child item to as shown in the following image. Your labels may differ based on process and customizations.
4. In the work item form that appears, enter a **Title** and any other required fields or details. Save the work item to close it.

To learn more, see Define features and epics, add child items.

View or find unparented work

To view or find unparented work:

1. Select the backlog level you want to inspect for unparented items.
2. Open **View options** and choose **Parents**.
3. Scroll to the bottom of the backlog and expand **Unparented Stories**, **Unparented Features**, or similar entries. Unparented work items are listed under these entries.



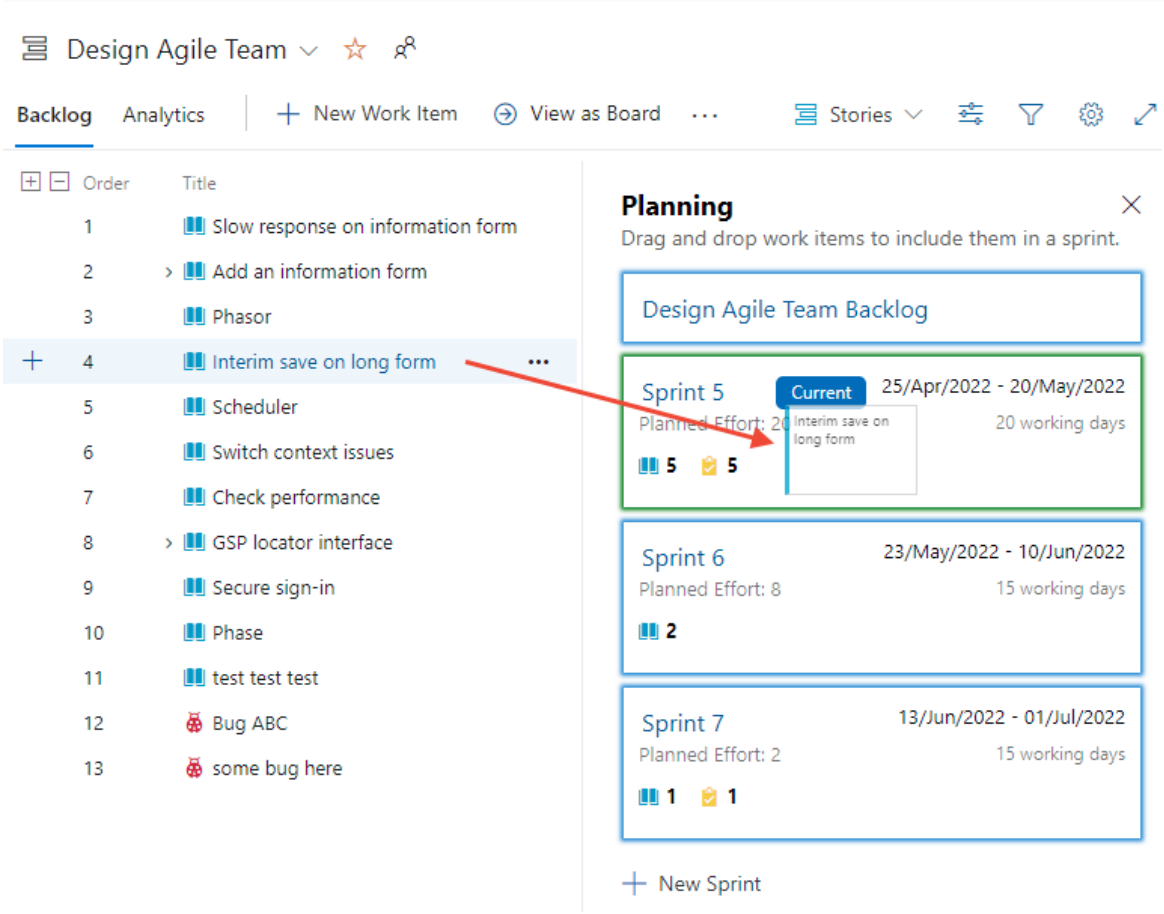
Assign work to a sprint or iteration

Similar to using the **Mapping** pane, you can use the **Planning** pane to assign one or more work items to an **Iteration Path** or sprint.

1. Make sure all **Iteration Paths** have been selected for your team that you want to show in the **Planning** pane.
2. Choose the backlog level that contains the work items you want to assign.
3. Open **View options**, turn off **Completed Child Items** and choose **Planning**.
4. Drag work items from the backlog to the portfolio item listed in the **Mapping** pane. The system creates a parent-child link in the background. The backlog item turns bold and then unbold as the system saves the

changes.

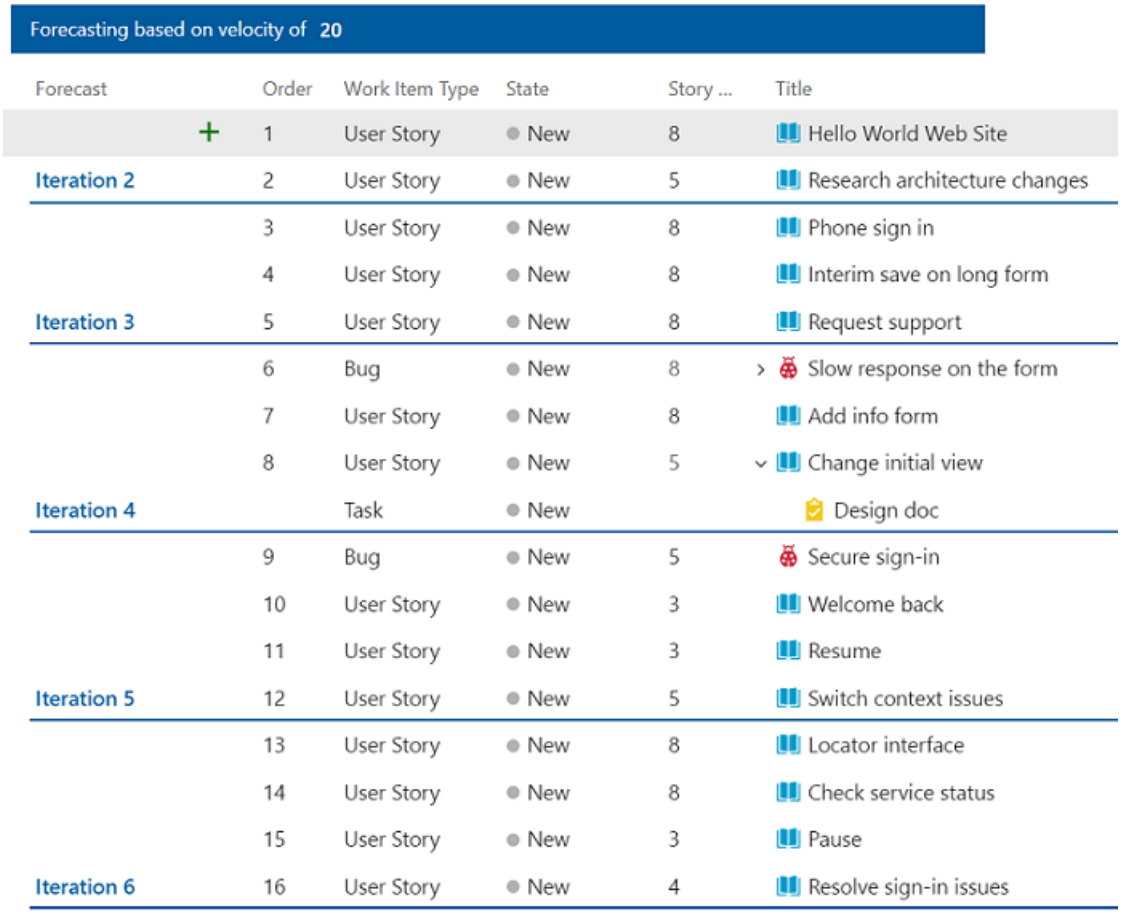
Note, you can select multiple backlog items and drag them to a portfolio item. To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the **Ctrl** key. Then, you can drag the selected items.



Forecast a backlog

The **Forecast** tool is only available for the product backlog. Use these steps to forecast your backlog. To use the **Forecast** tool, you must have **Basic** or higher level access. This feature isn't available to users granted **Stakeholder** access.

1. Make sure that future **Iteration Paths** have been selected for your team.
2. Choose the backlog level for your team.
3. (Optional) Add the **Story Points**, **Effort**, or **Size** field as a column based on the process your project uses.
4. Open **View options** and turn off **Parents** and **In Progress** items, and **Completed Child Items**. Turn on **Forecast**.
5. Enter a velocity estimate in the **Forecasting based on velocity** of box.
6. Review the forecast lines that appear, similar to the ones shown in the following image.



The forecast tool doesn't reference any iteration assignments made to the product backlog items.

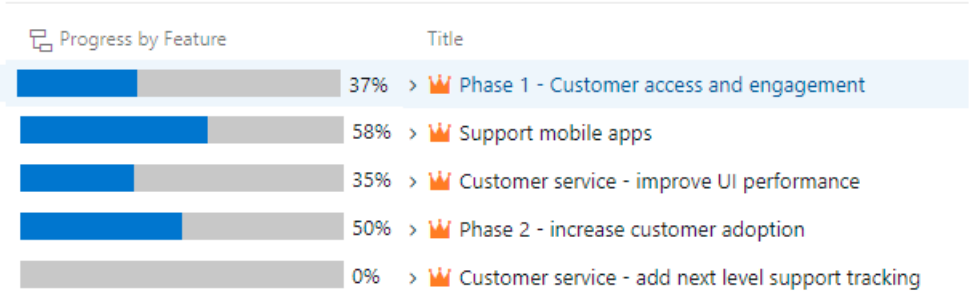
> **TIP**
> You can drag items to reprioritize them with forecast lines shown. You can also use the **Planning** pane with the **Forecast** tool turned on.

To learn more, see Forecast your product backlog.

Review progress made to your backlog

You can add a rollup progress bar, count of work items, or sum of any integer or numeric field as a column to the backlog. This option allows you to review progress made to parent work items based on the completion of their child items. These child items can be Tasks for User Stories, User Stories and Bugs for Features, or Features for Epics.

1. Select the backlog level you want to view progress on
2. Open **View options**, show **Completed Child Items**
3. Open **Column Options**, choose **Add a rollup column**, and select the progress bar or count to display.



It can take several moments for the progress bar or count to appear.

To learn more, see Display rollup progress or totals.

Related articles

- Set up your project's backlogs and boards
- Create your product backlog
- Define features and epics
- Organize your backlog and map child work items to parents
- Configure team settings

Bulk modify tools

- Bulk modify (web)
- Bulk add or modify (Excel)
- Import or update work items in bulk by using CSV files

Manage columns in a work item list in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Each column corresponds to a work item field. You can add and remove columns within work item lists to show the fields of interest to you. Or, you can drag a column to a new position. Your settings persist for each page you customize and are only valid for your views.

Specifically, you can do the following actions from the following list views:

Action	Backlogs	Sprint backlogs	Queries	Work items
Add or remove a column field	Yes	Yes	Yes	Yes
Add or remove the Parent field	Yes	Yes	Yes	Yes
Add or remove a rollup column	Yes	No	No	No
Sort on a column	No	No		

Yes

Yes

> **TIP**
> Unlike a query result, you can't sort a backlog by a column. However, you can use the **Create Query** link on each backlog to create a query that you can sort on any field column you choose from the **Sorting** tab of the Column options dialog. While you may be able to add a field to sort on, not all fields are supported. For example, selection of the **Parent**, **History**, **Description**, or other rich-text field will result in the display of an error message as you can't sort on these fields.

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. You can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. If your project uses the Inherited process model, you can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project.

About persistence of column options

Once you set the column options for a specific view, your settings persist until you change them. The following notes apply to specific views.

- Column options you set for a backlog apply only to the active team and backlog. Options you set for the product backlog differ from the options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query.
- Column options you set for one of the supported Work items views persists for a specific view only, such as **Assigned to me**, **Following**, **Mentioned**, and so on.

- Column options you set for a backlog apply only to the active team and backlog. Options set for the product backlog differ from those options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query. For Shared queries, your settings can get overwritten when someone else saves different column option settings.

> **NOTE**
> You can't set column options for other members of your team, nor can you set default column options.

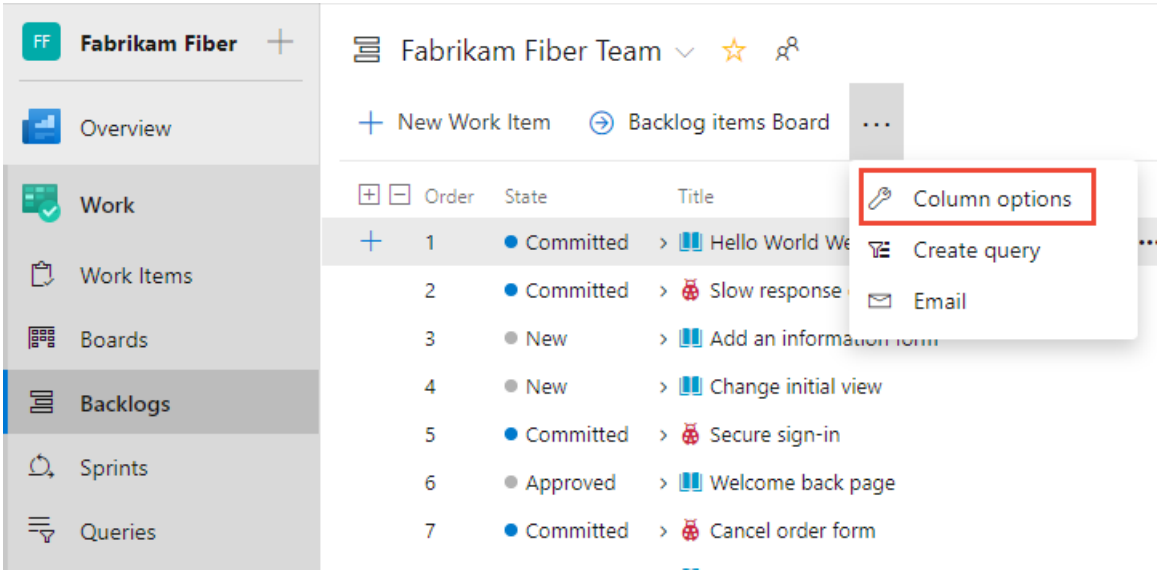
> **NOTE**
> You can't set column options for other members of your team. Also, for projects that use the Inheritance process model, you can't set default column options. For projects that use the On-premises XML process model, you can set the default column options for product, portfolio, and sprint backlogs. To learn how, see Process configuration XML element reference.

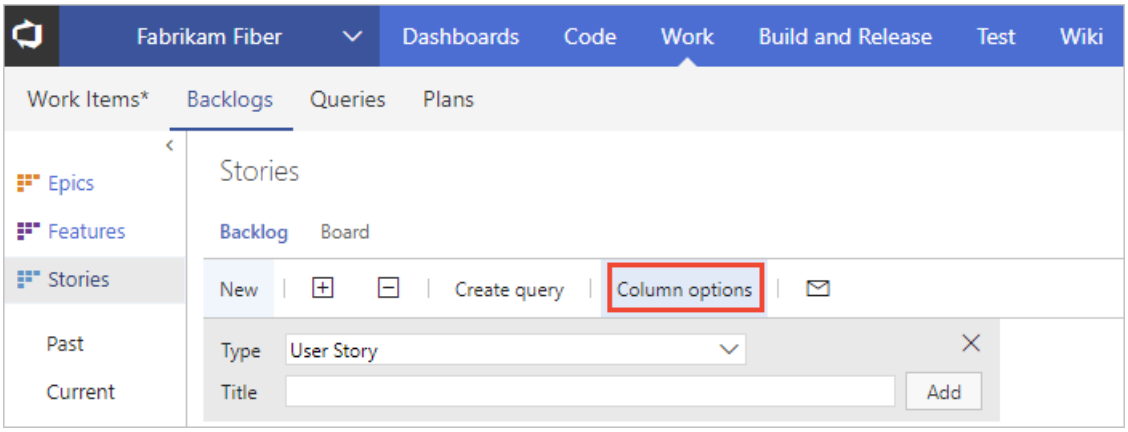
NOTE

You can't set column options for other members of your team. For projects that use the On-premises XML process model, you can set the default column options for product and portfolio backlogs. To learn how, see Process configuration XML element reference.

Open the Column options dialog

Start by opening the **Column Options** dialog. If you don't see the option, choose the ... and choose from the options provided.





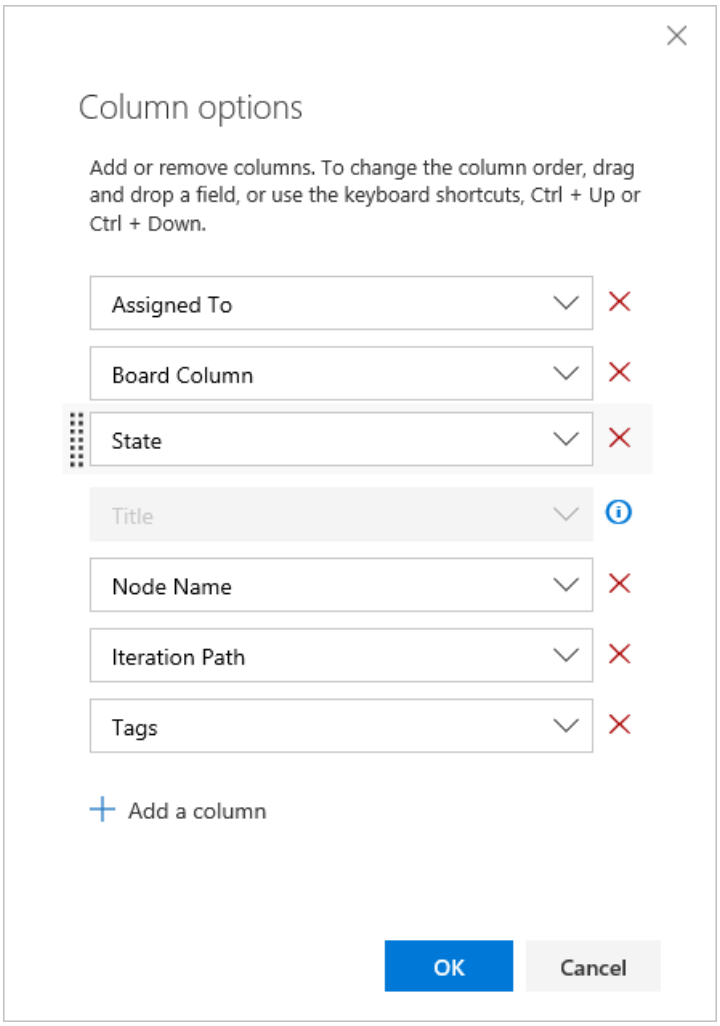
Add or remove columns

- Browser
- Visual Studio

In the Column options dialog, choose **Add a column** to add a field that isn't shown. To change the order of the fields, drag-and-drop the field where you want it within the set of selected fields. And, to remove a field, choose the ✕.

NOTE

The following dialog is available from TFS 2018.2 and later versions.

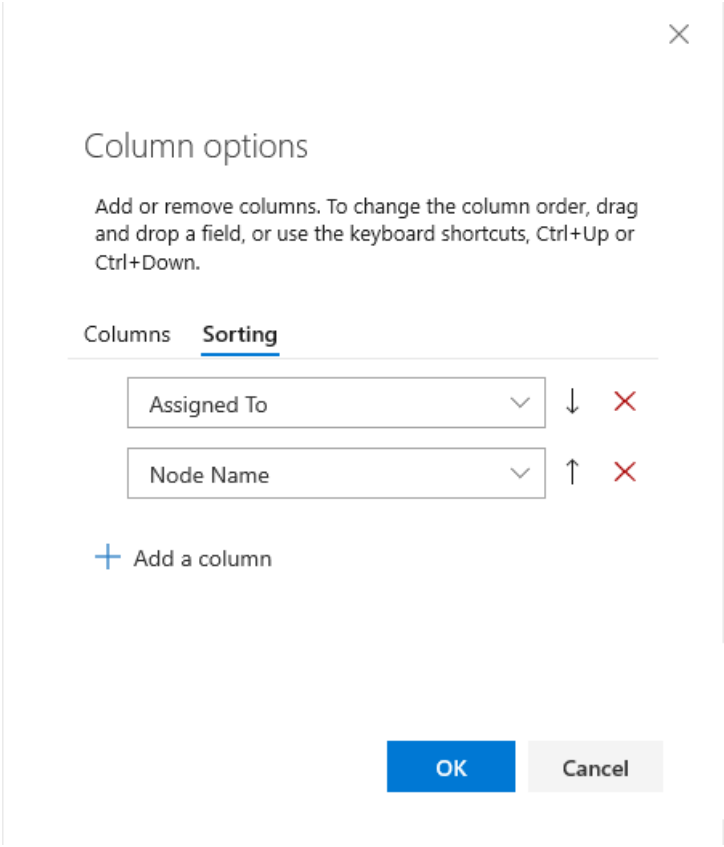


Add or remove rollup columns

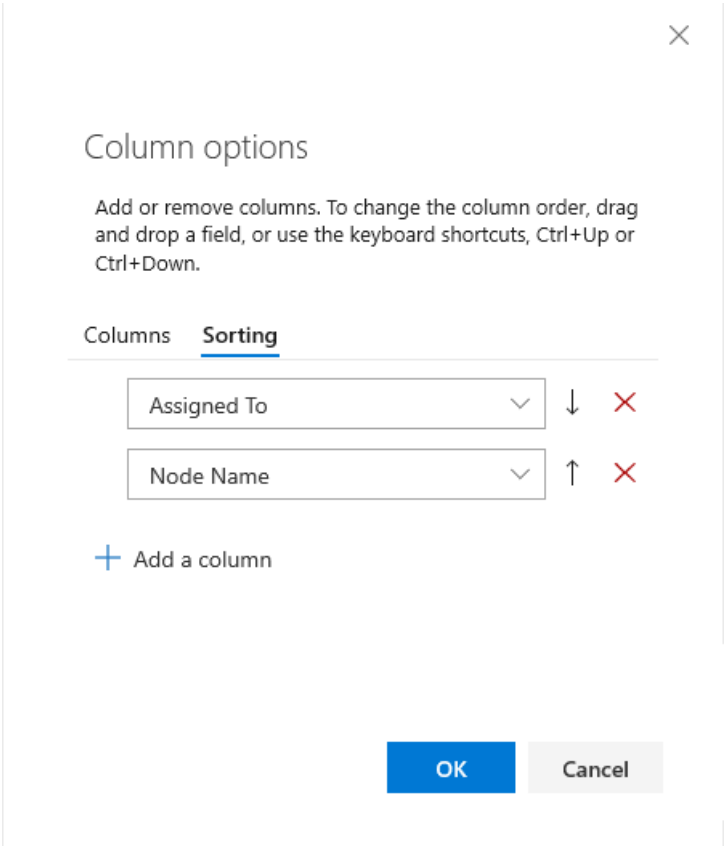
Rollup columns can display progress bars or the sum of numeric fields of child items. You can add them to any product or portfolio backlog. To learn more, see Display rollup progress or totals.

Sort on a column

You can sort query results and Work items views. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



You can sort query results. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



Use keyboard shortcuts to change the column order, column width, or sort options

You can change the column order, column size, or sort options by using the following keyboard commands:

- To change the column order, choose the field and drag it to a new location
- To resize a column, choose the column divider to the right of the field and drag to a new location
- For query results:
  - Add the field as a column in order to sort by that field
  - To sort by a column, hold down the SHIFT key and select on the field
  - To reverse the sort order, SHIFT+click on the field
  - To sort by multiple columns, SHIFT+click on each column in the order you want to sort

For other keyboard shortcuts, enter **?** to display available shortcuts based on the page you're on.

Related articles

- Display rollup progress or totals
- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- View, run, or email a work item query
- Create managed queries
- Customize a sprint Taskboard

- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- Create managed queries

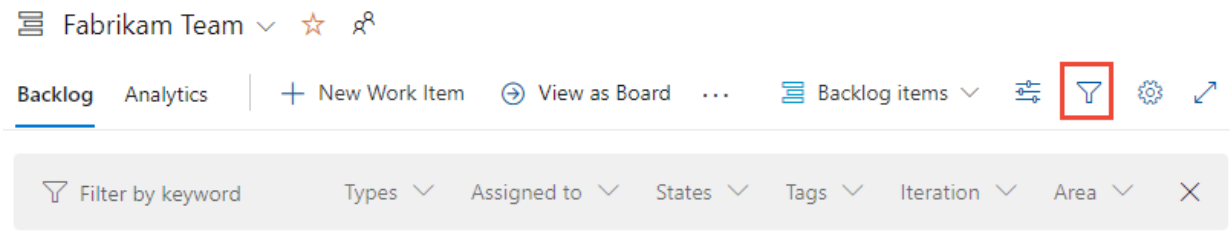
Interactively filter backlogs, boards, queries, and plans in Azure Boards

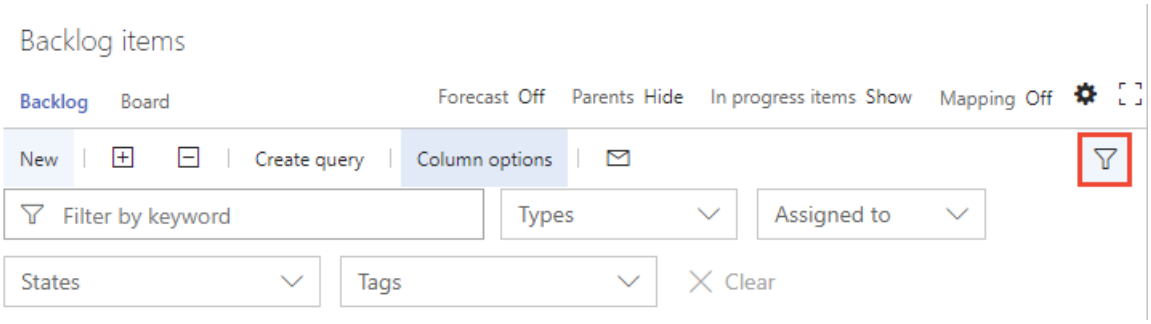
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

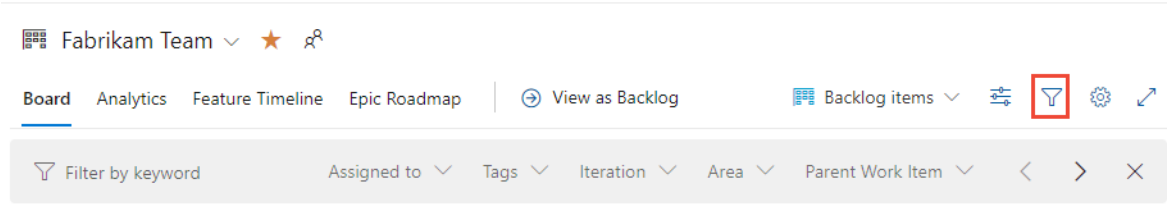
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

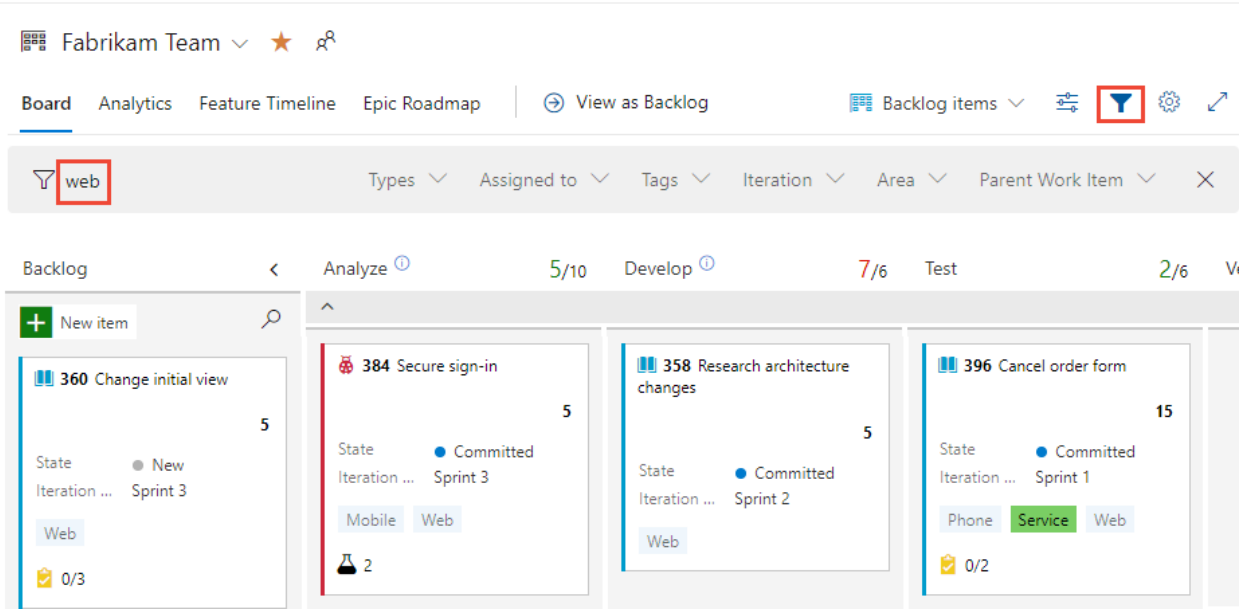
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

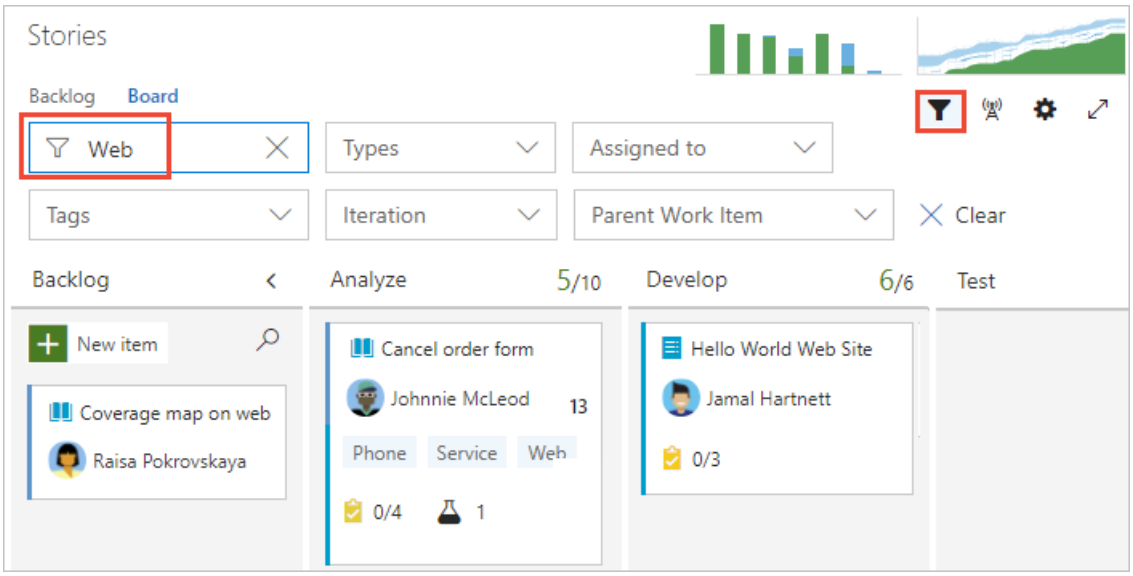
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

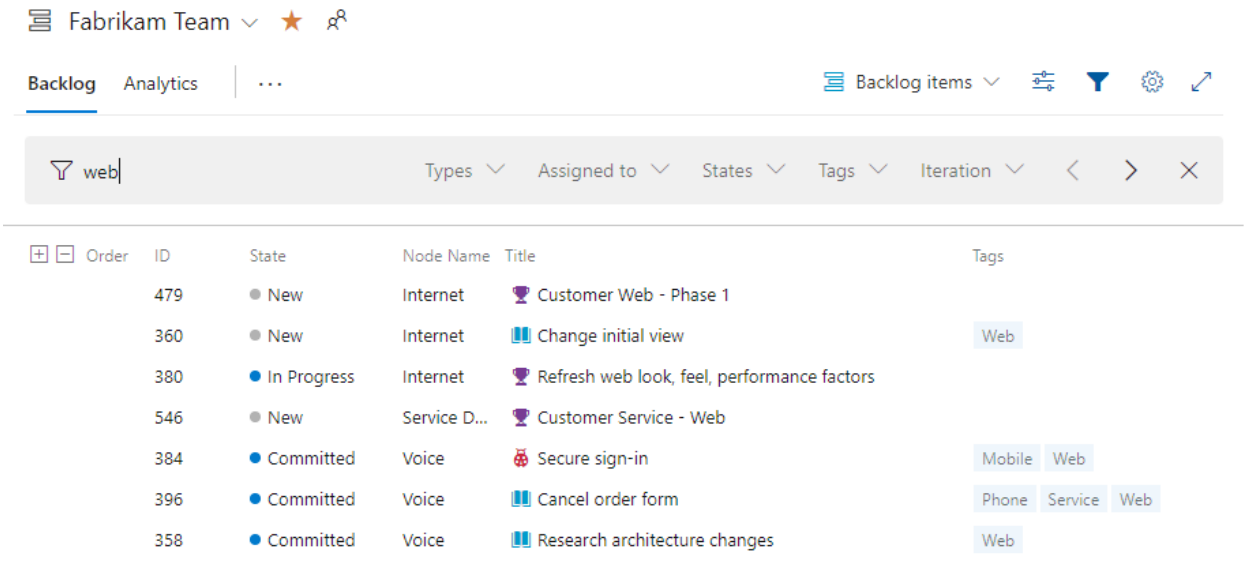


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

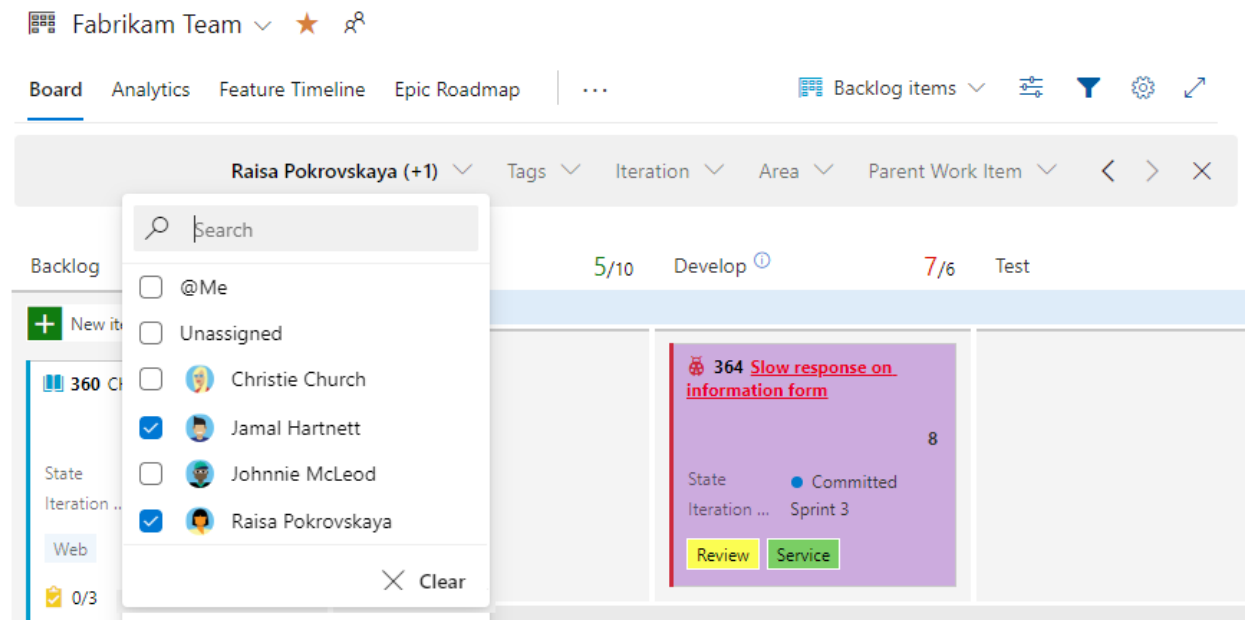
Features board

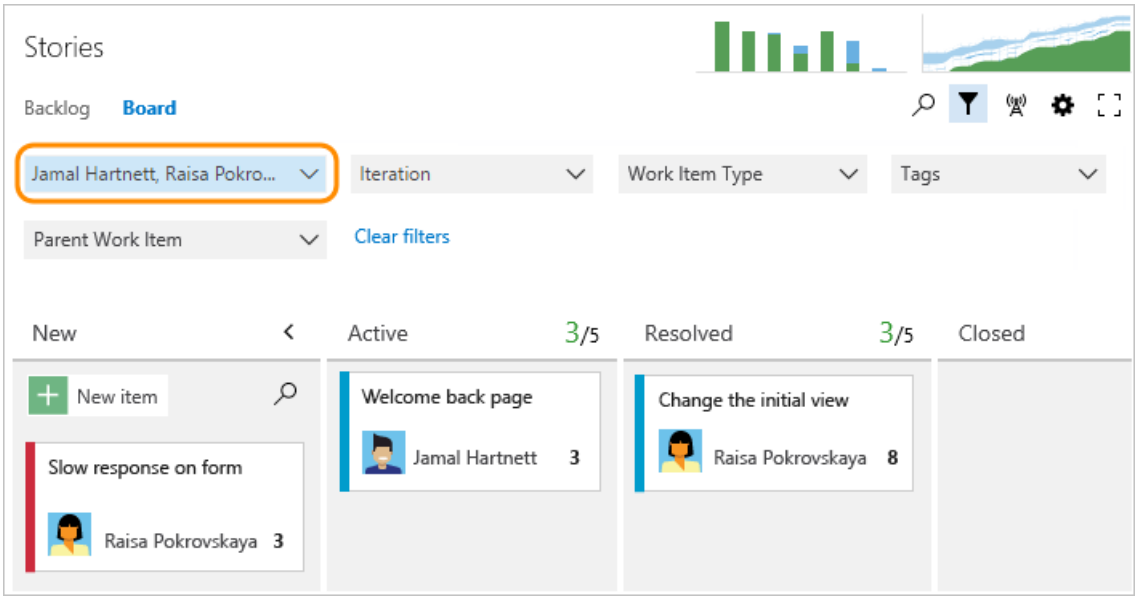
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

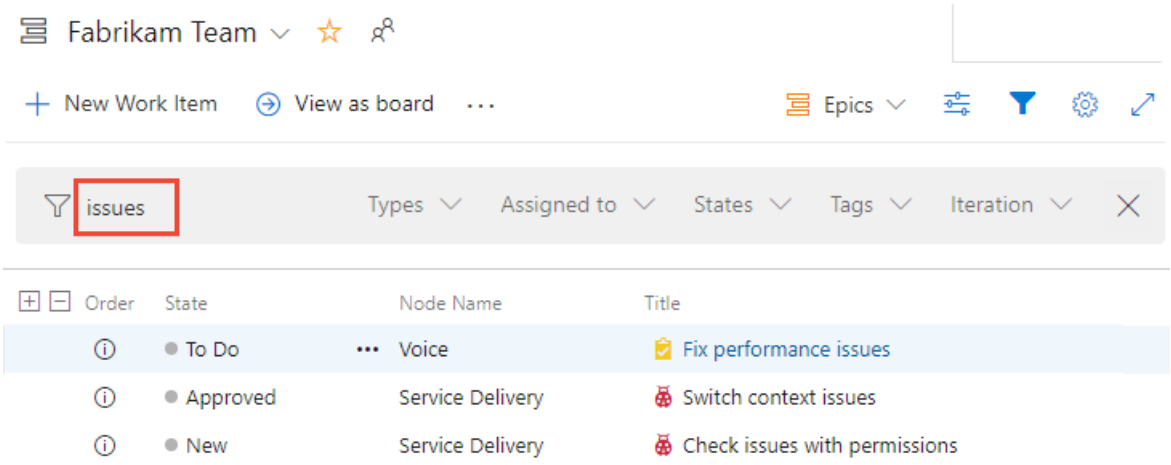
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

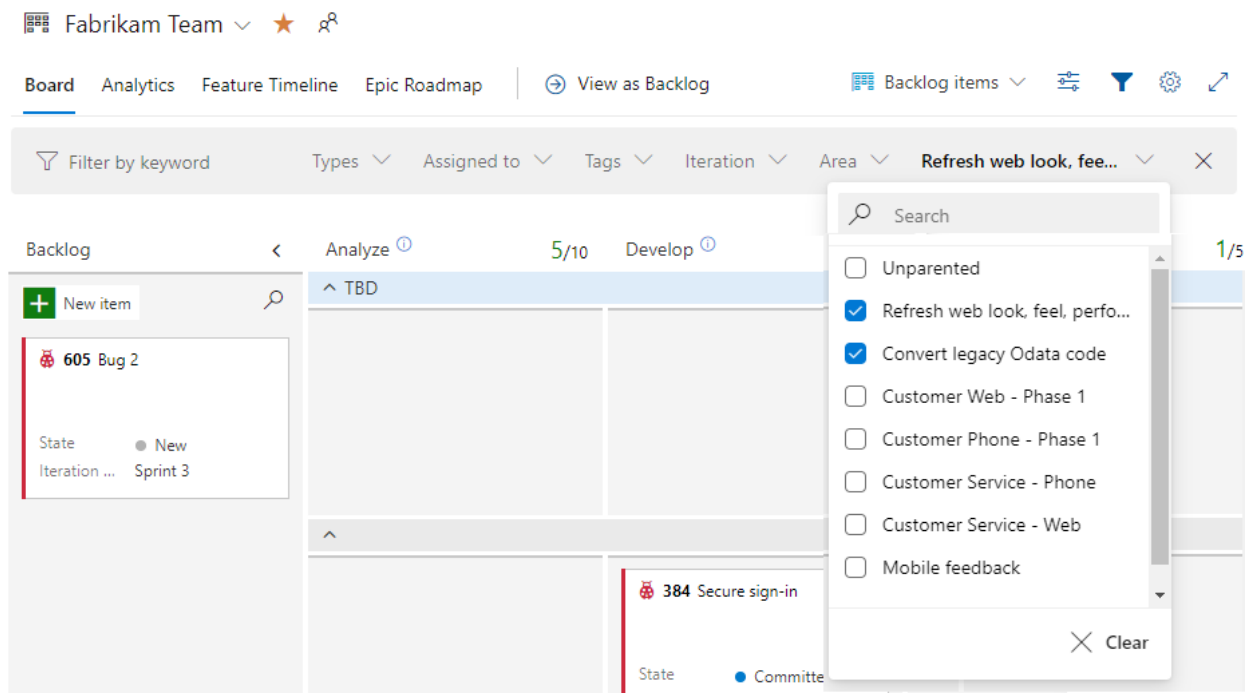
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

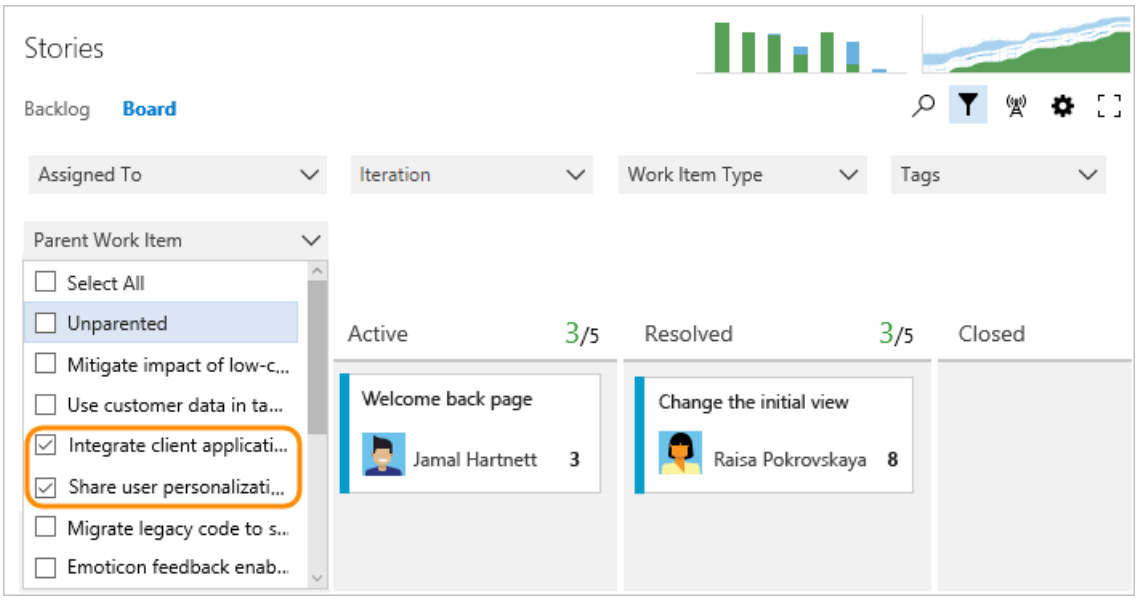
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



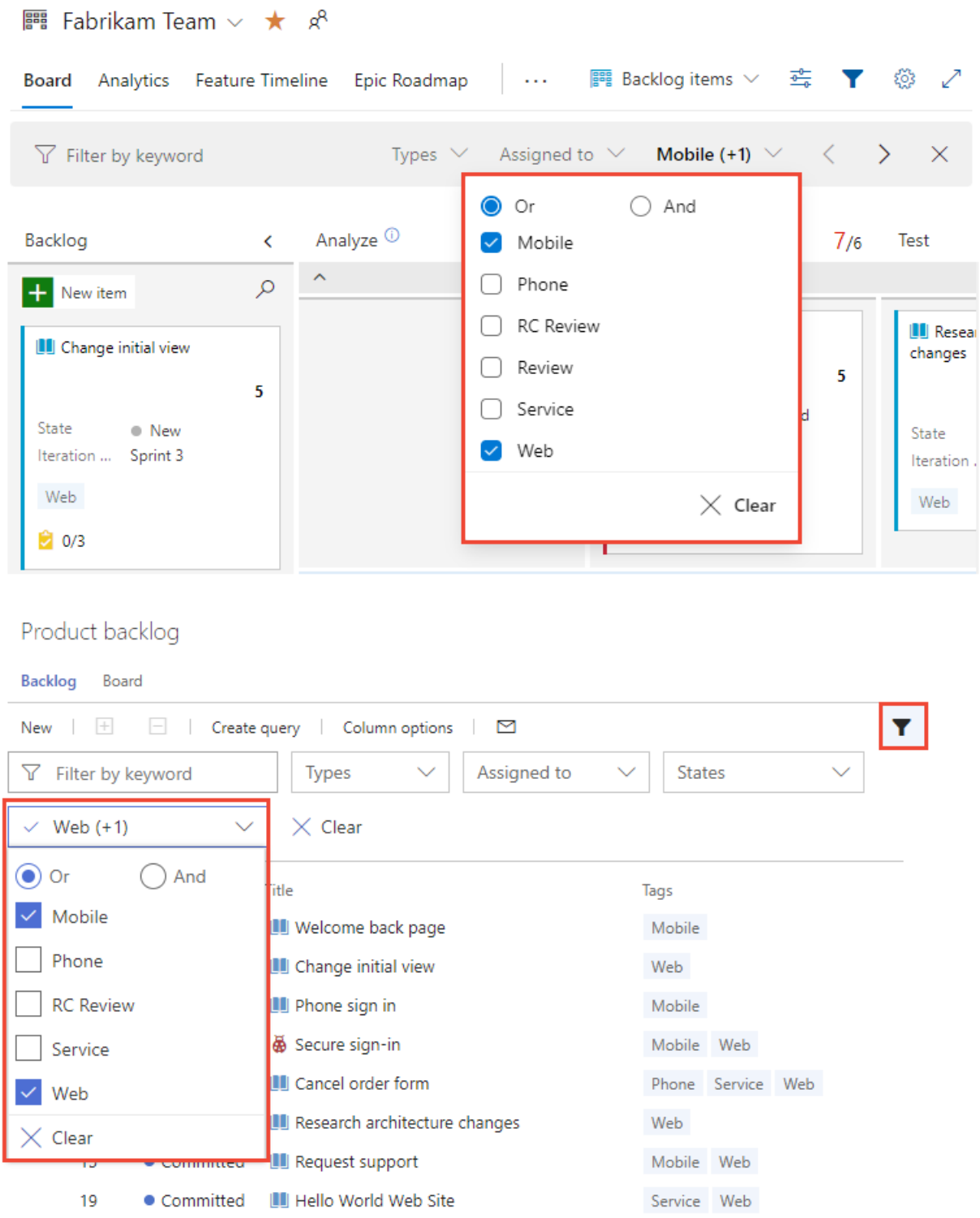


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

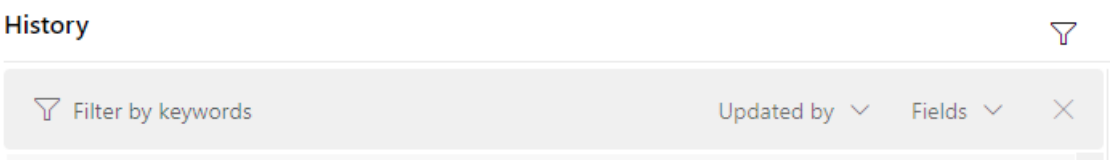


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

About teams and Agile tools

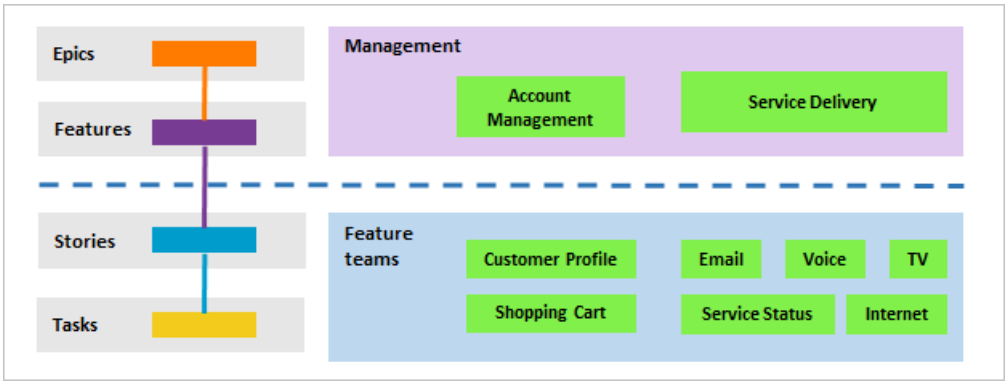
10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Learn how you can structure and use your teams and Agile tools to support your growing organization. When your team grows beyond its intended size—typically anywhere from 6 to 9 members—you might consider moving from a one team structure to a two-team structure. You can then set up a hierarchical team structure, which provides several advantages to managers for tracking progress across teams. For the step-by-step procedure to add a team, see Add another team.

> **NOTE**
> For guidance on configuring and customizing your project and teams to support your business needs, review Configuration and customization of Azure Boards.

Depending on the size of your organization and your tracking needs, you can set up a team structure similar to the following image shown. Do so by defining teams and their associated area path(s).



The following scenarios apply:

- each feature team can be associated with a single feature area path—such as *Customer Profile*, *Shopping Cart*, *Email*—or several area paths
- each management team, which focuses on a set of features, can choose several area paths to monitor
- each feature team has its distinct backlog to plan, determine priority, and track work
- portfolio or product owners can create their idea, road map, and goals for each release, monitor progress across their portfolio of projects, and manage risks and dependencies. For more information, see Portfolio management

About area paths, feature teams, and management teams

Area paths serve the following purposes in Azure Boards:

1. Filter the work items that appear on a team backlog or board.
2. Apply additional filtering to the work items that appear on a backlog or board, for details, see Interactively filter backlogs, boards, queries, and plans.
3. Group work that shares some relationship, such as belonging to the same product, feature, or other work-level grouping.
4. Restrict access to work by setting permissions. Permissions can be set for security groups to restrict

modification of work items based on area path.

Teams make their selections as follows:

- Feature teams can choose one or more area paths to specify which work items will appear on their backlogs and boards.
- Management teams typically choose all area paths that their feature teams work on.
- Additionally, management teams may only focus on Features and Epics, while Feature teams may focus only on product backlog items such as User Stories (Agile), Product Backlog Items (Scrum), and Requirements (CMMI).

> **NOTE**
> Teams can be assigned a maximum of 300 **Area Paths**. To learn more, see Work tracking, process, and project limits.

For steps to define area paths and assign them to a team, see Define area paths and assign to a team.

The default team

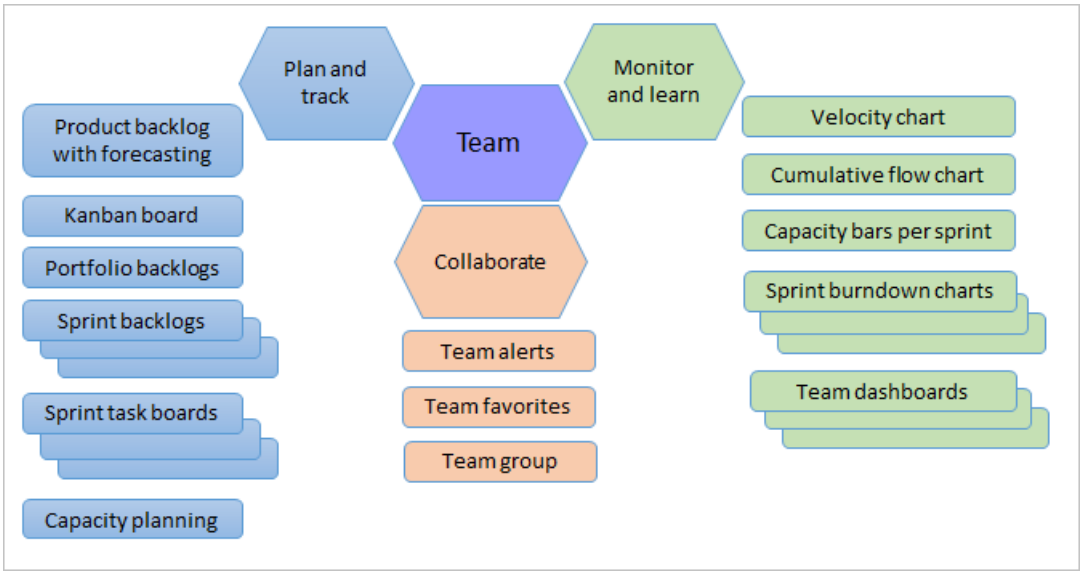
Each new project is configured with a default team with the project name. For example, the *Fabrikam* project is automatically configured with the *Fabrikam* team.

Backlogs, boards, and dashboards are automatically configured for this default team so that you can start defining work items and your backlog.

You can rename the default team and reassign a new team as the default.

Each team gets their own set of tools

Each team you create gets access to a suite of Agile tools and team assets. These tools let teams work autonomously and collaborate with other teams across the enterprise. Each team can configure and customize each tool to support how they work.



> **NOTE**
> In addition to team dashboards, you can add a project dashboard, which isn't specific to any one team. For more information, see Add, rename, and delete dashboards.

These tools automatically filter the set of work items they display by referencing the following items:

- default area path
- iteration path
- selected sprints

For more information about each tool and the configuration settings for each tool, see the following corresponding articles.

Area

Tool

Team configuration tasks

Backlogs

- Product backlog
- Features backlog
- Epics backlog
- Forecast

- Configure area paths
- Select active iteration paths (sprints)
- Select backlog levels
- Show bugs on backlogs & boards

Sprints and Scrum

- Sprint backlogs
- Sprint capacity
- Taskboard
- Sprint burndown

- Select active iteration paths (sprints)
- Set working days

Kanban boards

- Kanban board
- Features board
- Epics board
- Cumulative flow

- Configure area paths
- Select default iteration path
- Select backlog levels
- Show bugs on backlogs & boards

Widgets

- New work item
- Sprint burndown
- Sprint capacity
- Sprint overview
- Team members

- Configure area paths

- Select active iteration paths (sprints)
- Add team members

Other tools

- Favorites
- Work item templates
- Delivery plans
- Queries
- Velocity
- Dashboards
- Alerts

Not applicable

Many of these tools are built from system queries that reference the team area path. For instance, a team's default area path filters the work items that appear on a team's backlog. Work items created using an Agile tool auto-assign the areas and iterations based on team defaults.

Team defaults referenced by backlogs and boards

What work items appear on team backlogs and boards? When you add work items to a backlog or board, how are team defaults used to assign field values?

Teams are associated with one or more **Area Paths** and **Iteration Paths**. These paths determine what items appear on their backlogs and boards.

When you define a team, you define the team's:

- Selected Area Path(s)
- Default Area Path
- Selected Iteration Path(s)
- Backlog Iteration Path
- Default Iteration Path

All Agile tools reference the Area Path(s) defined for a team. The set of work items that appear on a backlog or board depend on the current **State** of a work item or its parent-child status.

Several tools also reference the team's default and selected Iteration Path or sprints. When you add a new work item from a team's backlog or Kanban board, the system assigns the team's **Default Area Path** and **Default Iteration Path** to it.

> **NOTE**
> New work items added through the **Work Items** page or the **New Work Items** widget on a team dashboard don't reference the **Default Iteration Path** assigned to the team. Instead, new work items are assigned the last **Iteration Path** selected by the user. New work items added through a team's **Sprints** backlog or taskboard are always assigned the **Iteration Path** associated with the selected sprint backlog or taskboard.

Agile tool

Area path (see note 1)

Iteration path

State
Portfolio or product backlogs
Selected area path(s)
Equal to or under team's backlog iteration path
Active (corresponds to a Proposed or InProgress state category, see notes 2, 3)
Kanban boards (see note 4)
Selected area path(s)
Equal to or under team's backlog iteration path
Any state (see notes 3, 5)
Sprint backlogs (see note 4)
Selected area path(s)
Team's selected iteration paths
Any state (see notes 3, 5)
Task boards (see note 4)
Selected area path(s)
Team's selected iteration paths
Any state (see notes 3, 5)
New work item widget
Default area path
Default iteration path
n/a

NOTE

1. Agile tools filter items based on the team's selected area path(s). Teams can choose whether to include or exclude items assigned to subarea paths.
2. Work items whose **State** equals *Closed*, *Done*, or *Removed* (corresponding to a *Completed* category state) don't appear on portfolio and product backlogs.
3. You can add custom workflow states and assign them to one of three state categories. The state categories"> determine which work items appear on backlog and board views.
4. Kanban boards, sprint backlogs, and taskboards only show the last node in a hierarchy, called the leaf node. For example, if you link items within a hierarchy that is four levels deep, only the items at the fourth level appear on the Kanban board, sprint backlog, and task board. To learn more, see parent-child links between items.
5. Work items whose **State** equals *Removed* don't appear on boards.

Structure hierarchical teams or scale agility within an enterprise

Although there's no concept of subteams, you can create teams whose area paths are under another team, which effectively creates a hierarchy of teams. To learn more, see Add another team.

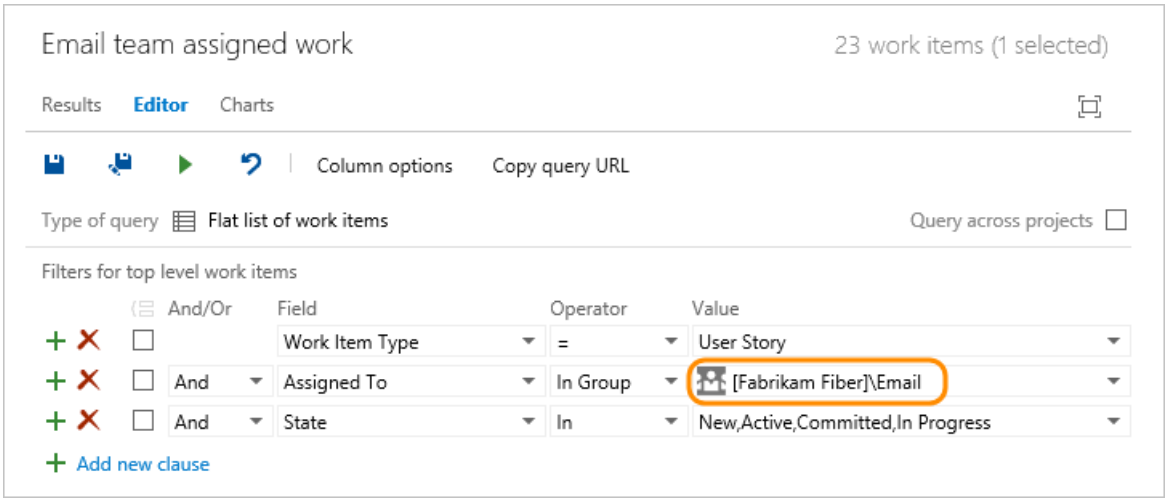
Also, the following articles walk you through the steps for configuring teams, area paths, and iterations to

support portfolio management or enterprise organizations:

- Portfolio management
- Implement Scaled Agile Framework to support epics, release trains, and multiple backlogs

Team groups

When you add a team, a security group is automatically created with the team name. You can use this group to filter queries. The name of team groups follows the pattern **[Project Name]\Team Name**. For example, the following query finds work assigned to members of the **[Fabrikam Fiber]\Email** team group.



You can also use the **@mention** control within discussions and pull requests to notify all members of a team. Start entering the name of a team or a security group, select the search icon, and then select from the options listed. To learn more, see Use @mentions to further discussion.

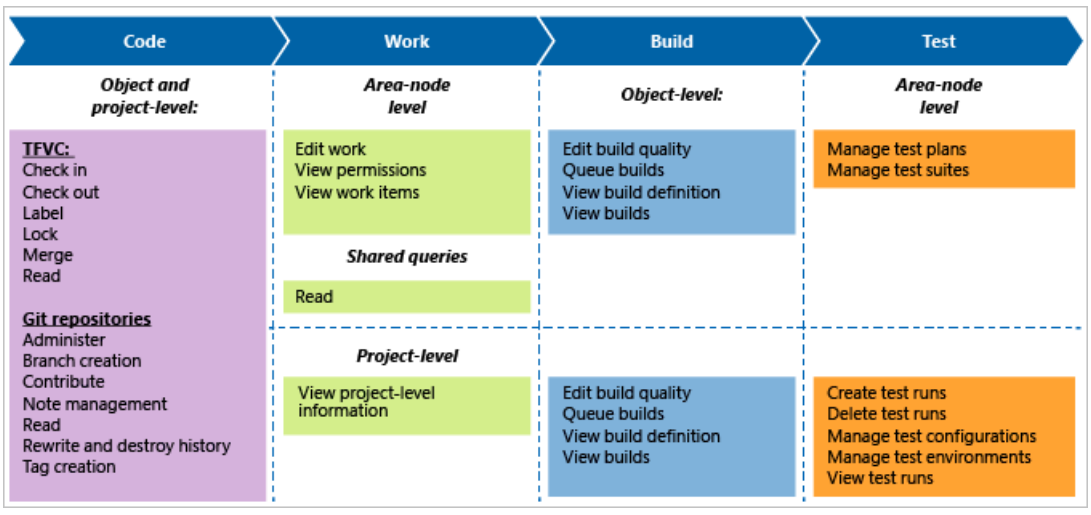
Work on more than one team

Can a user account belong to more than one team?

Yes. You can add user accounts as members of the project or to one or more teams added to the project. If you work on two or more Scrum teams, make sure you specify your sprint capacity for each team you work on.

Team member permissions

By default, team members inherit the permissions afforded to members of the project Contributors group. Members of this group can add and modify source code, create and delete test runs, and create and modify work items. Team members can collaborate on a Git project or check in work to the team's code base.



Limit access to select features by making a user a Stakeholder or limiting their access to read-only. For more information, see Permissions and access for work tracking.

Summary

- Every team owns their own backlog, to create a new backlog you create a new team
- Every backlog has a corresponding Kanban board you can use to track progress and update status
- The team's specified area and iteration paths determine which work items appear on the backlog and Kanban board—you can easily decide to include or exclude work items under a specific area path
- Each team can control how bugs show up on their backlogs and boards
- For an overview of all team assets and how to configure them, see Manage teams and configure team tools
- To have work done by several teams roll up in to a portfolio backlog, you'll want to setup the team hierarchy
- To add fields or work item types, see Customize your work tracking experience.

Related articles

- Add another team
- Configure team settings
- Work across projects

Create or add a team

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As your organization grows, you add teams to support that growth. You create a team in Azure DevOps that corresponds to a group of project members focused on specific products, services, or feature areas. You add teams to provide them the tools they need to manage their backlog, plan sprints, configure dashboards, define alerts, and set team favorites.

Each new project is configured with a default team with the project name. For example, the project named Fabrikam Fiber is configured with the default team Fabrikam Fiber Team. You can rename the default team and you can reassign a new team as the default.

For a good understanding on how to remain Agile as you add teams, review Scale Agile to Large Teams. For more information about team-configurable tools, see About teams and Agile tools.

> **NOTE**
> This article describes how to add a team or team members to a project defined in Azure DevOps. To learn about Microsoft Teams or the integration of Microsoft Teams with Azure Boards, see Welcome to Microsoft Teams or Use the Azure Boards app in Microsoft Teams.

Use this article to learn how to perform the following tasks:

- Create a team and add team administrators and team members
- Set the default team for a project
- List teams using Azure CLI
- Grant additional permissions to teams

- Create a team and add team administrators and team members
- Set the default team for a project
- Grant additional permissions to teams

To move work items assigned to one team to another team, see Move work items from one team to another.

Prerequisites

- To create a team or set the default team, you must be a member of the **Project Administrators** group. See Change project-level permissions. Only members of the Project Administrators group can add and delete teams.
- To add members to a team or change its configuration, you must be a team administrator or member of the Project Administrators group. To get added as a team administrator, see Add or remove a team administrator.
- To use Azure CLI commands, you must first install Azure CLI as described in Get started with Azure DevOps CLI.

- To create a team or set the default team, you must be a member of the **Project Administrators** group. See Change project-level permissions. Only members of the Project Administrators group can add and delete teams.
- To add members to a team or change its configuration, you must be a team administrator or member of the

Project Administrators group. To get added as a team administrator, see Add or remove a team administrator.

Create a team and add team members

From your web browser, you can view, create, and configure teams. Only members of the Project Administrators group can create teams.

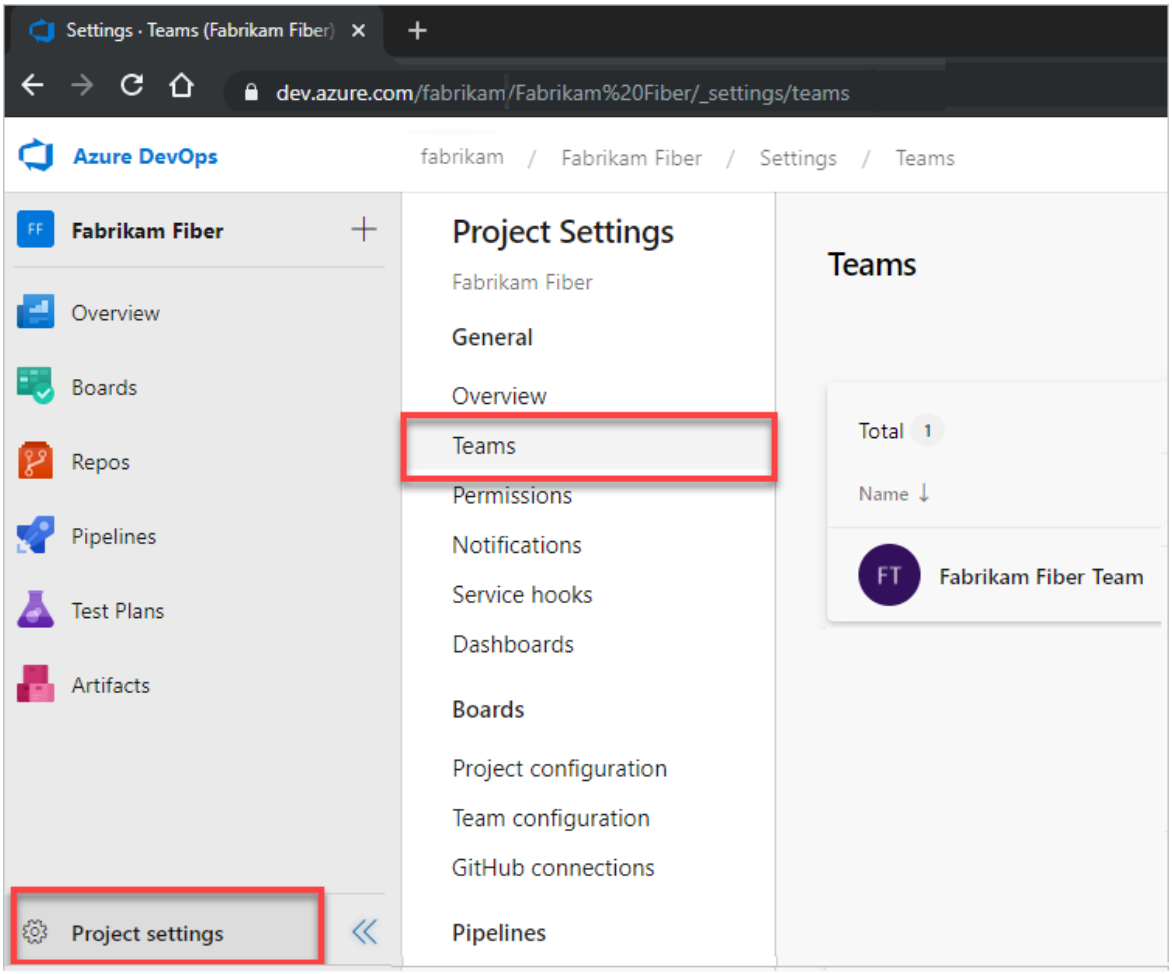
> **NOTE**
>
> When you create a team, you can automatically create the **Area Path** the team will use as a child node of the main project node. If you plan on creating a hierarchical team structure, you may want to first define the **Area Paths** at the project level, then create your teams, and then assign the **Area Path(s)** to be used by each team. To learn more about this team structure, see Configure a hierarchy of teams.

From the Azure CLI tool, you can list teams, create a team, update a team configuration, and delete a team.

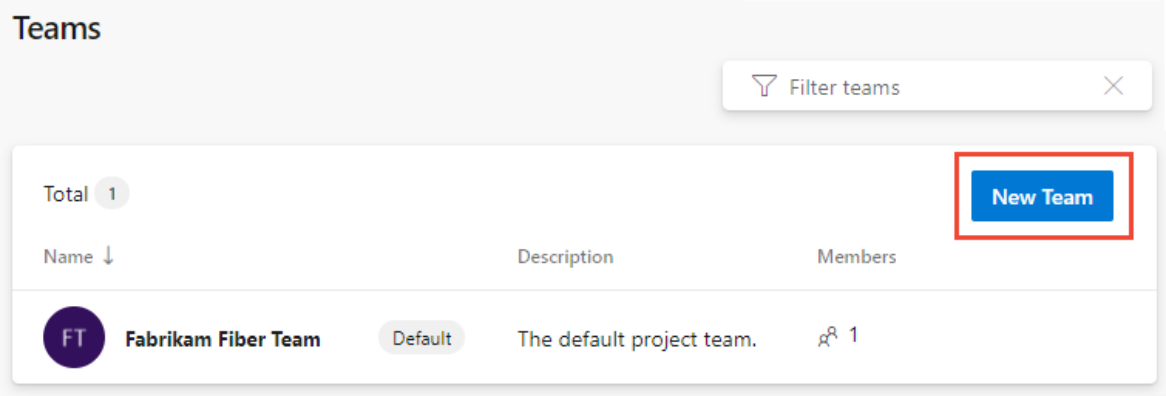
> **NOTE**
>
> To enable the new user interface for managing teams, enable the **New Teams Page** from the **Preview features** tool. To learn how, see Manage or enable features.

- New Teams UI
- Current UI
- Azure DevOps CLI

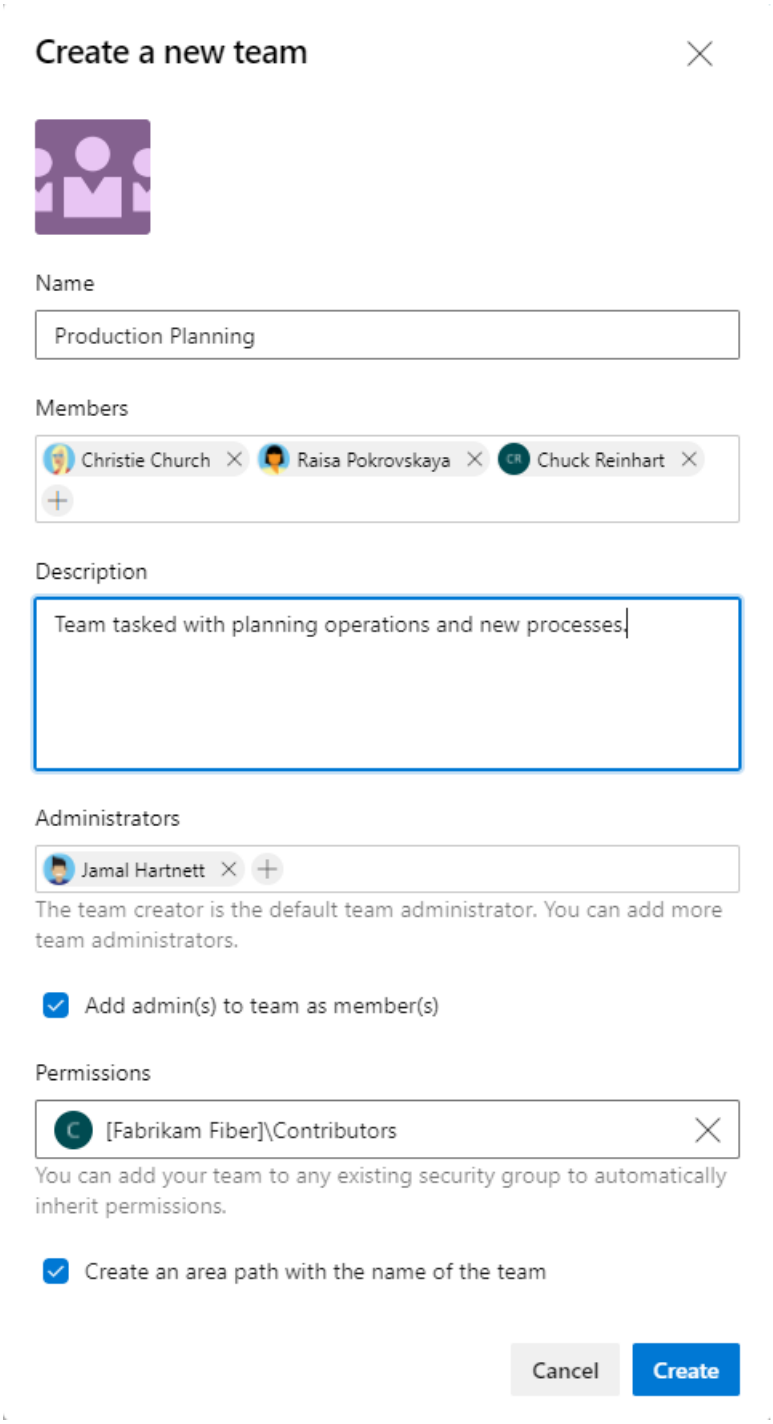
1. From the web portal, choose **Project settings** and open **Teams**.



2. Choose **New team**.



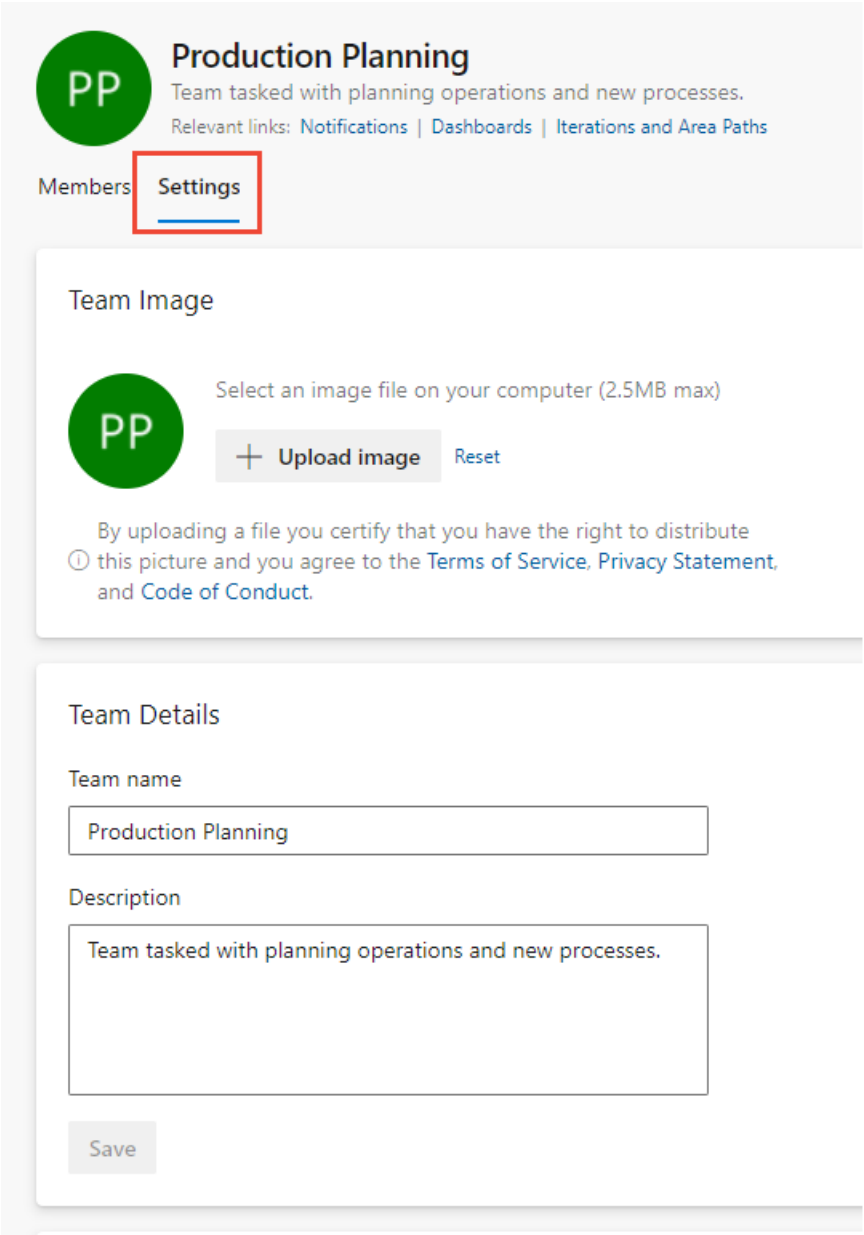
3. Enter a team name and the names of project members who you want to assign to the team. Optionally, enter a description. You must add at least one name as a team **Administrator**. Select **Create an area path with the name of the team**, or leave it unchecked and assign the **Area Path** for the team after it's created. You can choose an existing area path or add a new one at that time.



NOTE

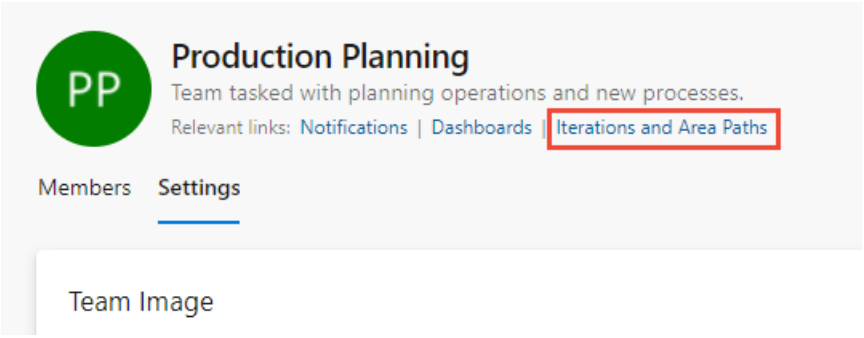
Consider adding one or more users as team administrators. Team administrators have the necessary permissions to add team members and configure all team settings—including backlogs, Kanban boards, and Taskboards. To learn more, see Manage and configure team tools.

4. When finished, select **Create**.
5. Select the team to add a team image, or configure other team elements.
6. Choose **Settings** to upload an image, change the team name, add a team administrator, or delete the team.



7. Choose **Save** to save any changes you made.

8. Choose **Iterations and areas** to set the **Area Paths** and **Iteration Paths** for the team to use.



To learn how, see the following articles:
- Define area paths and assign to a team
- Define iteration paths, also known as sprints, and configure team iterations.

Choose the **Current UI** tab. The New Teams Page UI is only available for Azure DevOps Services.

Upon creation of a team, the following items are automatically configured:

- Security group with the team name containing the members assigned to the team, for example, **[Fabrikam**

Fiber]\Production Planning. You can use this name to filter queries or to **@mention** all members of the team.

- Default notifications assigned to the team. To learn more, see Manage notifications for a team, project, or organization.
- Default dashboard named **Overview** assigned to the team. To configure the dashboard, see Add widgets to a dashboard.

> **IMPORTANT**
> Configuring the **Area Paths** and **Iteration Paths** used by the team is essential for many of the Azure Board tools to work, such as Backlogs, Boards, Sprints, and Delivery Plans. Team tools aren't available until the team's default area path is set. **Area Paths** and **Iteration Paths** are first configured for the project and then assigned or selected by the team.
>
> If you are moving from one team to two or more teams, you may want to review and revise the **Area Paths** assigned to the default project team.

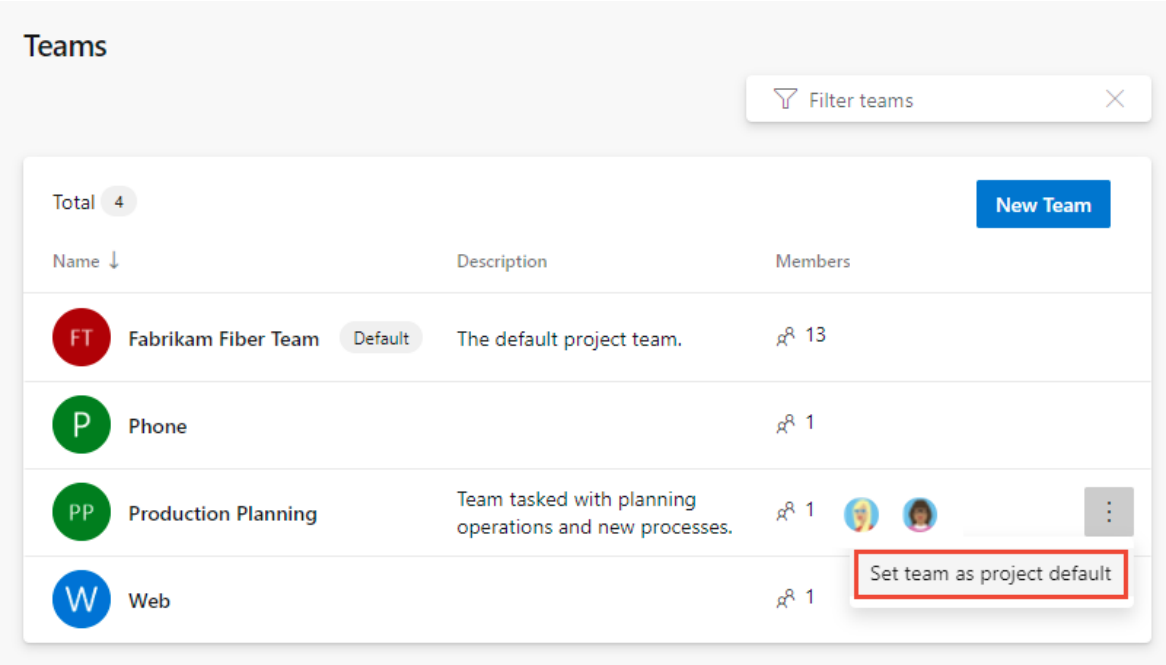
To configure other team features, see Manage teams and configure team tools.

Set the default project team

Each project has a default team assigned. You can change which project is assigned as the default with the following procedures.

- New Teams UI
- Current UI
- Azure DevOps CLI

1. Open **Project settings** > **Teams** as indicated in the previous section.
2. Choose ⋮ **More options** for the team you want to designate as the default, and choose **Set team as project default**.



Choose the **Current UI** tab. The New Teams Page UI is only available for Azure DevOps Services.

List teams with Azure CLI

You can list teams using Azure DevOps team list. To learn how to list team members, see Add users to a team or project, List team members.

```
az devops team list [--project]
                    [--skip]
                    [--top]
```

> **TIP**
> If you don't specify a **top** number, 100 teams are returned. To list all teams in a project, specify a number for **top** which is greater than the current number of teams defined.

Parameters

- **project**: Optional. Name or ID of the project. Example: --project "Fabrikam Fiber". You can configure the default project using `az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up via git config.
- **skip**: Optional. Number of teams to skip.
- **top**: Optional. Maximum number of teams to return.

Example

For example, the following command returns the 11 teams defined in the Fabrikam Fiber project. For addition output formats, see Output formats for Azure CLI commands.

Each team is assigned a unique ID.

```
az devops team list --project "Fabrikam Fiber" --output table
```

The table output listed below provides information on each of the attributes defined for the team.

```
ID                                    Name                Description
------------------------------------  ------------------  ------------------------------------------------
-------------------------
7f099146-29a2-4798-9949-77c9f5f79653  Account Management  Management team focused on creating and
maintaining customer services
2017b37a-486b-4222-ac84-b8b9eefa540e  Customer Profile    Feature team focused on securing account data
a90cd8f0-8e0d-42d6-aeb2-13442b826730  Email               Feature team delivering email apps
a48cb46f-7366-4f4b-baf5-b3632398ed1e  Fabrikam Team       The default project team. Was Fabrikam Fiber Team
e42fccbc-d96f-4c98-8168-7a85ecede548  Internet            Feature team developing web apps
b70aa504-33b4-4d17-a85d-0fbf4829a154  Phone               Feature team delivering phone apps
43e6bd2e-696f-492c-bbf7-9cde9cd420ea  Service Delivery    Management team responsible for ensure high
performance delivery of services
8920d2ec-eed1-4792-8934-82a57abce7c2  Service Status      Feature team focused on monitoring and addressing
service issues
9c676c8c-1910-4f73-b7b9-a946b5c551ae  Shopping Cart       Feature team managing shopping cart apps
64b86488-e105-4901-ba43-ffd48137bb93  TV                  Feature team developing TV apps
cda2b9b0-0335-4a0d-8bd0-67611d64ce9d  Voice               Feature team focused on voice communications
```

Grant team members additional permissions

For teams to work autonomously, you may want to provide them with permissions that they don't have by default. Suggested tasks include providing team administrators or team leads permissions to:

- Create and edit child nodes under their default area path

- Create shared queries and folders under the **Shared Queries** folder

For more information on setting permissions and access for select users, see Set permissions and access for work tracking.

If your deployment is integrated with SQL Server Reports, you need to Grant permissions to view or create SQL Server reports to team members.

Next steps

Move work items from one team to another team or Manage teams and configure team tools

Related articles

- Rename or remove a team
- About teams and Agile tools
- Add users to a team or project

Rest API resources

- Azure DevOps Teams CLI
- Teams (REST API)
- Work Items (REST API)

Add or remove a team administrator

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Learn how to add or remove an administrator for your team. It's always a good idea to have more than one user with administration permissions for a team. Team administrators can manage teams and configure team tools and manage projects. You may want to remove a user's administration permissions, for instance if the user is no longer active.

To add a team, see Add teams. To add or remove a project administrator, see Change project-level permissions.

Prerequisites

- To add or remove a user as a team administrator, you must be a member of the **Project Administrators** group, or a team administrator for the team you want to update.
- To be added as a team administrator, you must be granted **Basic** or higher access-level. Users granted **Stakeholder** access can't be added as a team administrator.

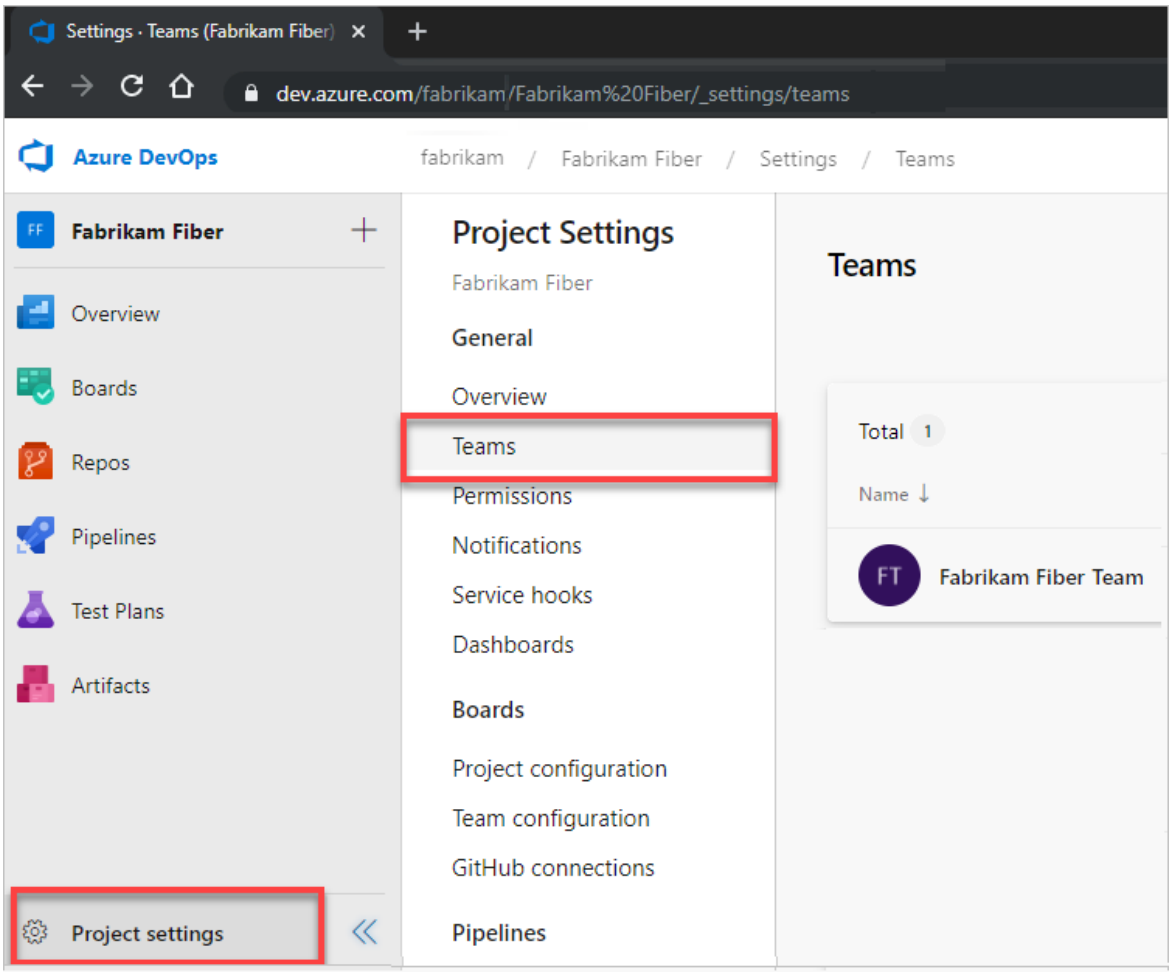
Add an administrator

To get added as a team administrator, ask another team administrator or a member of the **Project Administrators** group. See Look up a project administrator.

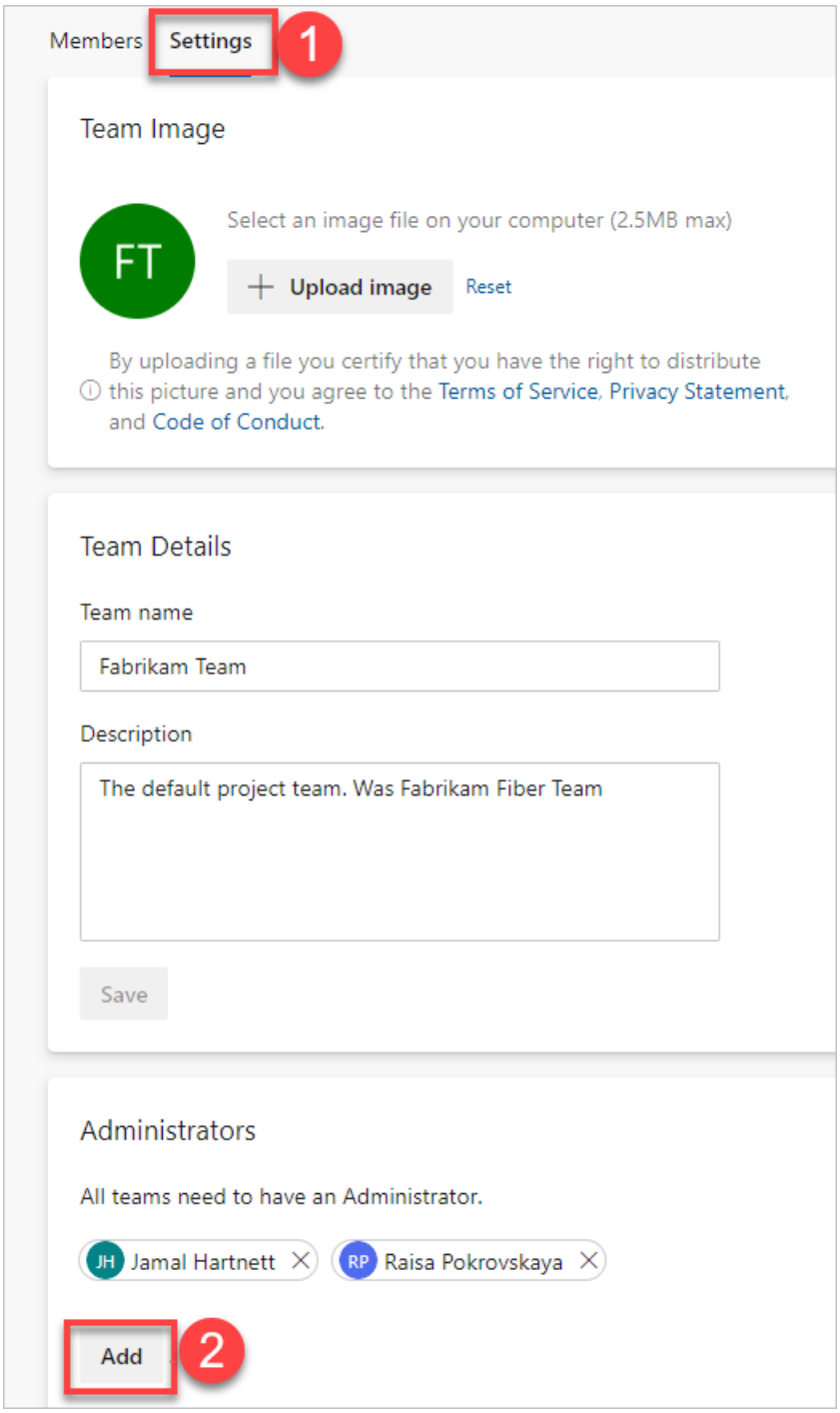
> **NOTE**
> To enable the user interface for the **New Teams Page**, see Manage or enable features.

- New Teams Page
- Current page

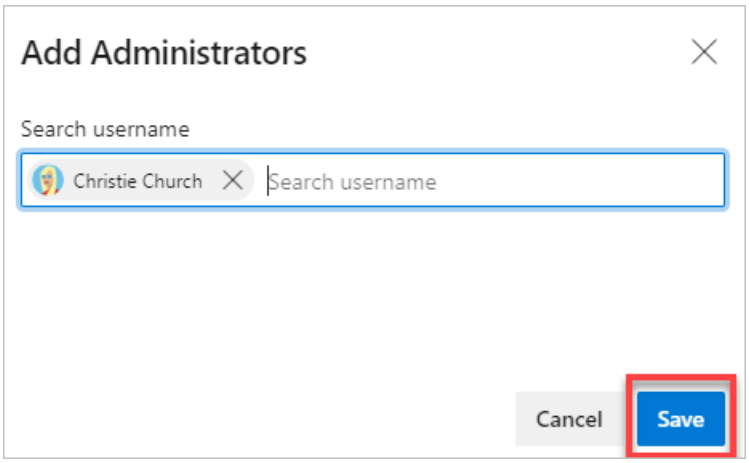
1. Sign in to your organization (`https://dev.azure.com/{yourorganization}`) and select a project.
2. Choose **Project settings**, and then choose **Teams**.



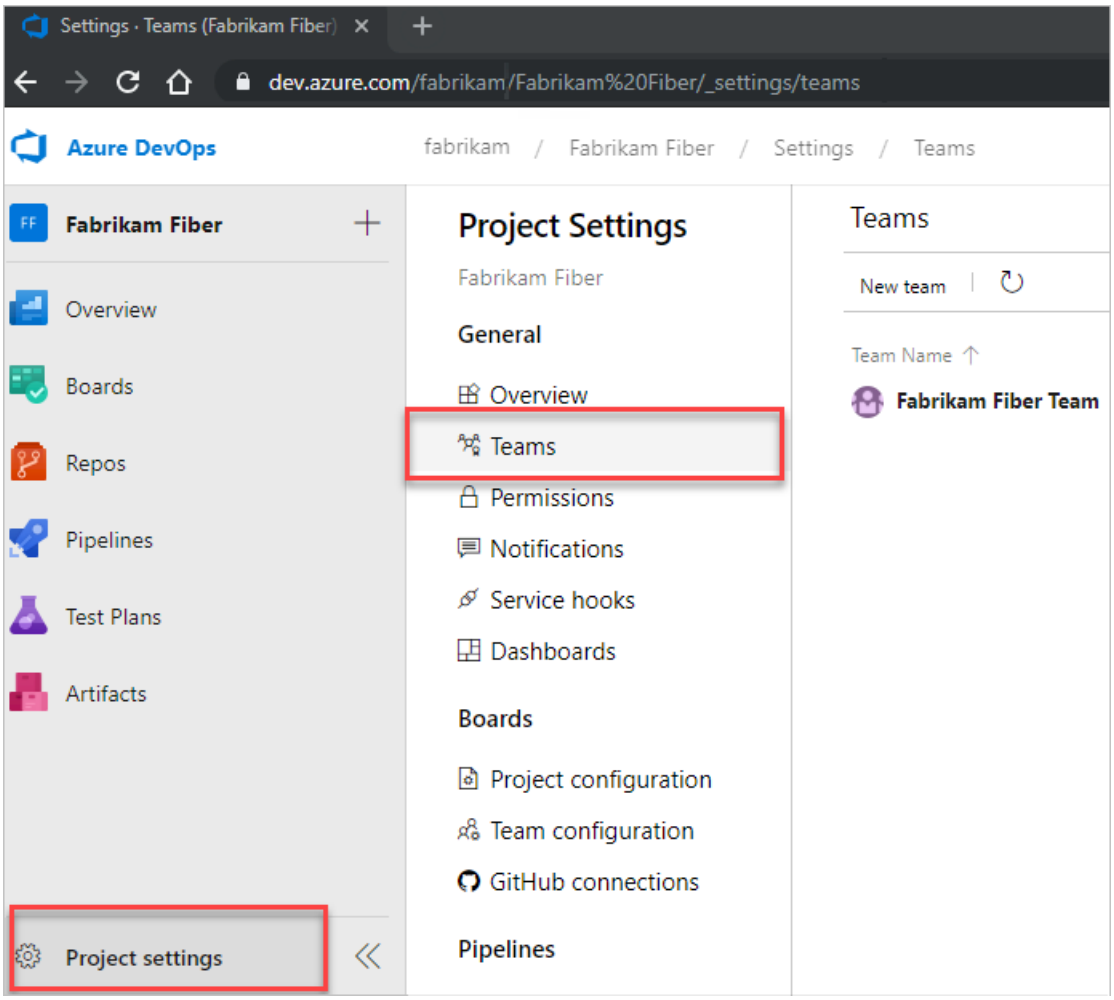
3. Select the team to configure, and then select **Settings** > **Add**.



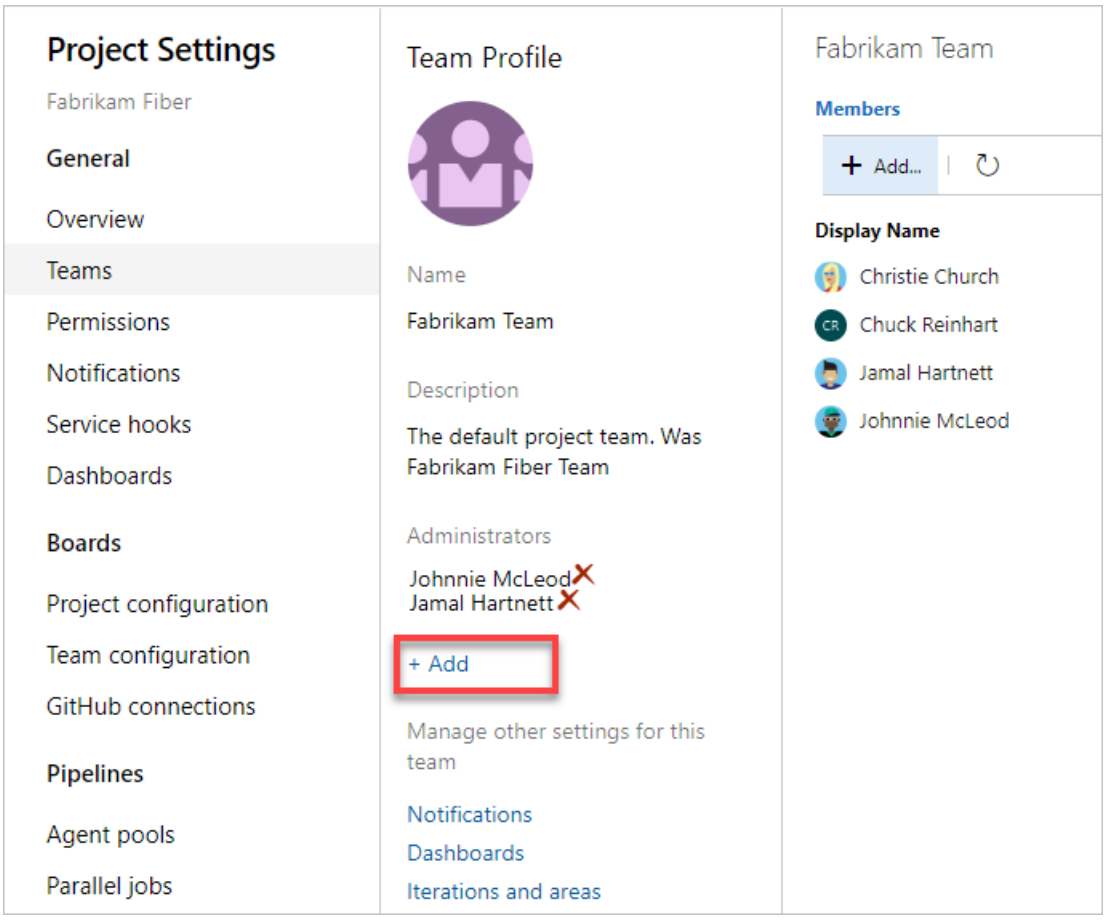
4. Enter the user's identity you want to add to the administrator role, and then select **Save**.



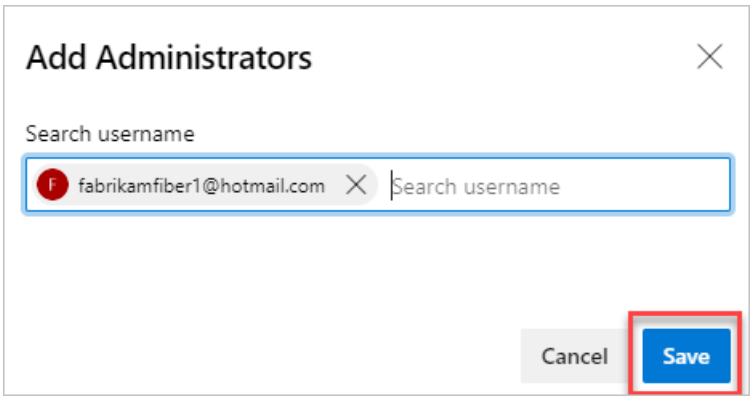
1. Choose **Project settings**, and then choose **Teams**.



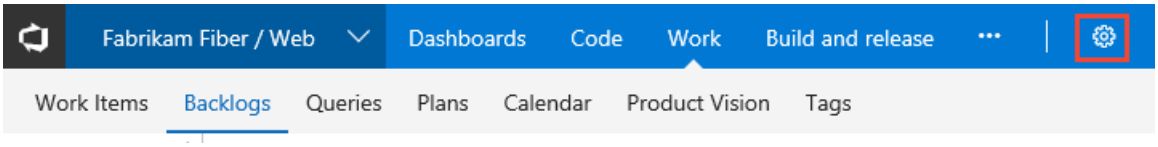
2. Select the team to configure, and then select **Settings** > **Add**.



3. Enter the user identity that you want to add to the administrator role, and then select **Save**.

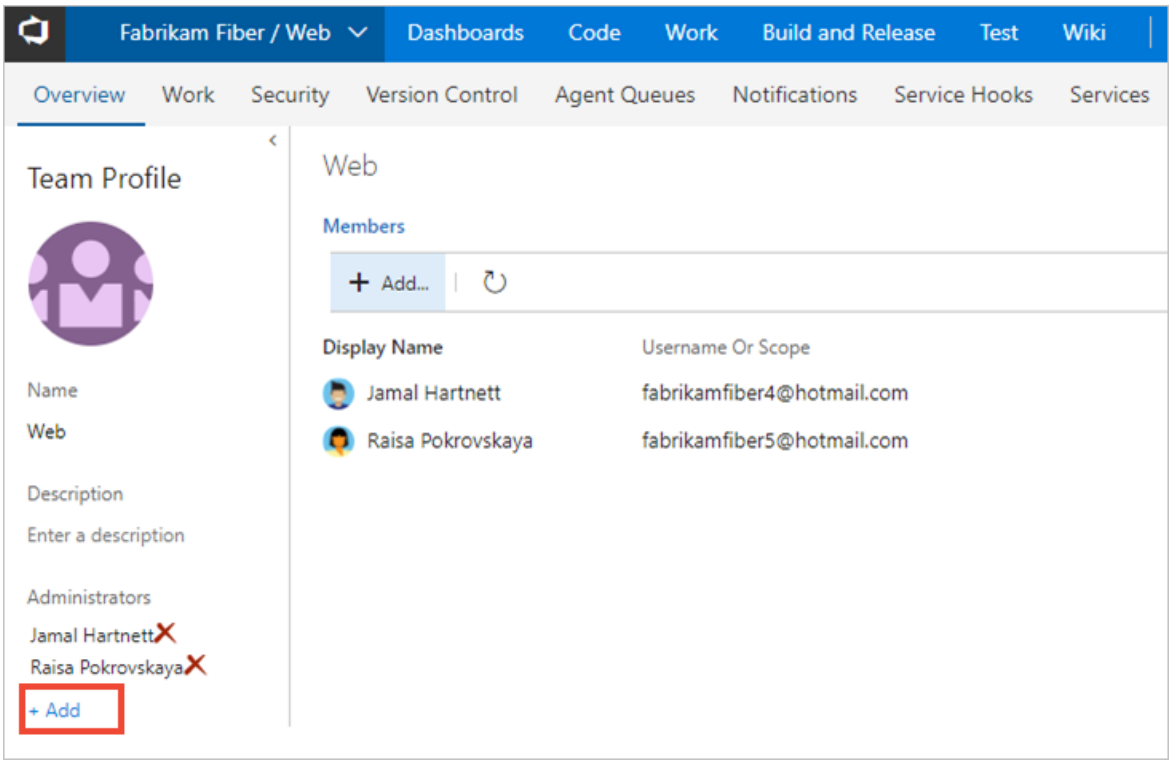


1. From the web portal and team context, choose **Team Settings** .

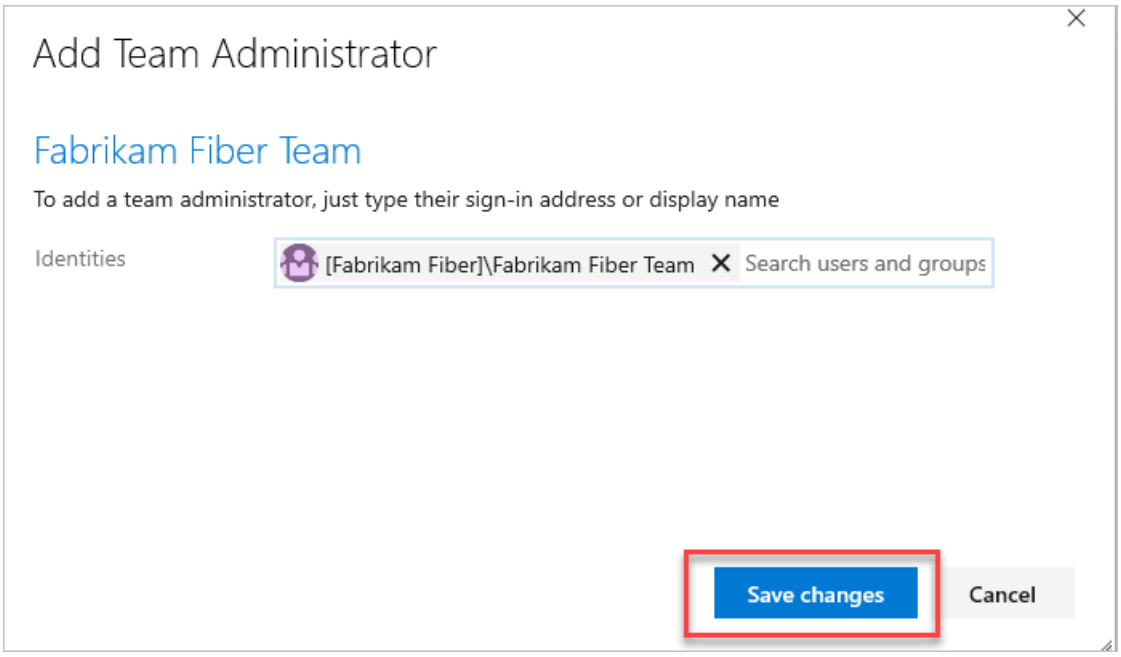


If you choose **Project settings** , then choose **Overview**, and select the team you want to configure.

2. Choose the **Add** link to open the dialog for adding user identities.



3. Enter the identities you want to add to the team administrator role.



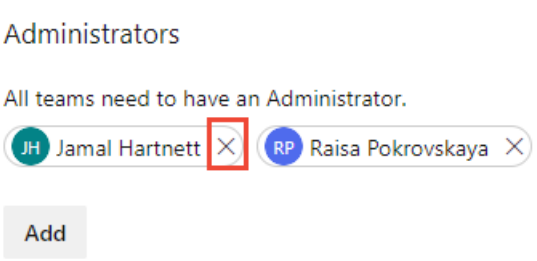
Remove an administrator

Each team must have at least one administrator. To remove an administrator, you must first add at least a second administrator.

Open the **Teams** page as described in the previous section.

- New Teams Page
- Current page

Choose **Settings** and scroll down to the Administrators section. Choose ✕ for the user that you want to remove as a team administrator.



From the Administrators section, choose ✕ for the user that you want to remove as a team administrator.



From the Administrators section, choose ✕ for the user that you want to remove as a team administrator.



Next steps

Manage teams and configure team tools

Related articles

- Add teams
- About teams & Agile tools
- Configure and customize Azure Boards
- Set team favorites

Add users or groups to a team or project

10/4/2022 • 21 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You add users to a team or project so they can contribute to the team and project. For enterprise organizations with large user bases, we recommend you use Azure Active Directory to add and manage new users through security groups. However, to enable flexibility for all size organizations, the following operations are supported:

- Team and project administrators can add new users to their team or project, unless the policy Allow team and project administrators to invite new users is disabled. New users are ones that haven't been added to the organization.
- When adding new users through the team and project user interfaces, the system automatically assigns an access level to the user.
- Adding users to a team or project automatically adds them to the Contributors group for the project. Members of the Contributors group have permissions to most features needed to contribute.
- By adding users to a team, you make team-specific tools aware of them, such as the team security group, Team Members widget, and sprint capacity planning tools.
- Once users have been added to a project or organization, you can browse for their display name or user name (email alias) from any people-picker tool.

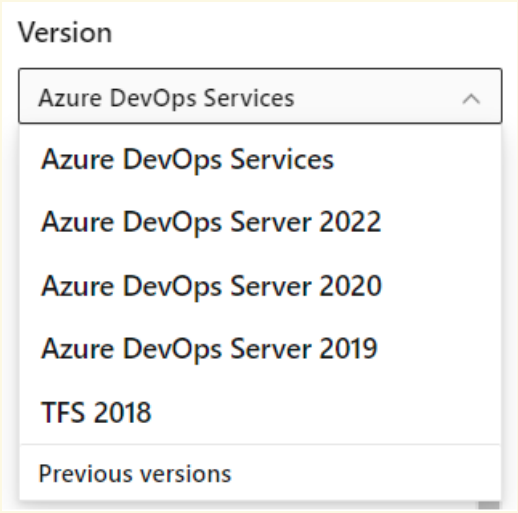
You add users to a team or project so they can contribute to the team and project. For enterprise organizations with large user bases, we recommend you use Active Directory or Windows Group to manage users through security groups. However, to enable flexibility for all size organizations, the following operations are supported:

- Team and project administrators can add existing users to their team or project. Existing users are ones known to the project collection through Active Directory or Windows group.
- Adding users to a team or project automatically adds them to the Contributors group for the project. Members of the Contributors group have permissions to most features needed to contribute.
- By adding users to a team, you make team-specific tools aware of them, such as the team security group, Team Members widget, and sprint capacity planning tools.
- Once users have been added to a project or organization, you can browse for their display name or user name (email alias) from any people-picker tool.

You add projects to an organization or project collection and you add teams to projects. To learn more, see:

- Create a project
- Add team, go from one default team to others

IMPORTANT



To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

Supported options for adding users

Depending on the interface you use, you can exercise different options for adding new or existing users to teams or projects.

Team and project administrators can add existing users to their team or project. Existing users are ones that are known to a project collection through the Active Directory or Windows Group created for the server that hosts the on-premises Azure DevOps Server.

Administrator level

Interface

Supported tasks

Team administrators

Team Members dashboard widget

Add new or existing users to a team. Send new users an invite.

Team administrators

Project Settings > Teams > Team > Members

Add existing users or groups to a team, or remove a member.

Project Administrators

Project **Summary** page, **Invite**

Add new or existing users. Send new users an invite. Optionally add users to one or more teams.

Project Administrators

Project Settings > Permissions > Groups > Group Members

Add existing users or groups to a security group. By adding to a team group, you effectively add them to the

team. Optionally remove a user from a group.

Project Collection Administrators

Organization Settings > Users

Add new users to an organization and send an invite. Must specify the access level. Optionally add them to select projects. Can use Group rules to further manage groups being added.

Project Collection Administrators

`az devops user` CLI

Add new users to an organization and send an invite. Must specify the access level.

Azure Active Directory Administrators

Azure Active Directory

Users you add to Azure Active Directory connected to Azure DevOps Services are added to the Project Collection Valid Users group. To learn more, see Connect your organization to Azure Active Directory.

Active Directory Administrators

Active Directory or Windows Group

Users you add to Active Directory or Windows Group connected to Azure DevOps are added as members of the Project Collection Valid Users group. They have access to all projects within a project collection. To learn more, see Set up groups for use in Azure DevOps on-premises.

Prerequisites

- You must have an organization and project. If you don't have a project yet, create one.
- To add users to or remove users from a team, you must be added as a team administrator, or be a member of one of the administrative groups.
- To add users to or remove users from a project, you must be a member of the **Project Administrators** group.
- When the organization is connected to Azure Active Directory, the Allow team and project administrators to invite new users policy must be enabled for team administrators or members of the Project Administrators group to add new users.
- To add users or manage users for an organization, you must be a member of the **Project Collection Administrators** group. Organization owners are automatically members of this group.

- If you don't have a project yet, create one
- To add users to or remove users from a team, you must be added as a team administrator, or be a member of one of the administrative groups.
- To add users to or remove users from a project,you must be a member of the **Project Administrators** group.
- To add users or manage users for a server, you must be a member of the Azure DevOps Administrators group.

If you're new to Azure DevOps, you may want to familiarize yourself with the information provided in these articles:

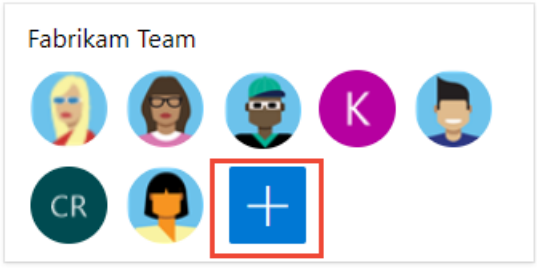
- Get started with permissions, access levels, and security groups
- About projects and scaling your organization
- Default permissions and access quick reference

- About teams and Azure Boards tools

Add a user from the Team Members widget

As a team administrator, you can add new or existing members from the **Team Members** dashboard widget. To add this widget to a dashboard, see Add widgets to a dashboard.

1. To invite someone to your team, choose the plus button on the Team Members widget.

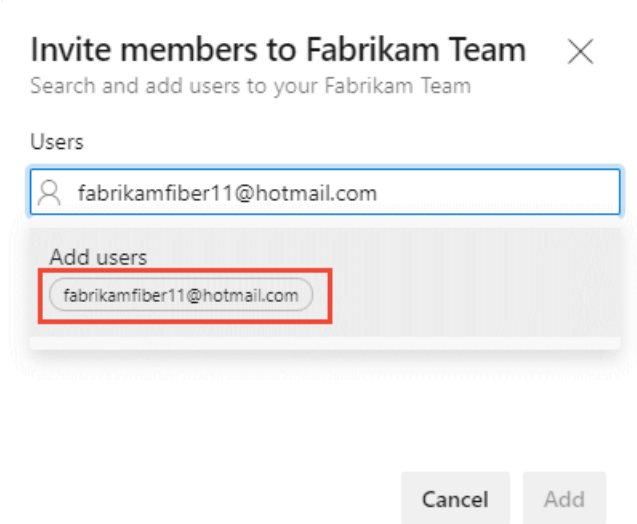


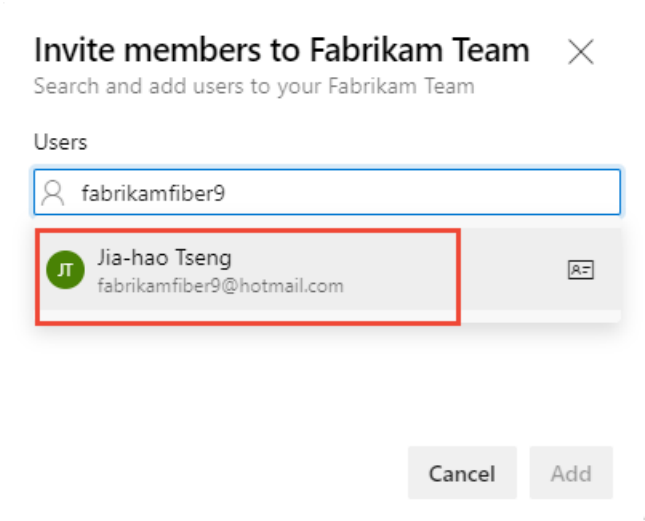
2. For new users, enter their email address. For existing users, type their name until it resolves as a known name to the system. You can add several email addresses or account names by separating them with a semicolon (;).

   Choose the entry listed under **Add users** to complete the entry.

   > **NOTE**
   > Any valid email address is acceptable. When the user accepts the invitation and signs into Azure DevOps, they register their email address as a Microsoft account and choose a password.

   Choose the name that appears to complete the entry.

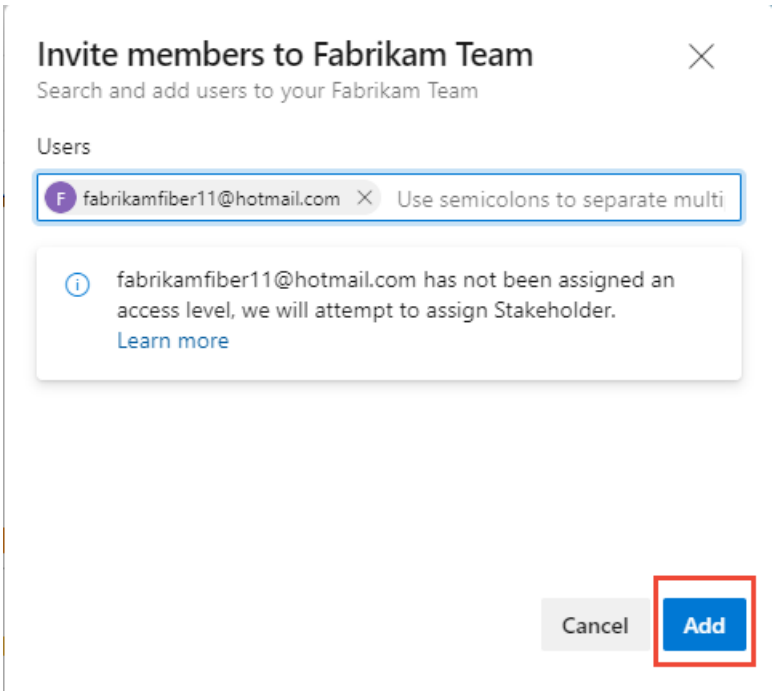


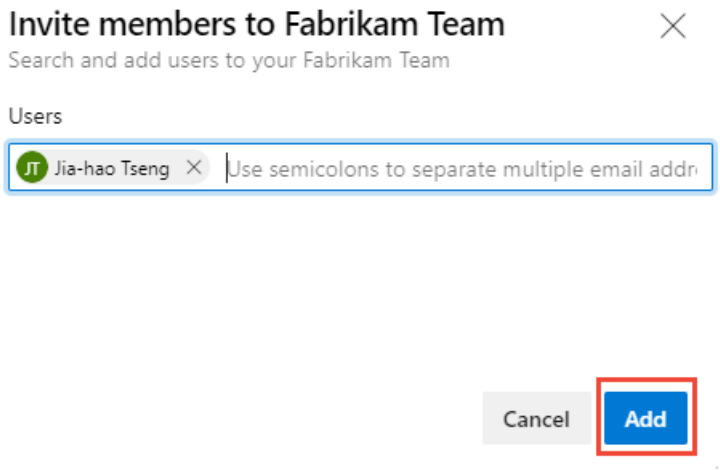


3. Complete the addition.

   When the user is unknown, you'll get a notification that an access level must be assigned. To complete the invitation, choose **Add**.

   Choose **Add** to complete adding the user. Known users don't receive an invitation.

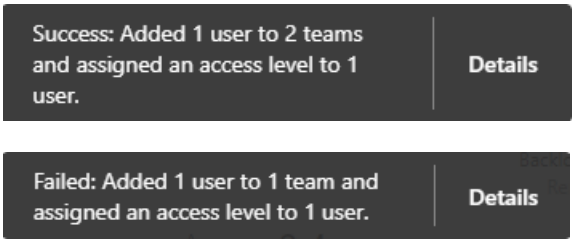




When adding a new user, the system assigns Stakeholder as the access level when all free five Basic access levels have been assigned. Active contributors to a project need to have Basic access as a minimum. A Project Collection Administrator can change the access level and resend invitations from the **Organization Settings > Users** page.

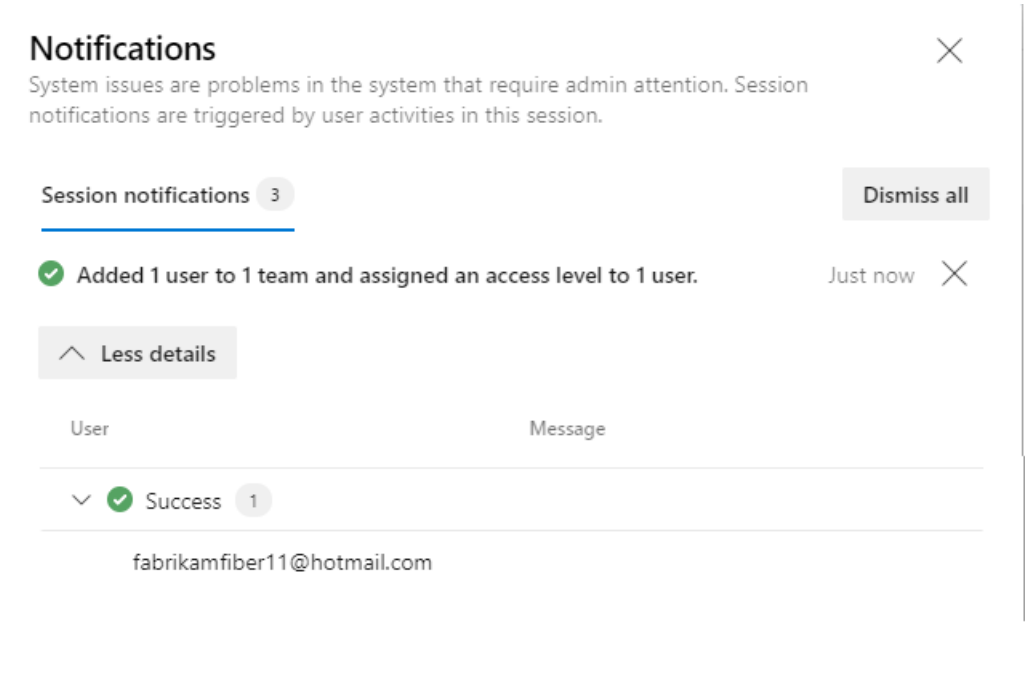
> **NOTE**
> Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see Permissions and access.

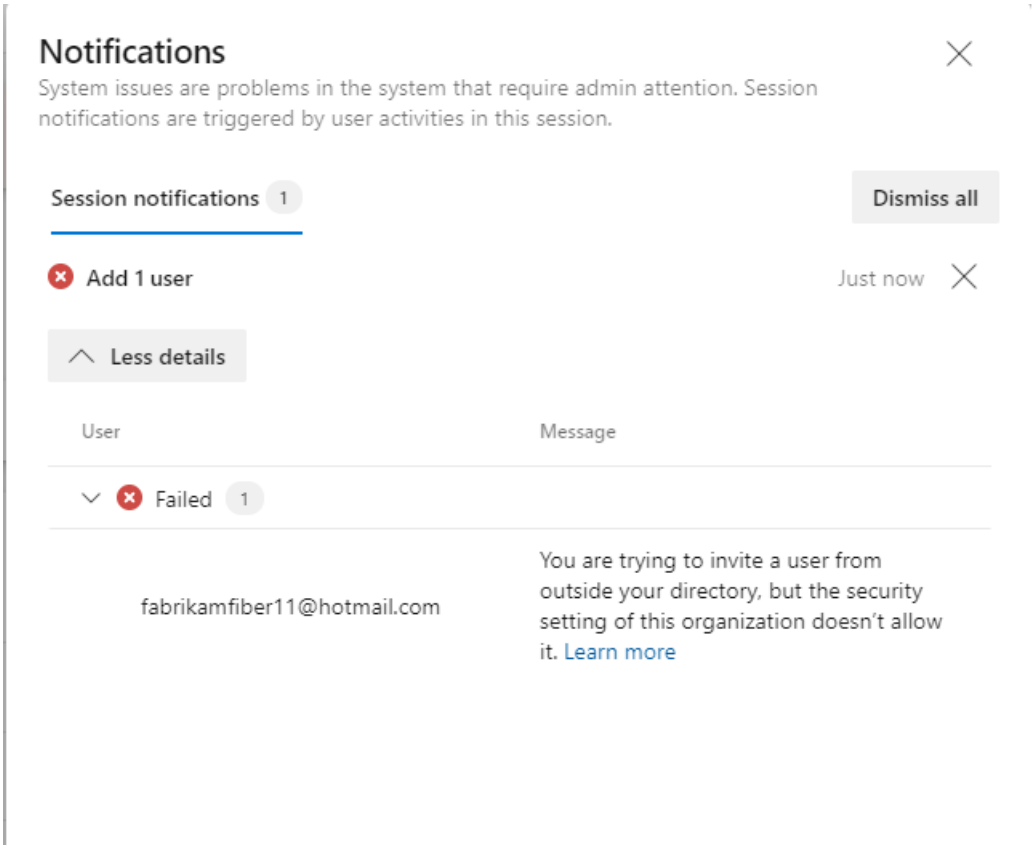
4. (Optional) A message will briefly display on the screen to indicate success or failure. Choose **Details** to open the notification and review details.



A success message indicates the status of adding the user to the system.

A failure message indicates why the addition of the user failed.





5. New users receive an email inviting them to sign in to the project. Existing users don't receive any formal notification.

Add users or groups to a team

Add existing users or security groups to a team from the **Project settings> Teams** page. From this interface you can view, add, or remove users and security groups to/from a team. To add a custom security group, see Add or remove users or groups, manage security groups.

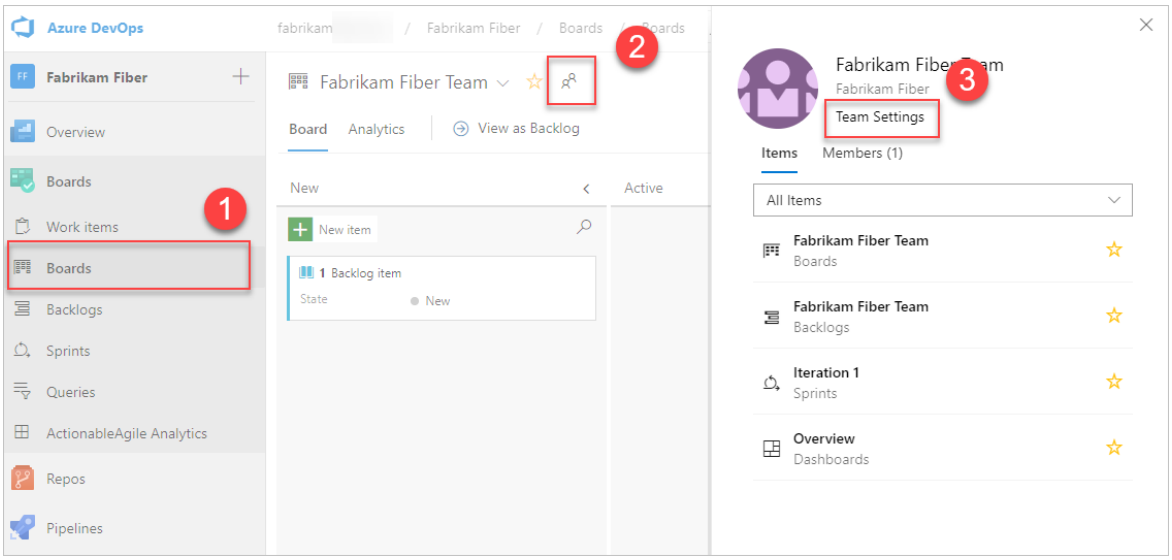
> **NOTE**
> To enable the new user interface for managing teams, enable the **New Teams Page** from the **Preview features** tool. To learn how, see Manage or enable features.

- Preview UI
- Current UI

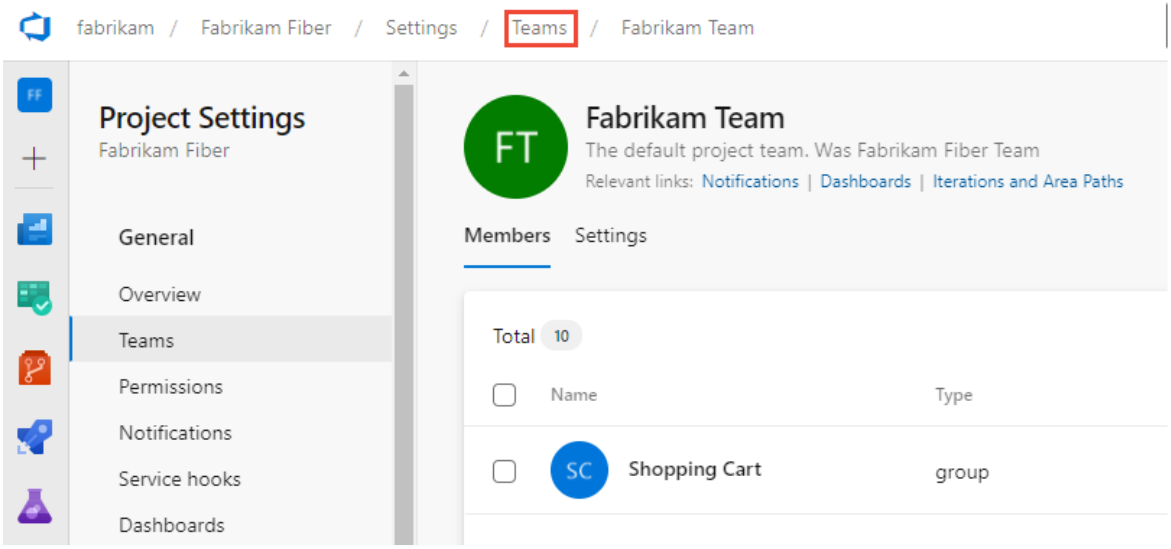
You can toggle between direct or expanded membership views. The **Direct Members** view displays users and groups that have been added to the team. The **Expanded Members** view replaces any Azure DevOps groups with the members that belong to those groups. Azure Active Directory or Active Directory groups aren't expanded.

1. Open a backlog or board for a team and choose the team profile icon. Then choose **Team Settings**.

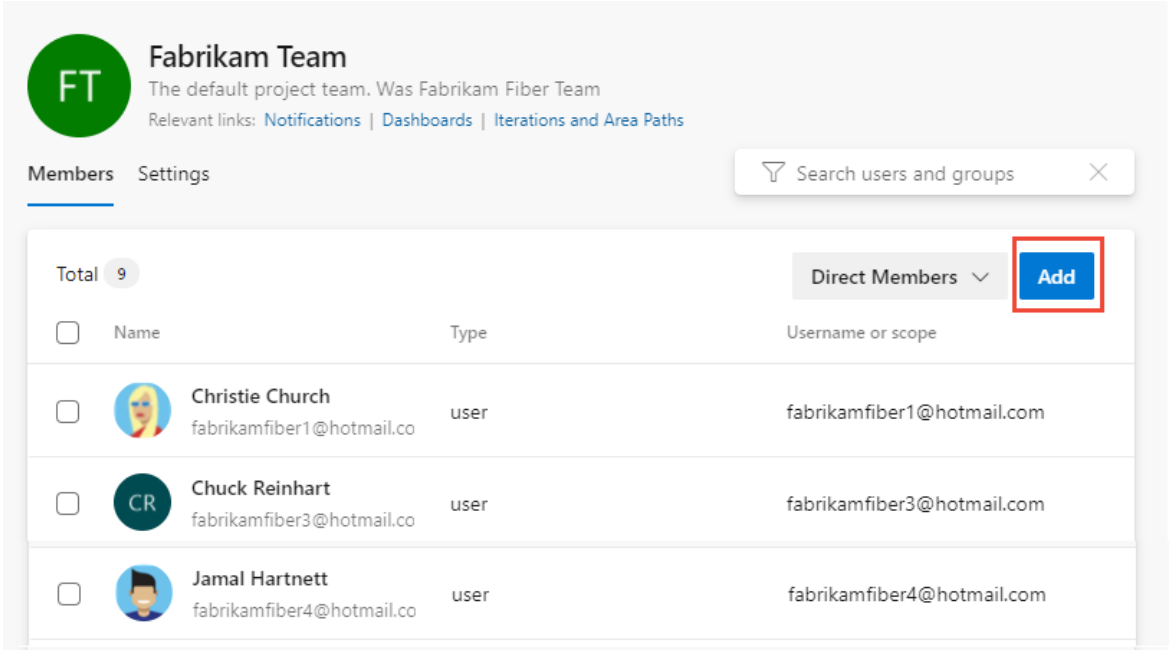
   Here we open the Board for the Web team and from there the team profile.



2. If you need to switch the team context, use the team selector within the breadcrumbs.

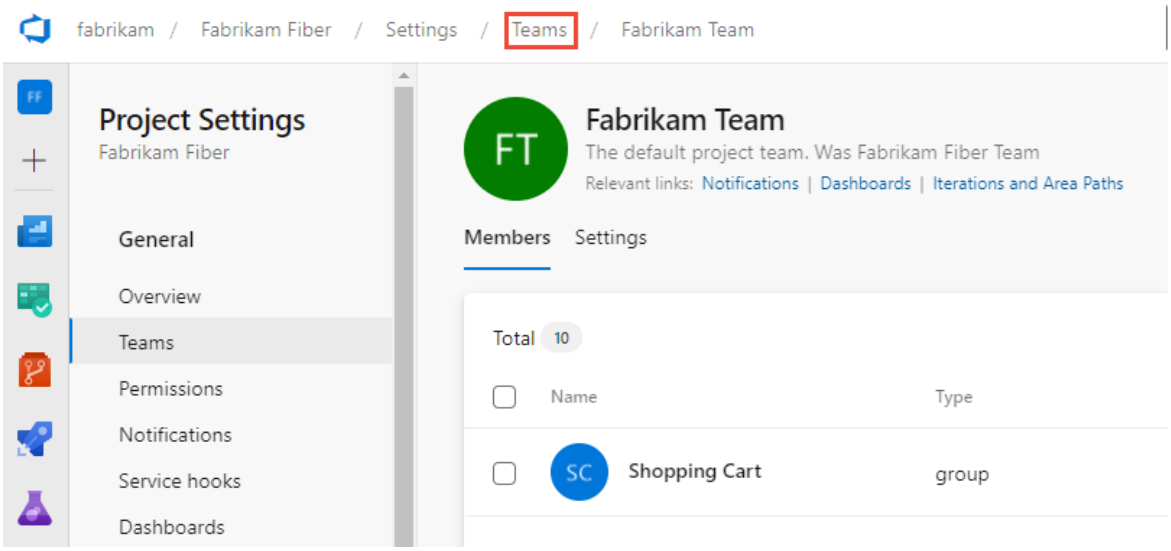


3. Choose **Add**.



4. Enter the sign-in addresses or display name for each account you want to add. You can also add a project security group—such as another team group, custom group, or Azure Active Directory group when used by the organization. Add them one at a time or all at the same time. You can enter several identities into

the text box, separated by commas.



> **TIP**
> You must enter user and group names one at a time. However, after entering a name, the account is added to the list, and you can enter another name in the Identities text box before choosing to save your changes.

You may need to choose the refresh icon to see your updates.

5. To add an account as a team administrator, choose the Settings page and then choose **Add** under the Administrators section. For details, see Add a team administrator

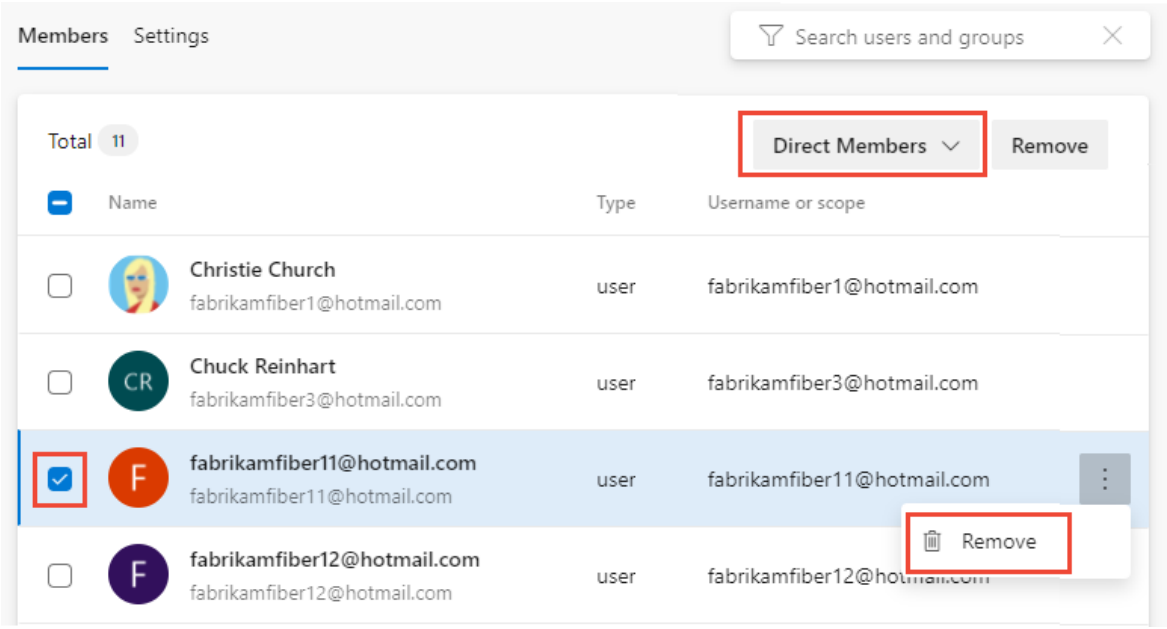
Choose the **Current page** tab for information on adding a user to a team. The **New Teams Page** preview feature is only available for Azure DevOps Services at this time.

Remove users or groups from a team

From the team's **Members** page, you can remove members.

- Preview UI
- Current UI

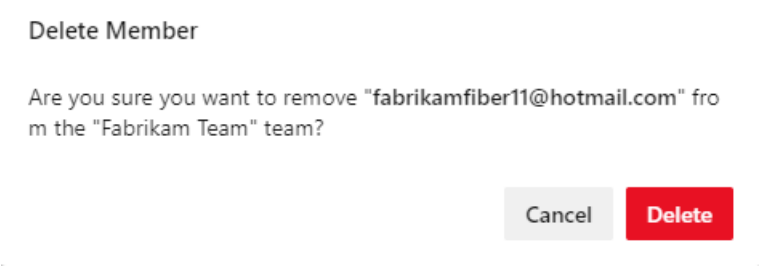
1. To remove members, open the team's **Members** page, choose **Direct Members**, check the checkbox of the user you want to remove, choose **More options**, and then choose **Remove**.



TIP

To remove a team administrator as a team member, you must first remove them as an administrator.

2. Confirm the removal by choosing **Delete** in the confirmation message.



Choose the **Current page** tab for information on adding a user to a team. The **New Teams Page** preview feature is only available for Azure DevOps Services at this time.

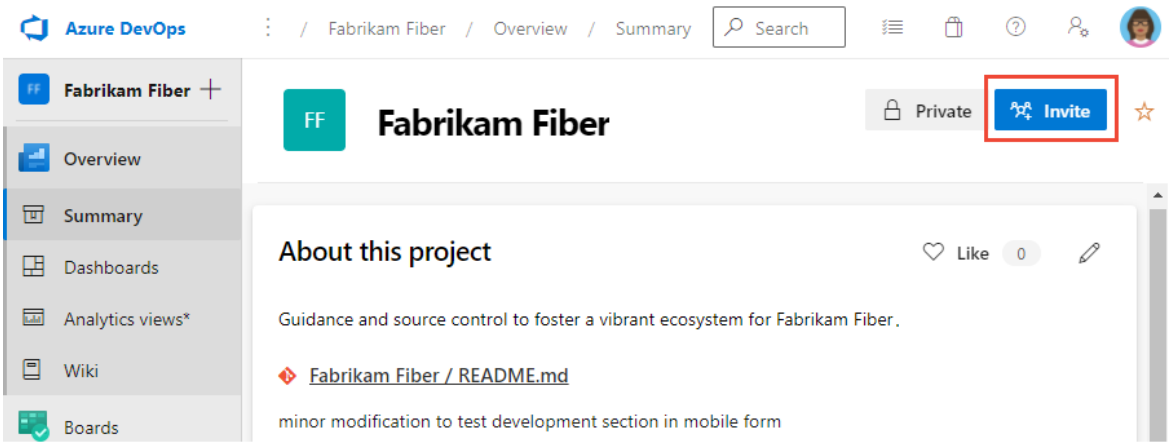
Invite users from the Summary page

As a member of the Project Administrators group, you can add members to a project from the **Summary** page and optionally add them to one or more teams. To learn more about the **Summary** page, see Share your project vision, view project activity.

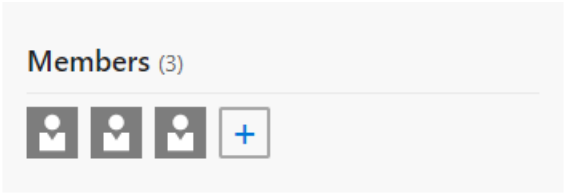
NOTE

For on-premises Azure DevOps, all email actions require an SMTP server to be configured.

1. Open the **Project>Summary** page, and choose **Invite**.



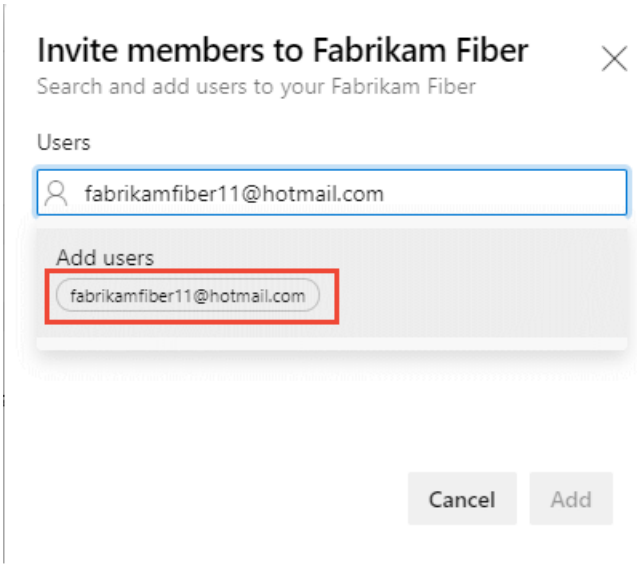
1. Open the **Project>Summary** page, and choose the **Add** button.

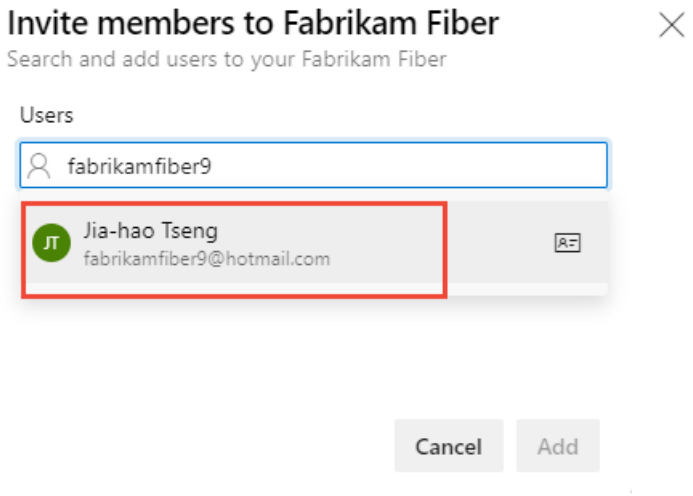


1. For new users, enter their email address. For existing users, type their name until it resolves as a known name to the system. You can add several email addresses or account names by separating them with a semicolon (;).

   Choose the entry listed under **Add users** to complete the entry.

   If you're adding a user known by the organization or collection, type the name or email address and then choose the name that appears to complete the entry.



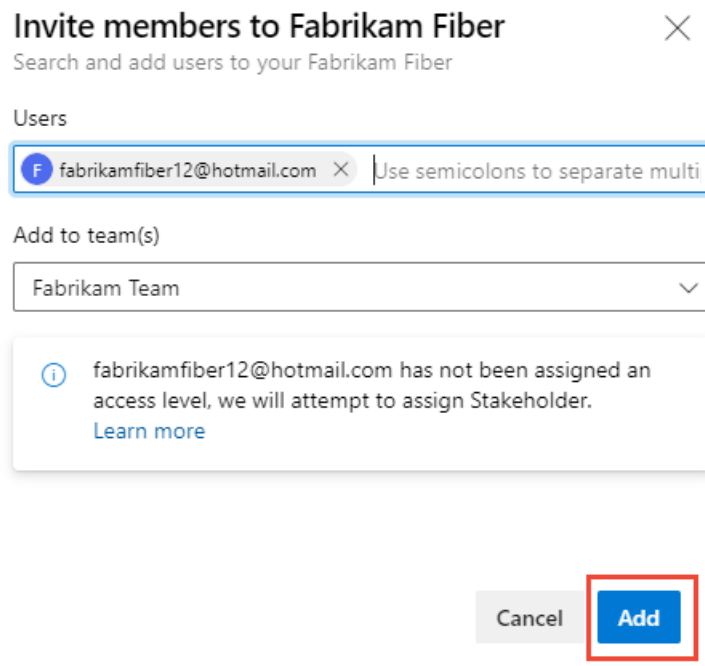


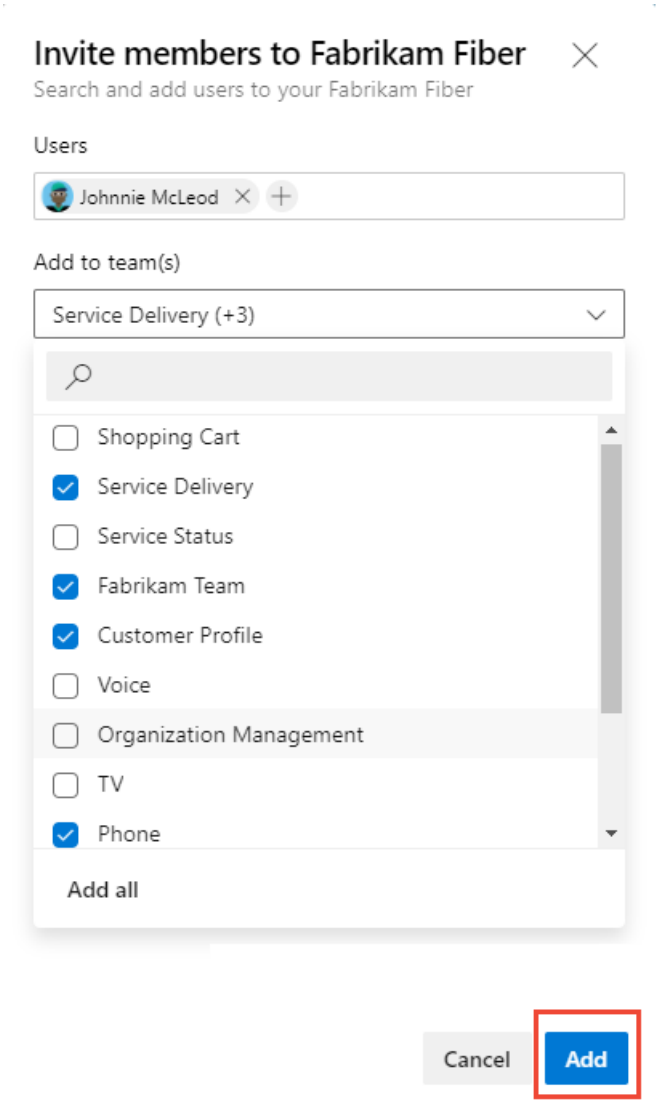
> **NOTE**
>
> Any valid email address is acceptable. When the user accepts the invitation and signs into Azure DevOps, they register their email address as a Microsoft account and choose a password.

2. Optionally, select the teams you want to add the user to and then choose **Add** to complete the invitation.

   When the user is unknown, you'll get a notification that an access level must be assigned. To complete the invitation, choose **Add**.

   Choose **Add** to complete the invitation.

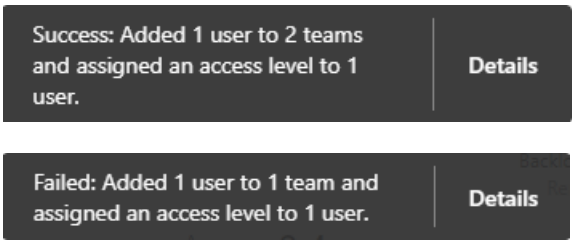




When adding a new user, the system assigns Stakeholder as the access level when all free five Basic access levels have been assigned. Active contributors to a project need to have Basic access as a minimum. A Project Collection Administrator can change the access level from the **Organization Settings > Users** page.

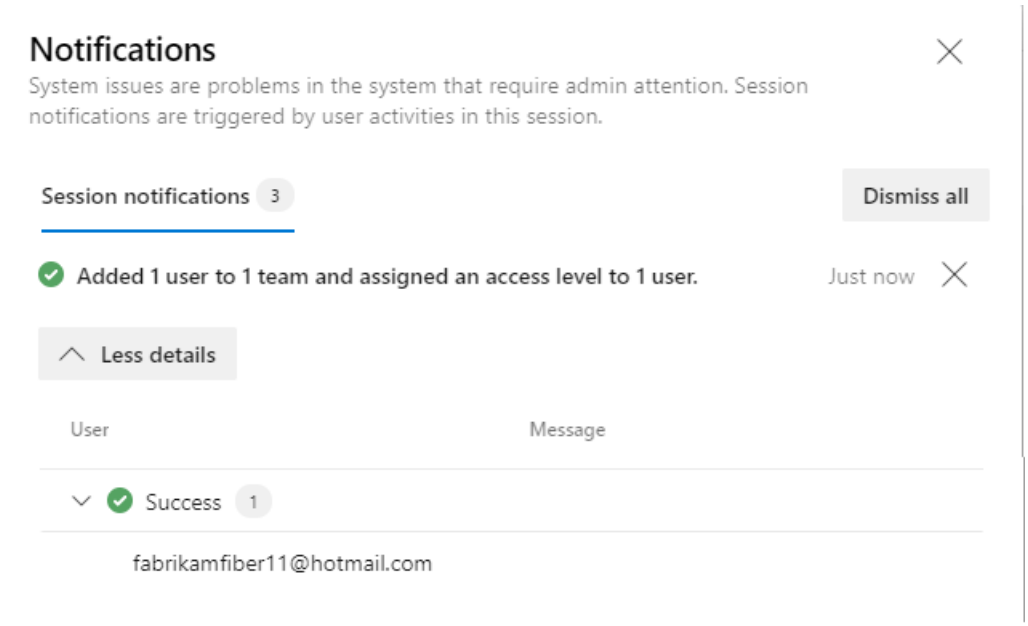
> **NOTE**
> Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see Permissions and access.

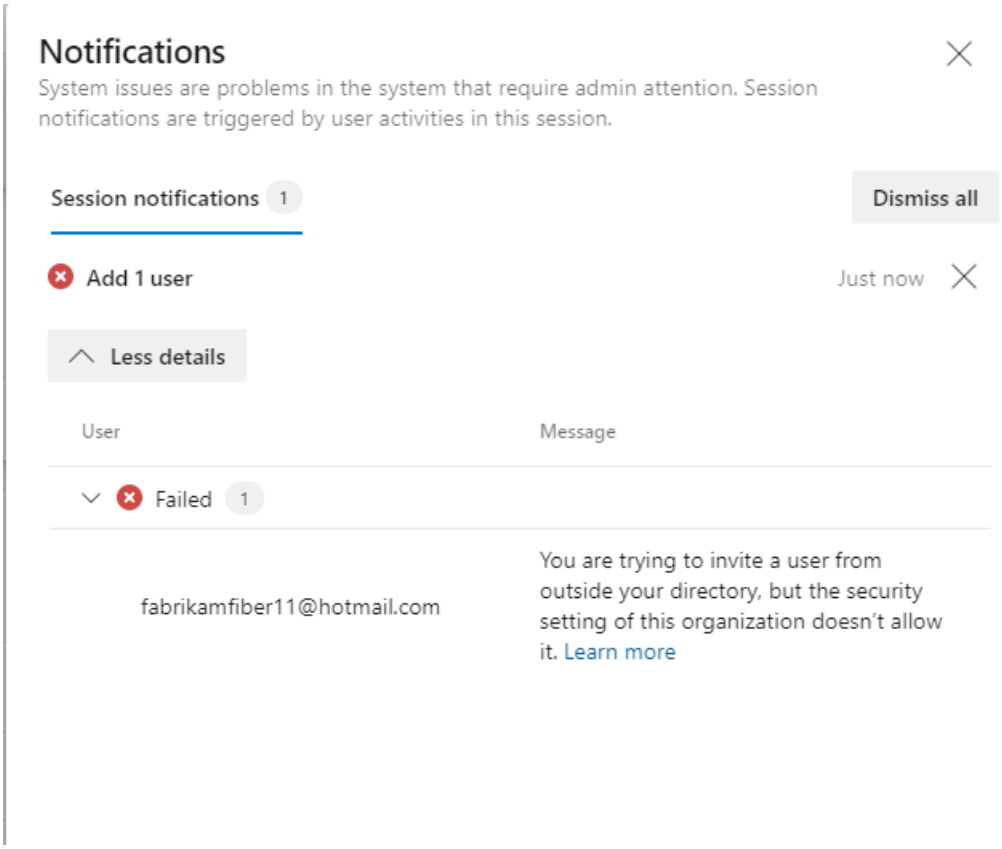
3. (Optional) A message will briefly display on the screen to indicate success or failure. Choose **Details** to open the notification and review details.



A success message indicates the status of adding the user to the system.

A failure message indicates why the addition of the user failed.





4. New users receive an email inviting them to sign in to the project. Existing users don't receive any formal notification.

Add users or groups to a project

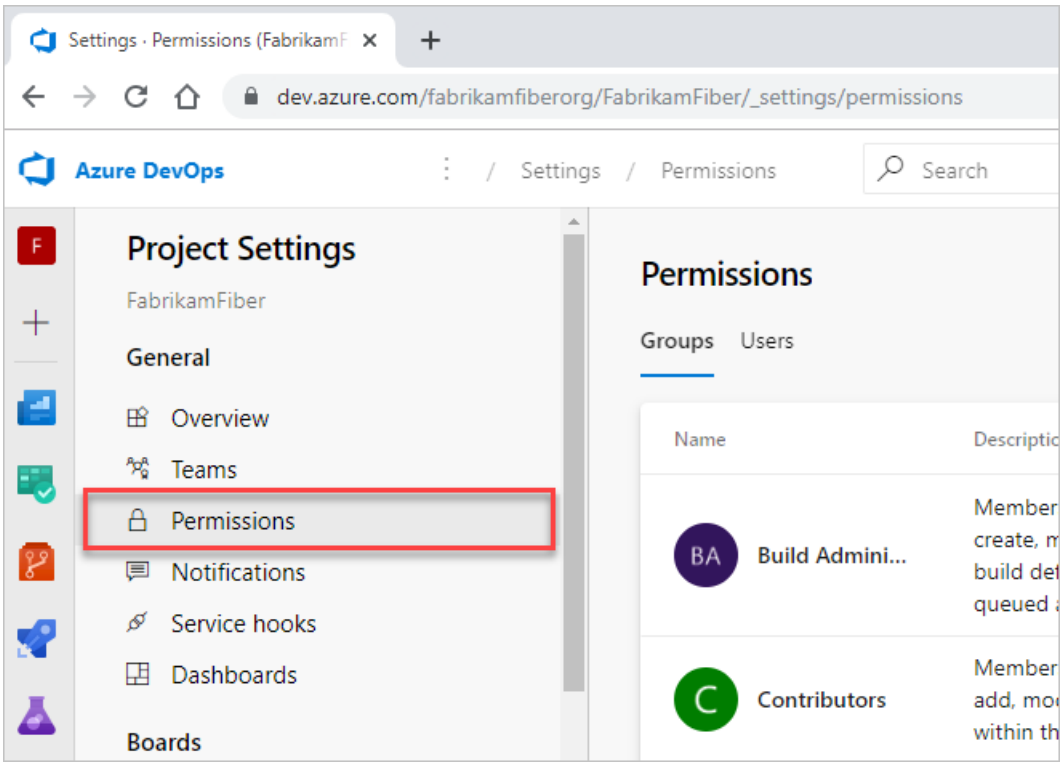
As a member of the **Project Administrators** group, you can add users or groups to a project from the **Project settings> Permissions** page by adding them to a security group. To add a custom security group, see Add or remove users or groups, manage security groups.

> **NOTE**
>
> To enable the new user interface for the **Project Permissions Settings Page**, see Enable preview features.

- Preview UI
- Current UI

1. Open the web portal and choose the project where you want to add users or groups. To choose another project, see Switch project, repository, team.

2. Choose **Project settings**, and then **Permissions**.

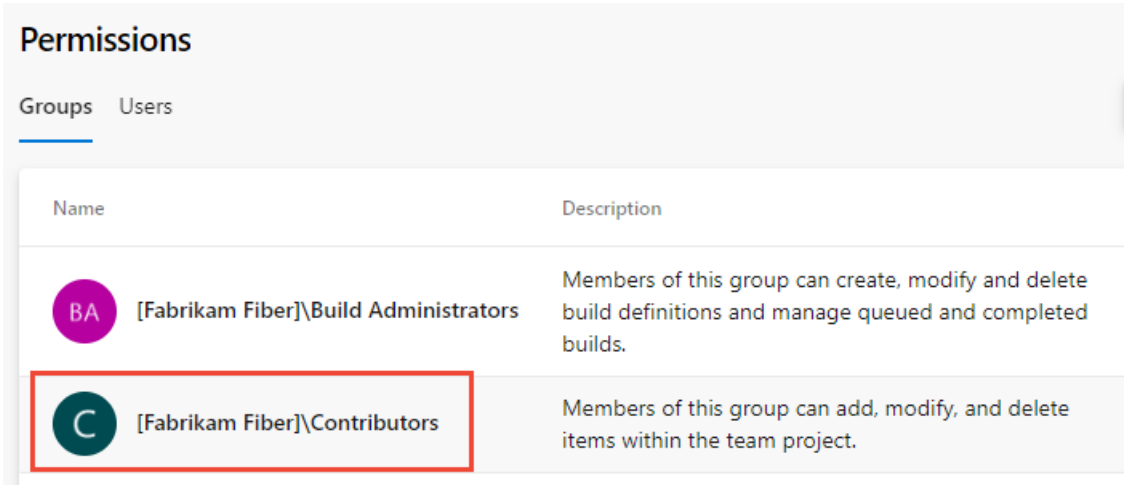


3. Under **Groups**, choose one of the following options:

   - **Readers**: To add users who require read-only access to the project, choose.
   - **Contributors**: To add users who contribute fully to this project or who have been granted Stakeholder access.
   - **Project Administrators**: To add users who need to administrate the project. To learn more, see Change project-level permissions.

   Or, you can choose any team group to add users to a specific team.

   Here we choose the **Contributors** group.



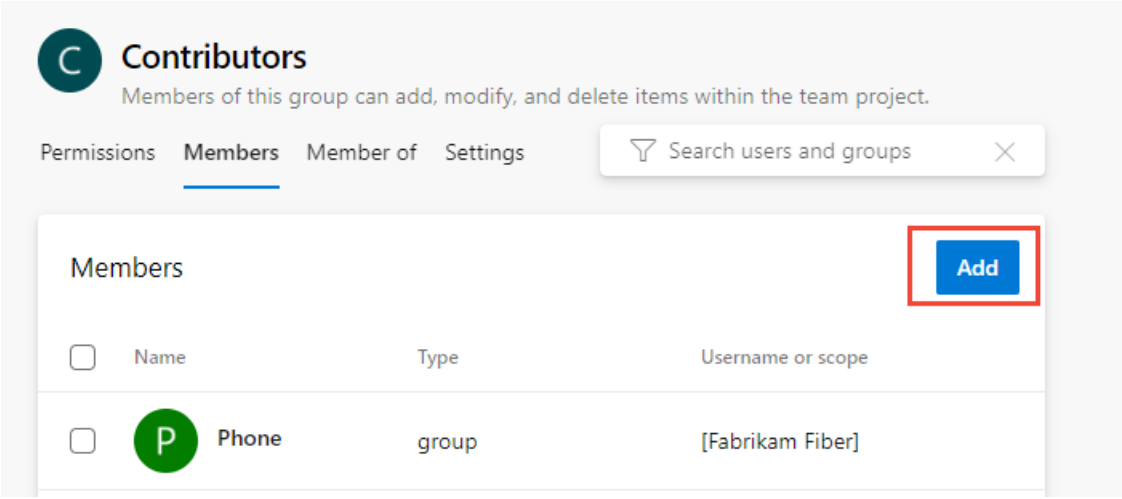
4. Next, choose the **Members** tab.

   The default team group, and any other teams you add to the project, get included as members of the

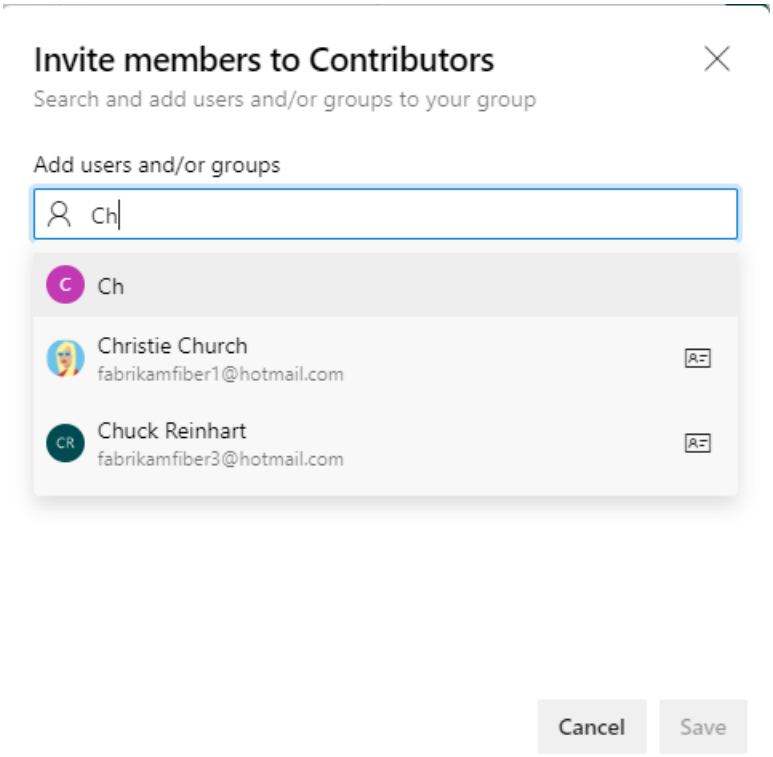
Contributors group. Add a new user as a member of a team instead, and the user automatically inherits Contributor permissions.

> **TIP**
> Managing users is much easier using groups, not individual users.

5. Choose **Add** to add a user or a user group.



6. Enter the name of the user account into the text box. You can enter several identities into the text box, separated by commas. The system automatically searches for matches. Choose the match(es) that meets your requirements.



> **NOTE**
> The first time you add a user or group to Azure DevOps, you can't browse to it or check the friendly name. After the identity has been added, you can just enter the friendly name.

Choose **Save** when done.

7. You may customize user permissions for other functionality in the project. For example, in areas and iterations or shared queries.

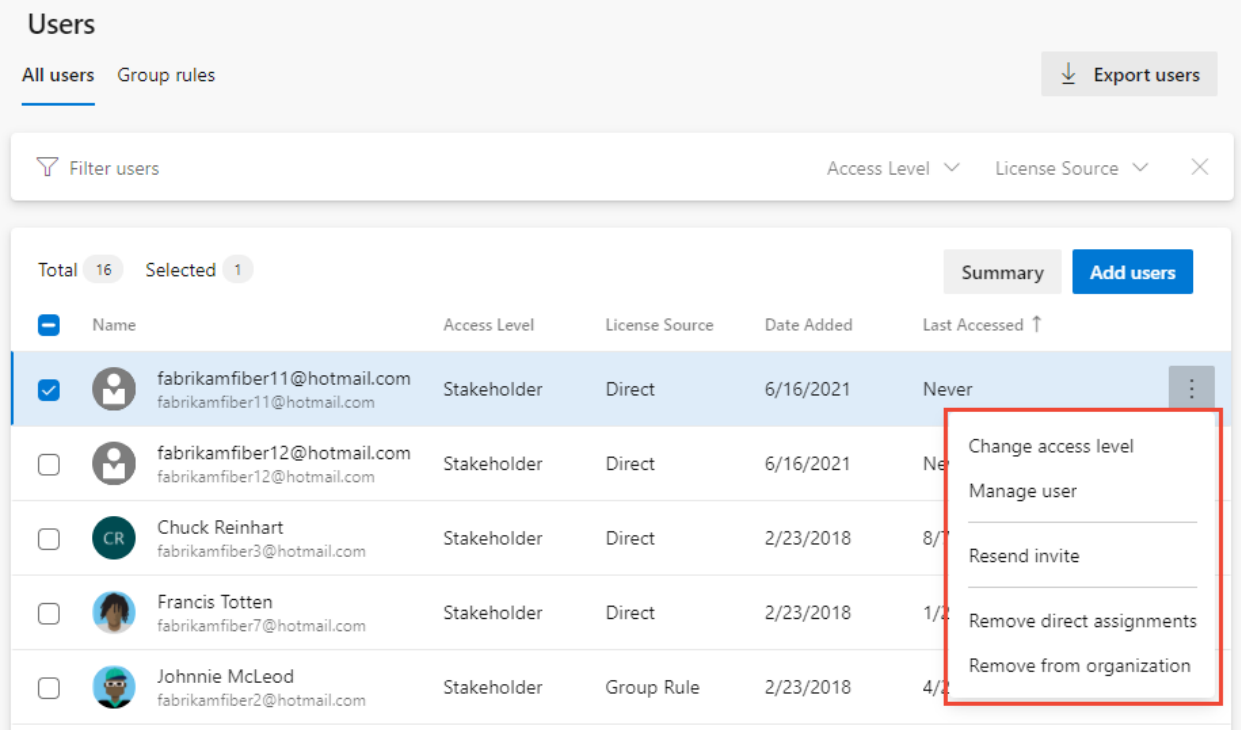
Choose the **Current page** tab for information on adding a user to a project. The Project Permissions Settings Page preview feature is only available for Azure DevOps Services at this time.

Manage users or resend invitations

Project Collection Administrators can update user assignments and resend invitations. The various options they have are:

- Change the access level
- Manage user - add them to select projects
- Resend invite
- Remove direct assignments
- Remove from organization

To learn more, see Add account users for Azure DevOps.



List team members or team details

From the Azure DevOps CLI command, you can see details about a team or list the individual members of that team. To first see a list of all teams in your organization, use the az devops team list command.

List team members | Show team details

> **NOTE**
> You can use the `az devops user` command to add users to an organization. There is no comparable command for adding users to a team or project.

List team members

You can list the individual members of a team in your organization with the az devops team list-member

command. To get started, see Get started with Azure DevOps CLI.

```
az devops team list-member --team
                           [--org]
                           [--project]
                           [--skip]
                           [--top]
```

Parameters

- **team**: Required. Name or ID of the team to show.
- **org**: Azure DevOps organization URL. You can configure the default organization using `az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using `git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.
- **project**: Name or ID of the project. You can configure the default project using `az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up using `git config`.
- **skip**: Optional. Number of members to skip.
- **top**: Optional. Maximum number of members to return.

Example

The following command lists the first five members of the team named **Fabrikam Team** and returns the details in table format.

```
az devops team list-member --team "Fabrikam Team" --top 5 --output table

ID                                    Name               Email
------------------------------------  -----------------  -------------------------
3b5f0c34-4aec-4bf4-8708-1d36f0dbc468  Christie Church    fabrikamfiber1@hotmail.com
19d9411e-9a34-45bb-b985-d24d9d87c0c9  Johnnie McLeod     fabrikamfiber2@hotmail.com
8c8c7d32-6b1b-47f4-b2e9-30b477b5ab3d  Chuck Reinhart     fabrikamfiber3@hotmail.com
d291b0c4-a05c-4ea6-8df1-4b41d5f39eff  Jamal Hartnett     fabrikamfiber4@hotmail.com
bd30c189-db0f-4dd6-9418-5d8b41dc1754  Raisa Pokrovskaya  fabrikamfiber5@hotmail.com
```

Show team details

You can view details about a team in your organization with the az devops team show command. To get started, see Get started with Azure DevOps CLI.

```
az devops team show --team
                    [--org]
                    [--project]
```

Parameters

- **team**: Required. Name or ID of the team to show.
- **org**: Azure DevOps organization URL. You can configure the default organization using `az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using `git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.
- **project**: Name or ID of the project. You can configure the default project using `az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up using `git config`.

Example

The following command shows information about the team in your organization named **Fabrikam Team** and returns the details in table format.

```
az devops team show --team "Fabrikam Team" --output table

ID                                    Name          Description
------------------------------------  ------------  -------------------------------------------------
a48cb46f-7366-4f4b-baf5-b3632398ed1e  Fabrikam Team  The default project team. Was Fabrikam Fiber Team
```

Add users or groups to an access level

For on-premises deployments, you may need to set the access level for a user or group, particularly if those groups don't belong to the default access level. To learn more, see Change access levels.

Add users or groups to SQL Server Reports

If your on-premises deployment is integrated with SQL Server Reports, you need to manage membership for those products separately from their websites. See Grant permissions to view or create SQL Server reports in Azure DevOps.

Next steps

Manage your project

Related articles

- Add users and manage access
- Resources granted to project members
- Manage your organization, Limit identity search and selection
- Manage your organization, Limit user visibility for projects and more
- Manage permissions with command line tool
- Grant or restrict access using permissions.

- Resources granted to project members
- Grant or restrict access using permissions.

Manage and configure team tools

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As a team administrator, you can customize your backlogs and board to best meet how your team works. If you need to have a team created, request a member of your Project Administrators group do so. It only takes a minute to add a new team. Team settings are managed by the team administrator role. Users assigned as team administrator can configure and manage all team tools.

Team administrators should do the following tasks:

- Add team members
- Add another team administrator
- Configure areas and iteration paths
- Configure backlogs, boards, and general settings

Also, consider the following optional tasks:

- Configure and manage team dashboards
- Configure team notifications

Prerequisites

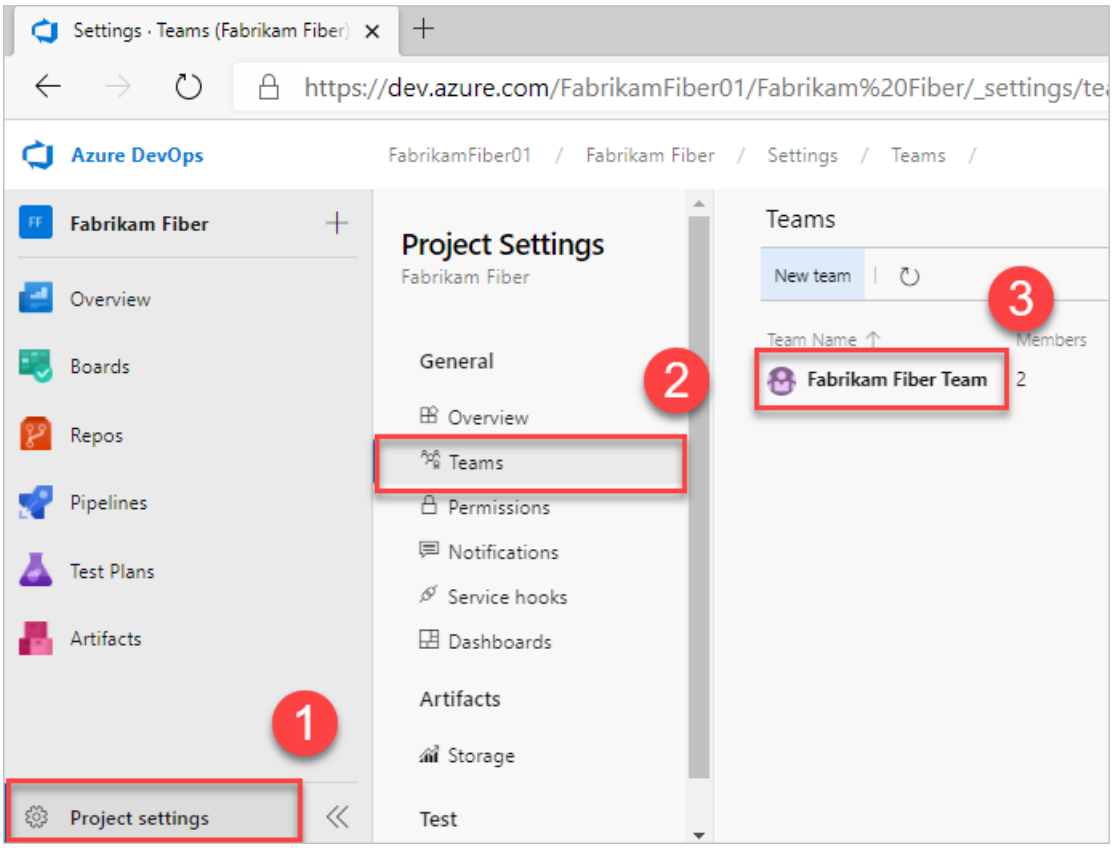
- To perform any team configuration task, you need to be added as a team administrator for the team to be modified, or be a member of the **Project Administrators** group. See Change project-level permissions.
- To add a team, you must be a member of the **Project Administrators** group. For more information, see Add teams.

> **NOTE**
> For guidance on configuring and customizing your project and teams to support your business needs, review Configuration and customization of Azure Boards.

Open your team profile

Open your team profile to quickly access items defined for your team.

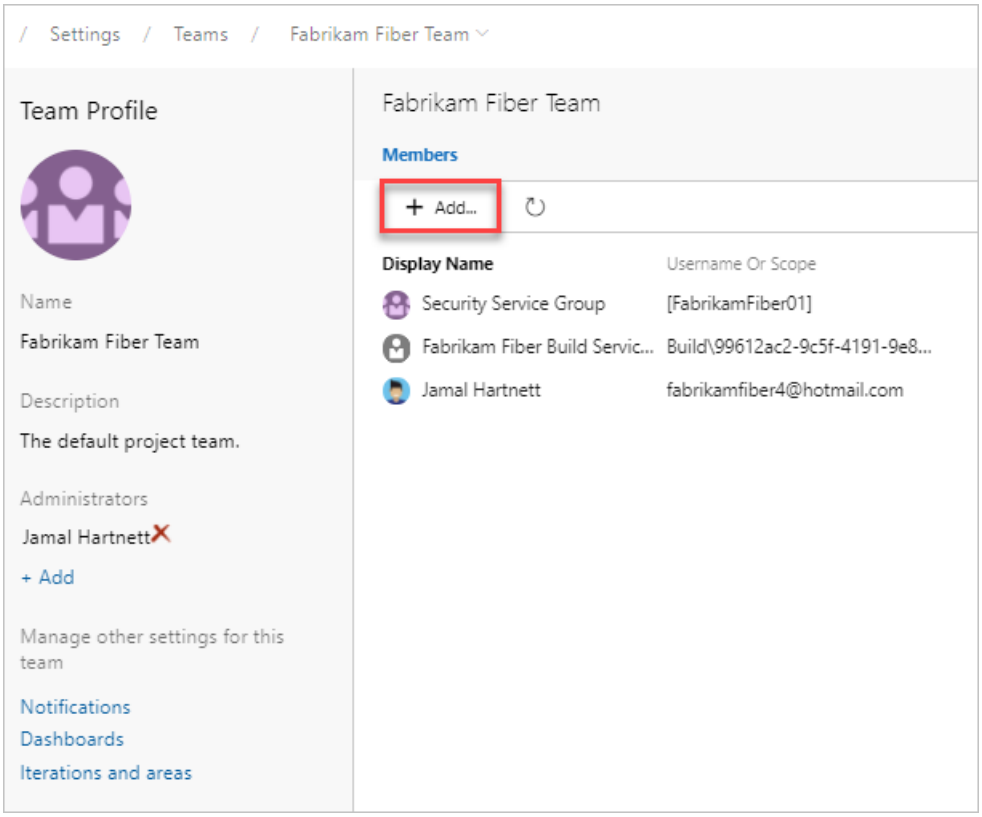
1. Sign in to your organization (`https://dev.azure.com/{yourorganization}`), and then open your project.
2. Select **Project settings** > **Teams** > your team name.



Add users to a team

Several tools, such as capacity planning, team alerts, and dashboard widgets, are team-scoped. These tools automatically reference the users that are as members of a team to support planning activities or sending alerts.

To add users to a team, see Add users to a project or specific team.



All members of a team can favorite team artifacts and define work item templates. For more information, see:

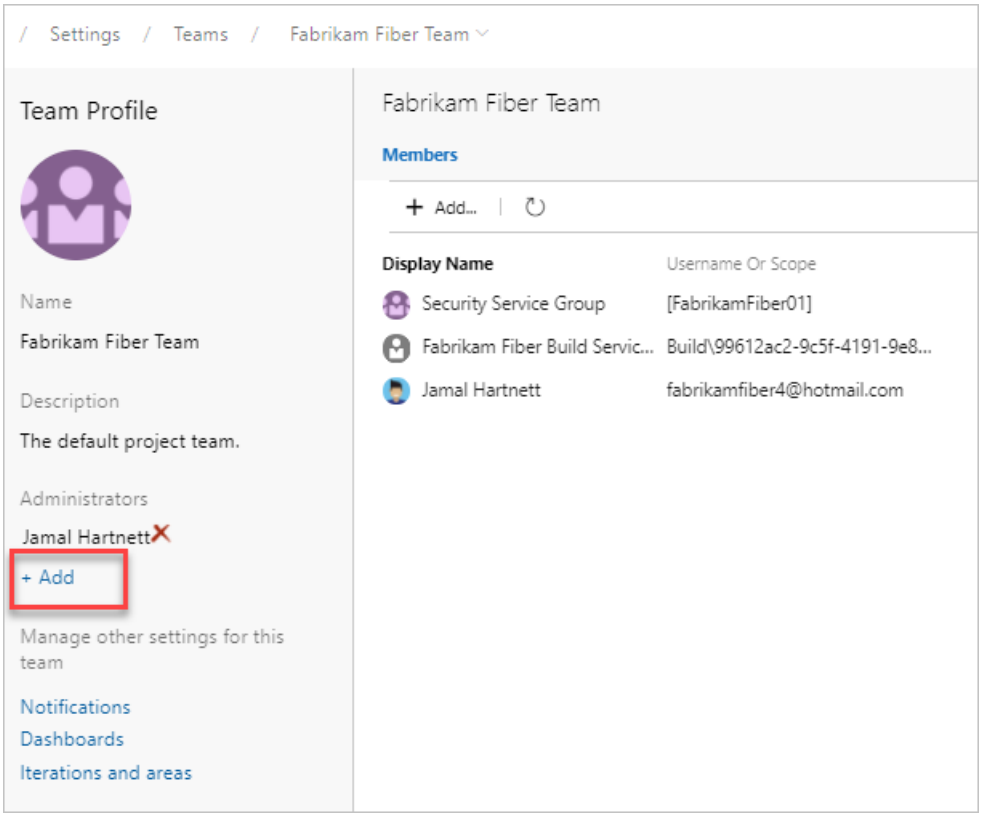
- Set personal or team favorites

- Use templates to add and update work items.

If team members don't have access to all the features they want, make sure they have the permissions needed for those features.

Add an administrator

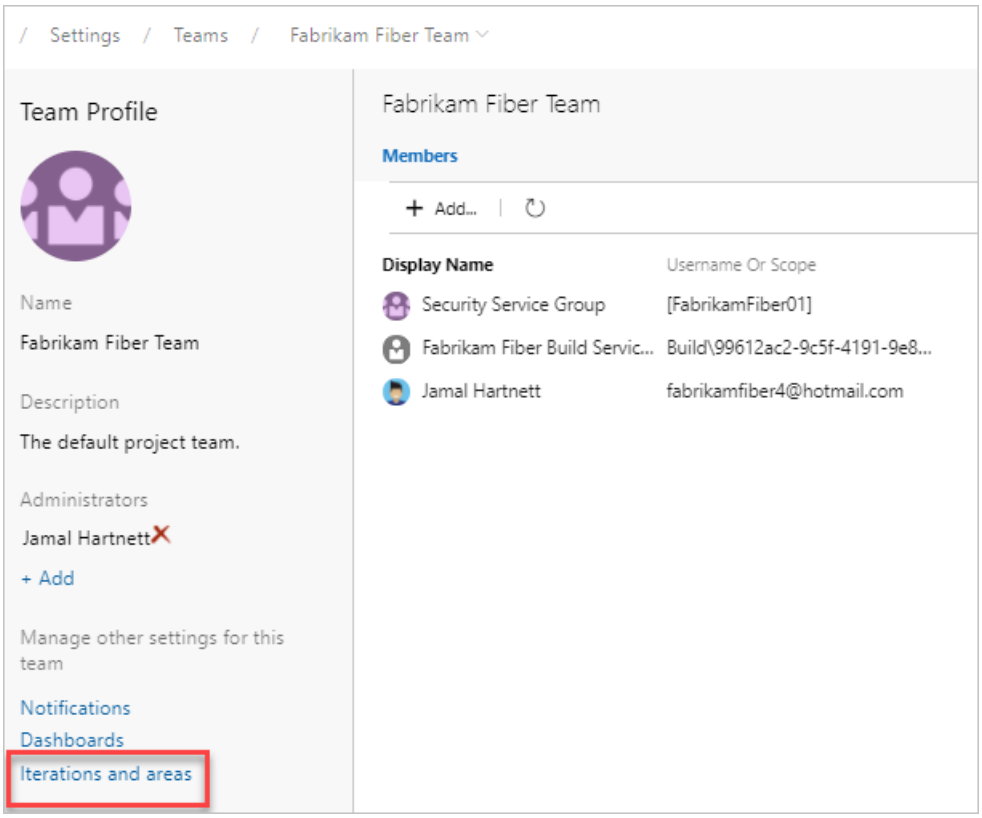
When you add a team to a project, a Project Administrator should add one or more team administrators.



Configure team areas and iterations

Many Agile tools depend on the area and iteration paths that are configured for the team. To learn more about configuring team areas and iterations, see About teams and Agile tools.

Once project administrators add area paths and iteration paths for a project, team administrators can select the area and iteration paths associated with their team. These settings affect many Agile tools available to the team.



Settings include making the following associations for each team:

- **Select team Area Paths**
  Can select the default area path(s) associated with the team. These settings affect many Agile tools available to the team.
- **Select team Iteration Paths or sprints** Can select the default area path(s) associated with the team. These settings affect many Agile tools available to the team.

For more information, see Define area paths and assign to a team and Define iteration paths and configure team iterations.

Configure team backlogs, boards, and general settings

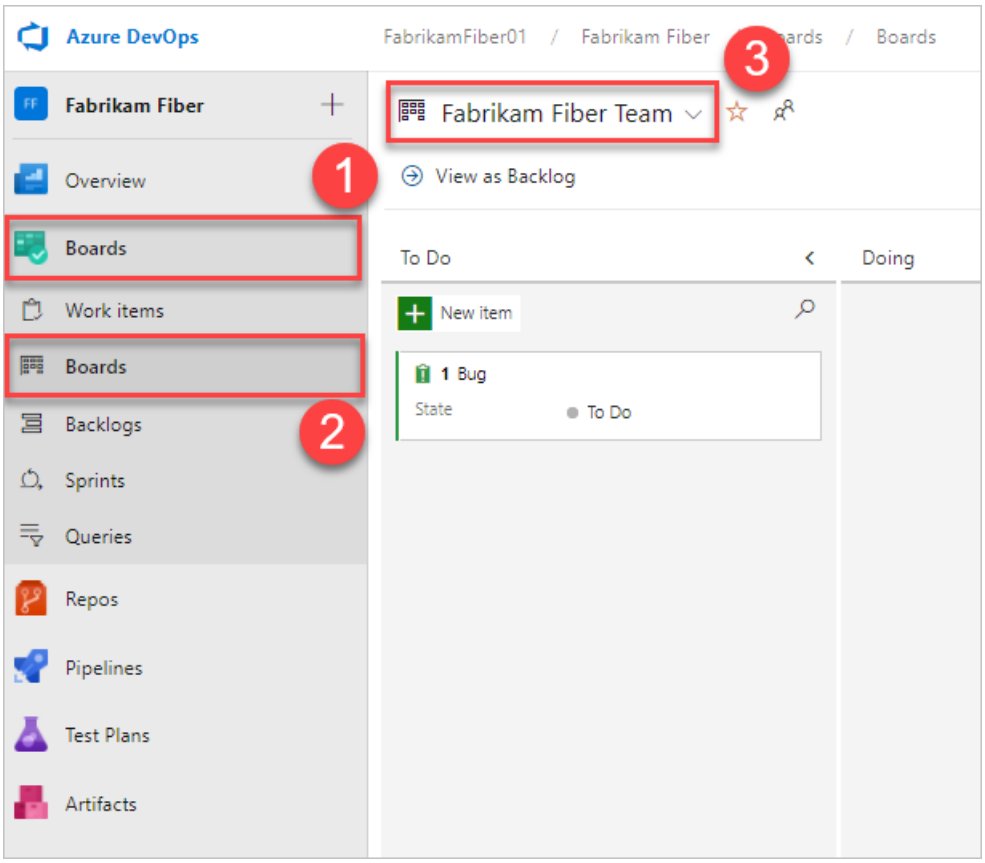
Team administrators can choose which backlog levels are active for a team. For example, a feature team may choose to show only the product backlog and a management team may choose to show only the feature and epic backlogs. Also, administrators can choose whether bugs are treated similar to user stories and requirements or as tasks.

Team administrators can also choose which days are non-working days for the team. Sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown.

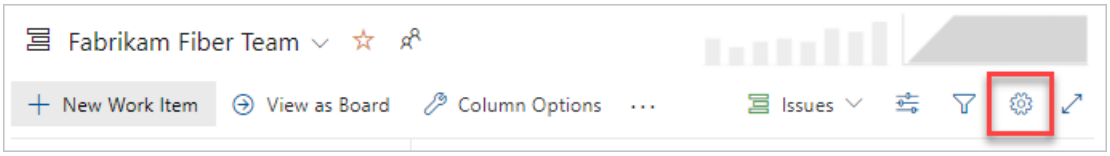
You can configure most of your team settings from the common configuration dialog.

> **NOTE**
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans.
> If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

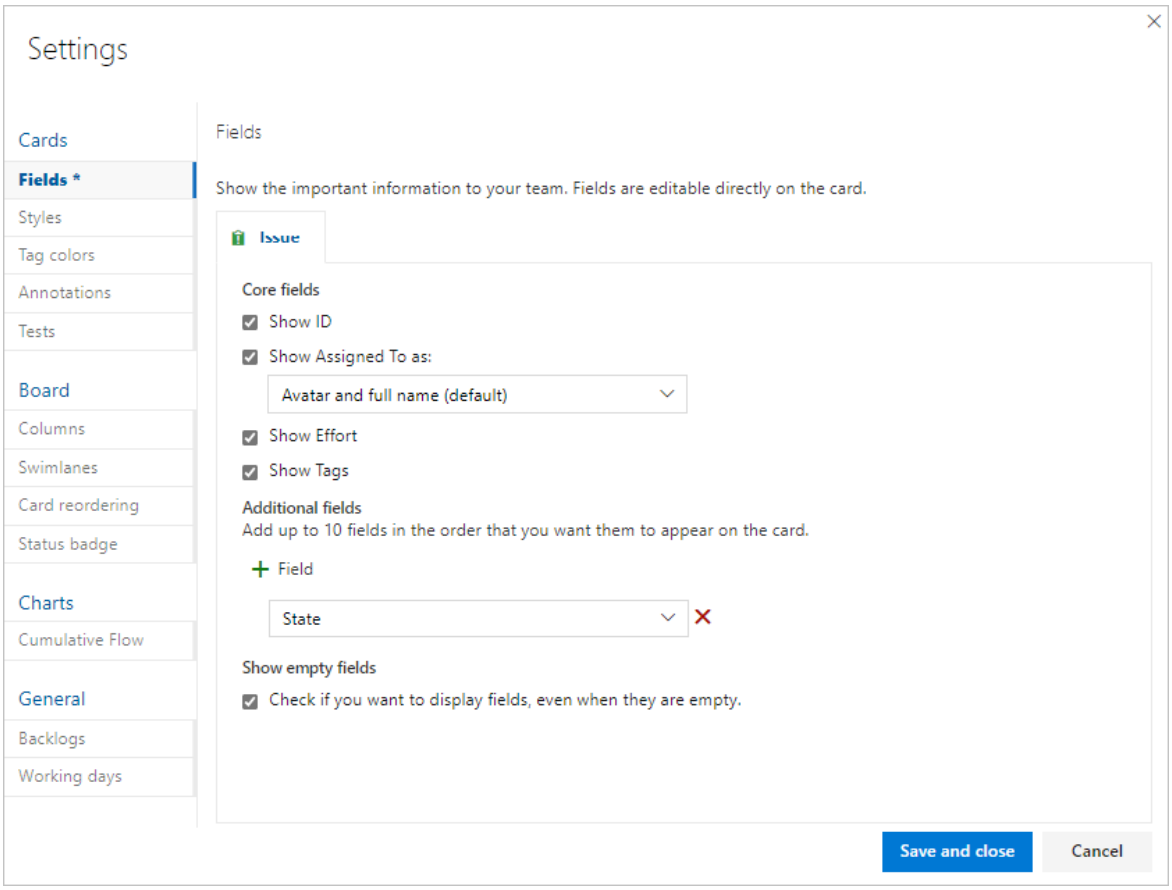
1. Check that you selected the correct project, and then choose **Boards** > **Boards**, and select the correct team from the team selector dropdown menu. For more information, see Use breadcrumbs and selectors to navigate and open artifacts.



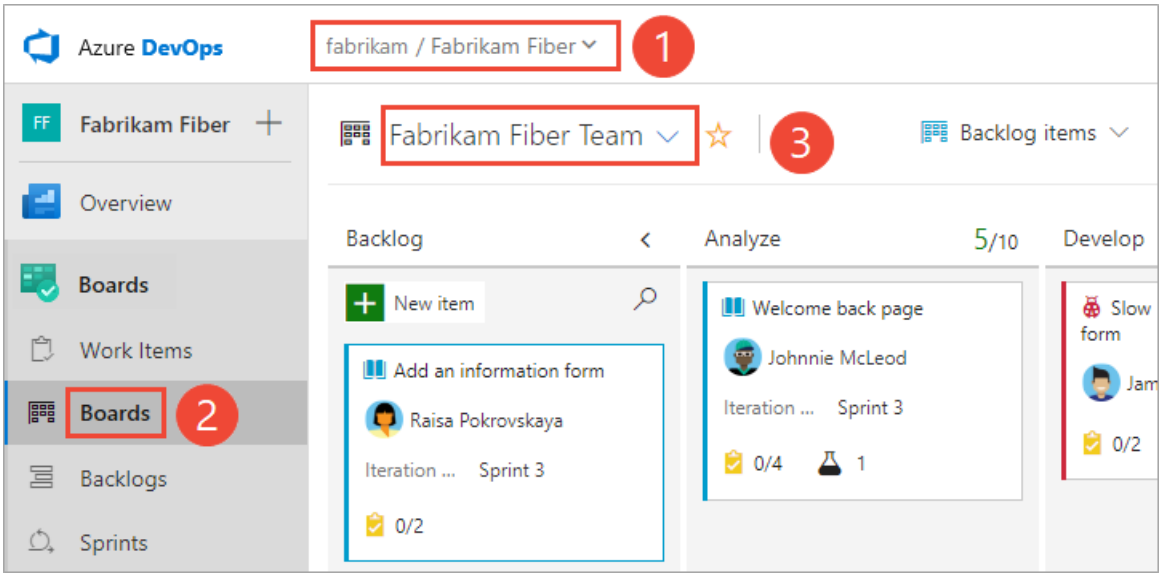
2. Choose **Team settings** to configure the board and set general team settings.



3. Choose a tab under any of the sections—**Cards, Board**, **Charts**, and **General**—to configure the cards or boards, the cumulative flow chart, or other team settings. When you're done configuring the settings, select **Save and close**.

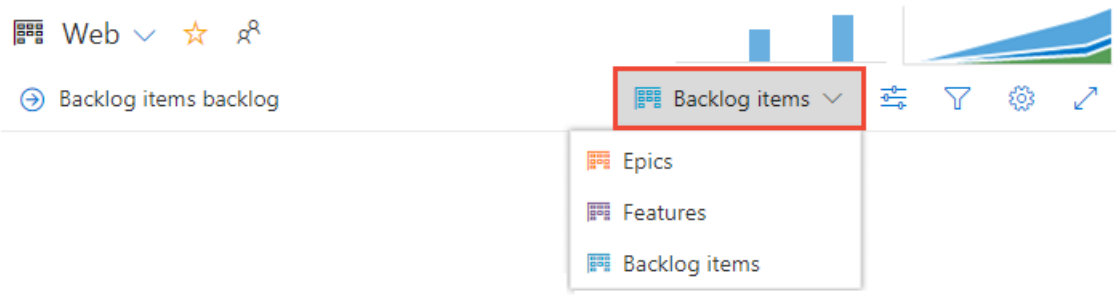


1. Check that you selected the right project, (2) choose **Boards** > **Boards**, and then (3) select the correct team from the team selector menu.



2. Make sure that you select the team backlog or board that you want to configure using the team selector. To learn more, see Use breadcrumbs and selectors to navigate and open artifacts.

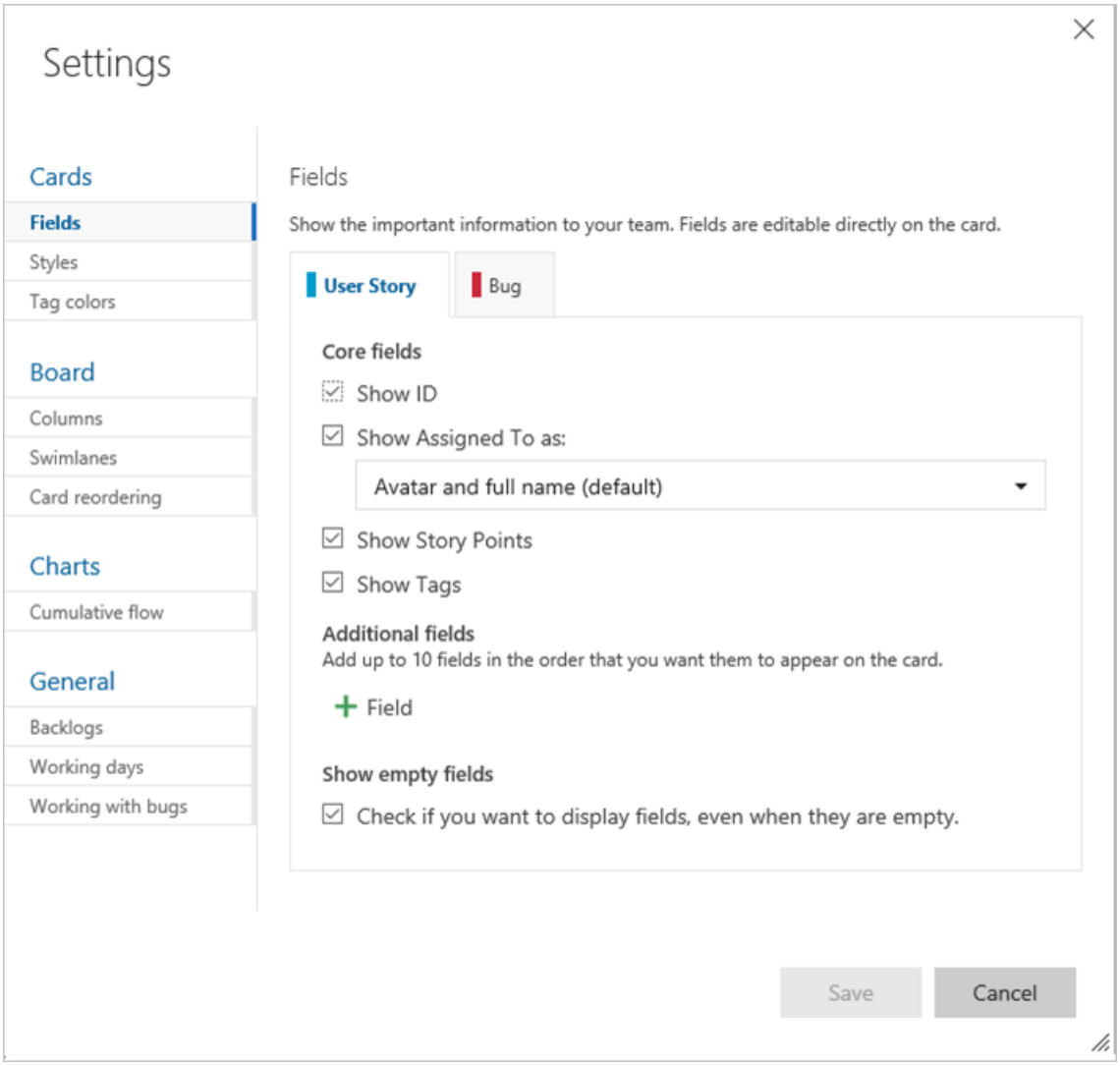
3. Choose the product or portfolio backlog from the board-selection menu.



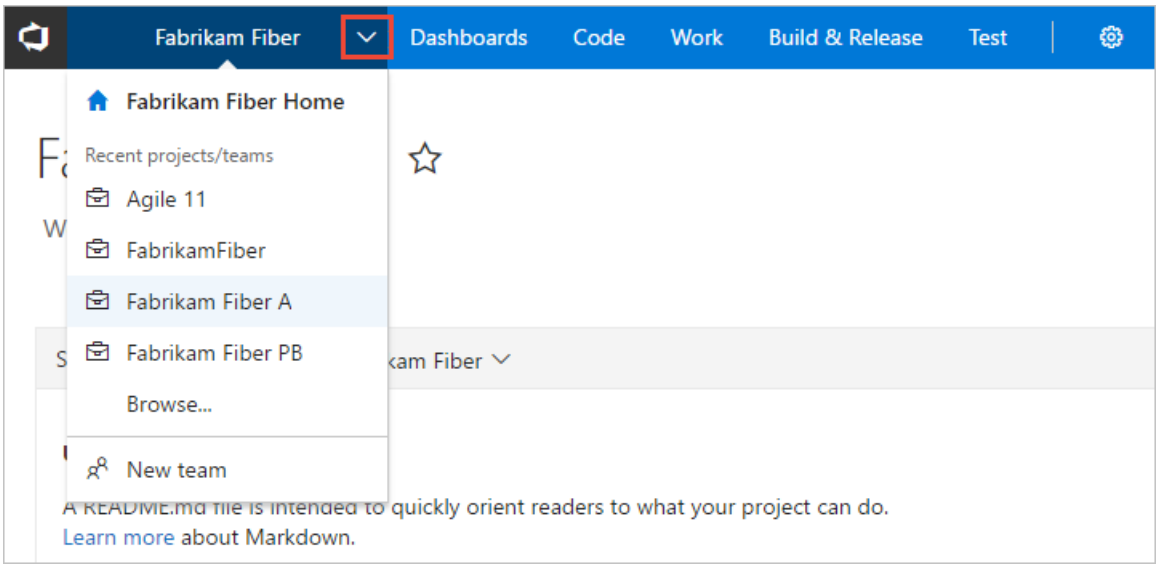
4. Choose **Team settings** to configure the board and set general team settings.



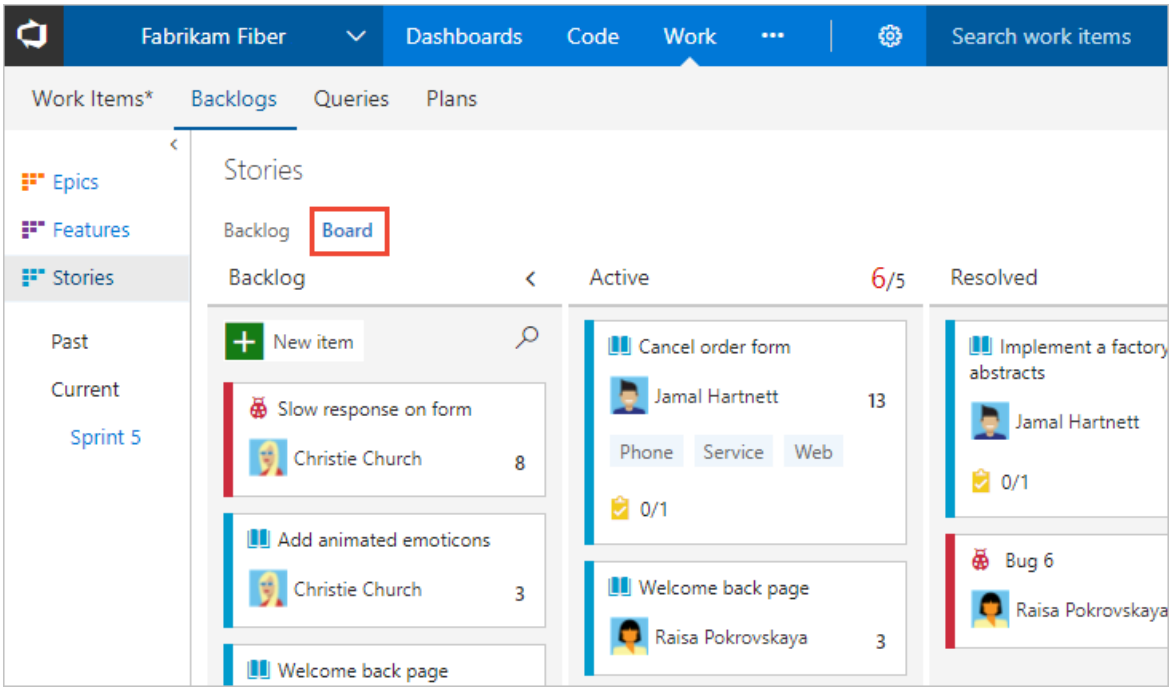
5. Choose a tab under any of the sections—**Cards, Board**, **Charts**, and **General**—to configure the cards or boards, the cumulative flow chart, or other team settings.



1. Make sure that you select the team from the project/team selector. You can switch your team focus to one that you've recently viewed from the project/team selector. If you don't see the team or project you want, choose **Browse...** or choose **Azure DevOps** to access the **Projects** page.



2. Open **Work** > **Backlogs** > **Board**.



3. Choose the board you want to configure and then choose **Team settings** to configure the board and set general team settings.

   For example, from the Kanban board ...



4. Choose a tab under **Cards** or **Board** to configure the cards and Kanban board columns and swimlanes.

   ![Common configuration dialog team settings]../.../boards/boards/media/customize-cards/common-config-141.png)

Team administrators can fully customize the team's Kanban boards associated with the product and portfolio backlogs. You configure a Kanban board by first defining the columns and WIP limits from the common configuration dialog. For guidance, see Kanban basics.

For more information on each configuration option, see the following articles:

General

- Backlogs
- Working days
- Working with bugs

Cards

- Add fields
- Define styles
- Add tag colors
- Enable annotations
- Configure inline tests

Boards

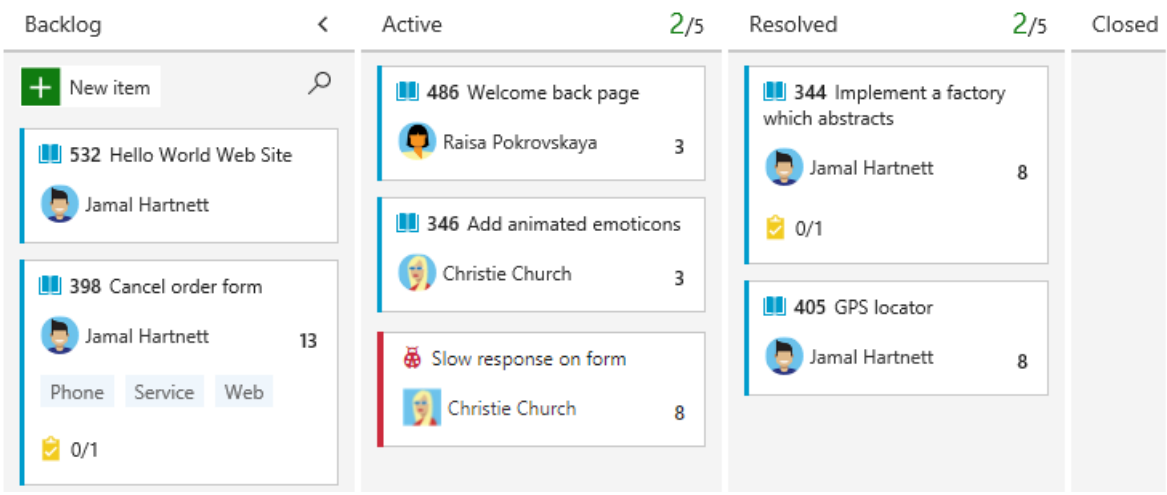
- Add columns
- Split columns
- WIP limits
- Definition of Done
- Add swimlanes
- Card reordering
- Configure status badges

- Add columns
- Split columns
- WIP limits
- Definition of Done
- Add swimlanes
- Card reordering

Chart

- Configure cumulative flow chart

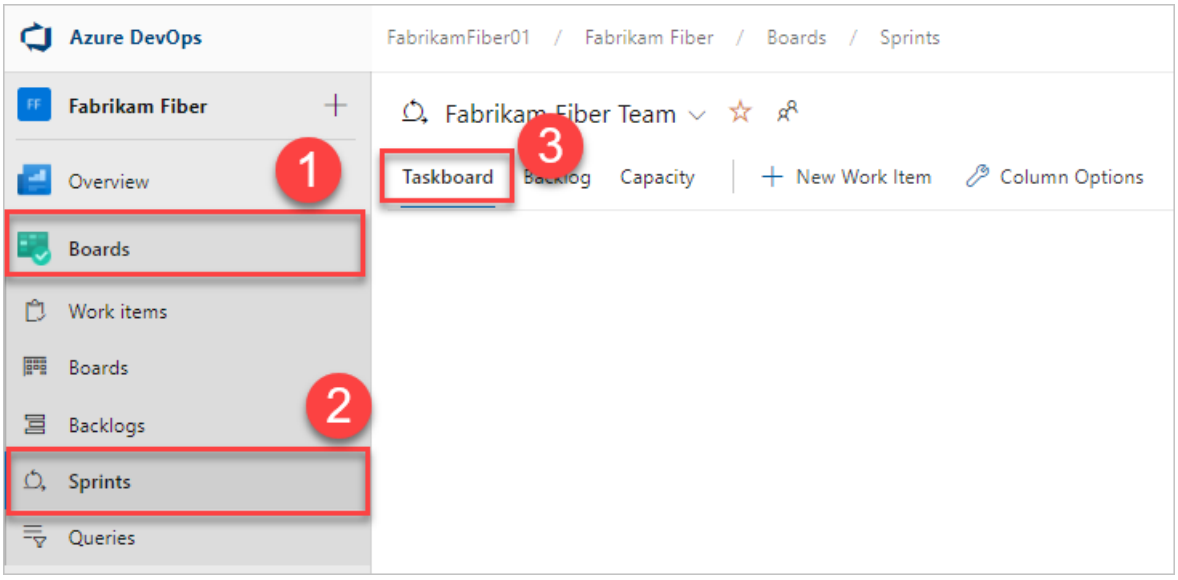
Kanban board



Configure sprint Taskboards

Similar to Kanban boards, each sprint Taskboard can be customized to support information-rich, color-coded cards as well as addition of customized columns. For details, see Customize sprint Taskboards.

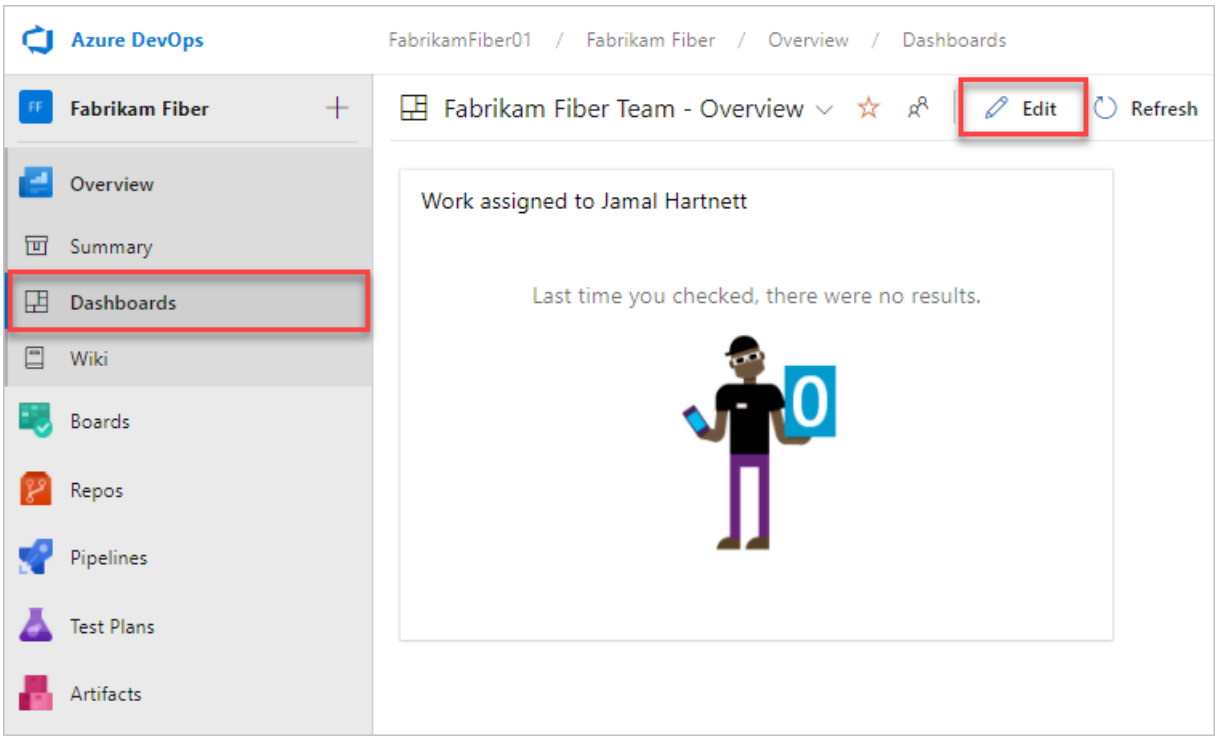
Similar to Kanban boards, each sprint Taskboard can be customized to support information-rich, color-coded cards. For details, see Customize sprint Taskboards.



Add and manage team dashboards

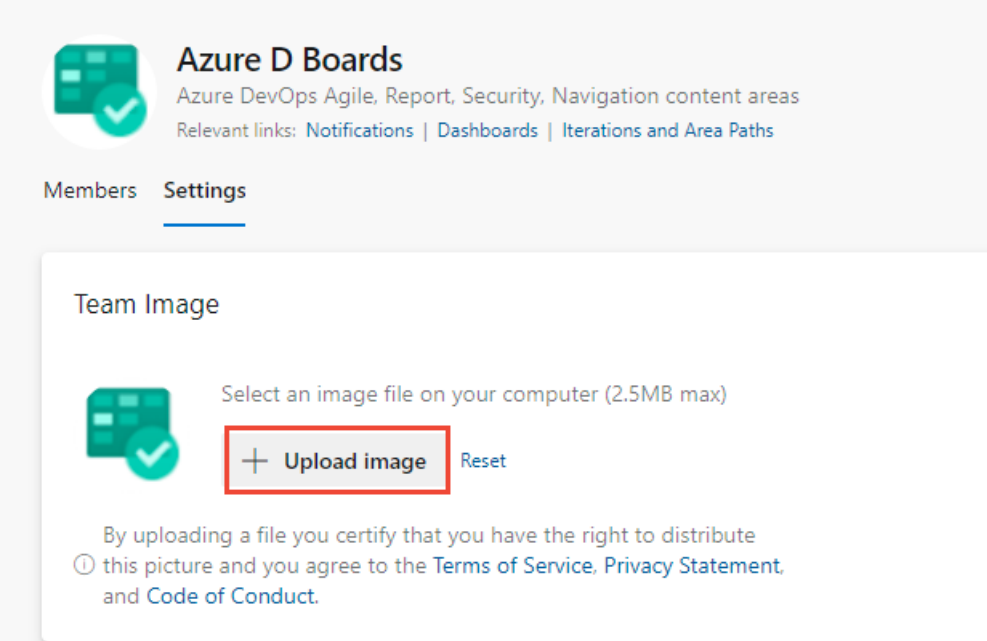
By default, all team members can add and edit team dashboards. In addition, team administrators can manage permissions for team dashboards. For details, see Add and manage dashboards.

Team administrators can add, configure, and manage permissions for team dashboards. For details, see Add and manage dashboards.



Update team name, description, and image

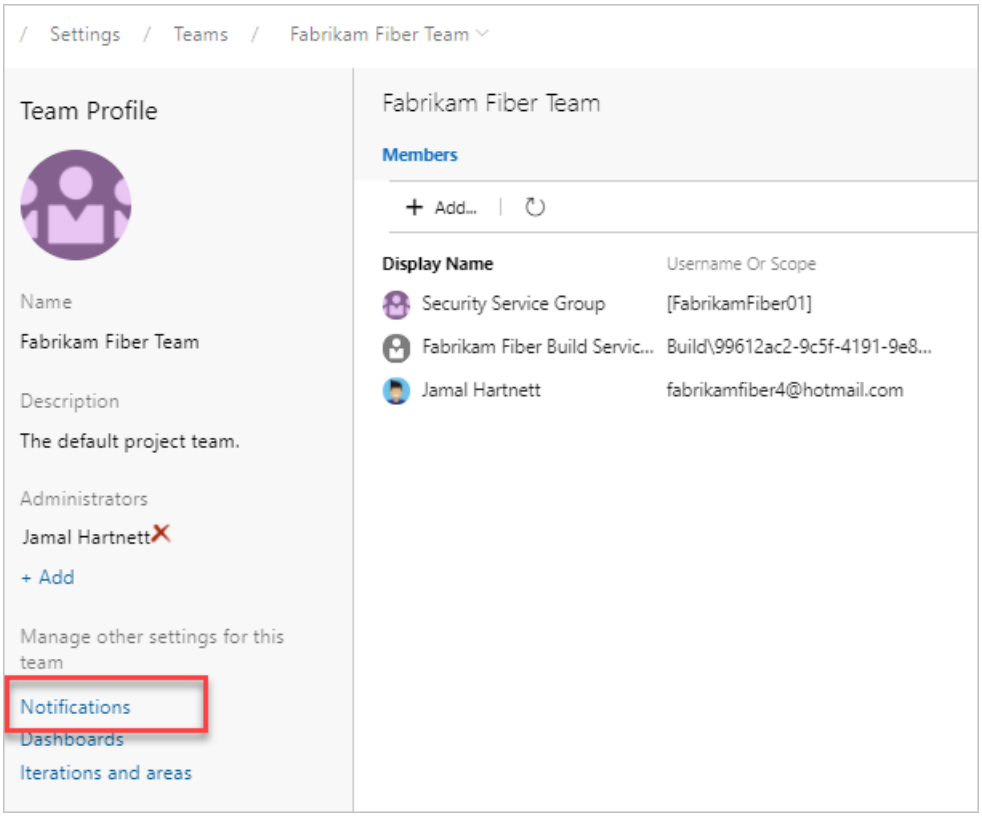
Team settings also include the team name, description, and team profile image. To add a team picture, select the image icon. The maximum file size is 2.5 MB.



Team settings also include the team name, description, and team profile image. To add a team picture. Open the Team Profile and choose the picture icon. The maximum file size is 4 MB.

Manage notifications

Team administrators can add and modify alerts so that the team can receive email notifications as changes occur to work items, code reviews, source control files, and builds. Many alerts are defined for each team. For details, see Manage team alerts.



Related articles

- About projects and scaling your organization

- About teams and Agile tools
- Add teams
- Add a team administrator

Set the Work Items experience

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Visual Studio 2019 | Visual Studio 2022

Visual Studio 2019 supports switching between the default view of the Team Explorer **Work Items** page and the legacy view. The default view is designed to match the web portal **Boards>Work Items** page. The legacy view supports the Work Items page available with the previous versions of Visual Studio as described in the linked articles listed below.

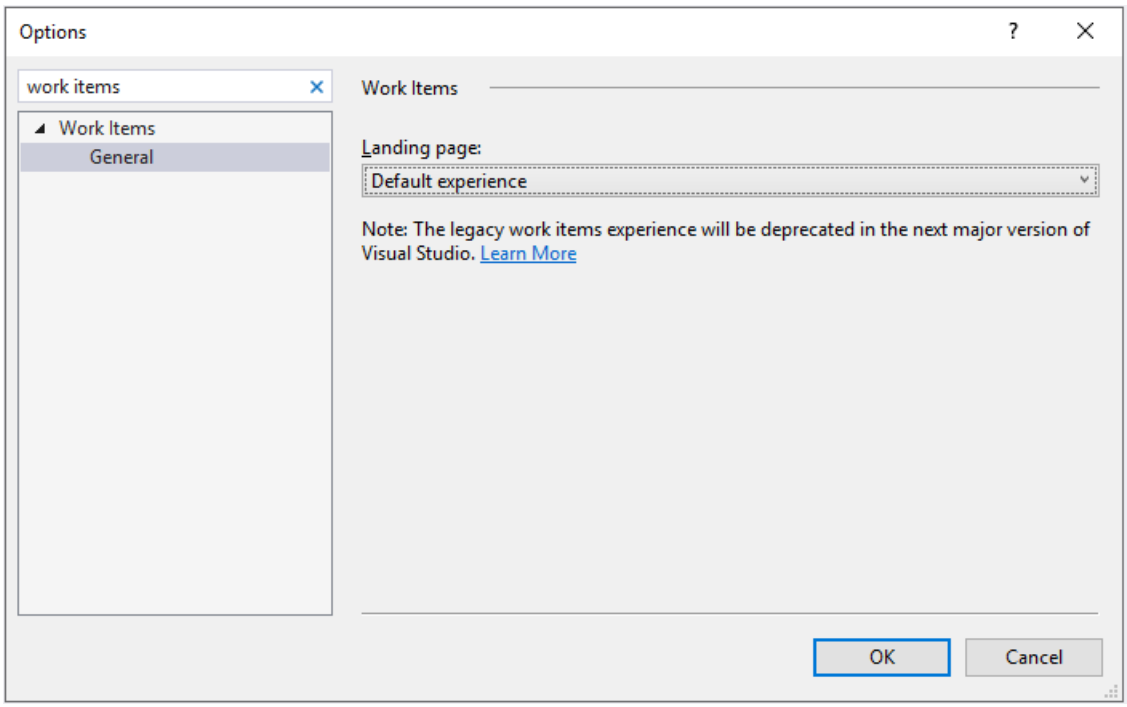
Each view supports the following tasks:

DEFAULT EXPERIENCE	LEGACY EXPERIENCE
- View and add work items	- Add work items - Use the query editor to list and manage queries - Organize query folders and set query permissions - Open query in Excel - Email query results list using Outlook - Create reports from query in Excel (on-premises Azure DevOps Server only)

> **NOTE**
> If you want to quickly access a list of work items based on their assignment to you, following, mentioned, or recent updates; then use the new default experience. However, if you rely on accessing queries from Team Explorer, then we recommend that you set your Landing page to the legacy option.

Change your Work Items view

1. From Visual Studio 2019 choose **Tools>Options**.
2. In the Options dialog, enter **work items** in the search box.
3. Choose your **Landing page** option from the menu.



To learn more about **Options**, see Options Dialog Box (Visual Studio).

New Git tool experience

Visual Studio 2019 now includes a new Git tool that provides an improved experience when connecting to a Git repository. When you enable this tool, the Team Explorer tool is effectively disabled when connected to a Git repository. You can acquire the new tool by downloading Visual Studio 2019 version 16.6. To enable and use the new tool, see Git experience in Visual Studio.

Related articles

- View and add work items using the Work Items page

Azure Boards-GitHub integration

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Use this guide to connect Azure Boards with one or more GitHub repositories. This connection uses the Azure Boards app for GitHub to support the integration between Azure Boards and GitHub. This app is free for both public and private repositories.

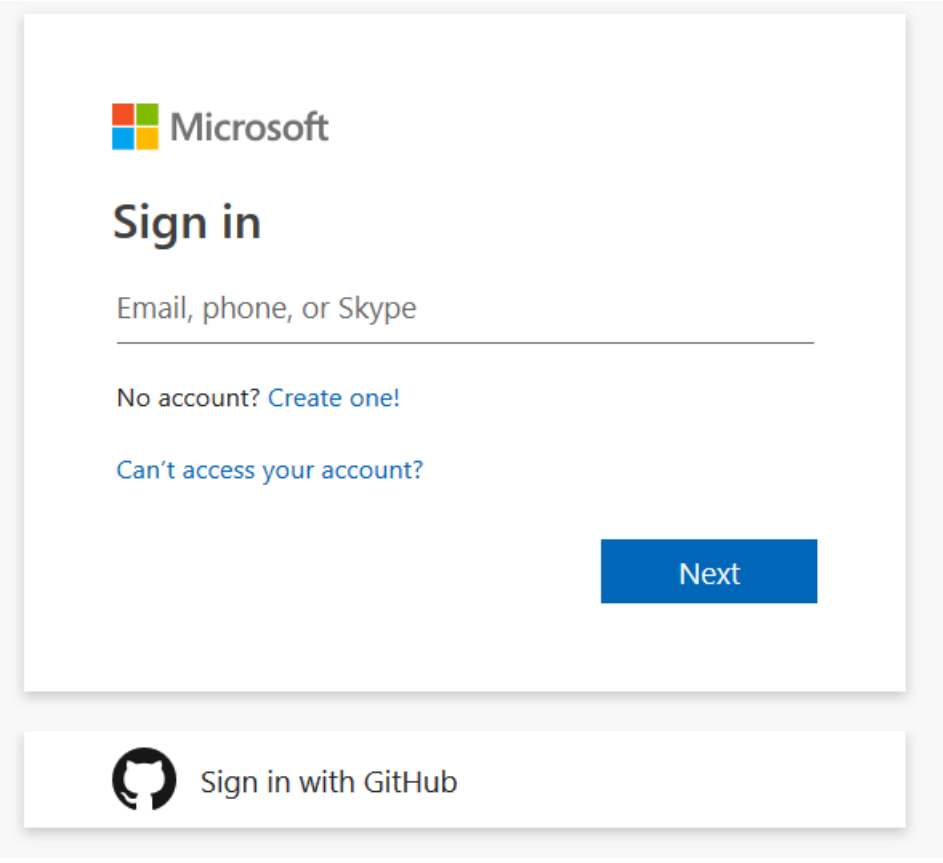
By connecting Azure Boards with GitHub repositories, you enable linking between GitHub commits, pull requests, and issues to work items. You can use GitHub for software development while using Azure Boards to plan and track your work. Azure Boards provides the scalability to grow as your organization and business needs grow.

NOTE

Azure DevOps only supports integation with GitHub repositories or Azure Repos Git repositories. Integration with other Git repositories is not supported.

Sign up for Azure Boards

If you haven't yet signed up for Azure Boards, you can do that now. See Sign up for free and invite others to collaborate on your project. You can also sign up and sign in using your GitHub credentials.



Along with accessing developer services such as Azure DevOps and Azure, you can use your GitHub account to

access all Microsoft online services, from Excel Online to XBox.

Supported integration scenarios

Azure Boards and Azure DevOps Services support integration with GitHub.com and GitHub Enterprise Server repositories. On-premises Azure DevOps Servers support integration with GitHub Enterprise Server repositories.

> **NOTE**
> By installing the Azure DevOps Server 2020.1.1 Patch 2, you can create connections from your Azure DevOps Server to GitHub.com repositories in addition to GitHub Enterprise Server repositories.

Azure Boards-GitHub integration supports the following connections:

- **From GitHub**:
  - Support integration for all repositories for a GitHub account or organization, or select repositories.
  - Add or remove GitHub repositories that participate in the integration and configure the project they connect to.
  - Suspend Azure Boards-GitHub integration or uninstall the app.
- **From Azure Boards**:
  - Connect one or more GitHub repositories to an Azure Boards project.
  - Add or remove GitHub repositories from a GitHub connection within an Azure Boards project.
  - Completely remove a GitHub connection for a project.
  - Allow a GitHub repository to connect to one or more Azure Boards projects within the same Azure DevOps organization or collection.

Azure Boards-GitHub integration supports the following operational tasks:

- Create links between work items and GitHub commit, pull requests, and issues based on GitHub mentions
- Support state transition of work items to a "Done" or "Completed" state when using GitHub mention by using `fix`, `fixes`, or `fixed`
- Support full traceability by posting a discussion comment to GitHub when linking from a work item to a GitHub commit, pull requests, or issue
- Show linked to GitHub code artifacts within the work item Development section
- Show linked to GitHub artifacts as annotations on Kanban board cards
- Support status badges of Kanban board columns added to GitHub repositories.

The following tasks aren't supported at this time:

- Query for work items with links to GitHub artifacts. However, you can query for work items with an `External Link Count > 0`.

Connect Azure Boards to GitHub repositories

You can start from Azure Boards or from GitHub to make the connection. You can connect up to 250 GitHub repositories to an Azure Boards project.

> **NOTE**
> Connection to more than 100 GitHub repositories requires Azure DevOps Server 2020.1 update or later version.

You can start from Azure Boards or from GitHub to make the connection. You can connect up to 100 GitHub repositories to an Azure Boards project.

> **NOTE**
> We recommend that you use the Azure Boards app for GitHub to configure and manage your connections to GitHub.com. The app provides a more streamlined configuration experience and has the advantage of authenticating and operating as the app rather than an individual. Once you have configured your connection, you can manage the connected repositories either from Azure Boards or GitHub.com.

To start from GitHub

- Install and configure the Azure Boards app for GitHub
  - Install the Azure Boards app for GitHub for your GitHub account or organization
  - Choose all or select repositories to participate in the integration
  - Approve and install the app
- Connect Azure Boards to GitHub (Cloud):
  - Specify the GitHub repositories to connect to an Azure Boards project
- Connect Azure DevOps Server to GitHub Enterprise Server (On-premises):
  - Specify the GitHub repositories to connect to an Azure Boards project

To start from Azure Boards

- Connect Azure Boards to GitHub (Cloud):
  - Add a connection from an Azure Boards project to select GitHub repositories
  - Approve and install the Azure Boards app for GitHub for your GitHub account or organization
- Connect Azure DevOps Server to GitHub Enterprise Server (On-premises):
  - Add a connection from an Azure Boards project to select GitHub Enterprise Server repositories
  - Approve and install the Azure Boards app for GitHub for your GitHub account or organization

Exercise Azure Boards-GitHub integration features

Once you've configured the connection, you can complete the following tasks.

- Link GitHub commits and pull requests to work items
- Configure status badges

Add or remove connections

- Add or remove GitHub repositories, remove connection
  Use the procedures provided here to add or remove GitHub connections to GitHub repositories from Azure Boards.
- Change repository access to Azure Boards
  Use the procedures provided here to change which GitHub repositories can be connected to Azure Boards for the selected GitHub account or organization, to suspend the integration, or to uninstall the app.

Restrictions

- Only connect a GitHub repository to one Azure DevOps organization and project.
  Connecting the same GitHub repo to projects defined in two or more Azure DevOps organizations can lead to unexpected **AB#** mention linking. For details, see Troubleshoot GitHub & Azure Boards integration.

Related articles

- What is Azure Boards?
- Link work items
- About work items
- Troubleshoot GitHub & Azure Boards integration
- Build GitHub repositories
- Build GitHub Enterprise Server repositories
- Trigger an Azure Pipelines run from GitHub Actions

Install the Azure Boards app for GitHub

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Installing the Azure Boards app for GitHub is the first step in connecting Azure Boards to your GitHub repositories. By connecting your Azure Boards projects with GitHub.com repositories, you support linking between GitHub commits and pull requests to work items. You can use GitHub for software development while using Azure Boards to plan and track your work.

For an overview of the integration that the Azure Boards app for GitHub supports, see Azure Boards-GitHub integration. Once you've installed the Azure Boards app for GitHub on your GitHub account or organization, choose which GitHub repositories you want to connect to from Azure Boards projects.

Prerequisites

- To install the Azure Boards app, you must be an administrator or owner of the GitHub organization.
- To connect to the Azure Boards project, you must have read permission to the GitHub repository. Also, you must be a member of the **Project Collection Administrators** group. If you created the project, then you have permissions.

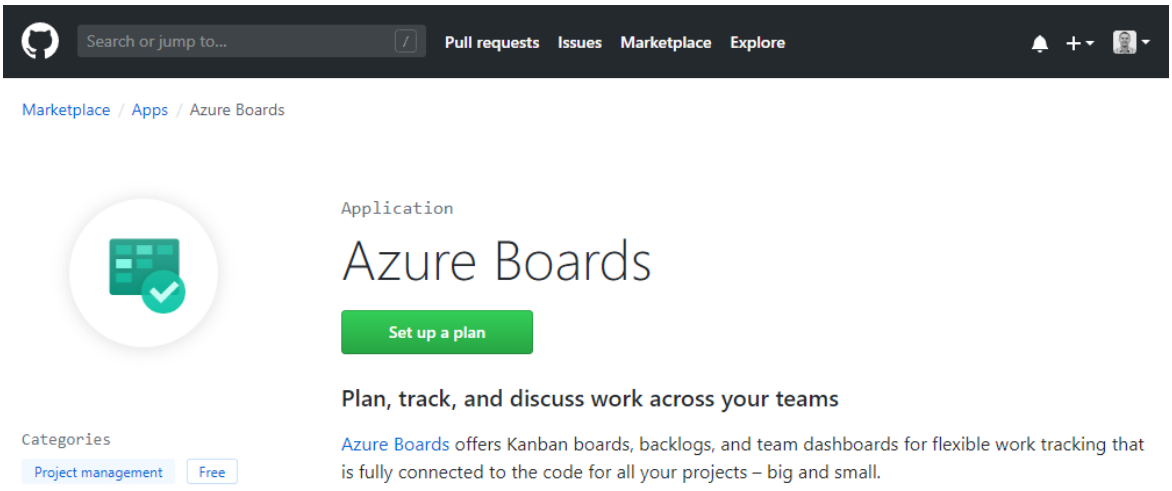
> **IMPORTANT**
>
> If your repository is already connected via another authentication type such as OAuth, you'll need to remove that repository from your existing connection before re-connecting it via the GitHub App. Follow the steps provided in Add or remove GitHub repositories before starting the GitHub App configuration.
>
> You can connect an Azure DevOps organization to multiple GitHub repositories so long as you are an administrator for those repositories. However, you shouldn't connect a GitHub repository to more than one Azure DevOps organization. To understand why, review Troubleshoot GitHub & Azure Boards connection, Unexpected results when linking to projects defined in two or more Azure DevOps organizations.

Install and configure the Azure Boards app

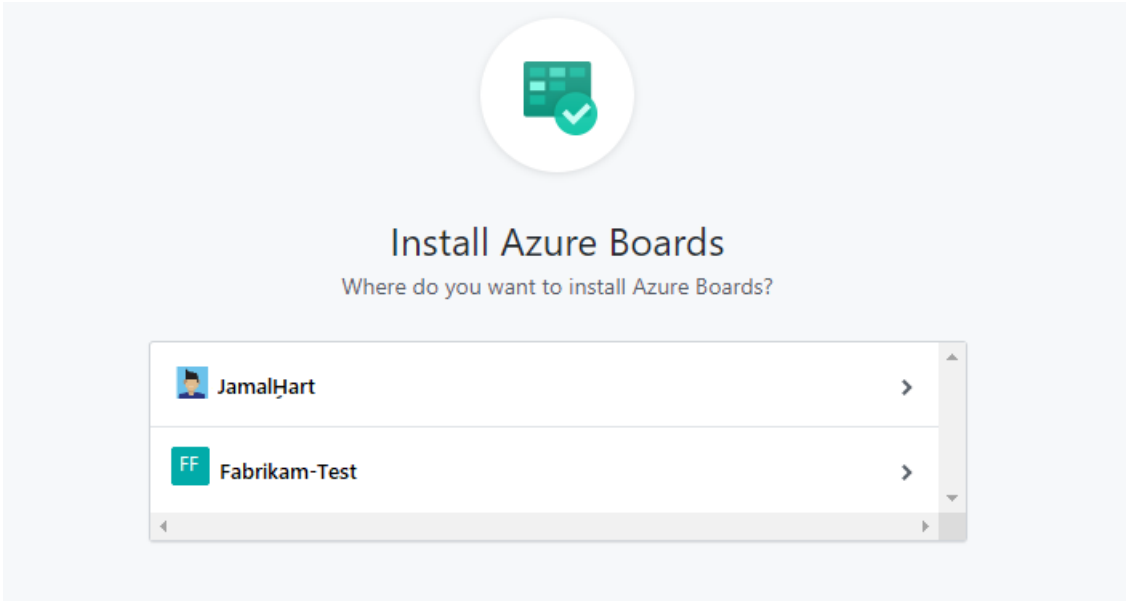
1. Go to the Azure Boards app in the GitHub Marketplace, `https://github.com/marketplace/azure-boards`.

   Azure Boards app

2. Choose **Set up a plan**.

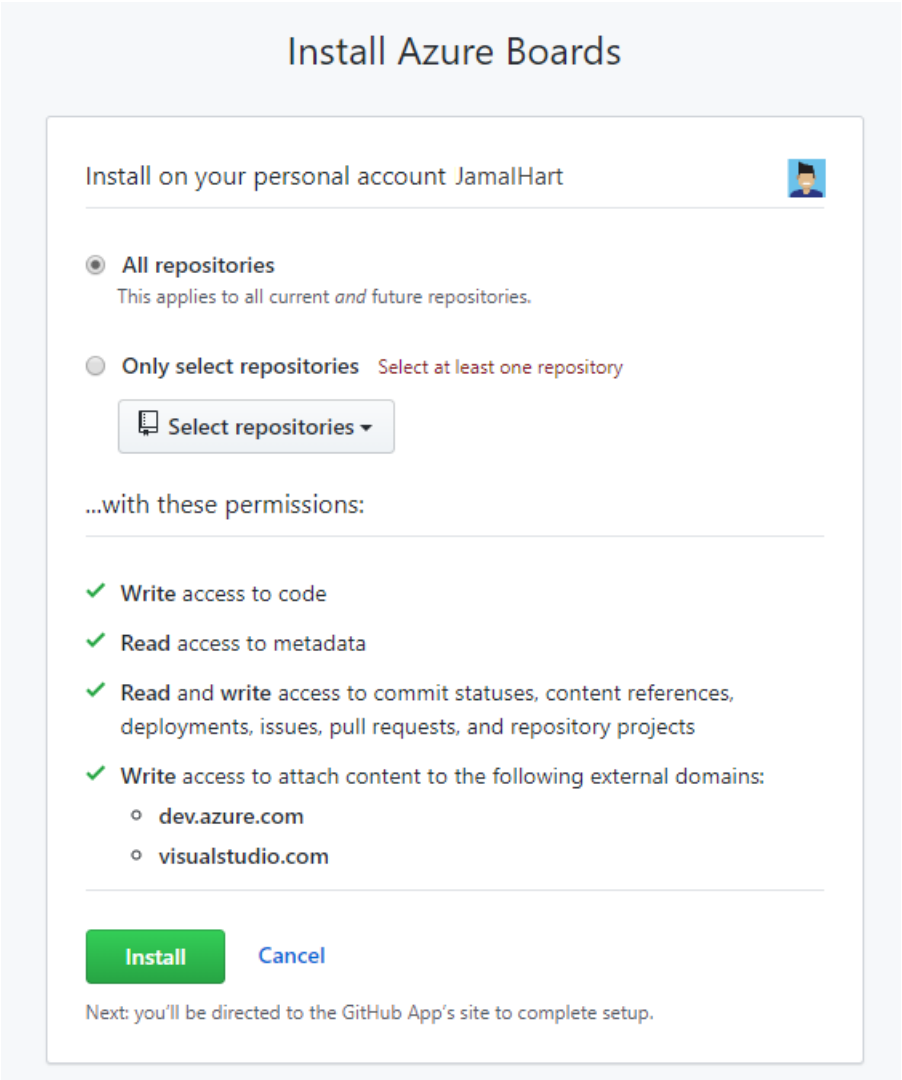


3. Choose the GitHub organization you want to connect to Azure Boards.

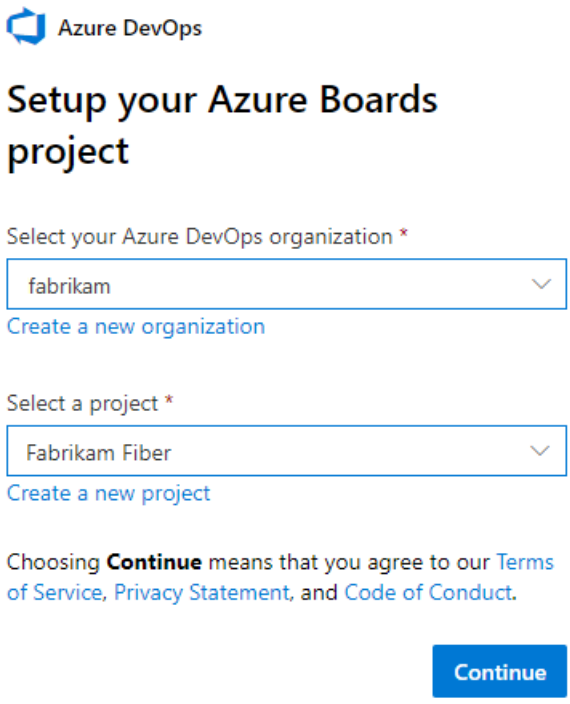


4. Choose the repositories you want to connect to Azure Boards.

   Here we choose to connect to all repositories.



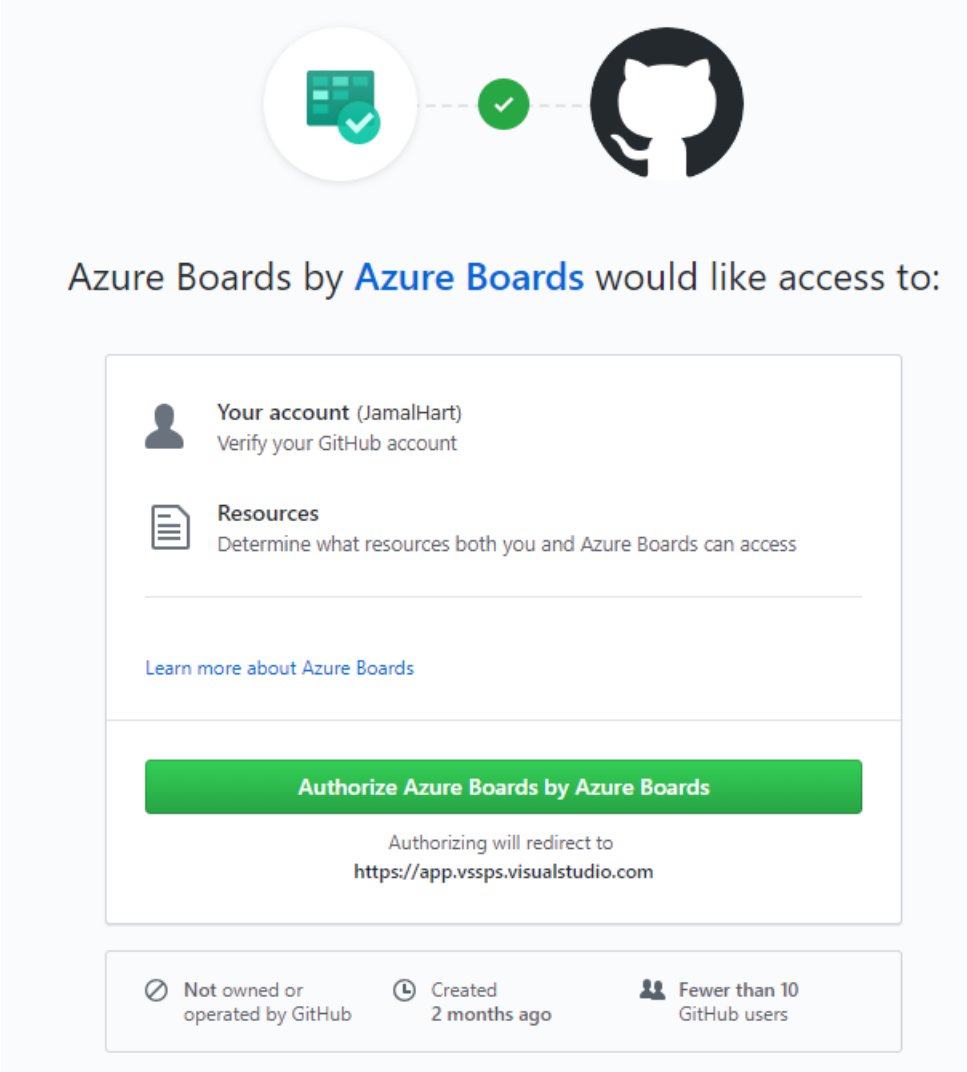
5. Choose the Azure DevOps organization and Azure Boards project you want to connect to GitHub.com.



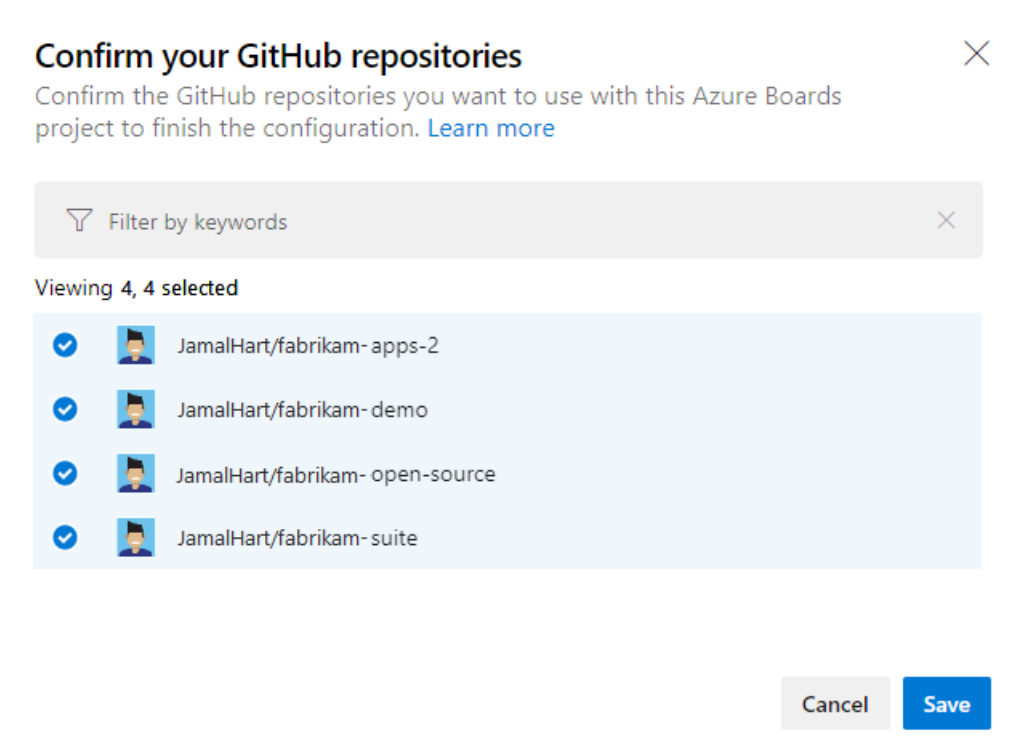
You can only connect one project at a time. If you have other projects you want to connect, you can do

that later as described in Configure other projects or repositories later in this article.

6. Authorize your Azure Boards organization to connect with GitHub.com.



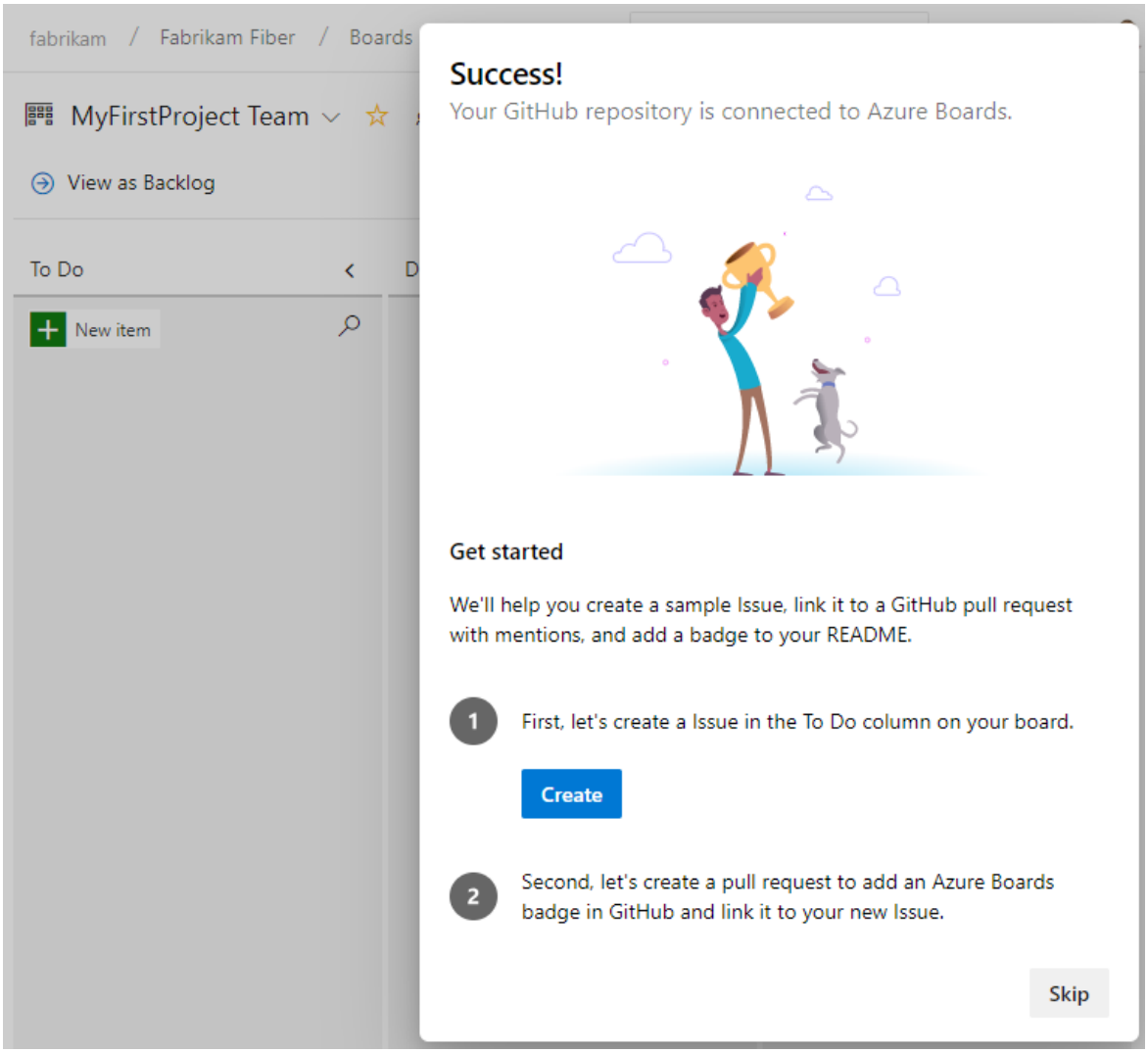
7. Confirm the GitHub.com repositories that you want to connect. Select each repository you want to connect to. Unselect any repositories that you don't want to participate in the integration.



Get started with the connection

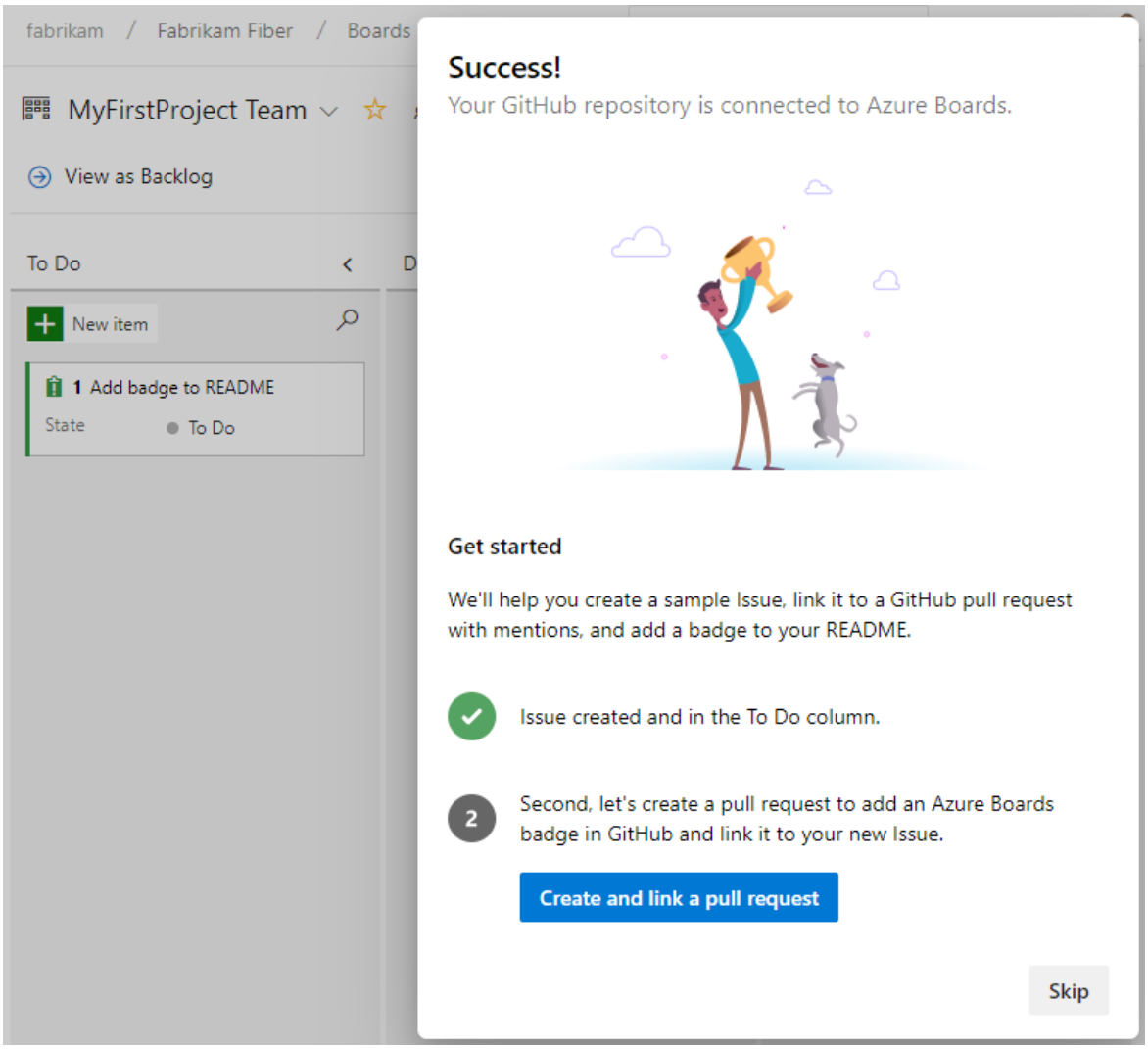
At this point, your Azure Boards-GitHub integration is complete. You can skip the next steps or run through them to understand the features supported with the connection.

1. Choose **Create** to add a work item—Issue (Basic), User Story (Agile), or Product Backlog Item (Scrum)—depending on the process model used by your Azure Boards project.



   A work item titled *Add badge to README* appears on your Azure Boards.

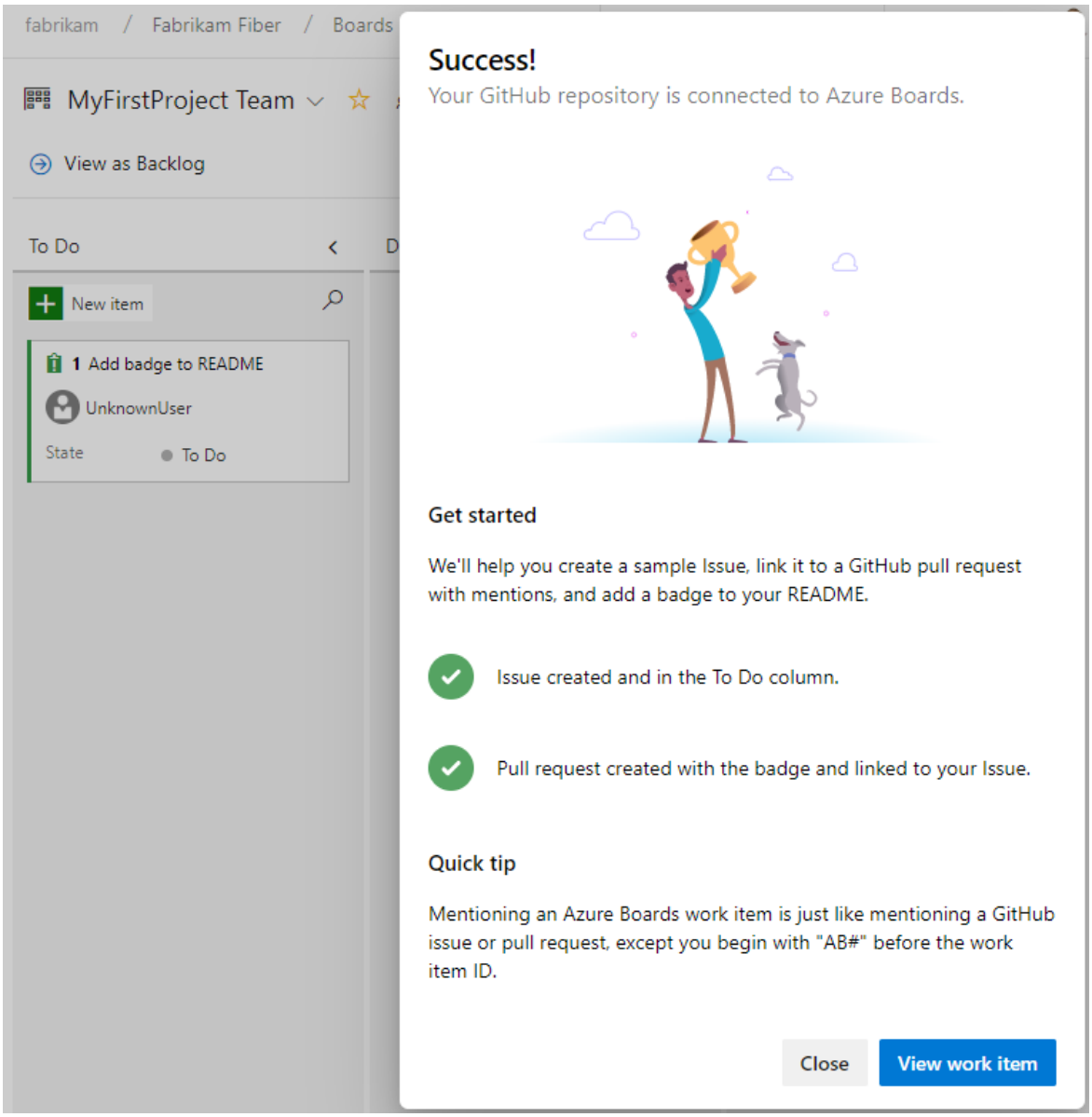
2. Next, choose **Create and link a pull request**.



This step runs the following actions in the background:

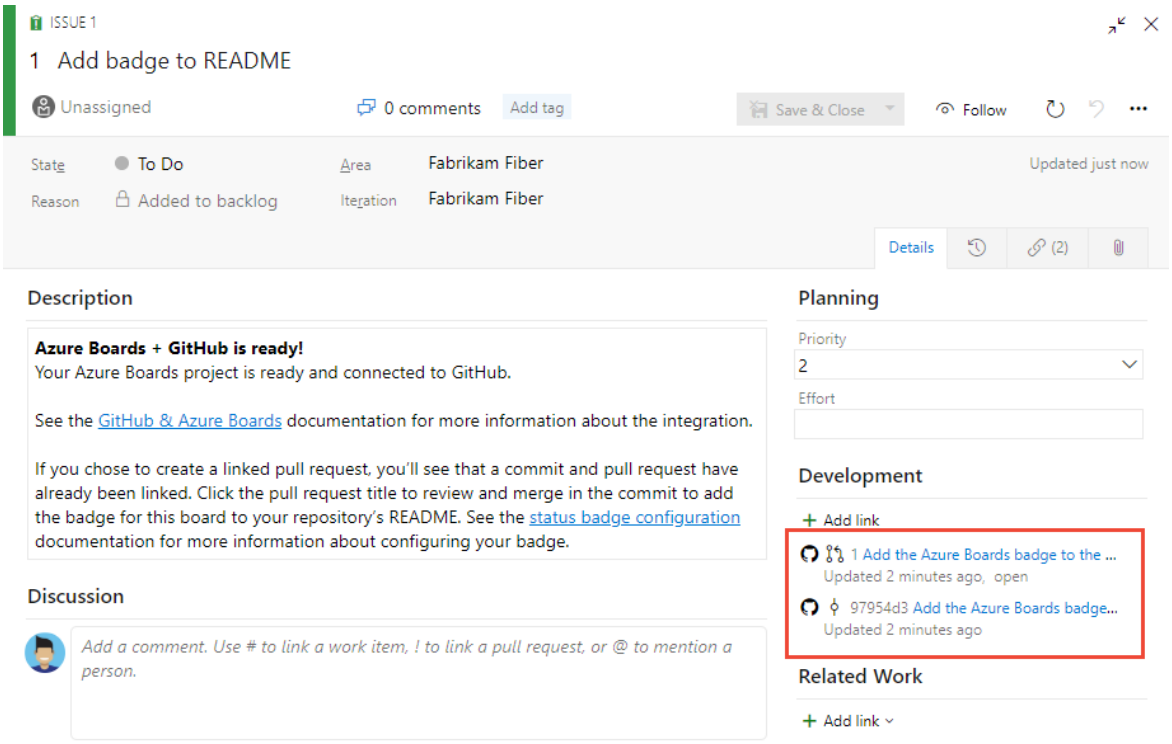
- Adds a badge to the README file of the first repository in the list of connected GitHub repositories
- Creates a GitHub commit for the update made by adding the badge to the README file
- Creates a GitHub pull request to merge the changes made to the README file
- Links the GitHub commit and pull request to the work item created in step 1.

3. Lastly, choose **View work item** to open the work item created in step 1. Note the links under the **Development** section that correspond to the commit and pull request created in GitHub.com

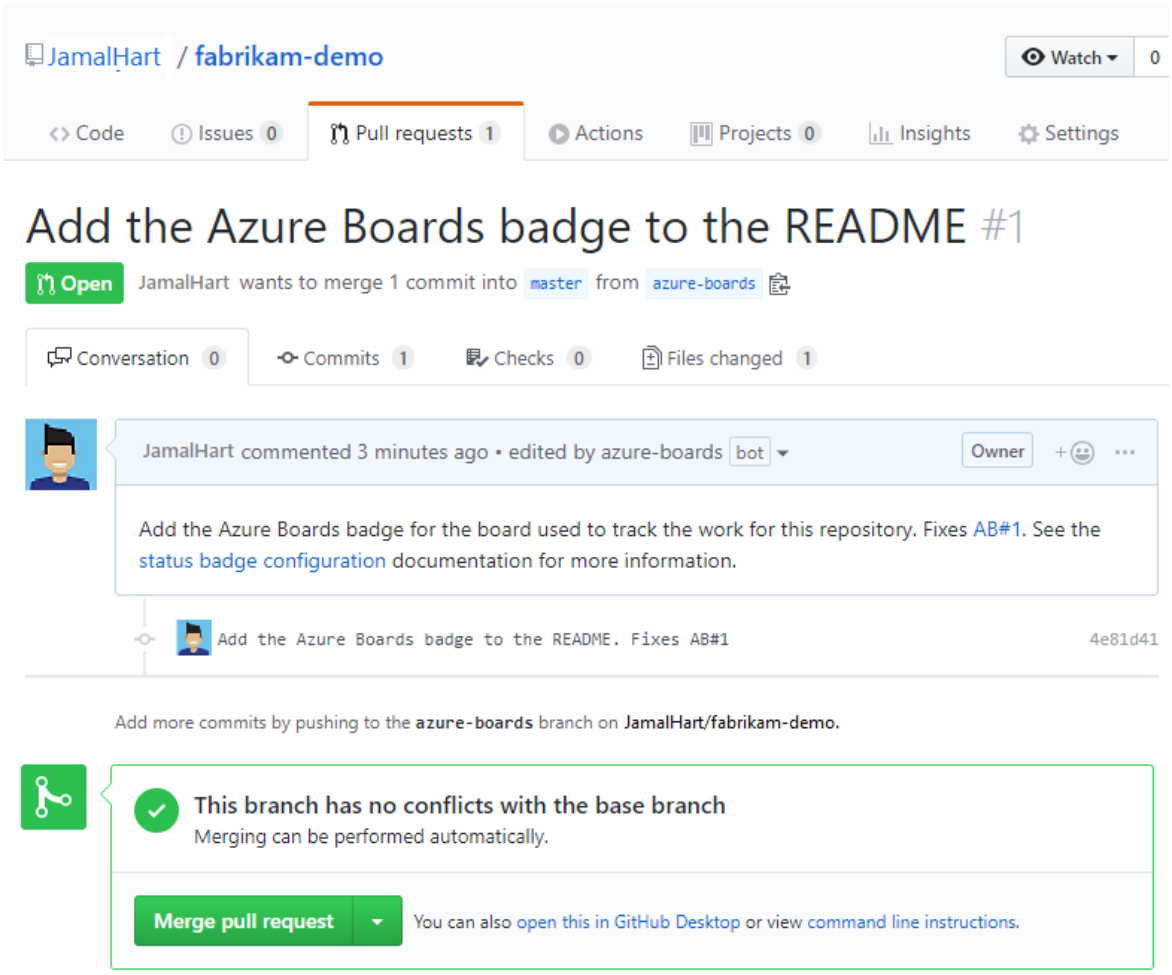


4. Choose the pull request link, the first link in the list, to open the pull request in GitHub.

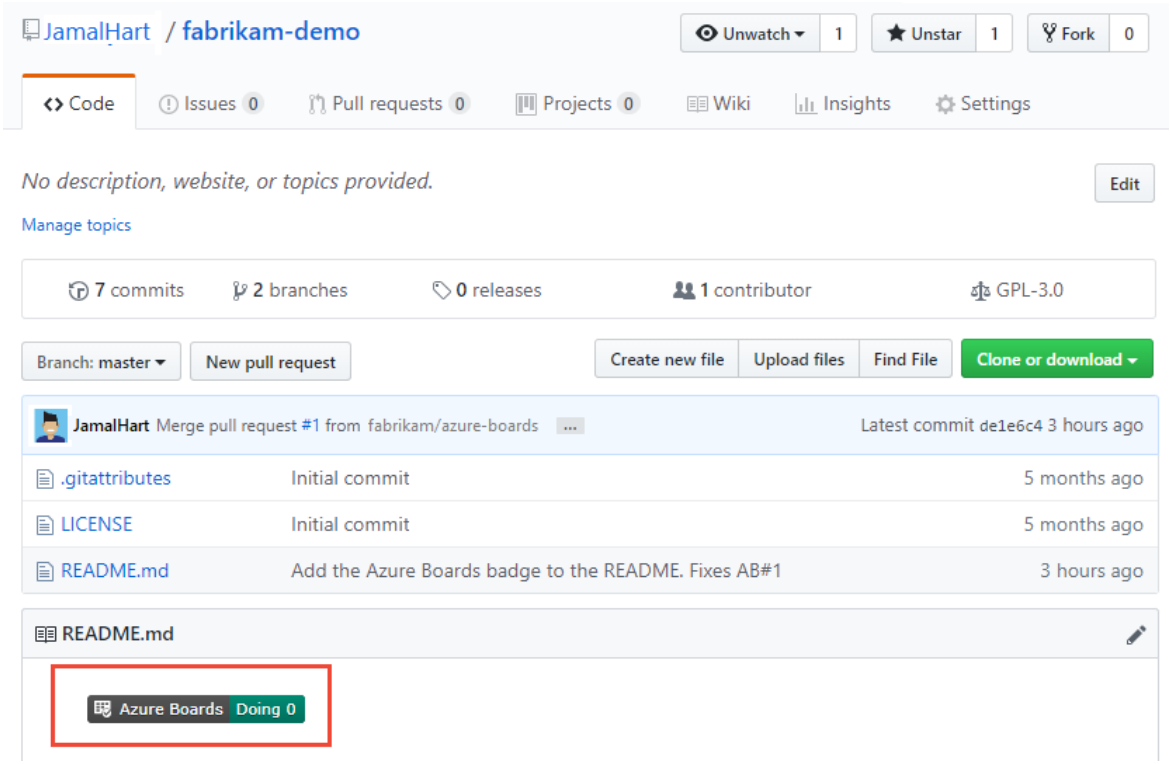
   The GitHub pull request opens in a new browser tab.



5. Go ahead and complete the pull request.



6. Navigate to your repository README file and view the badge that has been added.



To learn more about Azure Boards badges, see Configure status badges to add to GitHub README files.

Configure other projects or repositories

You can configure other Azure Boards/Azure DevOps projects, GitHub.com repositories, or change the current configuration from the Azure Boards app page. To learn more, see Change GitHub repository access, or suspend or uninstall the integration.

Add or remove repositories from Azure Boards

Once you've integrated Azure Boards with GitHub using the Azure Boards app, you can add or remove repositories from the web portal for Azure Boards. To learn how, see Add or remove GitHub repositories.

Try this next

Link GitHub commits and pull requests to work items

Related articles

- Drive Git development from a work item
- Change GitHub repository access, or suspend or uninstall the integration
- Add or remove GitHub repositories
- Configure status badges to add to GitHub README files

Connect Azure Boards to GitHub (Cloud)

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services

To support linking of GitHub commits and pull requests to Azure Boards work items, you connect your Azure Boards project to GitHub.com repositories. Then, use GitHub for software development while using Azure Boards to plan and track your work. For an overview of the Azure Boards app for GitHub, see Azure Boards-GitHub integration.

When you make the connection from Azure Boards, the list of GitHub repositories corresponds to ones that you allow Azure Boards to access. You can limit which repositories Azure Boards accesses overall, and limit what a particular project accesses or split the management of work across different Azure Boards projects.

> **NOTE**
> Azure Boards and Azure DevOps Services support integration with GitHub.com and GitHub Enterprise Server repositories. If you want to connect from an on-premises Azure DevOps Server, see Connect Azure DevOps Server to GitHub Enterprise Server.

Prerequisites

- Connect to an Azure Boards or Azure DevOps project. If you don't have a project yet, create one.
- You must be a member of the **Project Administrators** group. If you created the project, you have permissions.
- You must be an administrator or owner of the GitHub repository you'll connect to. You can connect to multiple GitHub repositories so long as you're an administrator for those repositories.

Authentication options

The following authentication options are supported based on the GitHub platform you want to connect to.

GitHub.com

GitHub Enterprise Server

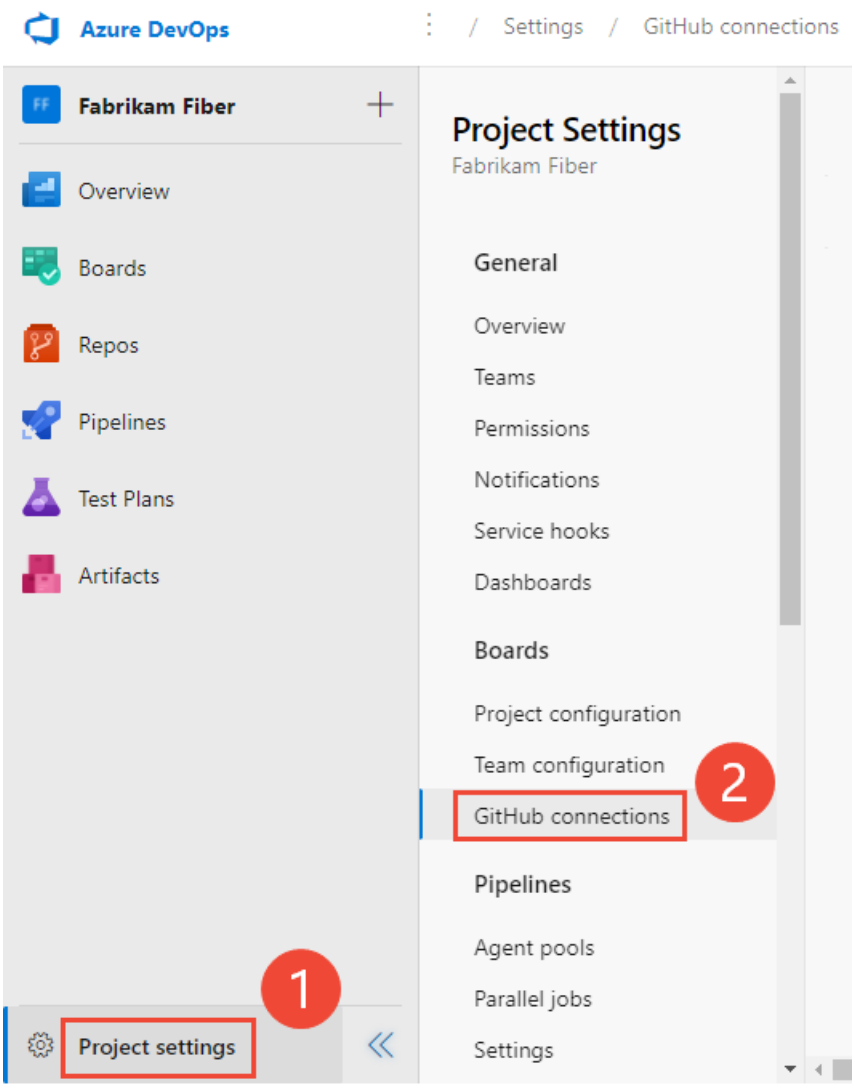
- GitHub.com user account (Recommended)
- Personal access token (PAT)

- OAuth (preferred, registration required)
- PAT
- Username plus password

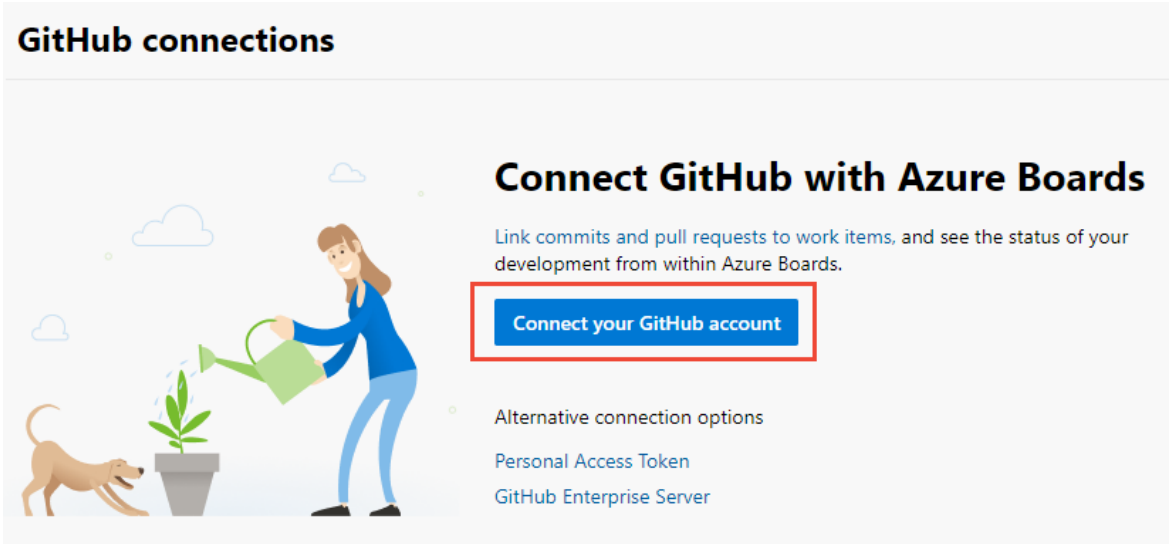
> **NOTE**
> If you choose to connect Github with PAT, make sure you configure single sign-on (SSO) for the PAT on your GitHub account. This is needed to be able to get a list of repositories of an organization with Security Assertion Markup Language (SAML) SSO authentication configured.

Open Project Settings>GitHub Connections.

1. Sign into Azure Boards for the project you want to connect to GitHub repositories.
2. Choose (1) **Project Settings** > (2) **GitHub connections**.



3. If it's the first time making a connection from the project, choose **Connect your GitHub account** to use your GitHub account credentials.



Otherwise, choose + **New connection**, and select your authentication method from the **New Connection** dialog.

When you connect using your GitHub account, use your GitHub account credentials to authenticate. If

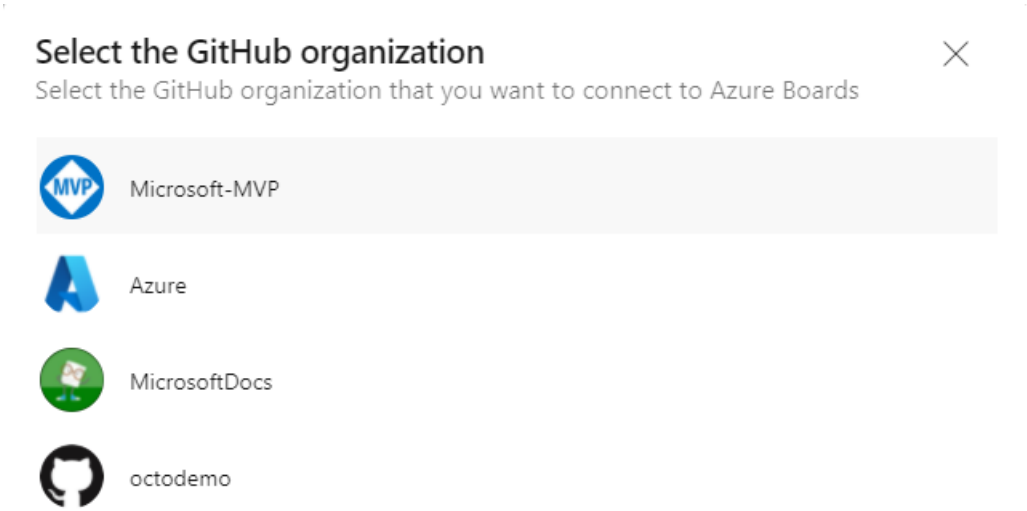
connecting using PAT, see Add a GitHub connection using PAT. If connecting to a GitHub Enterprise Server, see Register Azure DevOps in GitHub as an OAuth App.

Add a GitHub connection with GitHub credentials

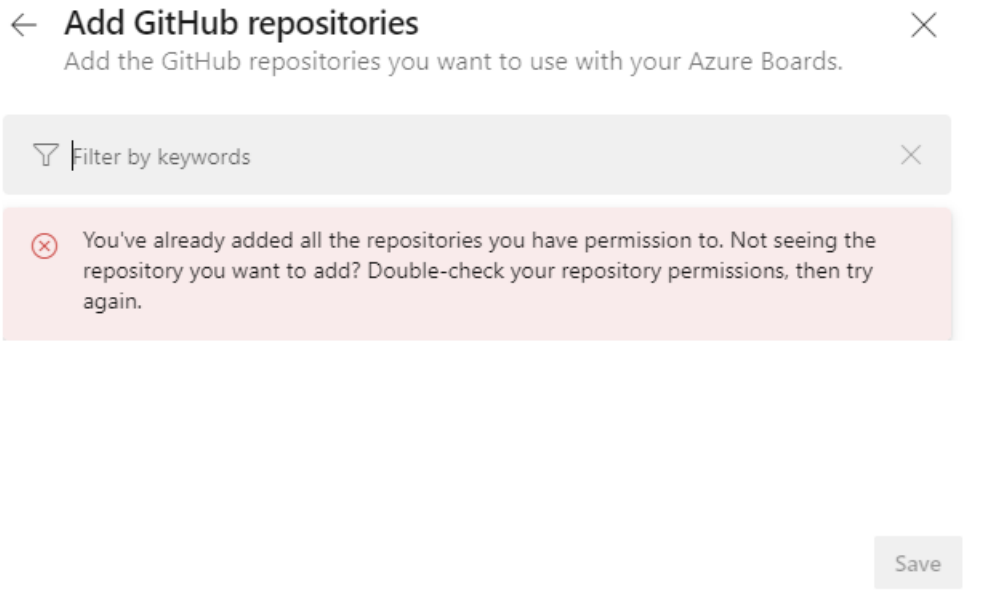
You can connect up to 250 GitHub repositories to an Azure Boards project.

1. If it's your first time connecting to GitHub from Azure Boards, you'll be asked to sign in using your GitHub credentials. Choose an account for which you're an administrator for the repositories you want to connect to.

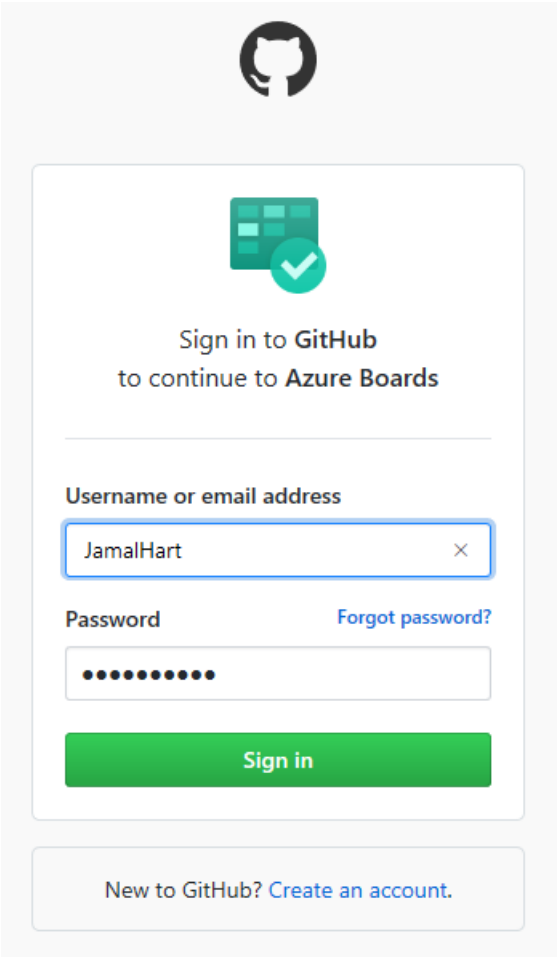
2. If you belong to more than one GitHub account or organization, choose the one whose repositories you want to connect. Only those organizations that you own or are an administrator for are listed.



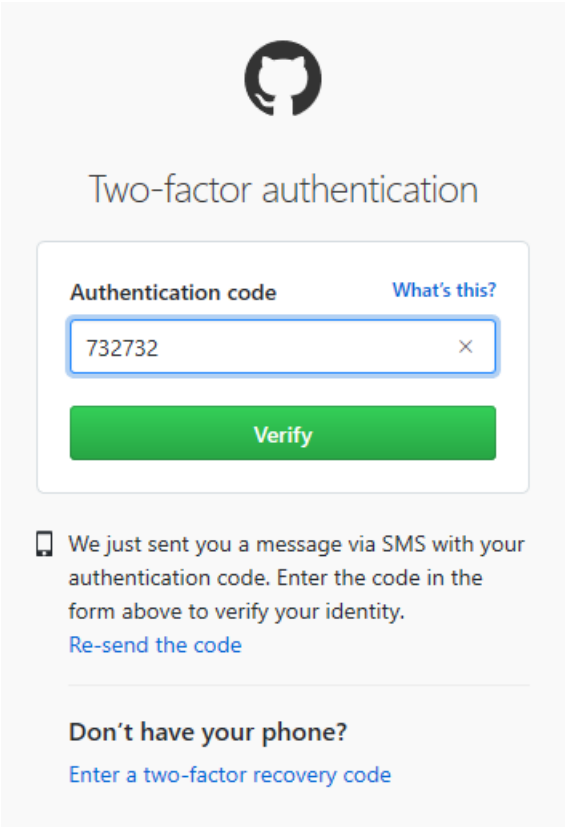
If all repositories for an organization have already been connected to Azure Boards, you'll see the following message.



3. When asked, enter your GitHub credentials.



If you have enabled two-factor authentication, enter the authentication code that GitHub sent you and choose **Verify**.

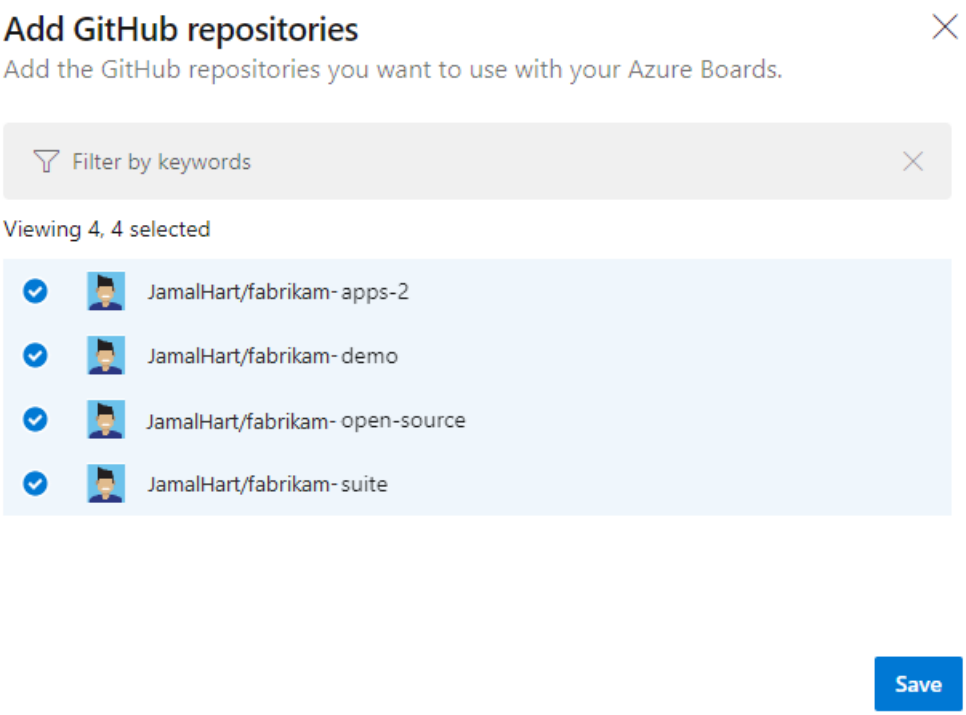


Otherwise, the system will automatically recognize your GitHub organization as your GitHub account has previously been associated with your Azure DevOps Services account.

Choose the repositories

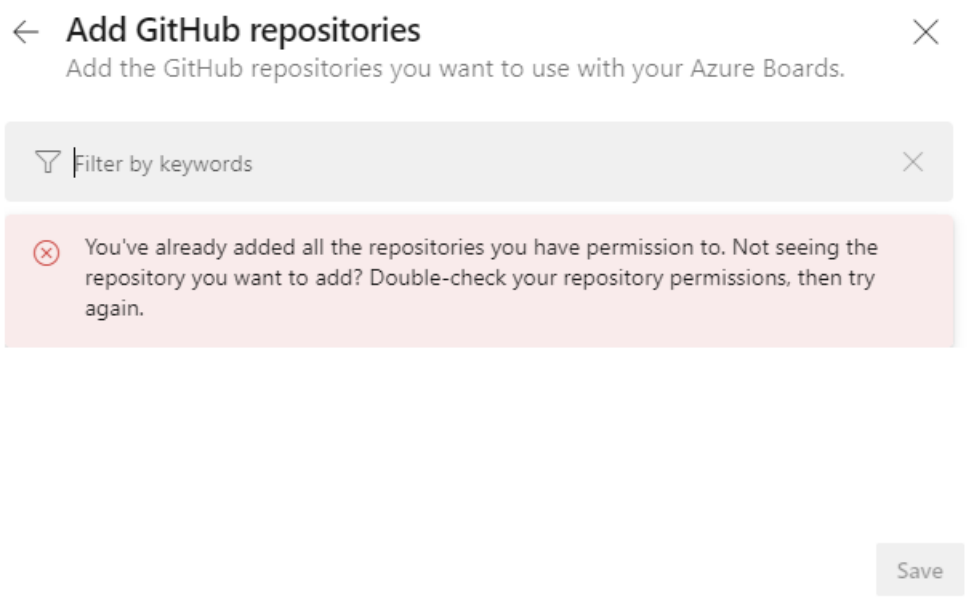
Once you've been authenticated, you can select the repositories you want to connect.

1. The **Add GitHub Repositories** dialog automatically displays and selects all GitHub.com repositories for which you're an administrator for the organization you selected. Unselect any repositories that you don't want to participate in the integration.



> **TIP**
> We recommend that you only connect a GitHub repo to projects defined in a single Azure DevOps organization. Connecting the same GitHub repo to projects defined in two or more Azure DevOps organizations can lead to unexpected **AB#** mention linking. For details, see Troubleshoot GitHub & Azure Boards integration.

If all repositories have been connected already to the current or other organization, then the following message displays.

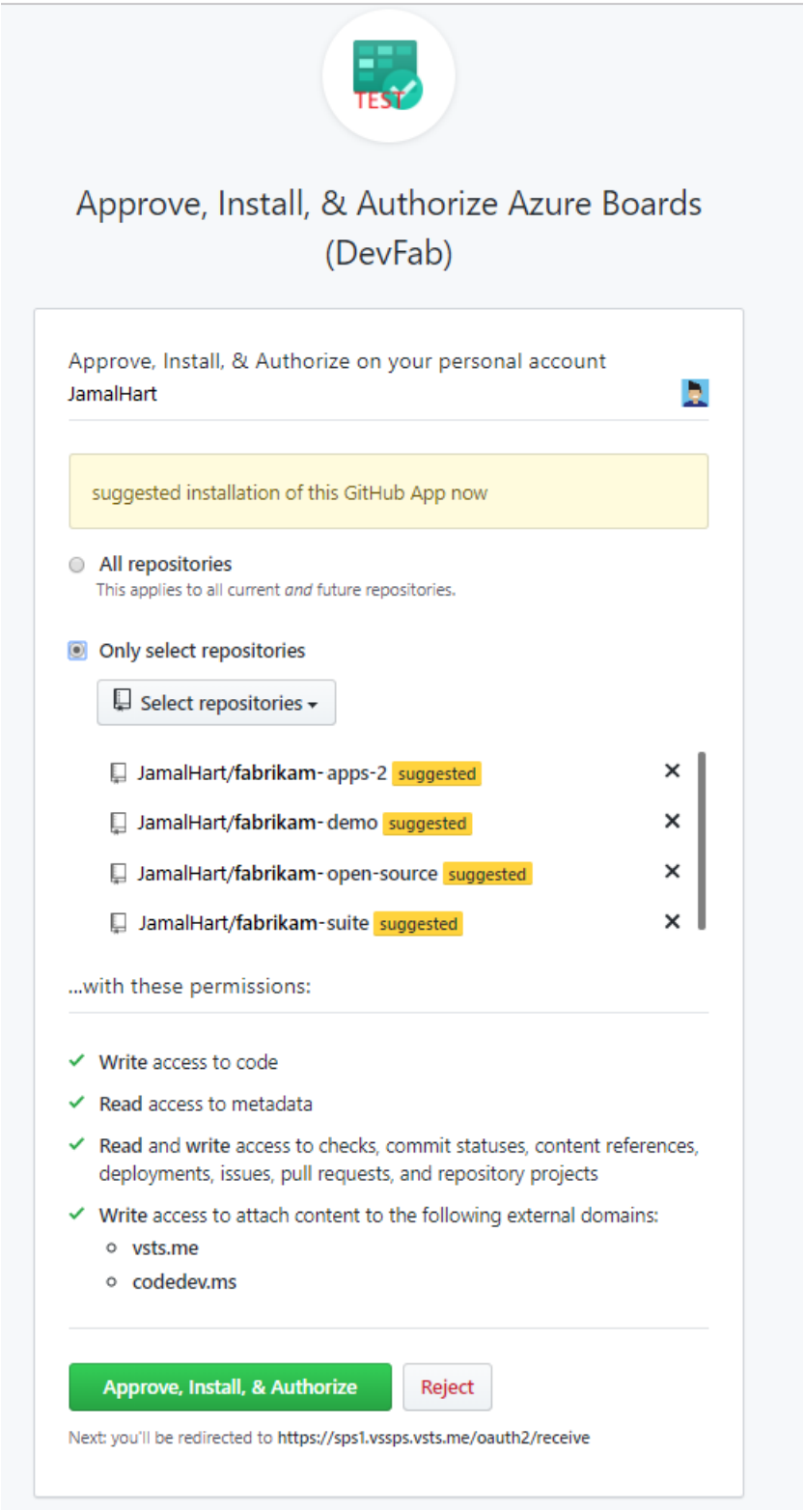


2. When done, choose **Save**.

Confirm the connection

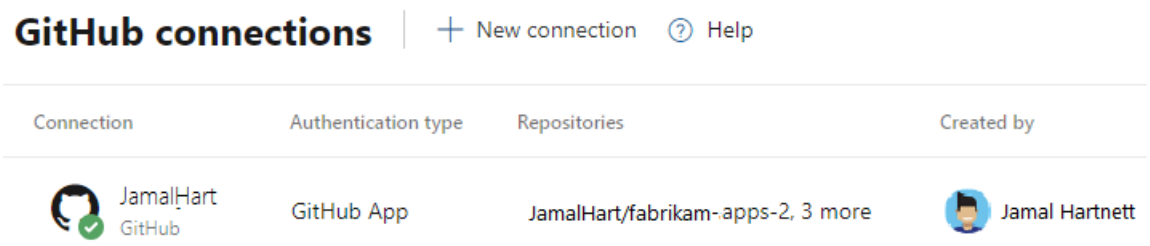
In this step, you install the Azure Boards app for GitHub on the GitHub account or organization associated with the repositories you chose to connect to.

1. Review the GitHub page that displays and then choose **Approve, Install, & Authorize**.



2. Provide your GitHub password to confirm.

3. When done, you should see the new connection with the selected repositories listed.



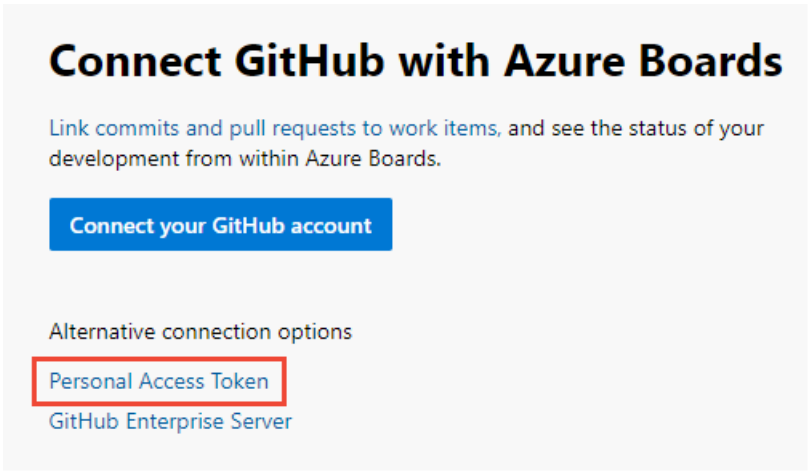
To change the configuration or manage the Azure Boards app for GitHub, see Change repository access to Azure Boards.

Add a GitHub connection using PAT

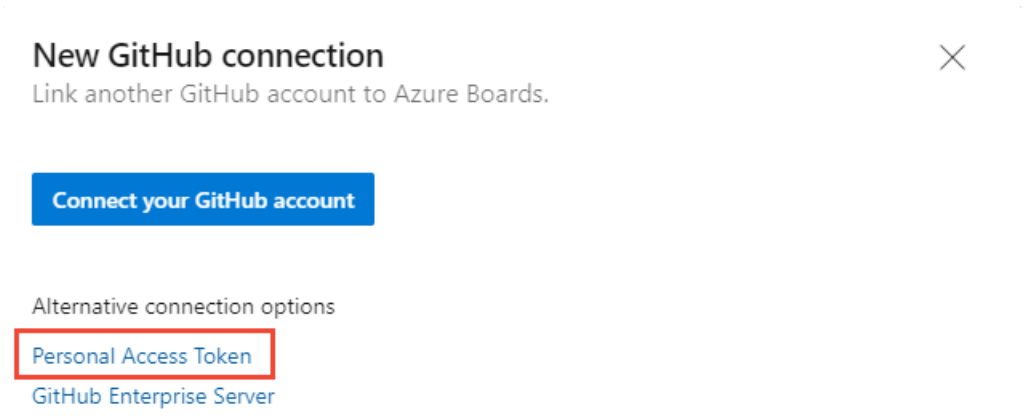
We recommend that you use your GitHub account credentials to connect to your GitHub repository. However, if you need to use a PAT, do so by following these procedures.

> **TIP**
> When creating your GitHub PAT, make sure that you include these scopes:
> `repo, read:user, user:email, admin:repo_hook` .

1. To choose a PAT when connecting a GitHub repository, choose **Personal Access Token** when making a first-time connection.

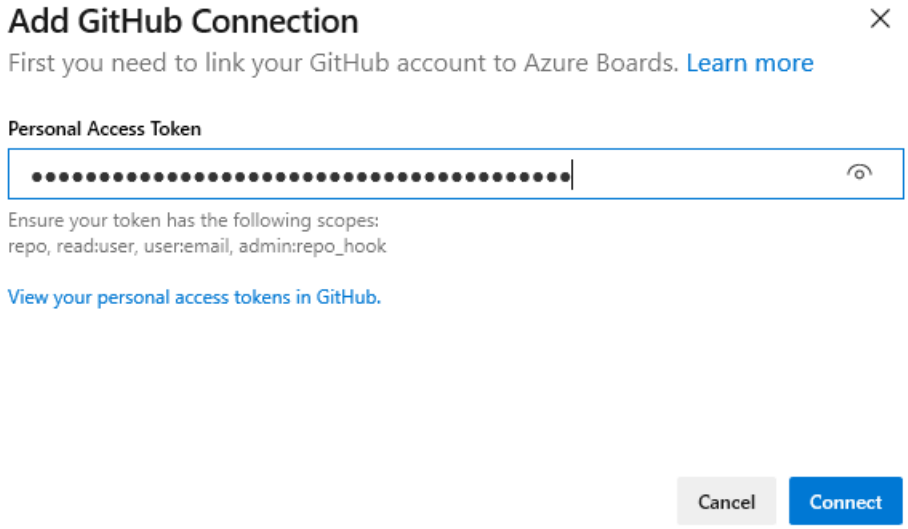


Or, from the New GitHub connection dialog.



To create a GitHub PAT, go to GitHub Developer Settings > Personal access tokens.

2. Enter the PAT and choose **Connect**.



3. Choose the repositories you want connected to the project by following the procedures outlined in Choose the repositories earlier in this article.

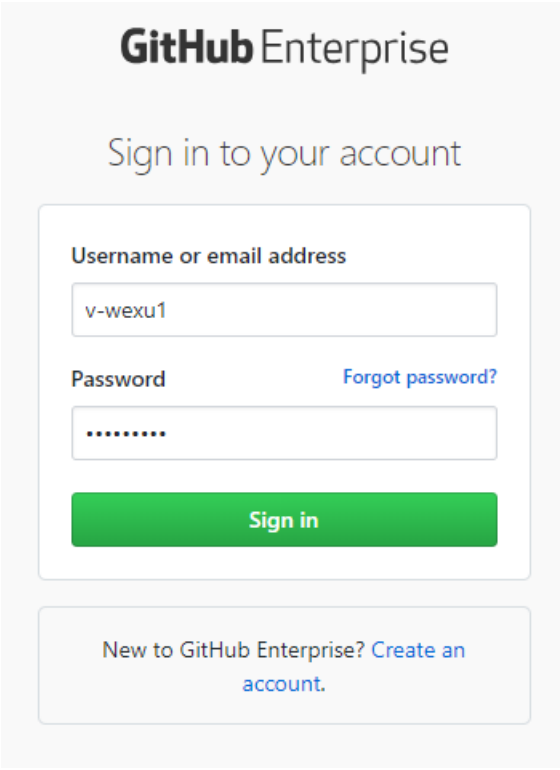
4. If it's the first time connecting to a GitHub account or organization from Azure Boards, you'll also be installing the Azure Boards app for GitHub. Complete the integration by following the procedures outlined in Confirm the connection earlier in this article.

Register Azure DevOps in GitHub as an OAuth App

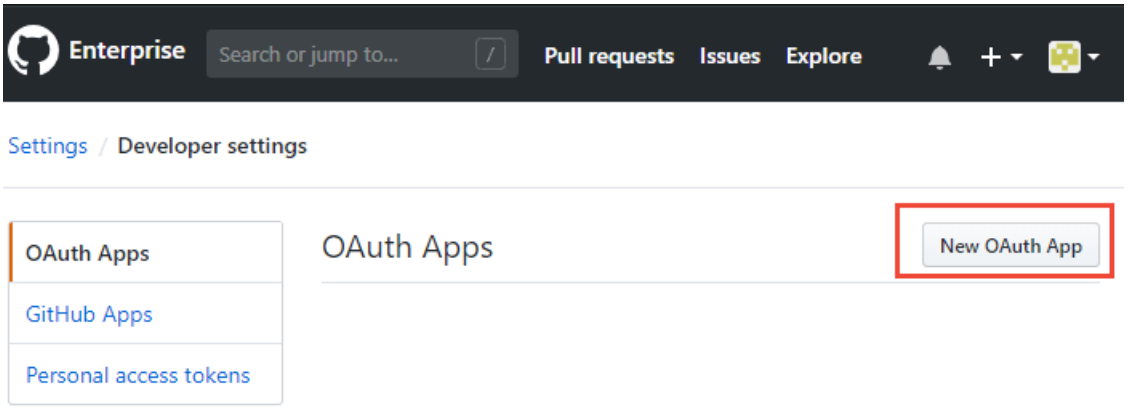
If you plan to use OAuth to connect Azure DevOps Services or Azure DevOps Server with your GitHub Enterprise Server, you first need to register the application as an OAuth App. For details, see Creating an OAuth App.

Register Azure DevOps Services

1. Sign into the web portal for your GitHub Enterprise server.



2. Open **Settings** > **Developer settings** > **Oauth Apps** and choose **New OAuth App**.



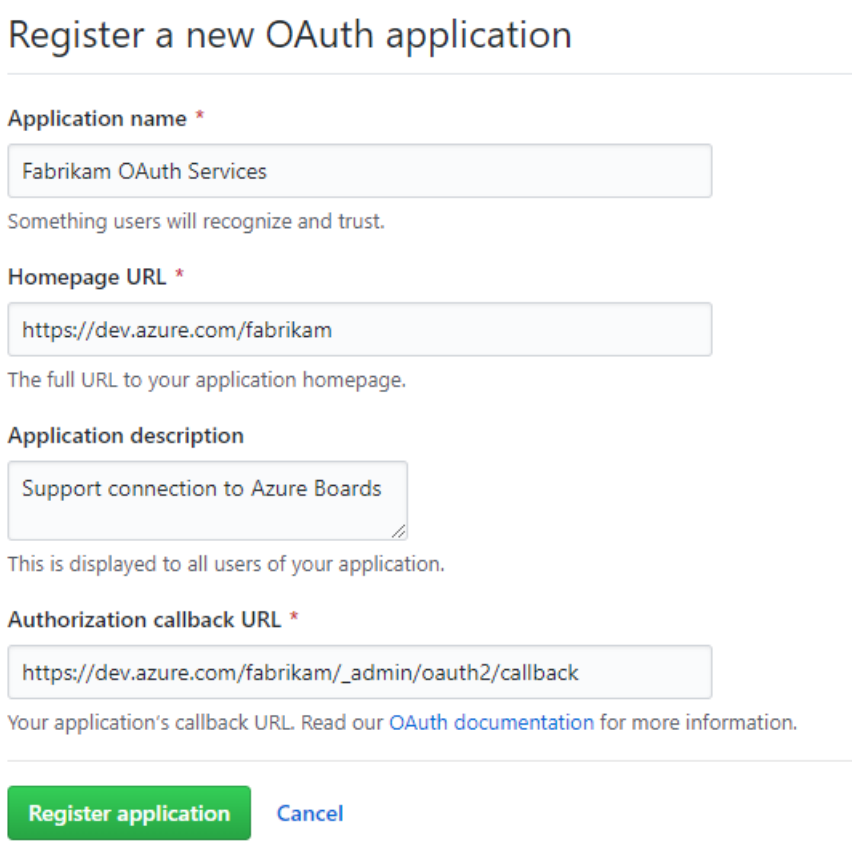
3. Fill out the form to register your Azure DevOps Server application.

   For the **Homepage URL**, specify the **Organization URL** of your organization.
   For the **Authorization callback URL**, use the following pattern to construct the URL.

   ```
   {Azure DevOps Services Organization URL}/_admin/oauth2/callback
   ```

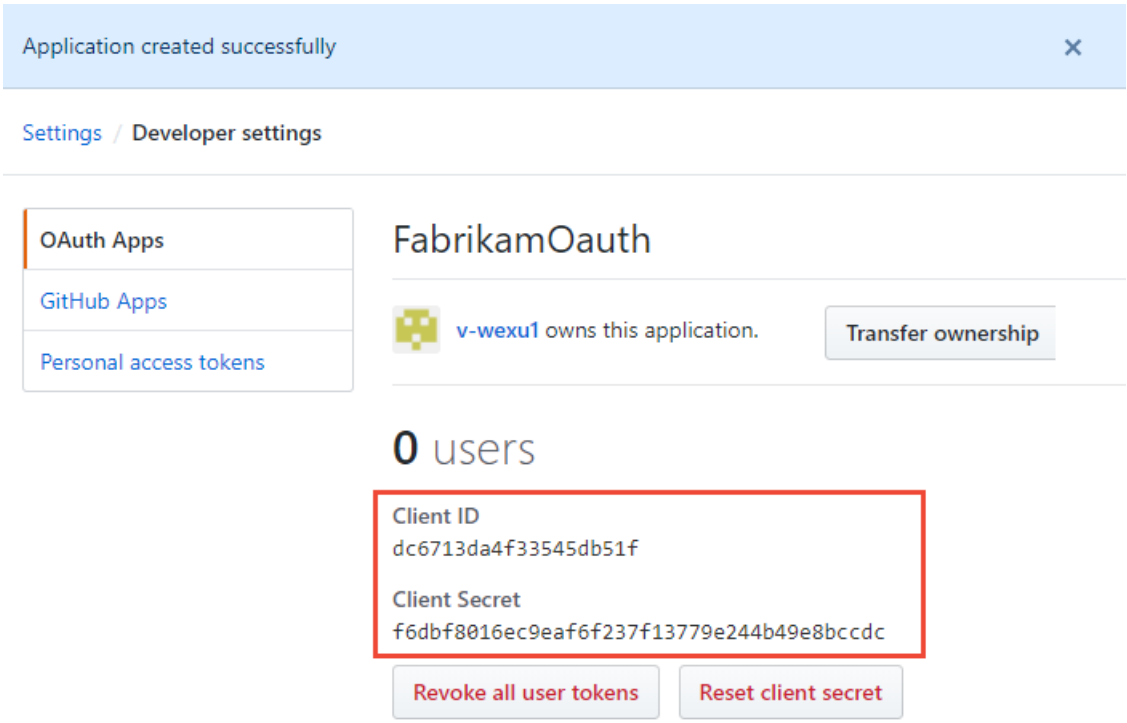
   For example:

   ```
   https://dev.azure.com/fabrikam/_admin/oauth2/callback
   ```



4. Choose **Register application**.

5. Upon success, you'll see a page that provides the **Client ID** and **Client Secret** for your registered OAuth application.

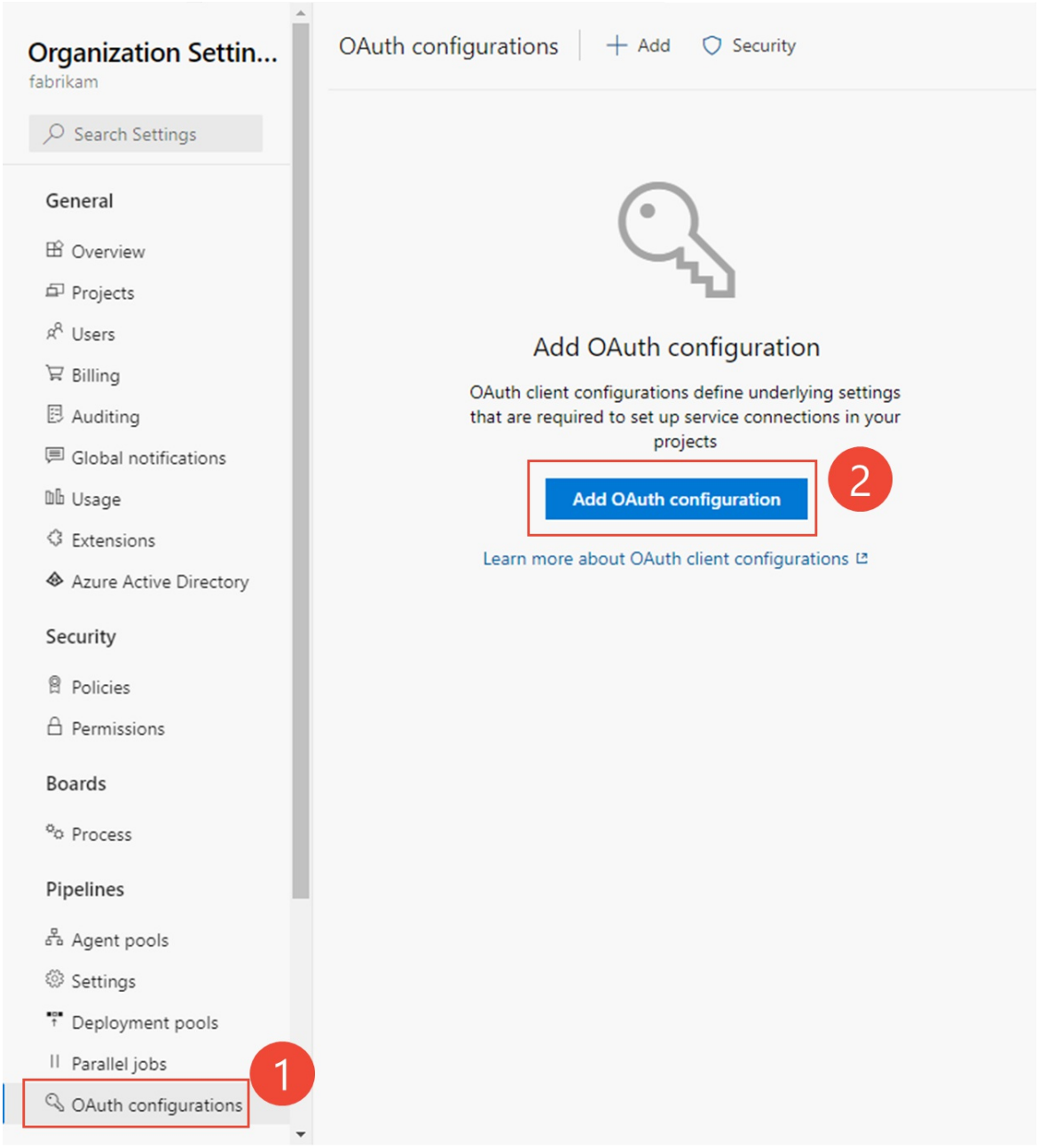


Register your OAuth configuration in Azure DevOps Services

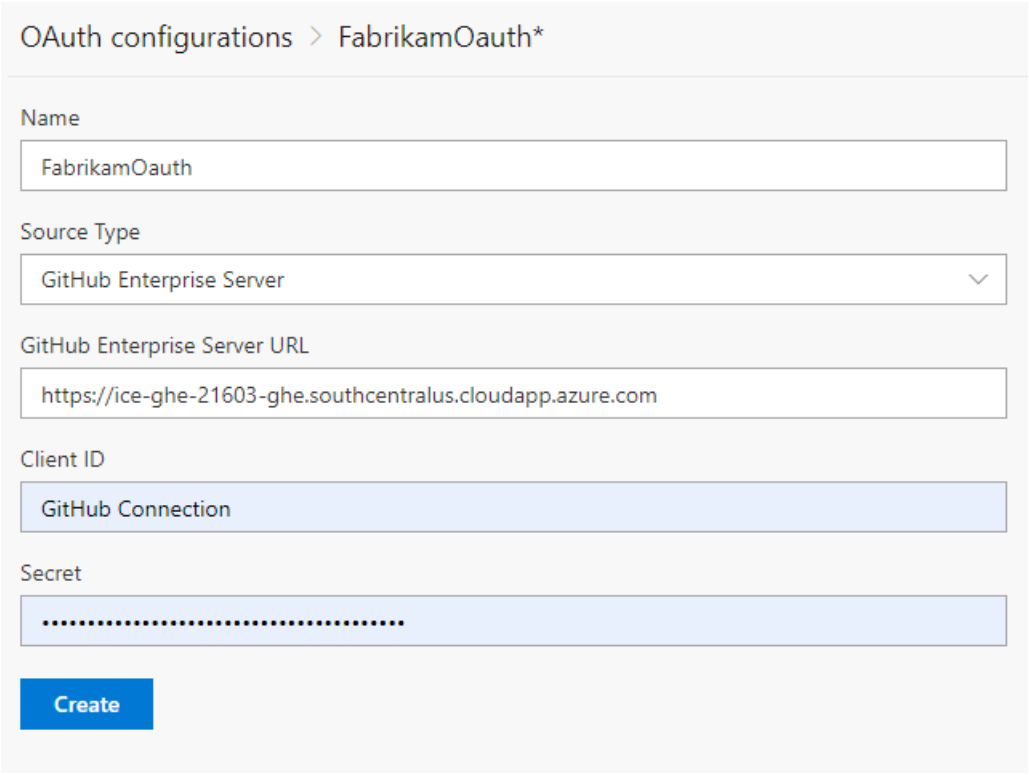
1. Sign into the web portal for Azure DevOps Services.

2. Add the GitHub Enterprise Oauth configuration to your organization.

3. Open **Organization settings > Oauth configurations**, and then choose **Add Oauth configuration**.



4. Fill in the form that appears, and then choose **Create**.



Connect Azure DevOps Services to GitHub Enterprise Server

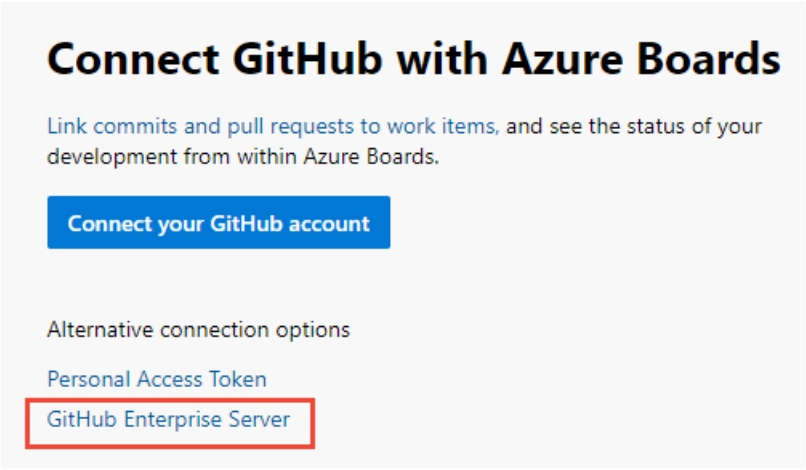
IMPORTANT

To connect Azure DevOps Services to your GitHub Enterprise Server, your GitHub Enterprise Server must be sufficiently accessible from the Internet. Make sure Azure DNS can resolve your GitHub Enterprise Server name and your firewall allows access from Azure Data Center IP addresses. To determine the IP address range, see Microsoft Azure Datacenter IP Ranges. A common error message encountered when connectivity issues exist is:

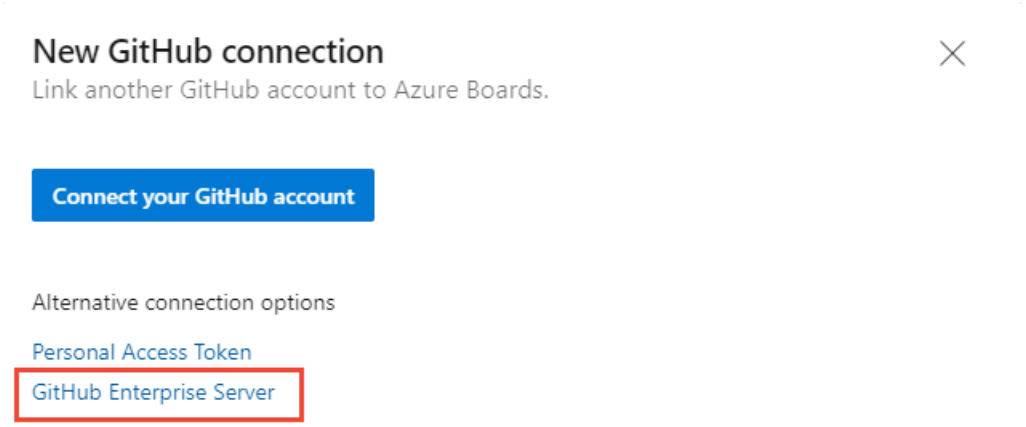
The remote name could not be resolved: 'github-enterprise-server.contoso.com'

If you encounter this error, check that your server is accessible. For more information, see Azure DNS FAQ.

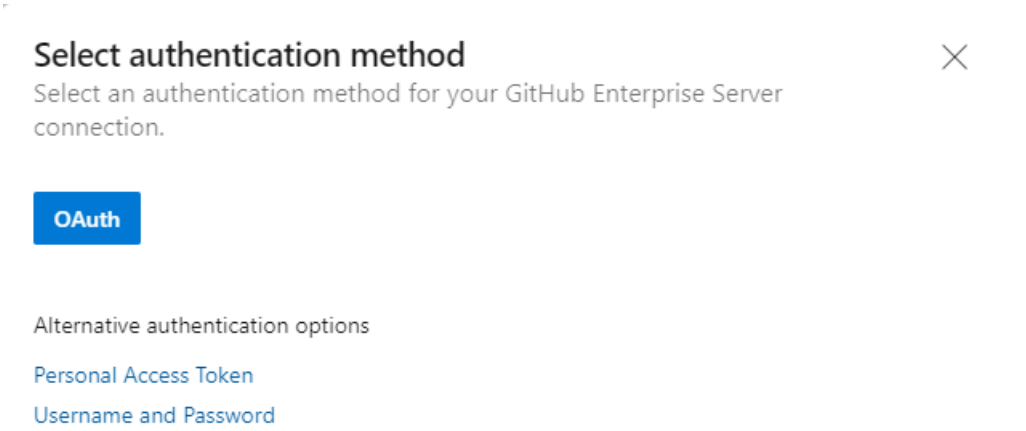
1. From the **Project Settings>GitHub connections** page choose GitHub Enterprise Server, choose **GitHub Enterprise Server** when making a first-ptime connection.



Or, from the **New GitHub connection** dialog, choose **GitHub Enterprise Server**.

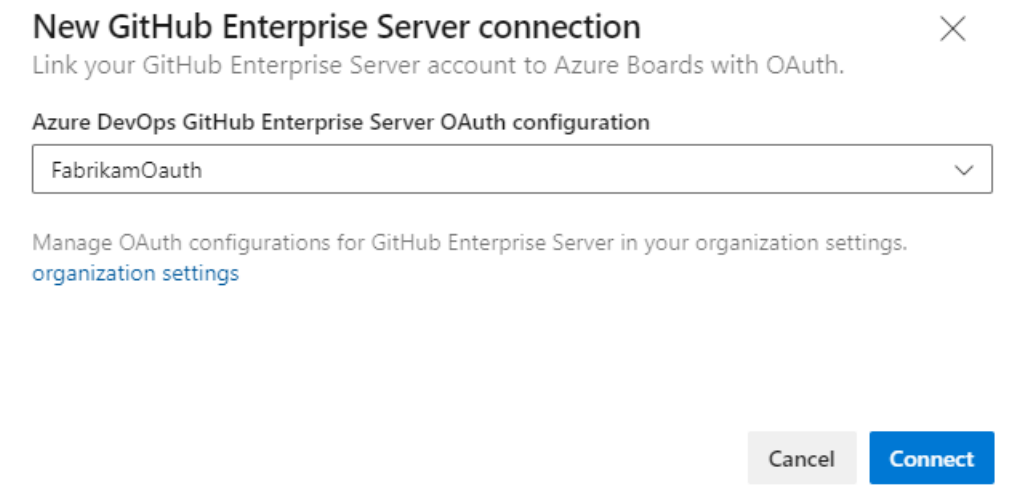


2. Select the authentication method.



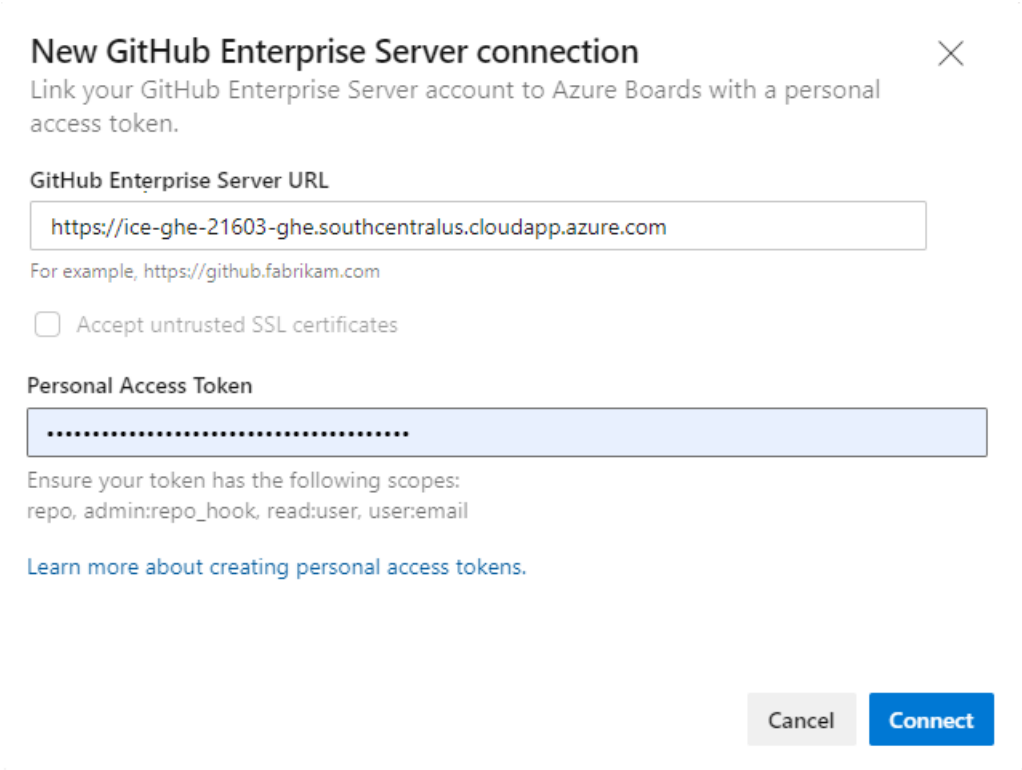
Connect using OAuth

Choose the configuration that you set up in Step 4 of Register your OAuth configuration in Azure DevOps Services, and then choose **Connect**.



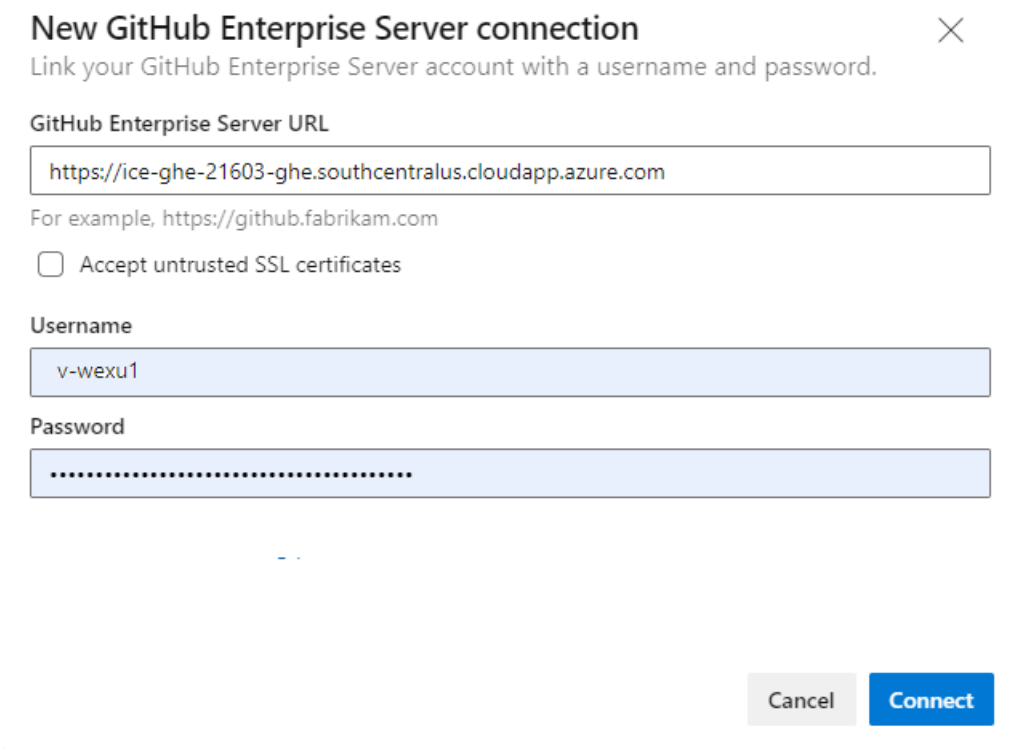
Connect using a Personal Access Token

Enter the URL for your GitHub Enterprise server and the Personal access token credentials recognized by that server. And then choose **Connect**.



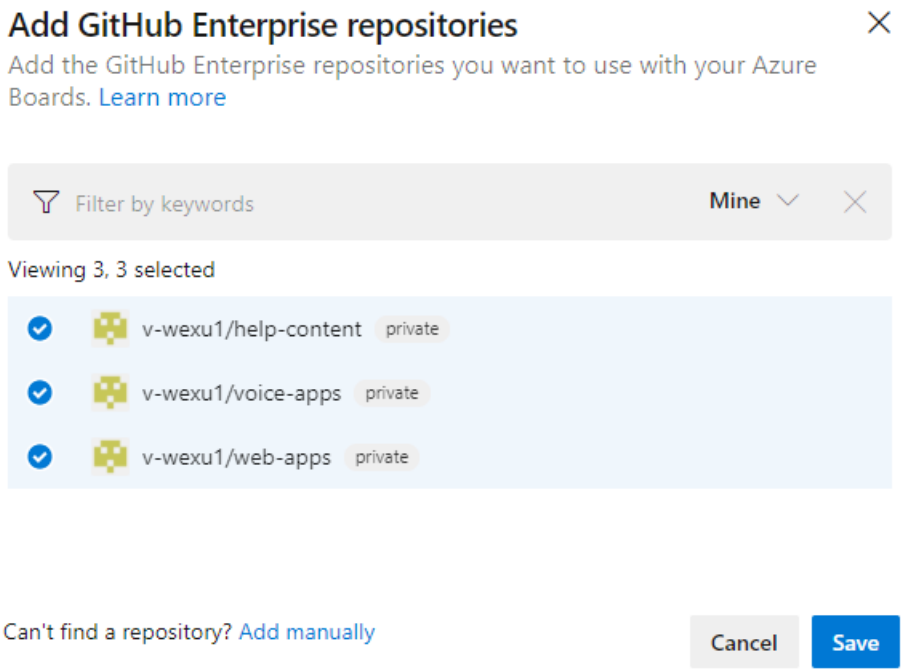
Connect using a Username and Password

Enter the URL for your GitHub Enterprise server and the administrator account credentials recognized by that server. And then choose **Connect**.



Choose the repositories to connect to

3. The dialog lists all repositories for which you have GitHub administration rights. You can toggle between **Mine** and **All** to determine if others appear, and then check the ones that you want to add. Choose **Save** when done.



> **TIP**
> You can only make a connection to repositories defined under one GitHub organization. To connect a project to other repositories defined in another GitHub organization, you must add another connection.

4. If it's your first time connecting to a GitHub account or organization from Azure Boards, you'll also be installing the Azure Boards app for GitHub. Complete the integration by following the procedures outlined in Confirm the connection earlier in this article.

Resolve connection issues

See Troubleshoot GitHub repository connection.

Try this next

Link GitHub commits and pull requests to work items

Related articles

- Add or remove GitHub repositories
- Install and configure the Azure Boards app for GitHub
- Configure status badges to add to GitHub README files
- Troubleshoot GitHub & Azure Boards integration
- Build GitHub repositories
- Build GitHub Enterprise Server repositories
- Trigger an Azure Pipelines run from GitHub Actions

Connect an on-premises Azure DevOps Server to a GitHub repo

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Server 2022 | Azure DevOps Server 2020 | Azure DevOps Server 2019

By connecting your Azure DevOps Server project to your GitHub repositories, you support linking between GitHub commits and pull requests to work items. You can use GitHub for software development while using Azure Boards to plan and track your work.

To connect to GitHub.com repositories, you must install Azure DevOps Server 2020.1.1 Patch 2. Without this patch, you can only connect to your GitHub Enterprise Server repositories.

> **NOTE**
> On-premises Azure DevOps Server 2020 supports integration with GitHub.com and GitHub Enterprise Server repositories. If you want to connect from Azure DevOps Services, see Connect Azure Boards to GitHub.

By connecting your Azure DevOps Server project with your GitHub Enterprise Server repositories, you support linking between GitHub commits and pull requests to work items. You can use GitHub Enterprise for software development while using Azure Boards to plan and track your work.

> **NOTE**
> On-premises Azure DevOps Server 2019 supports integration with GitHub Enterprise Server repositories. If you want to connect from Azure DevOps Services, see Connect Azure Boards to GitHub.

Prerequisites

- Install the Azure Boards app for GitHub on the GitHub organizations or account.
- Connect to an Azure Boards or Azure DevOps project. If you don't have a project yet, create one.
- You must be a member of the **Project Collection Administrators** group and the project's Contributors group. If you created the project, then you have permissions.
- You must be an administrator of the GitHub Enterprise Server you'll connect to.

Authentication options

The following authentication options are supported.

- PAT
- Username plus password

> **NOTE**
> OAuth is no longer supported for Azure DevOps Server 2020.

- OAuth (Recommended, registration required)
- PAT

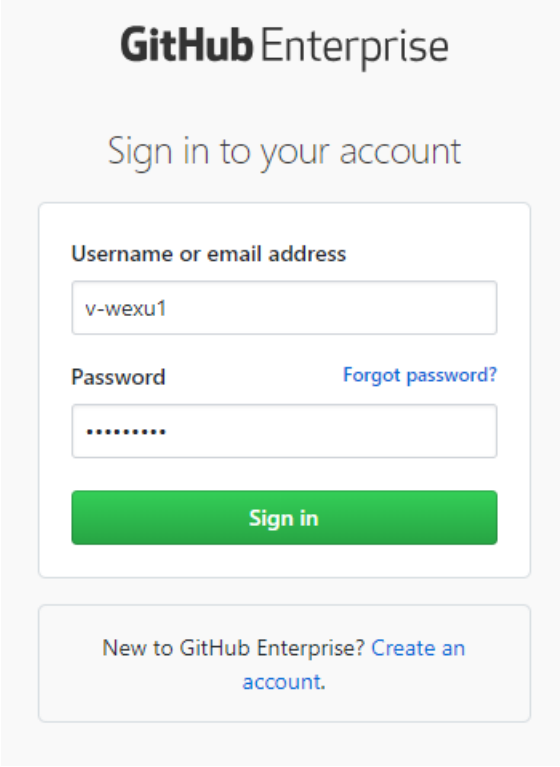
- Username plus password

Register Azure DevOps in GitHub as an OAuth App

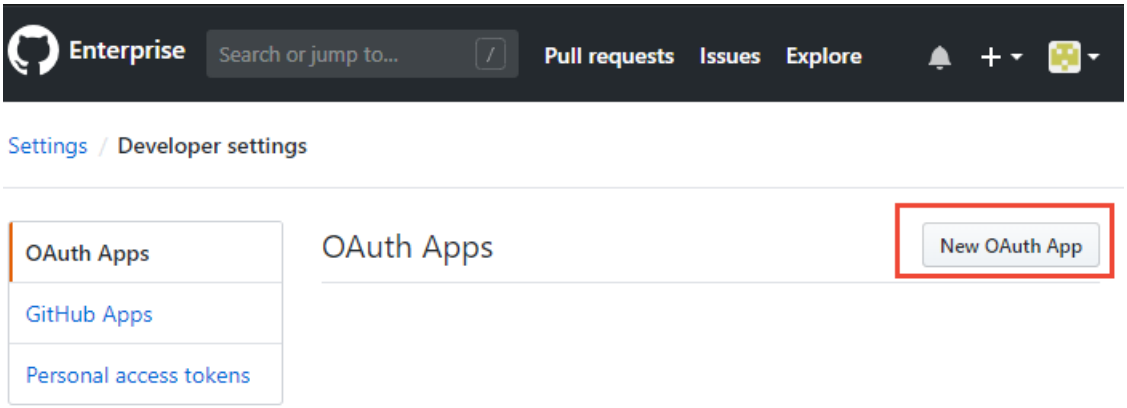
If you plan to use OAuth to connect Azure DevOps Server with your GitHub Enterprise Server, you first need to register the application as an OAuth App. For details, see Creating an OAuth App.

Register Azure DevOps Server

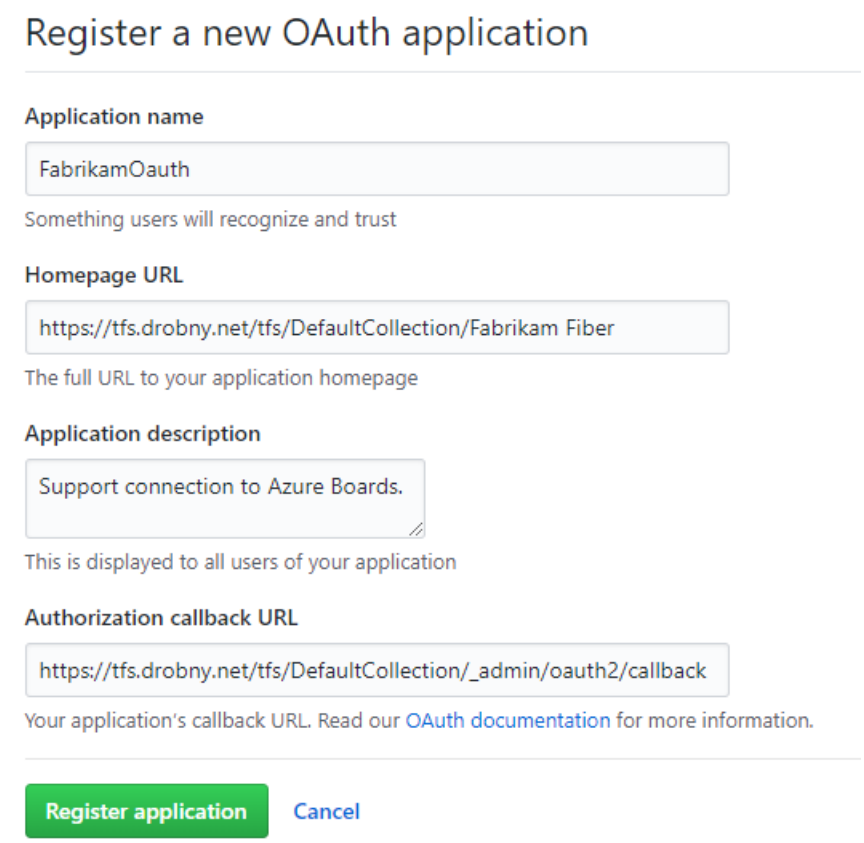
1. Sign into the web portal for your GitHub Enterprise server.



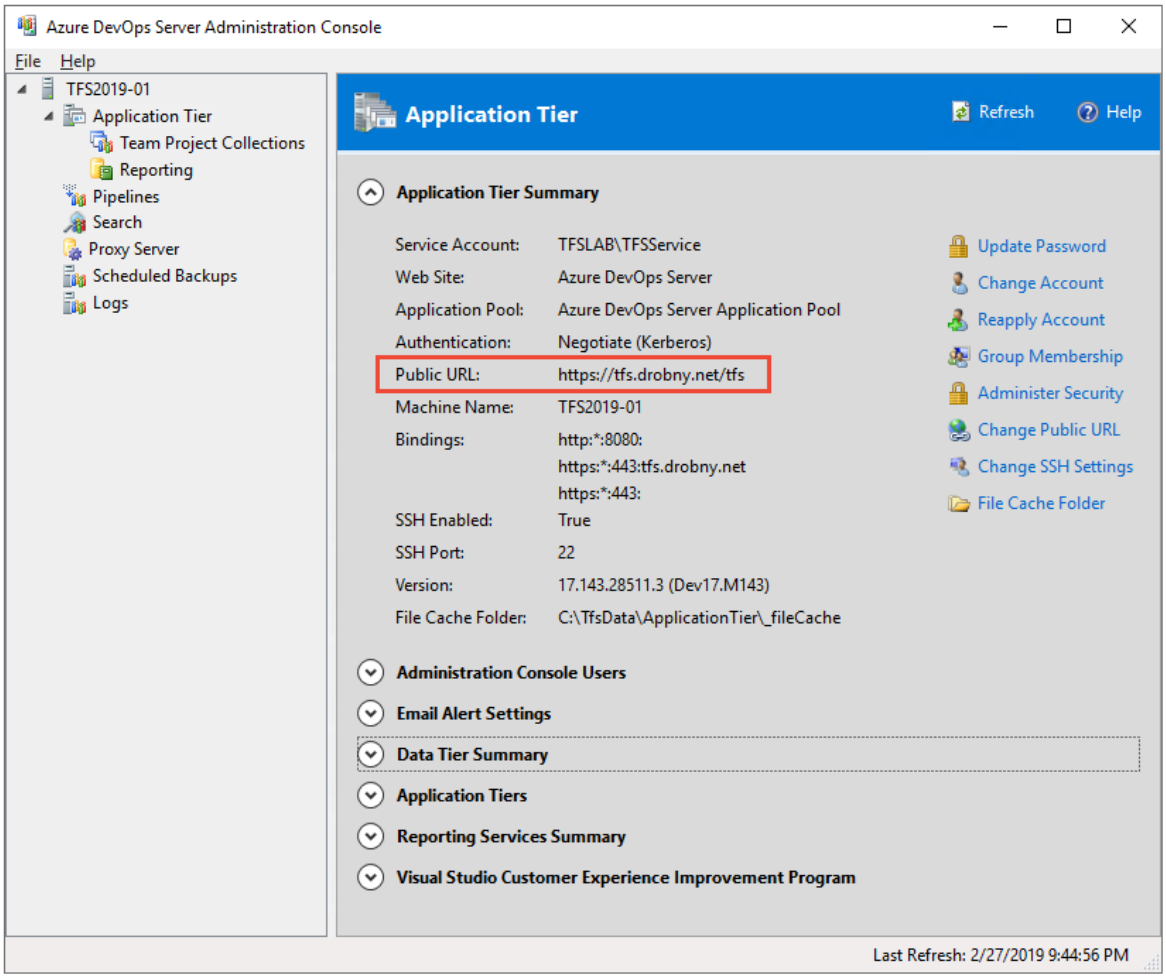
2. Open **Settings > Developer settings > Oauth Apps** and choose **New OAuth App**.



3. Fill out the form to register your Azure DevOps Server application.



For the **Homepage URL**, specify the **Public URL** of your project collection. You can find this URL by opening the Azure DevOps Administration Console and viewing the **Application Tier** node.



For the **Authorization callback URL**, use the following pattern to construct the URL.

```
{Azure DevOps Server Public Url}/{Collection Name}/_admin/oauth2/callback
```

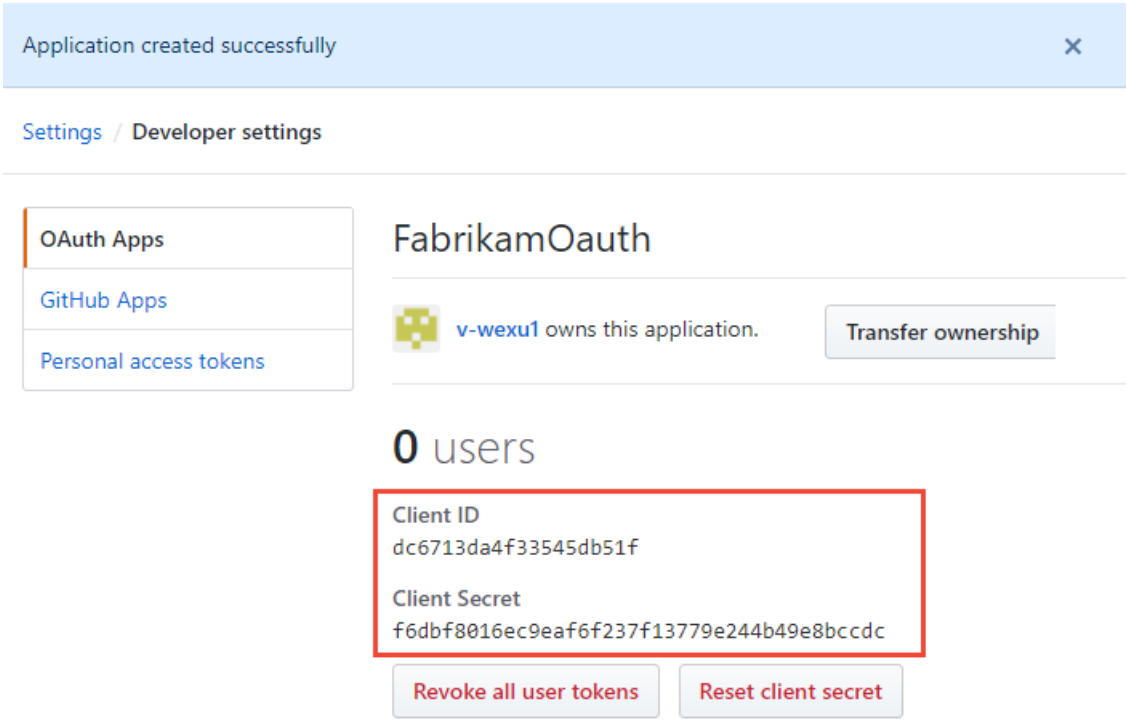
For example:

```
http://contoso/DefaultCollection/_admin/oauth2/callback
```

Or,

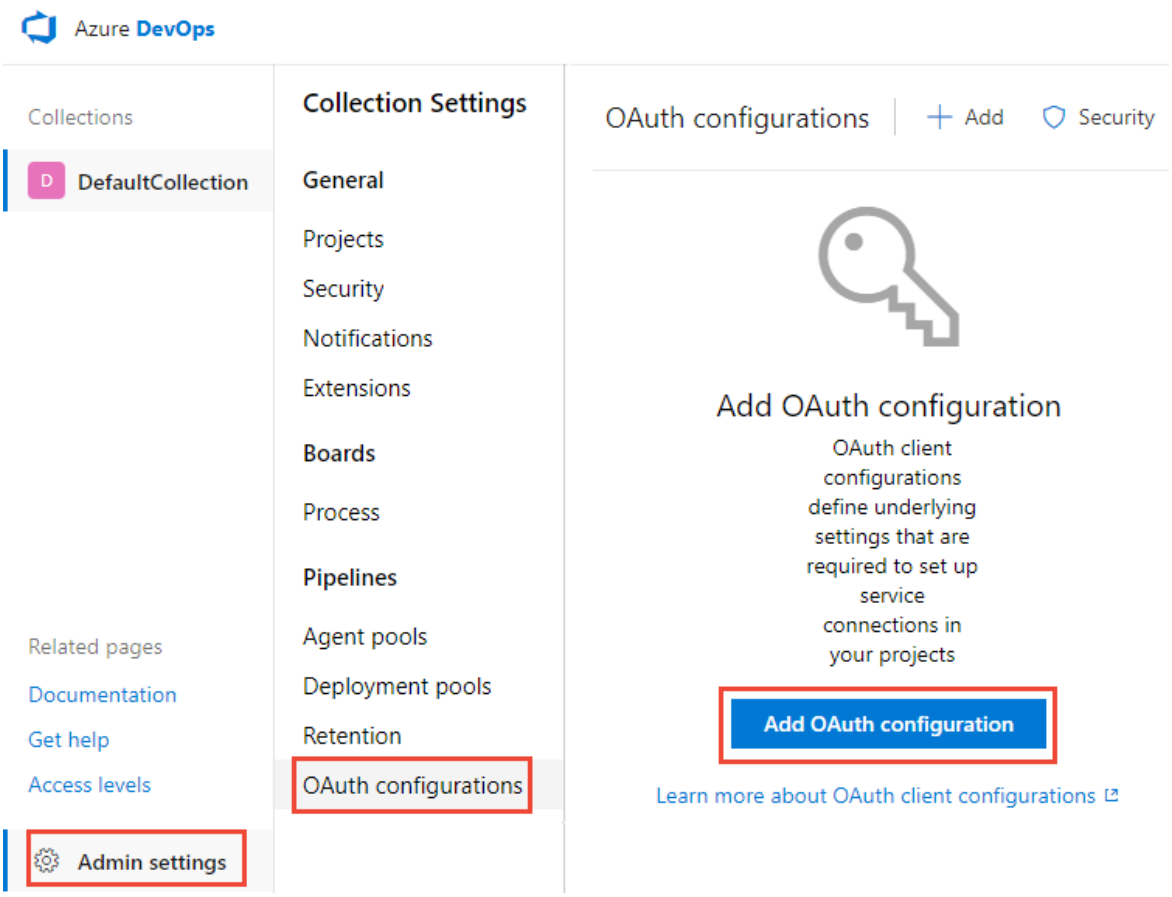
```
https://tfs.contoso.com/MyCollection/_admin/oauth2/callback
```

4. Choose **Register application**.
5. Upon success, you'll see a page that provides the **Client ID** and **Client Secret** for your registered OAuth application.

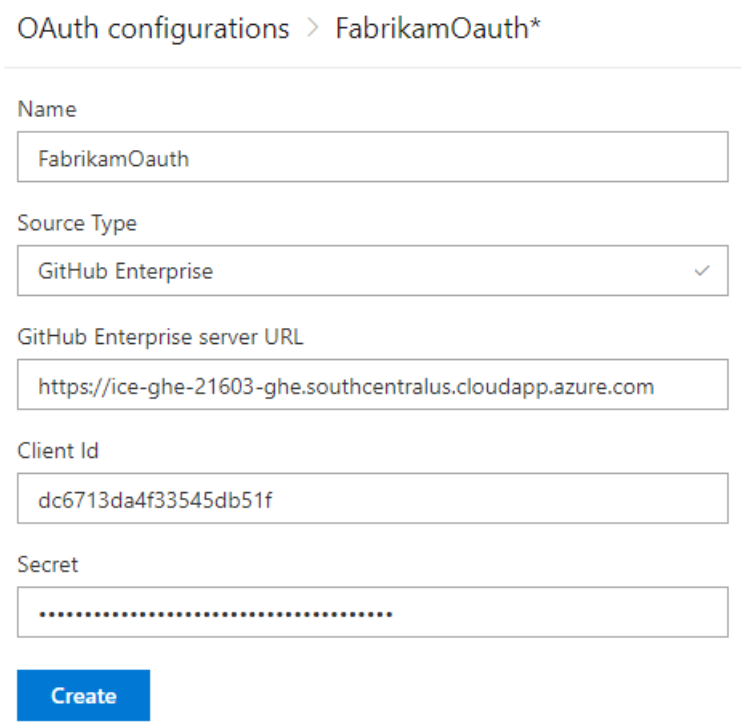


Register your OAuth configuration in Azure DevOps Server

1. Sign into the web portal for your Azure DevOps Server.
2. Add the GitHub Enterprise Oauth configuration to your Azure DevOps Server collection.
3. Open **Admin settings > Oauth configurations**, and choose **Add Oauth configuration**.



4. Fill in the form that appears, and then choose **Create**.



Connect Azure DevOps Server to GitHub Enterprise Server

You can connect up to 250 GitHub repositories to an Azure Boards project.

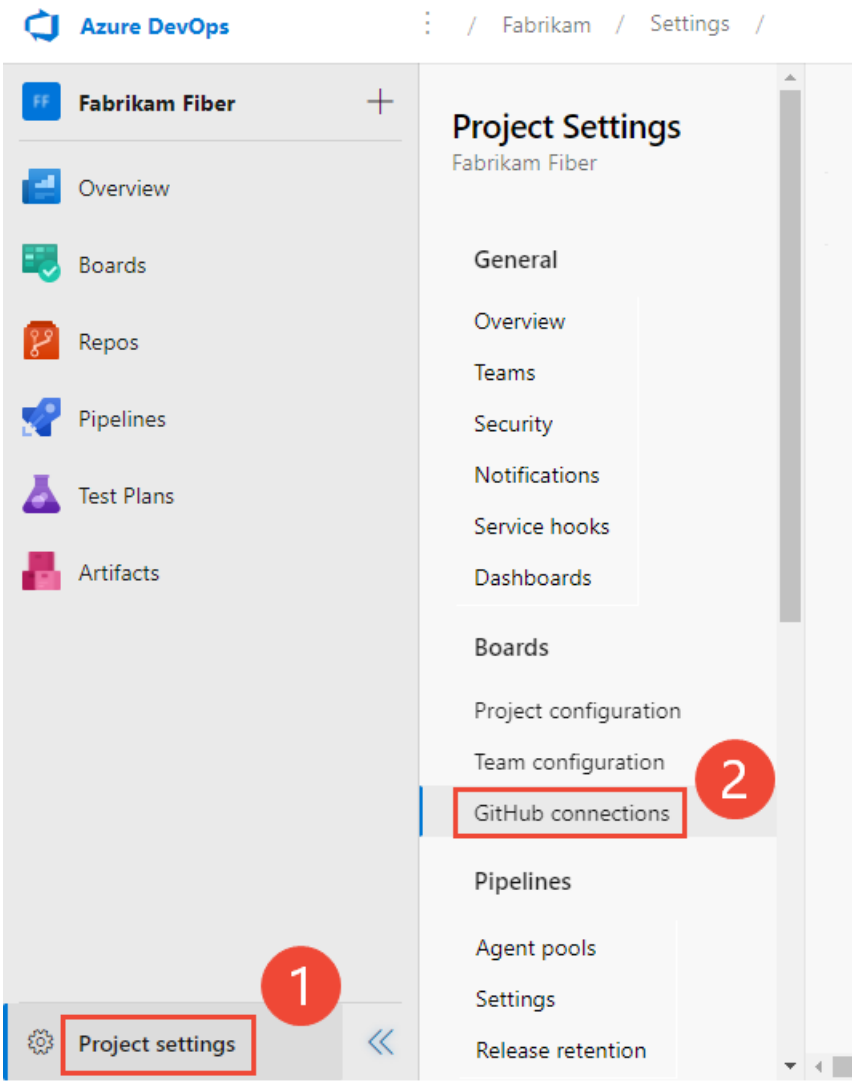
NOTE

Connection to more than 100 GitHub repositories requires Azure DevOps Server 2020.1 update or later version.
Connection to GitHub.com repositories requires Azure DevOps Server 2020.1.1 Patch 2 or later version.

You can connect up to 100 GitHub repositories to an Azure Boards project. This limit can't be changed.

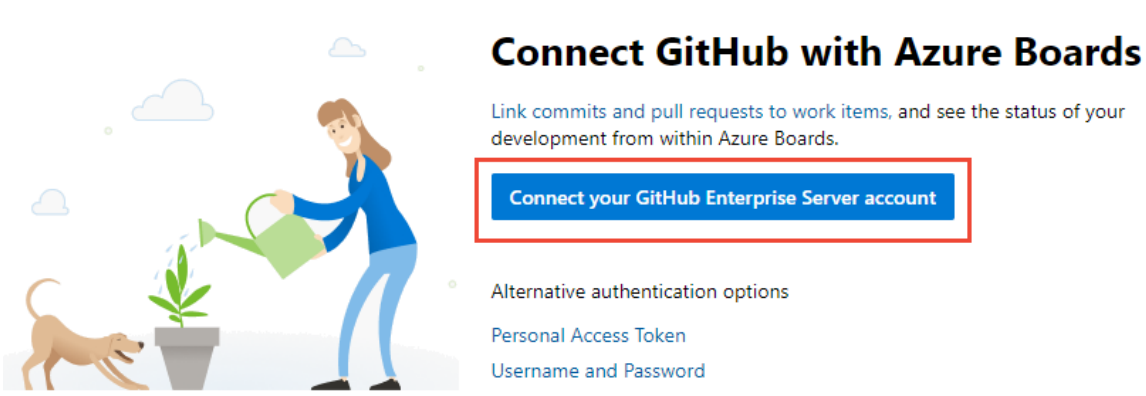
1. Open the web portal for your Azure DevOps Server.
2. Choose the Azure DevOps logo to open **Projects**, and then choose the Azure Boards project you want to configure to connect to your GitHub Enterprise repositories.

1. Choose (1) **Project Settings** > (2) **GitHub connections**.



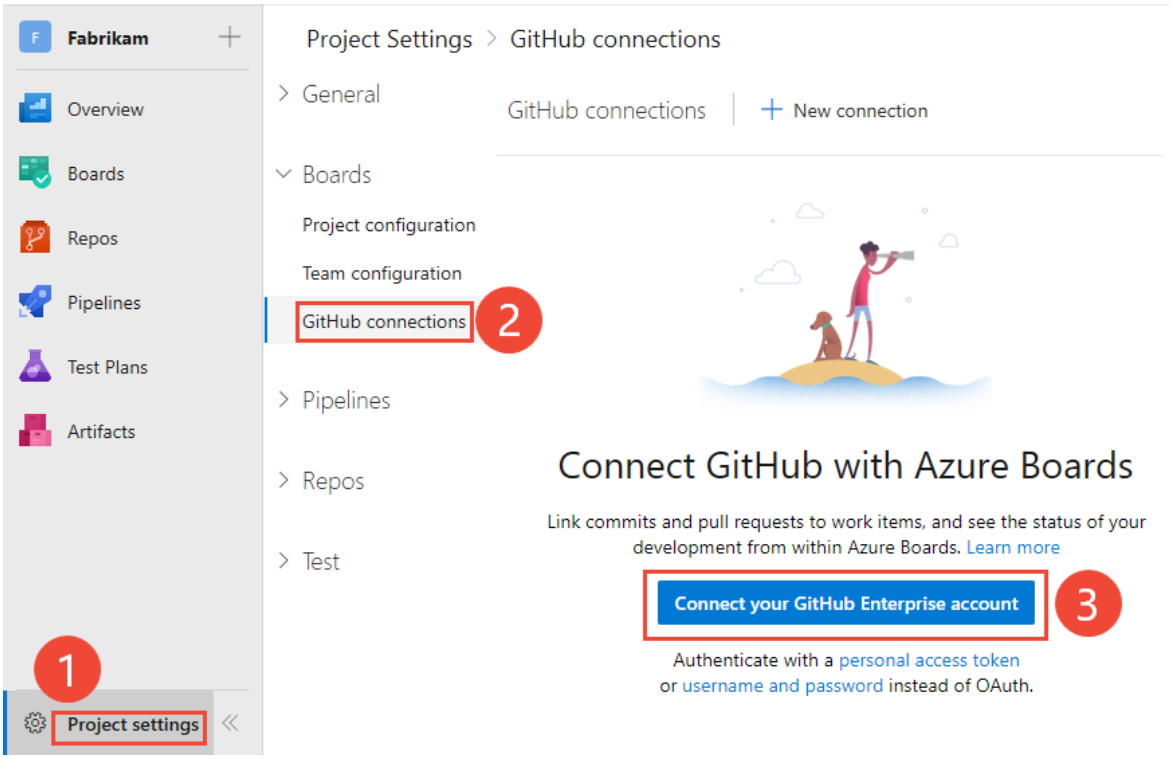
2. If it's the first time making a connection from the project, choose the authentication method you want to use to make the connection:

   - **Personal Access Token**, for details see Connect using a Personal Access Token.
   - **User Name and Password**, see Connect using a Username and Password.



Otherwise, choose + **New connection**, and select your authentication method from the **New Connection** dialog.

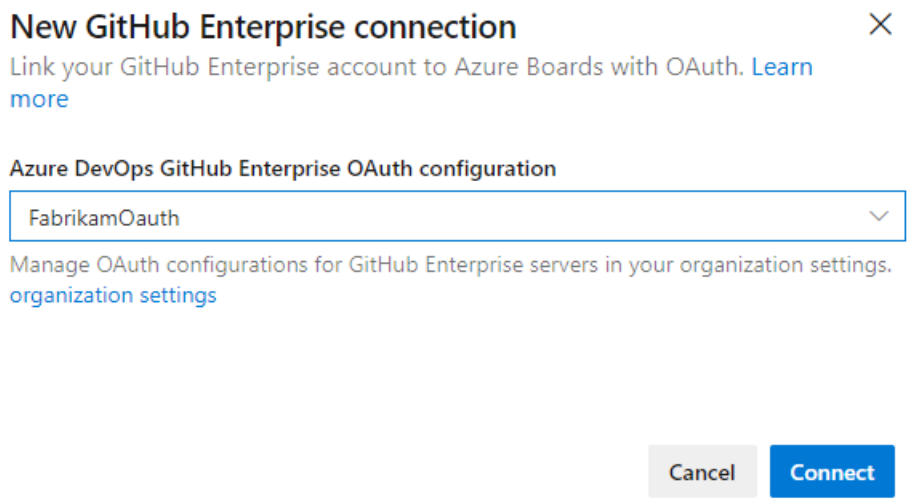
2. Choose (1) **Project Settings** > (2) **GitHub connections**, and then (3) **Connect your GitHub Enterprise account**.



Or, choose a **personal access token** or **username and password**, if you're using those credentials.

Connect using OAuth

- Choose the configuration that you set up in Step 4 of Register your OAuth configuration in Azure DevOps Server. Then, choose **Connect**.

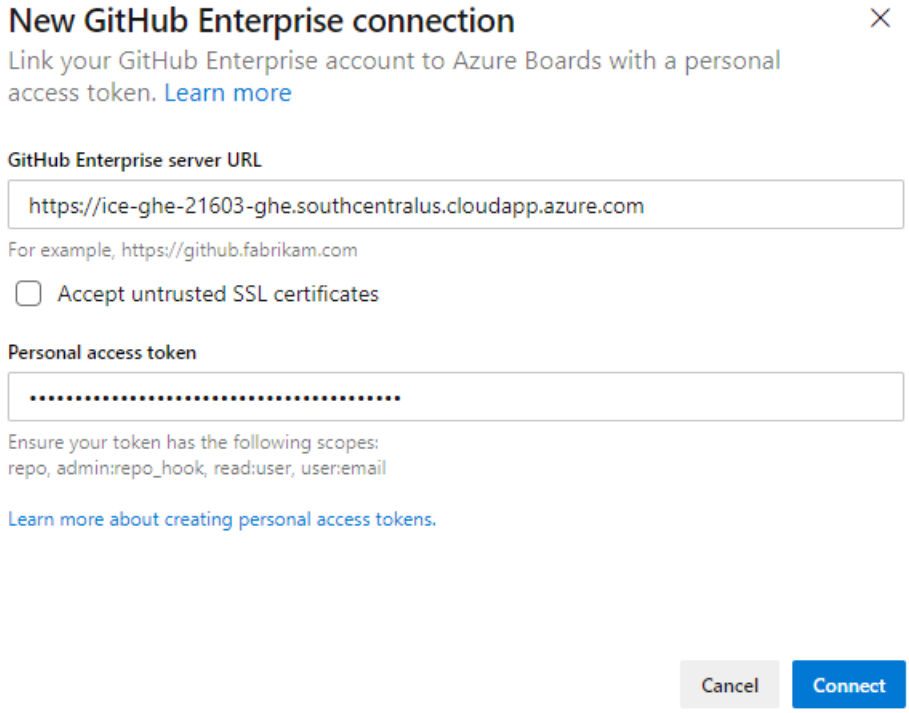


Connect using a Personal Access Token

1. To create a PAT, see Creating a personal access token.

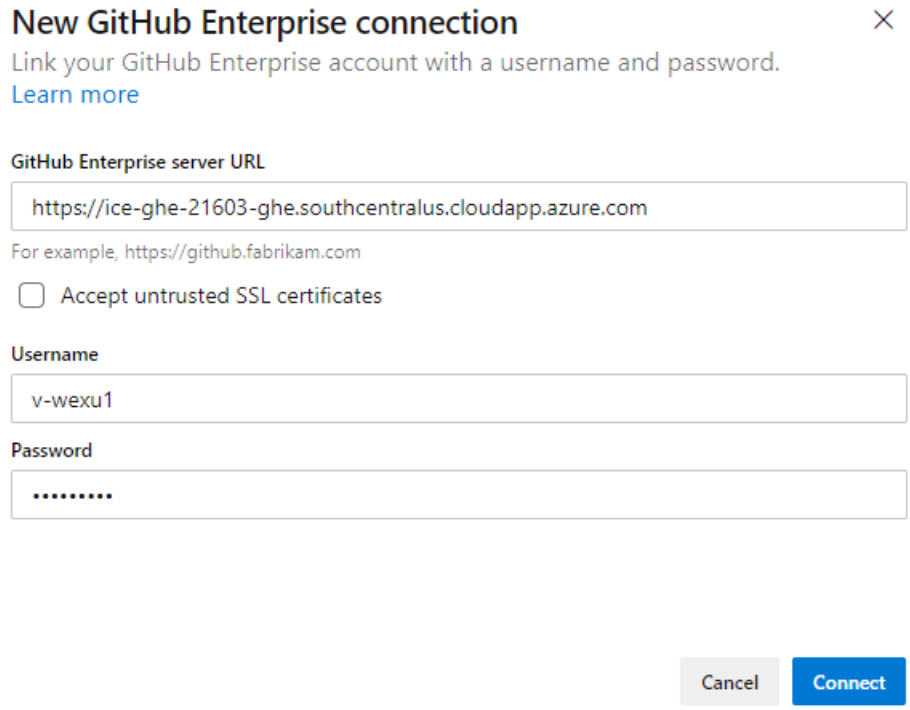
> **TIP**
> When creating your GitHub PAT, make sure that you include these scopes:
> `repo, admin:repo_hook, read:user, user:email`.

2. Enter the URL for your GitHub Enterprise server and the **Personal access token** credentials recognized by that server. And then choose **Connect**.

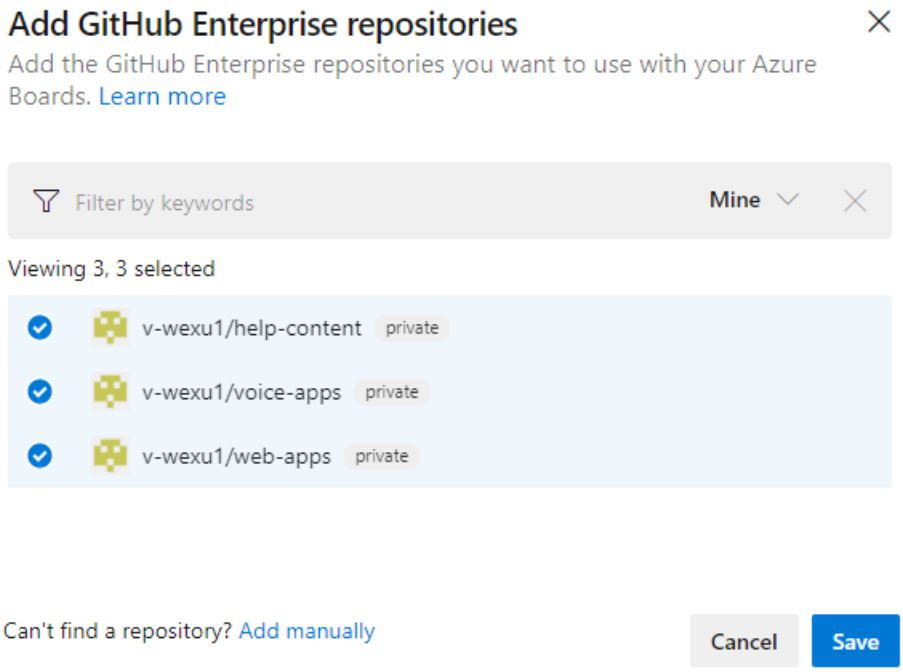


Connect using a Username and Password

1. Enter the URL for your GitHub Enterprise server and the administrator account credentials recognized by that server. And then choose **Connect**.



2. The dialog lists all repositories for which you have GitHub administration rights. You can toggle between **Mine** and **All** to determine if others appear, and then check the ones that you want to add. Choose **Save** when done.



3. If you're connecting to a GitHub account or organization from Azure Boards for the first time, you'll also be installing the Azure Boards app for GitHub. Complete the integration by following the procedures outlined in Confirm the connection.

Resolve connection issues

See Troubleshoot GitHub repository connection.

Try this next

Link GitHub commits and pull requests to work items

Related articles

- Add or remove GitHub repositories
- What is Azure Boards?
- Troubleshoot GitHub & Azure Boards integration
- Build GitHub Enterprise Server repositories
- Trigger an Azure Pipelines run from GitHub Actions

Link GitHub commits, pull requests, and issues to work items in Azure Boards

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Once you've connected your Azure Boards project with a GitHub repository, you can link work items to your GitHub commits and pull requests. You can add links using the **#mention** syntax familiar to GitHub users or by adding a GitHub Commit or GitHub Pull Request link type from the Azure Boards work item.

> **NOTE**
> With the Azure Boards app for GitHub, Azure Boards and Azure DevOps Services support integration with GitHub.com and GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

> **NOTE**
> With the Azure Boards app for GitHub, Azure DevOps Servers 2019 and later versions support integration with GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

Prerequisites

- Your Azure Boards project must be connected to the GitHub repository where the commits and pull requests you want to link to/from exist. For details, see Azure Boards-GitHub integration.
- You must be a Contributor to Azure Boards.
- You must be a Contributor to the GitHub repository.
- If your organization uses the Hosted XML process model to customize the work tracking experience, you'll need to update the work item types to link to and view the GitHub link types from the Development section in the work item form. For details, see Update XML definitions for select work item types.

Use `AB#` mention to link from GitHub to Azure Boards work items

From a GitHub commit, pull request or issue, use the following syntax to create a link to your Azure Boards work item. Enter the `AB#ID` within the text of a commit message. Or, for a pull request or issue, enter the `AB#ID` within the title or description (not a comment).

From a GitHub commit or pull request, use the following syntax to create a link to your Azure Boards work item. Enter the `AB#ID` within the text of a commit message or for a pull request, enter the `AB#ID` within the pull request title or description (not a pull request comment).

```
AB#{ID}
```

For example, `AB#125` will link to work item ID 125.

You can also enter a commit or pull request message to transition the work item. The system will recognize `fix, fixes, fixed` and apply it to the #-mention item that follows. Some examples are provided as shown.

Examples:

COMMIT MESSAGE	ACTION
`Fixed AB#123`	Links and transitions the work item to the "done" state.
`Adds a new feature, fixes AB#123.`	Links and transitions the work item to the "done" state.
`Fixes AB#123, AB#124, and AB#126`	Links to Azure Boards work items 123, 124, and 126. Transitions only the first item, 123 to the "done" state.
`Fixes AB#123, Fixes AB#124, Fixes AB#125`	Links to Azure Boards work items 123, 124, and 126. Transitions all items to the "done" state.
`Fixing multiple bugs: issue #123 and user story AB#234`	Links to GitHub issue 123 and Azure Boards work item 234. No transitions.

NOTE

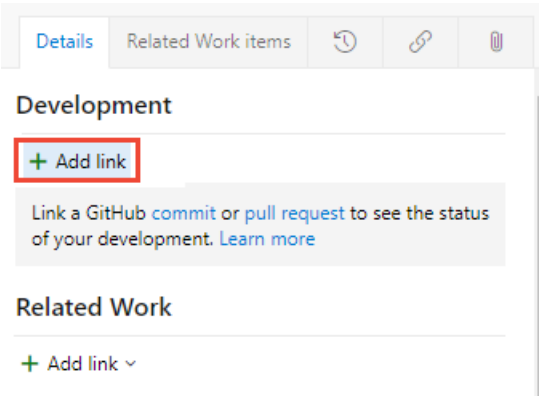
If you have connected the same GitHub repo to projects defined in two or more Azure DevOps organizations, you may see unexpected **AB#** mention linking. For details, see Troubleshoot GitHub & Azure Boards integration. For this reason, we recommend that you only connect a GitHub repo to projects defined in a single Azure DevOps organization.

Add link from a work item to a GitHub commit, pull request, or issue

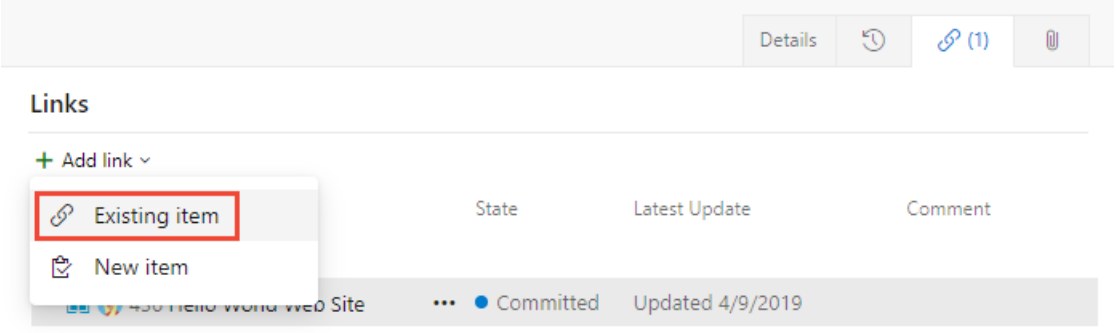
NOTE

Linking to a GitHub issue requires Azure DevOps Server 2019 Update 1 or later version.

1. To link to a commit or pull request, open the work item and choose **Add Link** under the Development section.

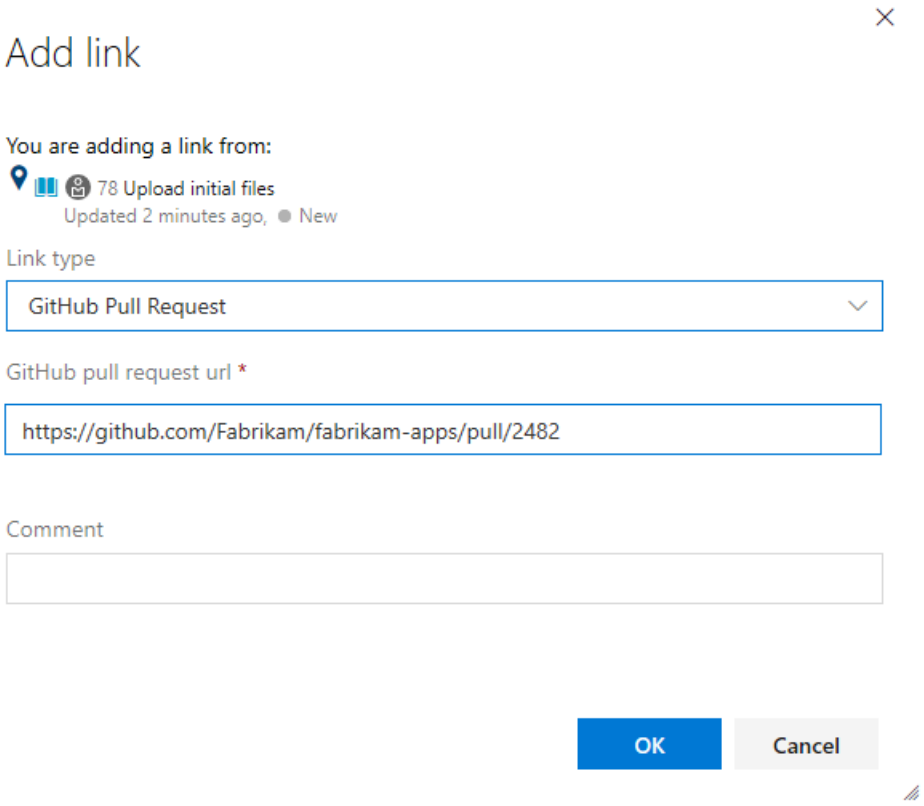


To link to an issue, choose the **Links** tab, and then choose **Add Link > Existing item**.



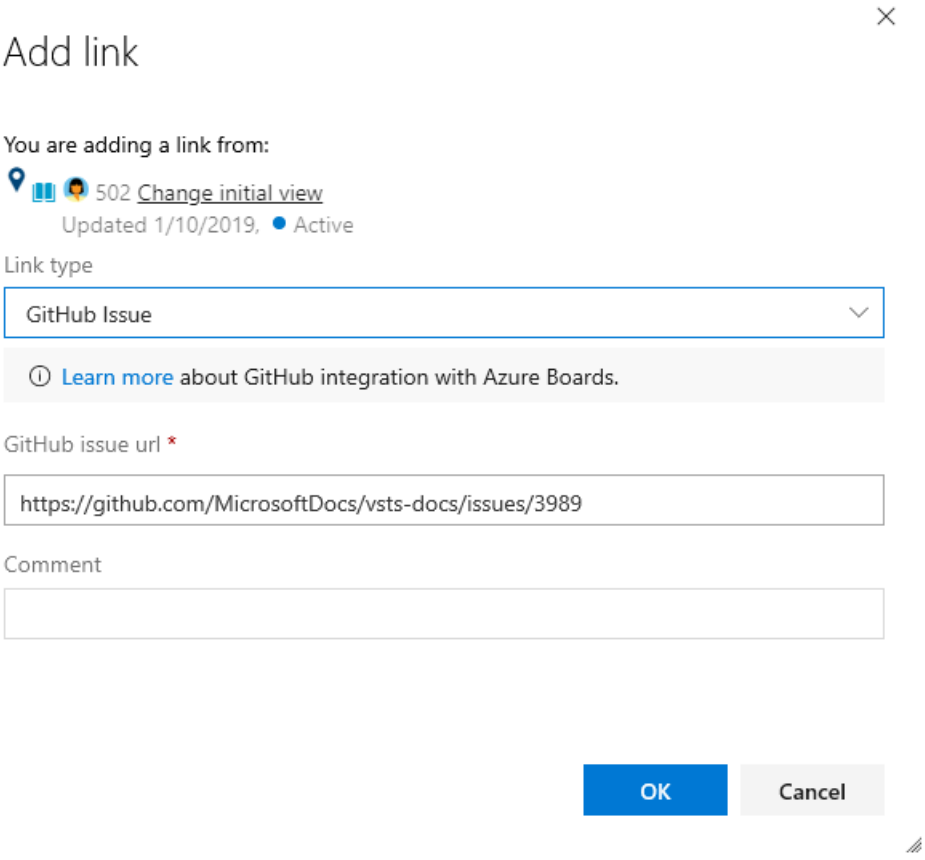
2. From the Add link dialog, select one of the **GitHub** link types, enter the URL to the commit, pull request, or issue and then choose **OK**.

   Here, we add a link to a GitHub pull request.



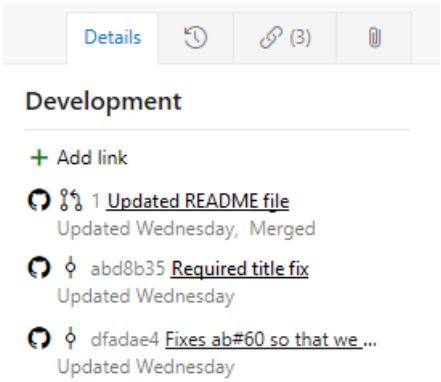
   Azure Boards completes a check to ensure that you've entered a valid link. The linked-to GitHub repository must be integrated with the project or the validation will fail.

   Here, we add a link to a GitHub issue.



View or open links from the Development section

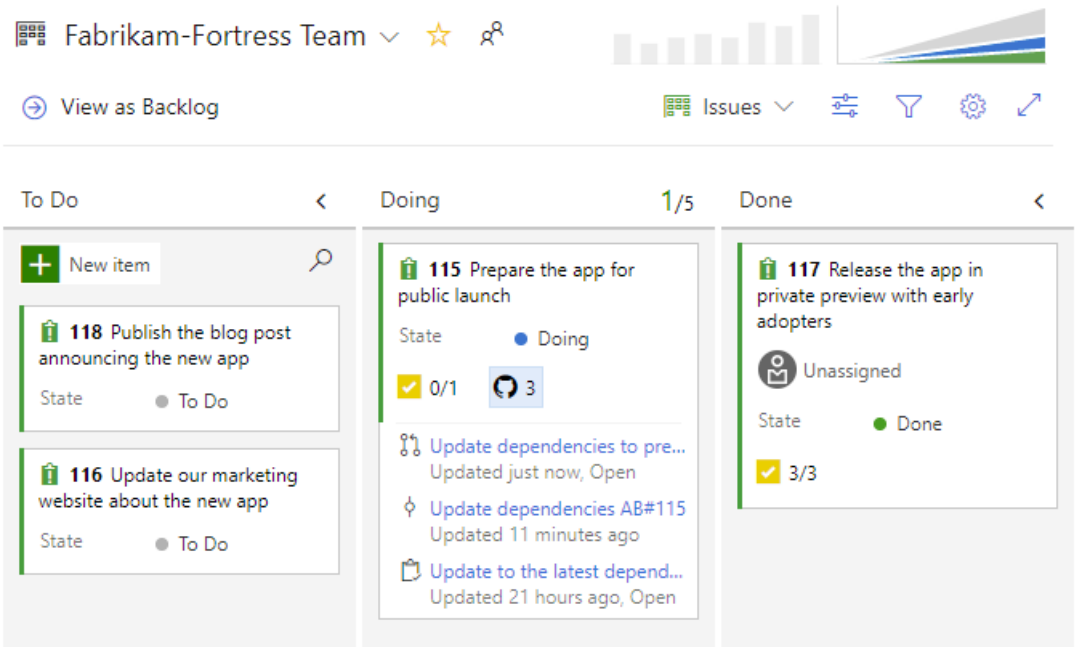
The Development section within the work item form lists the links created to GitHub commits and pull requests with the GitHub icon.



Choose the link provided to open the commit or pull request in GitHub.

View GitHub objects on Kanban board

With GitHub annotations enabled on the Kanban board, you can quickly open linked GitHub commits, pull requests, or issues for more detail. For details on enabling annotations, see Customize cards.



> **NOTE**
> GitHub annotations requires Azure DevOps Server 2019 Update 1 or later version.

Try this next

Configure status badges

Related articles

- Azure Boards-GitHub integration
- Linking, traceability, and managing dependencies
- Troubleshoot GitHub & Azure Boards integration

Configure status badges for your GitHub repo

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

You can add Markdown syntax to a GitHub repo `README.md` file to display your Kanban board's status in that repo. Show the status by adding the syntax you choose from your Kanban board settings.

> **NOTE**
> Requires Azure DevOps Server 2019 Update 1 or later version.

The syntax shown works whether you've connected your project to a GitHub.com or your GitHub Enterprise Server repository. For GitHub Enterprise Server, your server must be network accessible to Azure DevOps Services.

Prerequisites

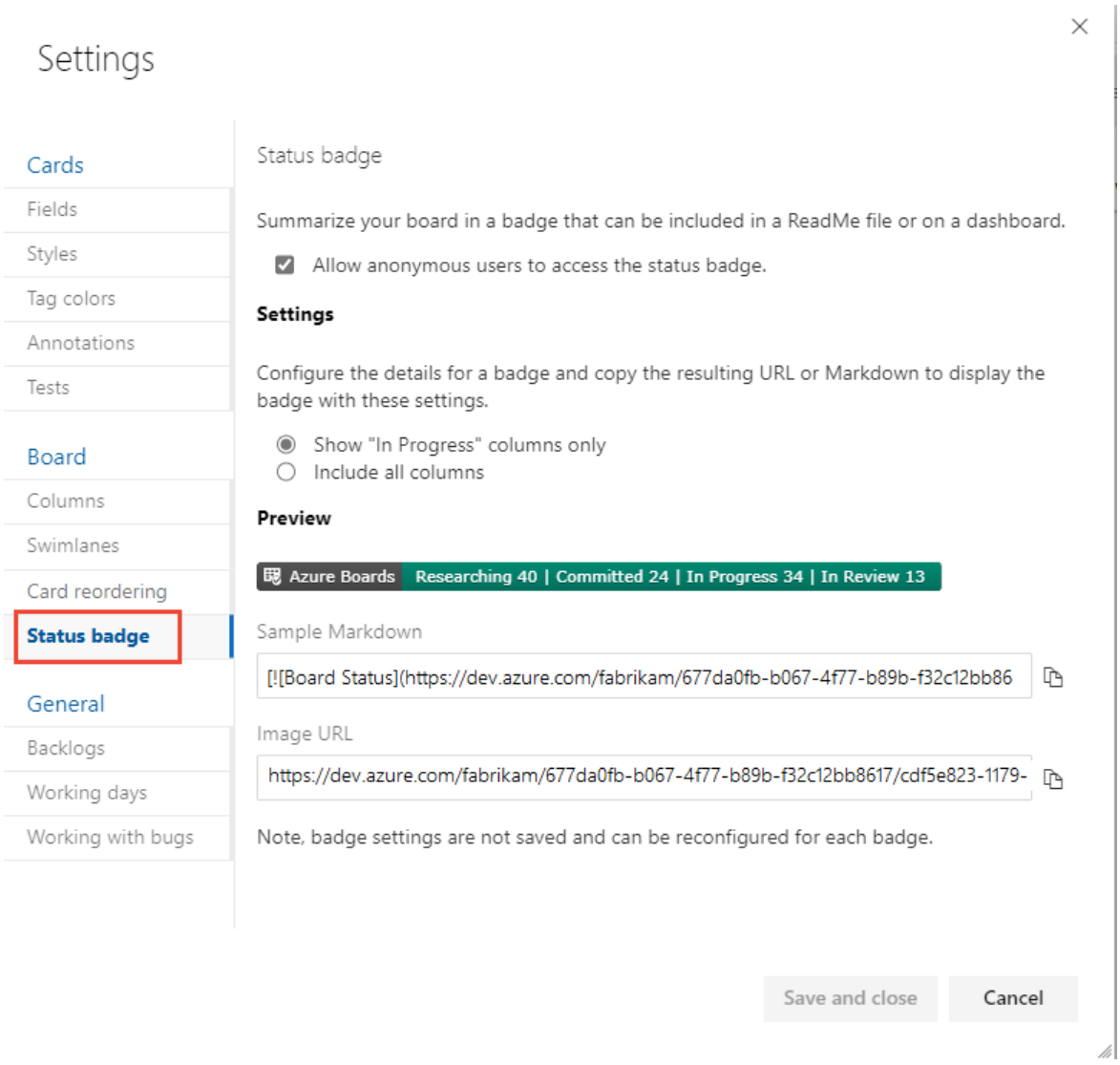
- Your Azure Boards project must be connected to the GitHub repository where the commits and pull requests you want to link to/from exist. For details, see Azure Boards-GitHub integration.
- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- You must be added to the team administrator role for the team's settings you want to modify, or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- To add the status badge to the GitHub.com repository, you must be a contributor of the repository.

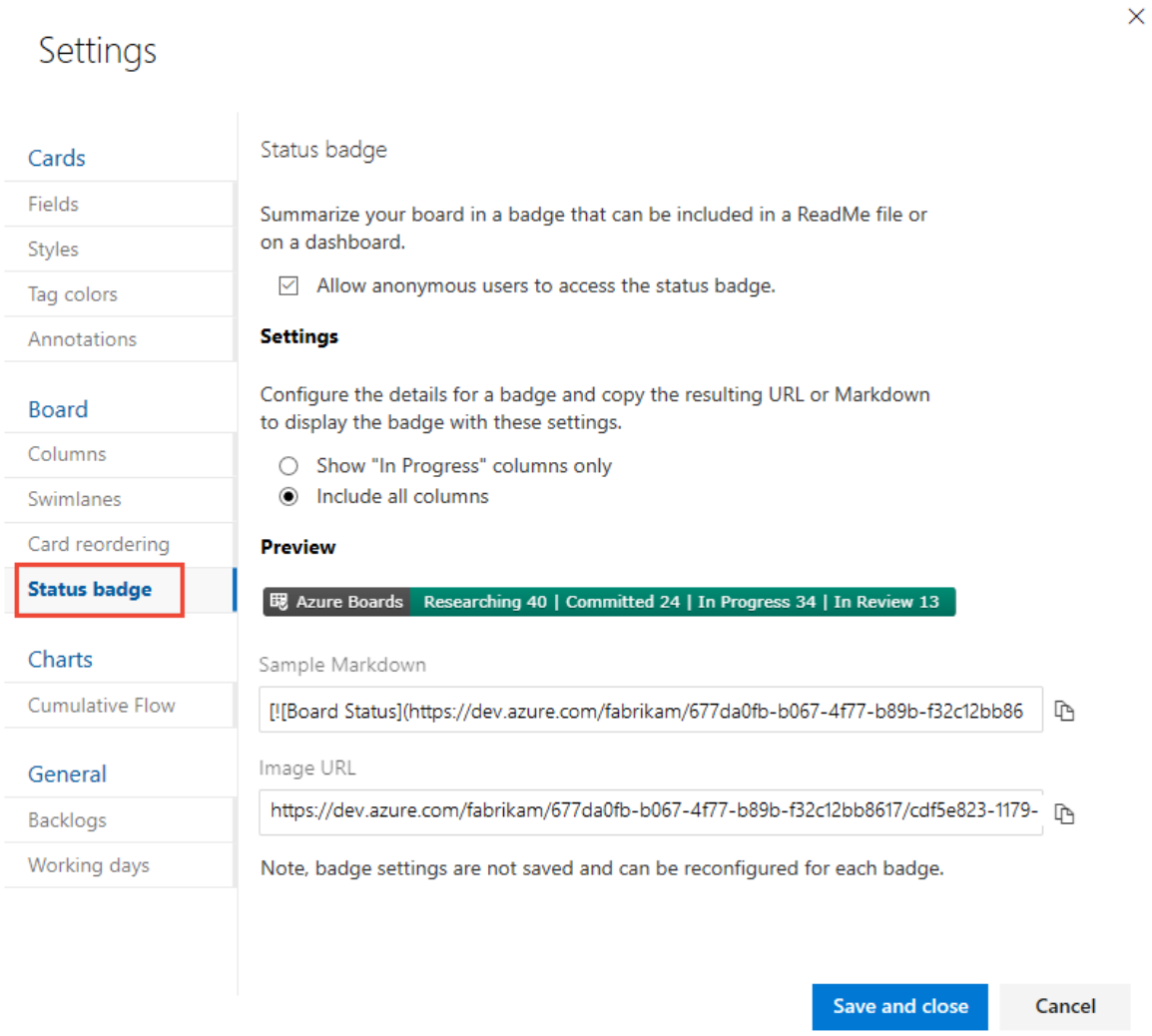
Open your Kanban board settings

1. Sign into Azure Boards.
2. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
3. Choose the gear icon to configure the board and set general team settings.



4. Choose **Status badge** and then check or uncheck the **Allow anonymous users to access the status badge**. When unchecked, users who aren't signed in can still view the status badge.


Settings
Cards
Fields
Styles
Tag colors
Annotations
Tests
Board
Columns
Swimlanes
Card reordering
Status badge
General
Backlogs
Working days
Working with bugs
Status badge
Summarize your board in a badge that can be included in a ReadMe file or on a dashboard.
Allow anonymous users to access the status badge.
Settings
Configure the details for a badge and copy the resulting URL or Markdown to display the badge with these settings.
Show "In Progress" columns only
Include all columns
Preview
Azure Boards
Researching 40 | Committed 24 | In Progress 34 | In Review 13
Sample Markdown
[![Board Status](https://dev.azure.com/fabrikam/677da0fb-b067-4f77-b89b-f32c12bb86
Image URL
https://dev.azure.com/fabrikam/677da0fb-b067-4f77-b89b-f32c12bb8617/cdf5e823-1179-
Note, badge settings are not saved and can be reconfigured for each badge.
Save and close
Cancel



5. Choose the badge type you want and choose the copy icon to copy the Markdown syntax for the badge.

   Show "In progress" columns only ignores the first and last columns.
   Include all columns includes the first and last columns of the board.

   Also, you can customize the set of columns by specifying `2` for the columnOptions and then a comma-delimited list of the board columns to appear. For example, `?columnOptions=2&columns=Proposed,Committed,In%20Progress,In%20Review`, as shown in the following syntax. For column labels that include spaces, you must encode the space with `%20`. For example, `In%20Progress`.

   ```
   [![Board Status](https://dev.azure.com/fabrikam/677da0fb-b067-4f77-b89b-f32c12bb8617/cdf5e823-1179-4503-9fb1-a45e2c1bc6d4/_apis/work/boardbadge/6fa7b56f-d27c-4e96-957d-f9e7b0f56705?columnOptions=2&columns=Proposed,Committed,In%20Progress,In%20Review)](https://dev.azure.com/fabrikam/677da0fb-b067-4f77-b89b-f32c12bb8617/_boards/board/t/cdf5e823-1179-4503-9fb1-a45e2c1bc6d4/Microsoft.RequirementCategory/)
   ```

   And which will cause a badge similar to the following to display.

6. When done, choose **Save and close** or **Cancel**.

> **NOTE**
>
> The only setting that you can configure is the **Allow anonymous users to access the status badge**. The badge type under **Settings** only switches the Markdown syntax for you to copy from the **Sample Markdown** and **Image URL** values.

7. Open the README file in your GitHub repo and paste the syntax you copied to have the badge display.

   You should see the same preview image that you selected with values that correspond to your Kanban board. For example:

Related articles

- Add columns to your Kanban board
- Customize cards
- Configure team settings
- Troubleshoot GitHub & Azure Boards integration

Add or remove GitHub repositories

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Once you make a connection to a GitHub repository from Azure Boards, you can add or remove repositories under the same GitHub account or organization. Or, you can completely remove the connection to all repositories.

You can manage which GitHub repositories can participate in Azure Boards integration. For more information, see Change GitHub repository access, or suspend or uninstall the integration.

> **NOTE**
> With the Azure Boards app for GitHub, Azure Boards and Azure DevOps Services support integration with GitHub.com and GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

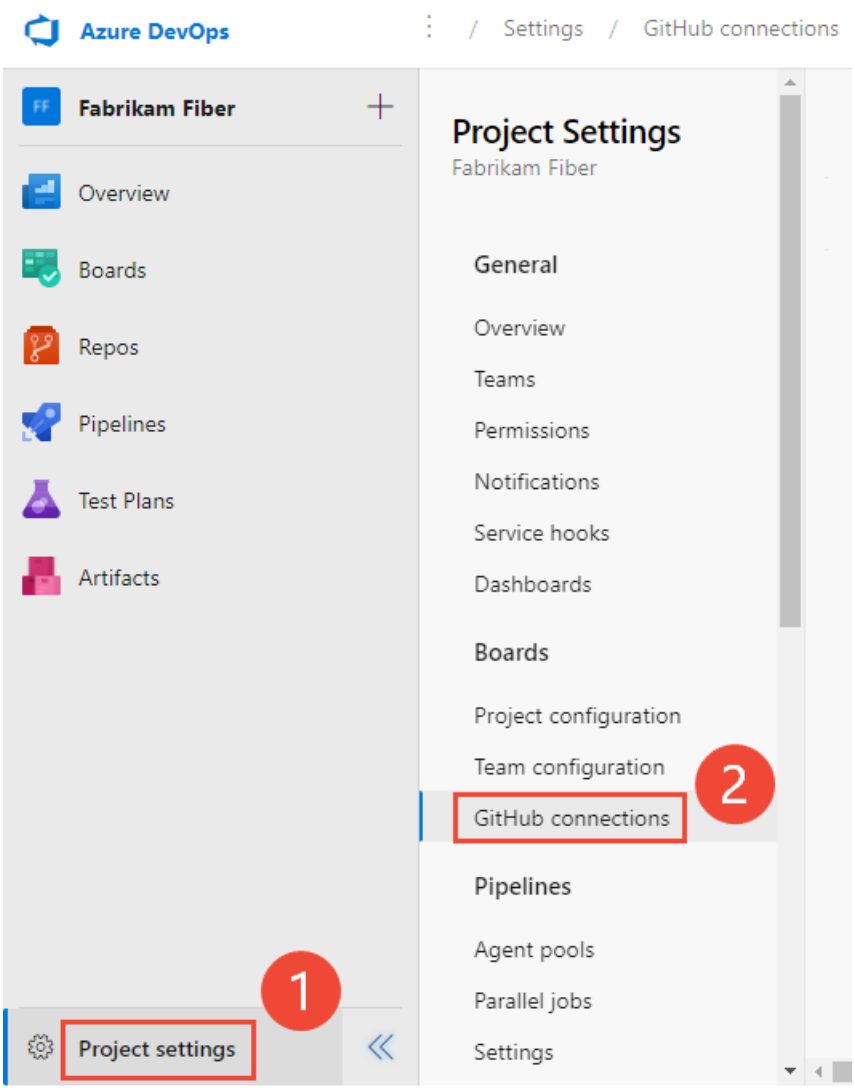
> **NOTE**
> With the Azure Boards app for GitHub, Azure DevOps Servers 2019 and later versions support integration with GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

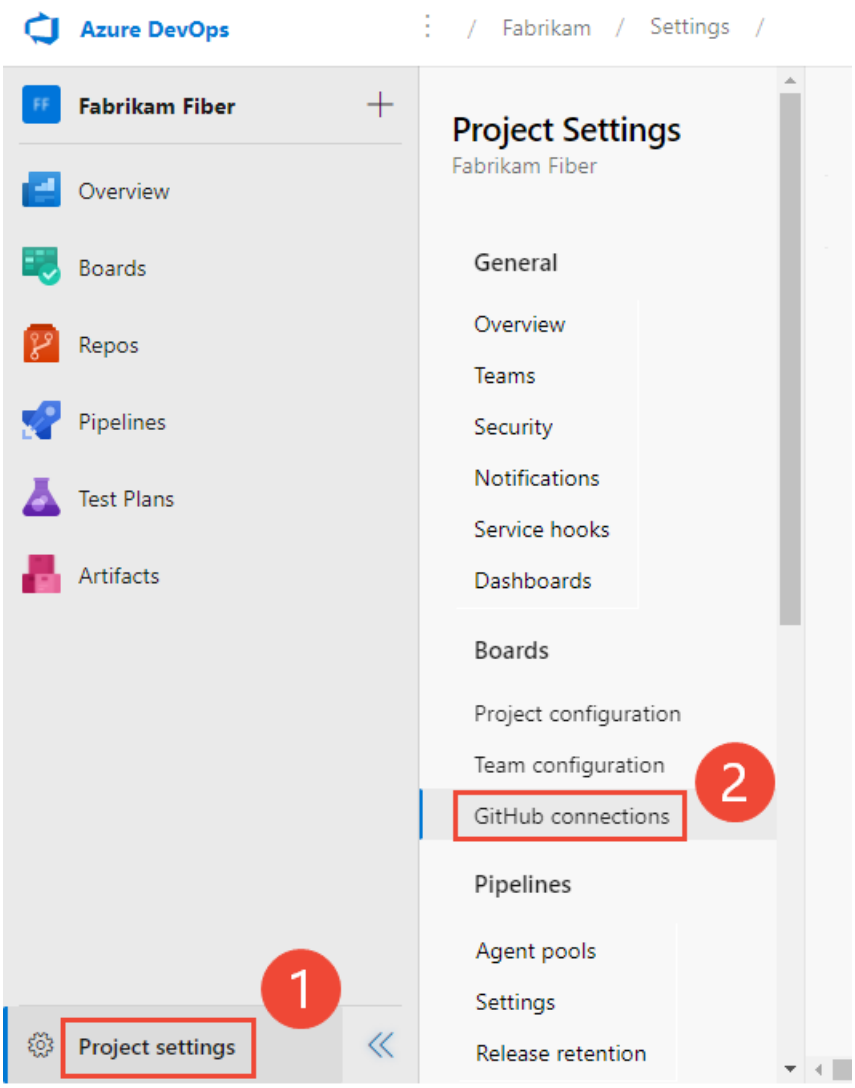
Prerequisites

- To add repositories to a connection, you must be the person who created the connection.
- To remove repositories or remove a connection, you must be the person who created the connection or belong to the **Project Collection Administrators** group.
- To add repositories, you must be an administrator or owner of the GitHub repository you'll connect to. You can connect to multiple GitHub repositories so long as you're an administrator for those repositories.

Open Project Settings>GitHub Connections

1. Sign into the web portal for your Azure Boards project.
2. Choose (1) **Project Settings** > (2) **GitHub connections**.

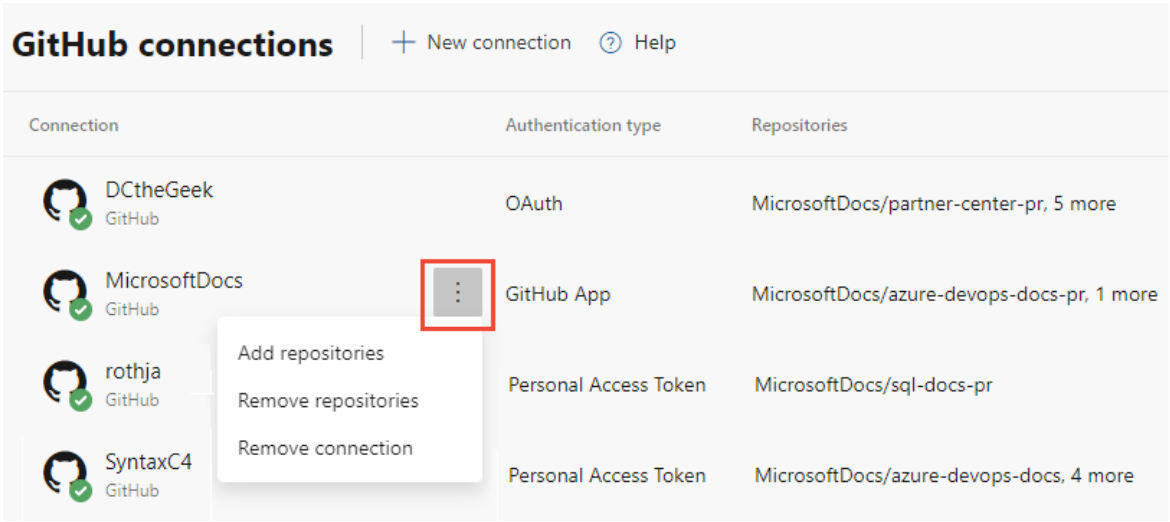

Azure DevOps
Settings
GitHub connections
Fabrikam Fiber
Overview
Boards
Repos
Pipelines
Test Plans
Artifacts
Project settings
1
Project Settings
Fabrikam Fiber
General
Overview
Teams
Permissions
Notifications
Service hooks
Dashboards
Boards
Project configuration
Team configuration
GitHub connections
2
Pipelines
Agent pools
Parallel jobs
Settings



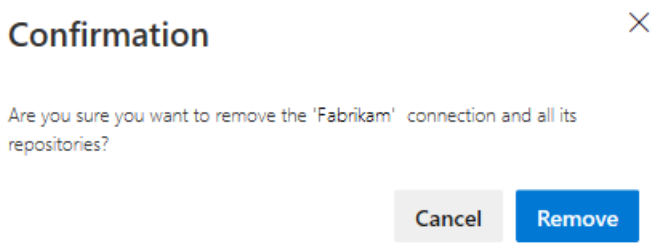
Add or remove repositories, or remove a connection

> **TIP**
> If you encounter a problem with a connection, we recommend that you remove the connection and start over with a new connection.

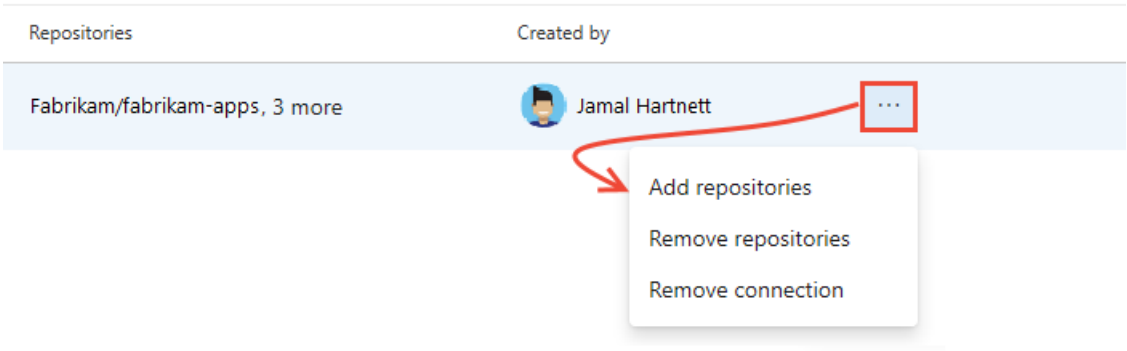
1. To add or remove repositories, choose ⋮ **More options** for the connection and choose **Add repositories** or **Remove repositories** from the menu.



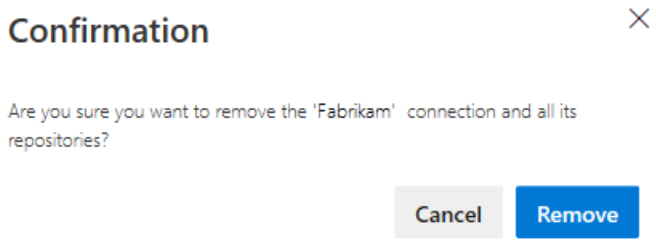
2. To remove all repositories and the connection, choose the **Remove connection** option. Then, choose **Remove** to confirm.



1. To add or remove repositories, open the ··· actions icon for the connection and choose **Add** repositories or **Remove repositories** from the menu.



2. To remove all repositories and the connection, choose the **Remove connection** option. Then, choose **Remove** to confirm.



Try this next

Change GitHub repository access, or suspend or uninstall the integration

Related articles

- Azure Boards-GitHub integration
- Troubleshoot GitHub & Azure Boards integration

Manage GitHub repository access with the Azure Boards app

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Once you install the Azure Boards app for GitHub, you can change the configuration and suspend operations. You can also uninstall the app. To learn more about Azure Boards and GitHub, see GitHub integration overview.

> **NOTE**
> With the Azure Boards app for GitHub, Azure Boards and Azure DevOps Services support integration with GitHub.com and GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

> **NOTE**
> With the Azure Boards app for GitHub, Azure DevOps Servers 2019 and later versions support integration with GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

Prerequisites

The procedures provided in this article only apply when you've installed the Azure Boards app for GitHub. For details, see Install and configure the Azure Boards app for GitHub.

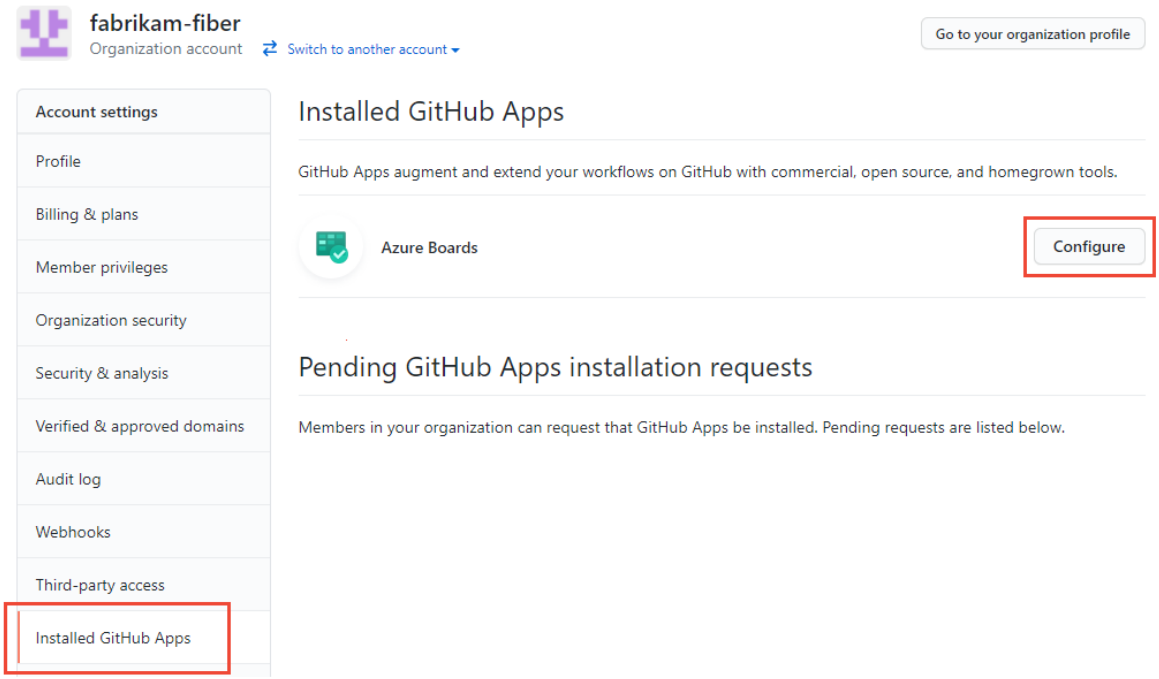
- To manage the Azure Boards integration, you must be the owner or administrator of the GitHub organization. You can connect to multiple GitHub repositories so long as you're an administrator for those repositories.

Change repository access

1. Sign into the web portal for your GitHub organization and open **Account settings**.

   ```
   https://github.com/organizations/fabrikam-fiber/settings/profile
   ```

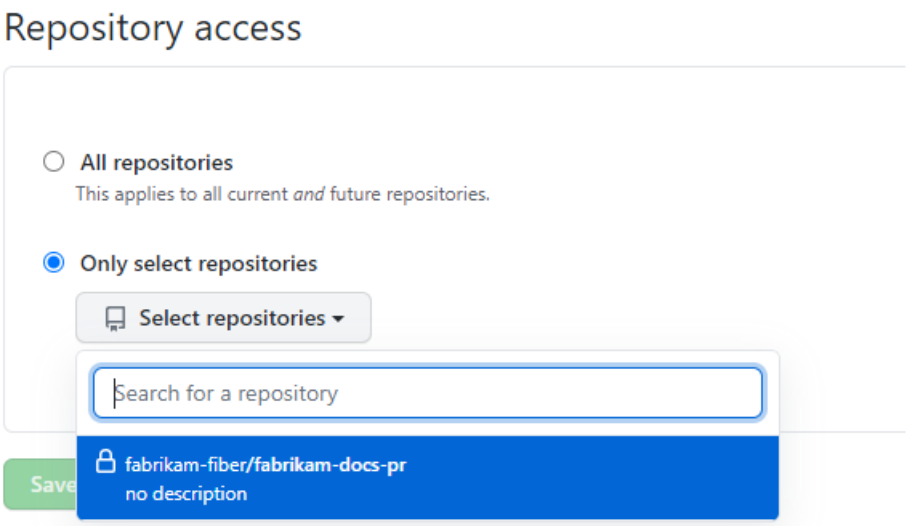
2. Choose **Installed GitHub Apps** and then **Configure** next to **Azure Boards**.



The Azure Boards configuration page opens.

3. Scroll down to the **Repository access** section.
4. Choose the option you want, **All repositories** or **Only select repositories**.

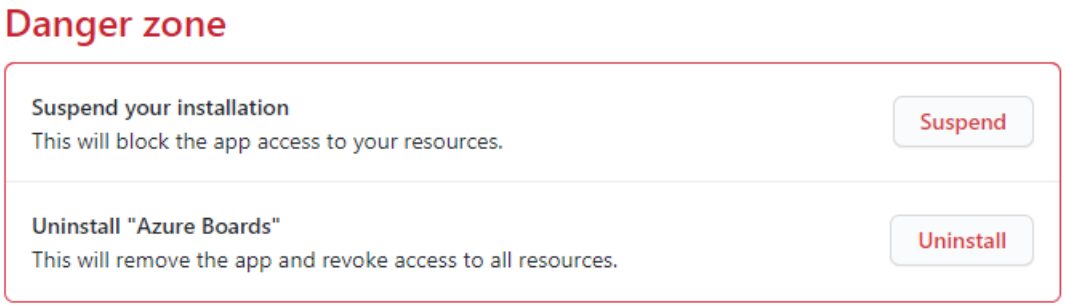
   If you choose **Only select repositories**, select the repositories you want to participate in integration with Azure Boards.



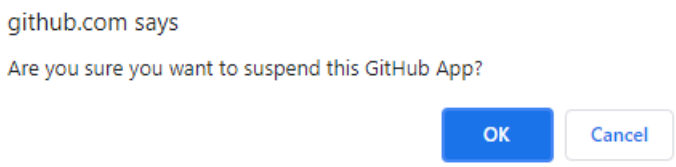
5. Choose **Save** when done.

Suspend or uninstall Azure Boards integration

1. Starting from step 2 in the previous procedure, scroll down to the **Danger zone** section.

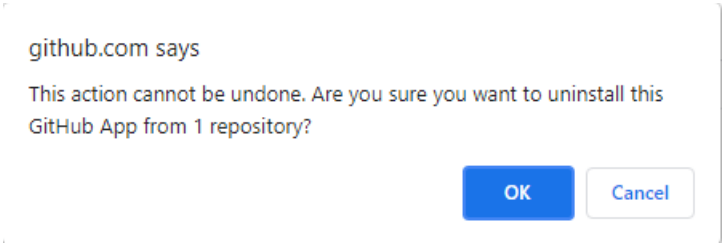


2. To suspend the integration, choose **Suspend**. From the popup confirmation window, choose **OK** to confirm the suspension.



   To unsuspend the integration, choose **Unsuspend**.

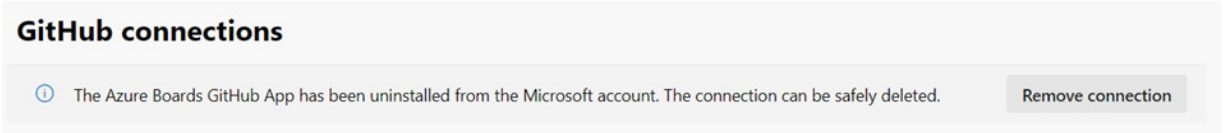
3. To uninstall the Azure Boards app, choose **Uninstall**, and then choose **OK** from the popup confirmation window.



Update Azure Boards' GitHub connections

If you change the repositories that the Azure Boards app for GitHub supports, you may be redirected to Azure Boards GitHub connections. A good practice is to remove the repositories in Azure Boards that can no longer connect to GitHub. For details, see Add or remove GitHub repositories.

If you uninstall the Azure Boards app for GitHub, you'll see the following message in Azure Boards, **Project Settings > GitHub** Connections. Choose **Remove connection** to remove all GitHub connections that have been previously made. See also, Add or remove GitHub repositories.



Try this next

Add or remove GitHub repositories

Related articles

- Connect Azure Boards to GitHub
- Connect Azure DevOps Server to GitHub Enterprise Server
- Troubleshoot GitHub & Azure Boards integration

Troubleshoot an Azure Boards-GitHub integration

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

The Azure Boards-GitHub integration relies on various authentication protocols to support the connection. Changes to a user's permission scope or authentication credentials can cause revocation of the GitHub repositories connected to Azure Boards.

For an overview of the integration that the Azure Boards app for GitHub supports, see Azure Boards-GitHub integration.

Supported authentication options

The following authentication options are supported based on the GitHub platform you want to connect to.

Platform

GitHub.com

GitHub Enterprise Server

Azure DevOps Services

- GitHub.com user account
- Personal access token (PAT)

- OAuth
- PAT
- Username plus password

Azure DevOps Server 2020

Not applicable

- PAT
- Username plus password

Azure DevOps Server 2019

Not applicable

- OAuth
- PAT
- Username plus password

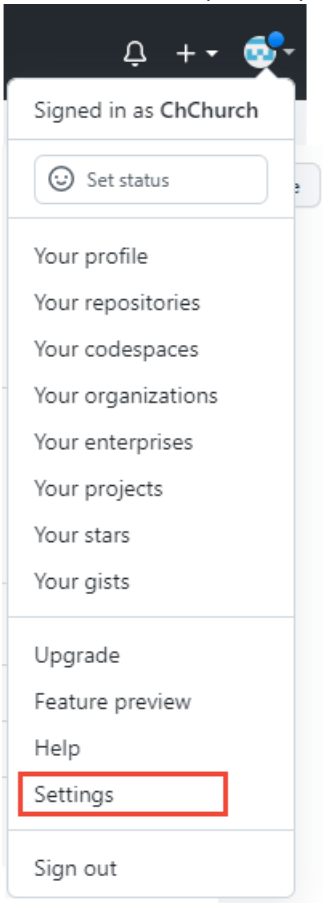
> **NOTE**
> With the Azure Boards app for GitHub, Azure Boards and Azure DevOps Services support integration with GitHub.com and GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

> **NOTE**
> With the Azure Boards app for GitHub, Azure DevOps Servers 2019 and later versions support integration with GitHub Enterprise Server repositories. Integration with other Git repositories is not supported.

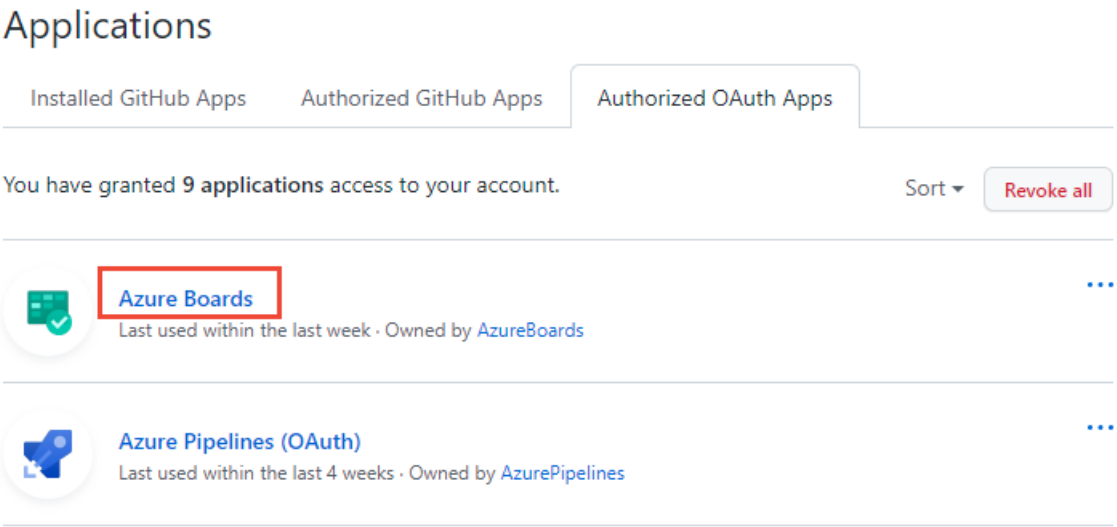
Grant Azure Boards organization access

If the integration between Azure Boards and GitHub isn't working as expected, verify you've granted organization access.

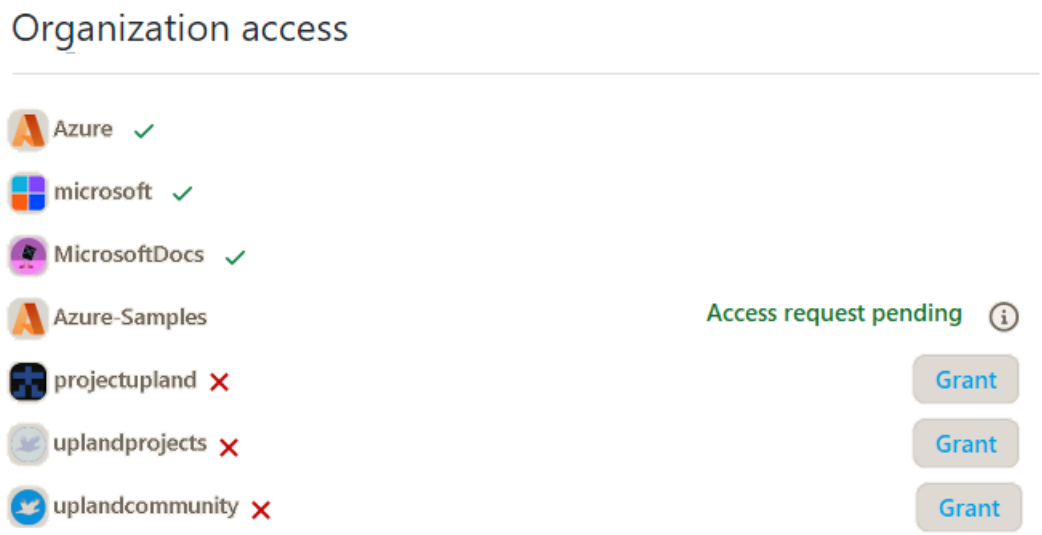
1. From GitHub web portal, open **Settings** from your profile menu.



2. Next, choose **Applications** under **Integrations**.
3. Choose the **Authorized OAuth Apps** tab, and then choose **Azure Boards**.

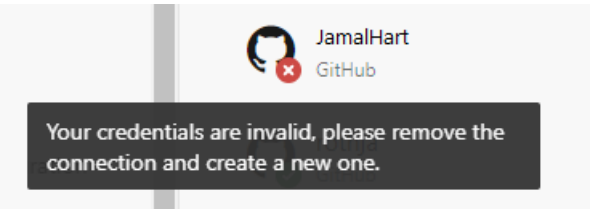


4. Under **Organization access**, resolve any issues that may appear. Choose **Grant** to grant access to any organizations that show as having an **Access request pending**.



Resolve connection issues

When the Azure Boards connection to GitHub no longer has access, it shows an alert status in the user interface with a red-X. Hover over the alert and it indicates that the credentials are no longer valid. To correct the problem, remove the connection and recreate a new connection.



When the Azure Boards connection to GitHub no longer has access, it shows an alert status in the user interface with a red-X that has a tooltip such as, *Unable to connect to GitHub*.

To resolve the problem, consider the following items:

- **If the connection is using OAuth**:
  - The Azure Boards application had its access denied for one of the repositories.
  - GitHub might be unavailable/unreachable. This unavailability could be because of an outage in either service or an infrastructure/network issue on-prem. You can check service status from the following links:
    - GitHub
    - Azure DevOps

    To resolve the first issue, delete and recreate the connection to the GitHub repository. This recreated connection will cause GitHub to prompt to reauthorize Azure Boards.
- **If the connection is using a PAT**:
  - The PAT may have been revoked or the required permission scopes changed and are insufficient.
  - The user may have lost admin permissions on the GitHub repo.

    To resolve, recreate the PAT and ensure the scope for the token includes the required permissions: `repo, read:user, user:email, admin:repo_hook`.

Resolve broken GitHub Enterprise Server connection

If you've migrated from Azure DevOps Server to Azure DevOps Services with an existing GitHub Enterprise Server connection, your existing connection won't work as expected. Work item mentions within GitHub may be delayed or never show up in Azure DevOps Services. This problem occurs because the callback url associated with GitHub is no longer valid.

To resolve the problem, consider the following solutions:

- **Remove and re-create the connection**: Remove and re-create the connection to the GitHub Enterprise Server repository. Follow the sequence of steps provided in Connect from Azure Boards documentation.
- **Fix the webhook url**: Go to GitHub's repository settings page and edit the webhook url to point out to the migrated Azure DevOps Services organization url:
  `https://dev.azure.com/{OrganizationName}/_apis/work/events?api-version=5.2-preview`

Connecting to multiple Azure DevOps organizations

If you connect your GitHub repository to two or more projects that are defined in more than one Azure DevOps organization, such as dev.azure.com/Contoso and dev.azure.com/Fabrikam, you may get unexpected results when using **AB#** mentions to link to work items. This problem occurs because work item IDs aren't unique across Azure DevOps organizations, so **AB#12** can refer to a work item in either the Contoso or Fabrikam organization. So, when a work item is mentioned in a commit message or pull request, both organizations attempt to create a link to a work item with a matching ID (if one exists).

In general, a user intends an **AB#** mention to link to a single work item in one of the projects. However, if a work item of the same ID exists in both accounts, then links are created for both work items, likely causing confusion.

Currently, there's no way to work around this issue, so we recommend that you connect a single GitHub repository only to a single Azure DevOps organization.

> **NOTE**
> When making the connection by using the Azure Boards app for GitHub, the app prevents you from connecting to two different organizations. If a GitHub repository is incorrectly connected to the wrong Azure DevOps organization, you'll need to contact the owner of that organization to remove the connection before you'll be able to add the repository to the correct Azure DevOps organization.

Update XML definitions for select work item types

If your organization uses the Hosted XML or On-premises XML process model to customize the work tracking experience and you want to link to and view the GitHub link types from the Development section in the work item forms, you'll need to update the XML definitions for the work item types.

For example, if you want to link user stories and bugs to GitHub commits and pull requests from the **Development** section, you need to update the XML definitions for user stories and bugs.

Follow the sequence of tasks provided in Hosted XML process model to update the XML definitions. For each work item type, find the `Group Label="Development"` section, and add the following two lines in the following code syntax to support the external links types: **GitHub Commit** and **GitHub Pull Request**.

```
            <ExternalLinkFilter Type="GitHub Pull Request" />
            <ExternalLinkFilter Type="GitHub Commit" />
```

When updated, the section should appear as shown.

```
<Group Label="Development">
   <Control Type="LinksControl" Name="Development">
      <LinksControlOptions ViewMode="Dynamic" ZeroDataExperience="Development" ShowCallToAction="true">
         <ListViewOptions GroupLinks="false">
         </ListViewOptions>
         <LinkFilters>
             <ExternalLinkFilter Type="Build" />
             <ExternalLinkFilter Type="Integrated in build" />
             <ExternalLinkFilter Type="Pull Request" />
             <ExternalLinkFilter Type="Branch" />
             <ExternalLinkFilter Type="Fixed in Commit" />
             <ExternalLinkFilter Type="Fixed in Changeset" />
             <ExternalLinkFilter Type="Source Code File" />
             <ExternalLinkFilter Type="Found in build" />
             <ExternalLinkFilter Type="GitHub Pull Request" />
             <ExternalLinkFilter Type="GitHub Commit" />
         </LinkFilters>
      </LinksControlOptions>
   </Control>
</Group>
```

Related articles

- Add or remove GitHub repositories
- Change repository access to Azure Boards

Track user stories, issues, bugs, and other work items in Azure Boards

10/4/2022 • 15 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Track the features and requirements you're developing, code defects or bugs, and other details using work items. Work items are similar to GitHub issues, but offer different types to track various types of information.

If you're just getting started, read the information provided in this article. To jump right in and start tracking work on a Kanban board, see Plan and track work. For a quick reference to various work item tasks and key concepts, see Work item quick reference.

Plan and track work

Use work items to track anything you need to track. Each work item represents an object stored in the work item data store. Each work item is based on a work item type. And work item types have a unique identifier within an organization or project collection. The available work item types depend on the process you used when creating your project (Agile, Basic, Scrum, or CMMI).

Use work items to track anything you need to track. Each work item represents an object stored in the work item data store. Each work item is based on a work item type. And work item types have a unique identifier within an organization or project collection. The available work item types depend on the process you used when your project was created (Agile, Scrum, or CMMI).

Tasks you can do using work items

- Use different work item types (WITs) to track different types of information. Specific tools include:
  - Add backlog items, such as Issues (Basic process), User Stories (Agile), Product Backlog Items (Scrum), Requirements (CMMI)
  - Define Features and Epics
  - Define, triage, and manage Bugs
  - Add Tasks to backlog items
- Update the work item form to add information, update status, reassign another project member or sprint, link work items, attach files, and add comments
- Specify how bugs should be tracked, either as requirements or as tasks
- Add and modify work items using the web portal and other supported clients
- Assign a work item to one and only one project member
- Assign work items to a sprint via the iteration path
- Link work items to other work items or Azure DevOps objects
- Run impromptu search or queries to find or list work items
- Capture and apply work item templates to quickly fill in work item field
- Add and customize work item types
- Modify work items

Required permissions and access

As a member added to the Contributors group of a project, you can use most features provided under **Boards** or **Work**. Users with Basic access have full access to all features. Users with Stakeholder access are limited to

certain features. For details, see Stakeholder access quick reference.

To learn more about permissions and access, see Permissions and access for work tracking and Stakeholder access quick reference.

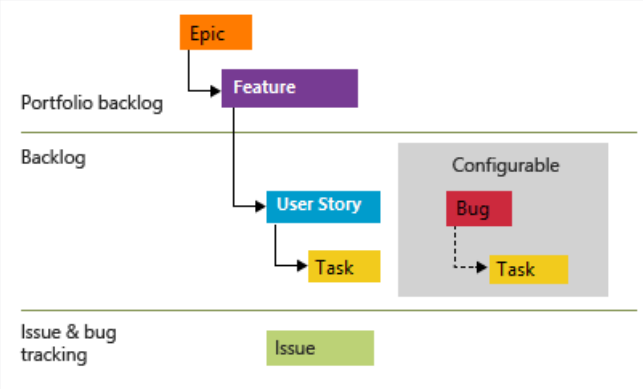
To add users to a project, see Add users to a project or team.

Work item types (WITs)

To track different types of work, different WITs are defined. The work item types available are based on the process used when your project was created—Agile, Basic, Scrum, or CMMI—as illustrated in the following images. The items in your backlog might be called user stories (agile), issues (Basic), product backlog items (Scrum), or requirements (CMMI). All four types are similar: they describe the customer value of the work and the work to do.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

Each work item type belongs to a category. Categories are used to group work item types and determine which types appear on backlogs and boards.

CATEGORY	WORK ITEM TYPE	CONTROLS BACKLOGS/BOARDS
Epic	Epic	Epic portfolio backlogs and boards
Feature	Feature	Feature portfolio backlogs and boards
Requirement	User Story (Agile) Issue (Basic) Product Backlog Item (Scrum) Requirement (CMMI)	Product backlogs and boards and Sprints backlog
Task	Task	Sprints Taskboards
Bug	Bug	Dependent on how bugs are tracked

Along with the work items types that appear on backlogs and boards, there are work item types that track testing, reviews, and feedback. These types, listed in the following table by category, are available for all processes.

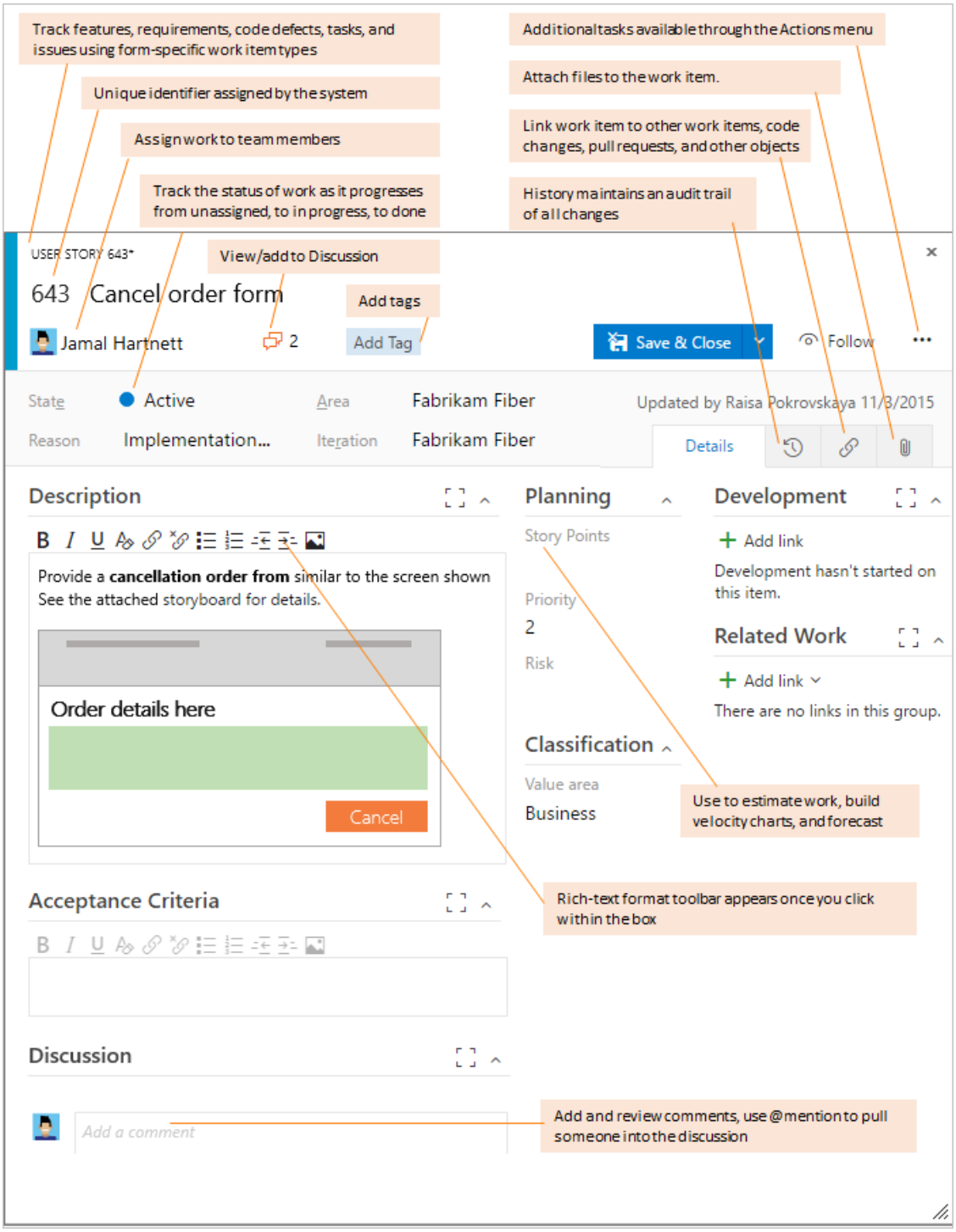
Category	Work item type	Used to track specified types of work
Code Review Request	Code Review Request	Tracks a code review request against code maintained in a Team Foundation version control (TFVC) repository. To learn more, see Day in the life of a Developer: Suspend work, fix a bug, and conduct a code review.
Code Review Response	Code Review Response	A code review response is created for each person who's requested to provide review comments.
Feedback Request		Feedback requests track requests for feedback generated through the feedback request form. See Get feedback.
Feedback Response	Feedback Response	A feedback response is created for each person and for each item for which feedback is provided through the Microsoft Feedback Client. See Get feedback.
Shared Step	Shared Step	Shared steps are used to repeat tests with different data.
Shared Parameter		Shared Parameters specify different data and parameters for running manual test cases. See Repeat a test with different data.
Test Case	Test Case	Each test case defines a manual test.
Test Plan	Test Plan	Test plan group test suites and individual test cases together. Test plans include static test suites, requirement-based suites, and query-based suites.To learn more, see Create test plans and test suites.
Test Suite	Test Suite	Test suites group test cases into separate testing scenarios within a single test plan. Grouping test cases makes it easier to see which scenarios are complete. See Create test plans and test suites.

Ideally, work items are only created from the specific tool designed to support their usage. So, to prevent users from creating work items manually, there's a Hidden Types category. All of the categories listed in the previous table are added to the Hidden Types category except for the Test Case category.

Work item form

Each work item supports tracking data contained in work item fields. Also, it captures changes as updates are made within the **History** field and comments made in the **Discussion** section. To learn more about each field, see Work item field index.

Each form contains multiple controls, as shown below and described in Work item form controls.



Track work in the web portal

You can add and update work items from the web portal. To track work using other clients, see Best tools for

adding, updating, and linking work items.

Web portal and clients that support tracking work items

You can add and update work items from the web portal and various clients. For an overview of all clients that connect to your project, see Tools and clients that connect to Azure DevOps.

Web portal

Use the web portal to accomplish the following tasks.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.
- **Backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
- **Sprints**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others, performing bulk updates, or import/export operations.
- **Delivery Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.
- **Backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
- **Sprints**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others, performing bulk updates, or import/export operations.
- **Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view. Requires installation of the Delivery Plans extension.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.
- **Backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
- **Sprints**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others, performing bulk updates, or export operations.
- **Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view. Requires installation of the Delivery Plans extension.

- **Work items**: Use to quickly find work items assigned to you or pivot or filter work items based on other criteria, such as work items that you follow, that you're mentioned in, or that you viewed or updated.
- **Backlogs**: Which provide access to:
  - **Product** and **Portfolio backlogs**: Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs.
  - **Boards**: Use to implement Kanban practices and visualize the flow of work for a team.

    - **Sprint backlogs**: Use to plan work for a team to perform during a specific time frame referred to as a sprint.
- **Queries**: Use to define a set of filter criteria to list work items for the purposes of sharing with others or performing bulk updates.
- **Plans**: Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined that are assigned to sprints (iteration path) of selected teams against a calendar view. Requires installation of the Delivery Plans extension.

Track bugs as requirements or tasks

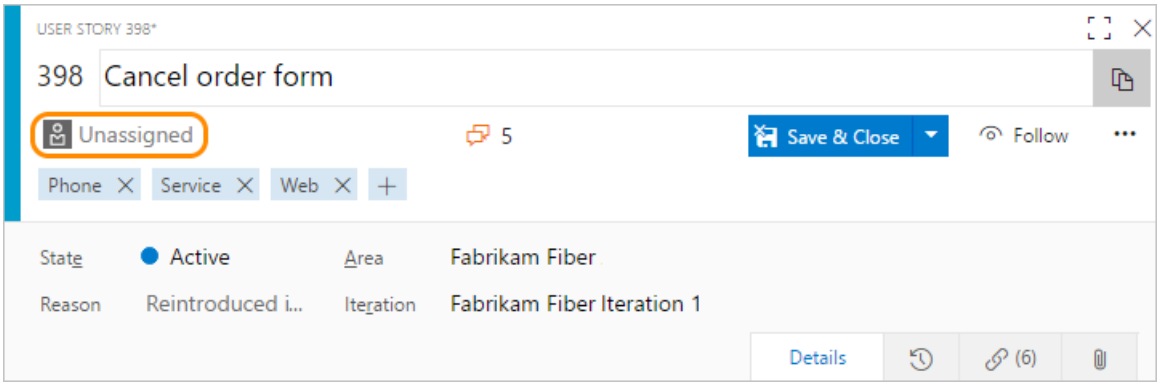
Many Scrum teams treat bugs the same as any backlog item or user story. Others see bugs as work that belongs to implementing a story, and then treat them as a task. Bugs, like product backlog items (PBIs) and user stories, represent work that needs doing. So, should you track your bugs along with other items in the product backlog items? Or, should you track your bugs as tasks linked to those backlog items? How does your team estimate work?

Based on how your team answers these questions, they can choose how they want to track bugs from one of these three choices. To change the team setting, see Show bugs on backlogs and boards.

For an overview of all team settings, see Manage teams and configure team tools.

Assign work items to a project member

You can only assign a work item to one person at a time. The **Assigned To** field is a person-name field designed to hold a user identity recognizable by the system. Within the work item form, choose the **Assigned To** field to select a project member. Or, you can begin typing the name of a project member to quickly focus your search to a select few.



Anyone who has write access to a project can assign work items, including users with Basic and Stakeholder access.

Note the following:

- You can assign a work item only to users that have been added to a project or team
- You can assign a work item to one and only one user at a time. If work is split across two or more users, consider creating separate work items that you'll assign to each user responsible for the work to complete
- Over time, the drop-down menu of person-name fields will display the names you have most recently selected
- Some drop-down menus that support assignment from a team backlog or board are automatically limited to users assigned to the team
- The system shows the display name and adds the user name when required to disambiguate identical display names

- You can assign several work items at once from the backlog or query results, see Bulk modify work items for details.

Integration with Azure Active Directory

When your system is configured with Azure Active Directory (Azure AD), then the system will synchronize person-name fields with these directories. Person-name fields include Activated By, Assigned To, Closed By, Created By, and Resolved By.

You can grant access to a project by adding security groups that you created in Azure AD or by adding accounts to existing or custom groups defined from the collection setting **Security** pages. For more information, see Add or delete users using Azure Active Directory.

Integration with Active Directory

When Azure DevOps Server is configured with Active Directory (AD), Azure DevOps synchronizes person-name fields with these directories. Person-name fields include Activated By, Assigned To, Closed By, Created By, and Resolved By.

You can grant access to a project by adding security groups you create in AD or by adding accounts to existing or custom groups defined in the collection setting **Security** pages. To learn more, see Set up groups for use in Azure DevOps Server deployments.

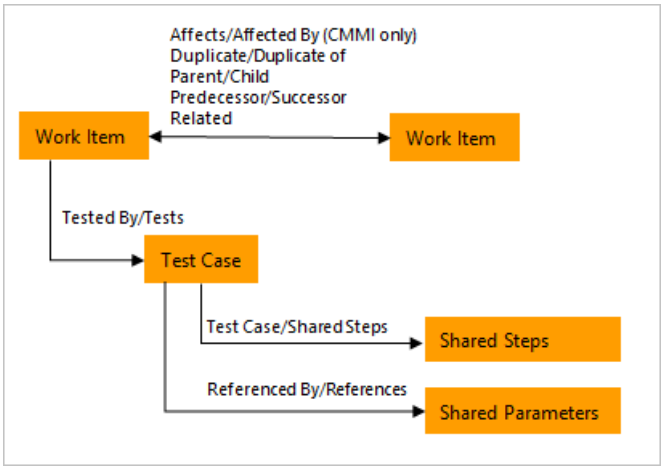
> **NOTE**
> To minimize the list of names that appear in the drop-down menus of person-name fields, you can scope the field to only those groups that you want to appear in the menu. You do this by adding one or more of the following child elements to the **FIELD** definition in the work item type definition: **ALLOWEDVALUES**, **PROHIBITEDVALUES**, and **VALIDUSER**. For details, see Define pick lists.

Assign work items to a sprint

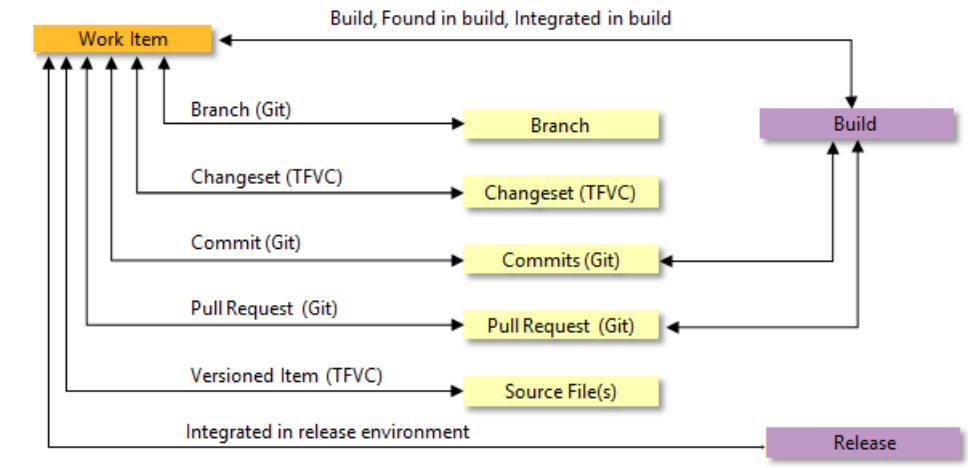
To schedule work items to be worked on during a specific time period, you assign the **Iteration Path**. First, you define the Iteration Paths for use in the project, and then each team selects the Iteration Paths that they'll use. To learn more, see Assign work to sprints.

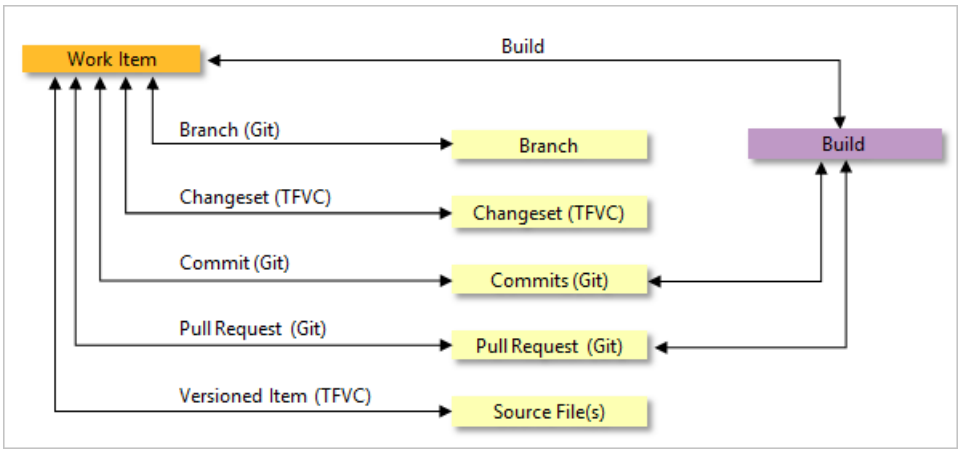
Link work items to other work or objects

As shown in the following image, you can link work items to other work items. Depending on the work item type, the link type differs. Also, special link types support a hierarchy of work items using parent-child links and predecessor-successor links to track dependencies.



Also, several tools support linking to other objects, such as builds, releases, commits, pull requests, and more. The following image demonstrates.





For a complete list of link types and supported features, see Linking, traceability, and managing dependencies and Link type reference.

Find or list work items

Use the search box to do an impromptu search for finding specific work items based on select field criteria. Or, create a query to do a managed search, which lists work items based on your query criteria. With managed searches, you can do other tasks, such as a work items triage, creating a trend or status chart, adding to the dashboard, and more.

To learn more, see these articles:

- About managed queries
- View, run, or email a query
- About managed queries
- Work item query charts

Use work item templates to quickly fill in forms

With work item templates, you can quickly create work items that have pre-populated values for your team's commonly used fields. For example, create a task template that sets the area path, iteration path, and discipline or activity whenever you use it to create a task.

To learn more, see Use templates to add and update work items.

Once you have a template defined, you can share it via email or a dashboard.

Customize a work item type

You can add or modify the fields contained within a work item type or add a custom work item type. To learn more, see Customize an inheritance process.

You can add or modify the fields contained within a work item type or add a custom work item type. To learn more, see:

- For project collections that use the Inheritance process model: Customize an inheritance process.
- For project collections that use the On-premises XML process model: Customize the On-premises XML process model.

You can add or modify the fields contained within a work item type or add a custom WIT. To learn more, see Customize the On-premises XML process model.

Try this next

Add a work item

Related articles

- Web portal navigation
- Backlogs, portfolios, and Agile project management
- About Kanban and Agile project management
- Keyboard shortcuts
- Agile, Scrum, and CMMI processes
- Work item field index
- Use categories to group work item types

Key concepts and work item tasks in Azure Boards and Azure DevOps

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use this index to quickly access concepts related to work items and information on adding and updating work items—such as users stories, features, tasks, and bugs.

Key concepts

- Agile glossary
- Agile process
- Area Paths
- Autocomplete work items
- Assigned to
- Basic process

- Chart for work items widget
- Charts
- Client tools
- CMMI process
- Customization process models
- Dependencies
- Delivery plans

- Filtering
- Following
- Inheritance process model
- Iteration Paths
- Keyboard shortcuts

- Link types
- Linking and traceability
- Mobile browser
- New Boards Hub
- New work item widget
- On-premises XML process model
- Permissions and access
- Process guidance
- Process models

- Queries
- Recycle bin

- Remote linking
- Rich text fields
- Rollup
- Scrum process
- State categories

- Queries
- Recycle bin
- Rich text fields
- Rollup
- Scrum process
- State categories

- Queries
- Recycle bin
- Requirements
- Rich text fields
- Rollup
- Scrum process
- State categories

- Tags
- Track bugs as requirements or tasks
- Track dependencies

- Visual Studio work item experience

- Work item fields
- Work item form
- Work item templates
- Work item types
- Work tracking limits
- Workflow

Work item user tasks

Tasks listed below are available to users with Contributor permissions and Basic access.

- Add a work item
- Add Epics
- Add Features
- Add items to a backlog
- Add items to a Kanban board
- Add tags
- Add tasks
- Add to discussion
- Apply a template to a work item
- Assign work to a team member

- Attach files

- Bulk add or remove tags
- Bulk modify work items (Excel)
- Bulk modify work items (Web)

- Capture work item as a template
- Change work item type
- Copy or clone a work item
- Copy work item URL
- Copy list of work items
- Create a branch
- Create a work tracking chart

- Define a work item template
- Define work item URL
- Delete work item tags
- Delete work items
- Display rollup

- Define a work item template
- Define work item URL
- Delete work item tags
- Delete work items

- Email work item list
- Export a work item list
- Filter a backlog, board, or plan
- Filter the History tab
- Follow a work item
- Forecast work items
- Get notified of work item changes
- Group work items

- Link to cross-organization work items
- Link to development objects
- Link to GitHub commits and pull requests
- Link to work items from a wiki
- Link work items
- List work items
- List work items in a wiki

- Manage bugs
- Manage issues or impediments
- Manage work item tags
- Map work items
- Move work items to a sprint
- Move work items to another project

- Open work items
- Print work items
- Prioritize backlog items
- Query work item history
- Query for work items

- Reassign work items
- Remove work items
- Request feedback
- Restore deleted work items

- Start storyboarding
- Track dependencies
- Triage work items

- Update status of tasks (Taskboard)
- Update status of work items (Kanban board)
- Use #ID to link
- Use @mentions

- View history
- View work items (mobile)
- View work items (web)
- View work assigned to me
- View work I'm following
- View work I've recently viewed or updated
- View work recently completed
- View work recently created
- View work where I'm mentioned

- View history
- View work items (mobile)
- View work items (web)

Administrative customization tasks

Tasks listed below must be performed by an administrator who has the necessary permissions, as they affect all users and teams within a project.

You customize work item types using the Inheritance process model.

You customize work item types using either the Inheritance process model or On-premises XML process model, depending on the selection made when you created the project collection on which the project is defined.

Inherited process model

- Add a checkbox (Boolean) field
- Add a custom field
- Add a custom work item type

- Add/remove custom fields
- Add/remove custom groups
- Add/remove custom pages
- Add/remove a custom control
- Add/remove custom rules to a field
- Add a person-name/Identity
- Add a picklist (drop-down menu)
- Add a rich-text (HTML) field
- Add, edit, or remove a WIT workflow state

- Change a field label
- Change the WIT color or description
- Change the reference process from Agile to Scrum
- Change the reference process from Basic to Agile
- Change the reference process from Scrum to Agile
- Create a project

- Define Area Paths
- Define Iteration Paths
- Delete field
- Delete a WIT
- Enable/disable a WIT

- Modify a default pick list
- Move the field within the layout
- Remove a field from form
- Restrict modification
- Set Required/Default options

- Add a custom field
- Add a custom work item type (WIT)
- Add/remove custom fields from a WIT
- Add/remove custom groups on a WIT form
- Add/remove custom pages on a WIT form
- Add/remove a custom control on a WIT form
- Add/remove custom rules to a field
- Add a person-name/Identity
- Add a picklist (drop-down menu)
- Add a rich-text (HTML) field
- Add, edit, or remove a WIT workflow state

- Change a field label
- Change the WIT color or description
- Create a project

- Define Area Paths
- Define Iteration Paths

- Delete field
- Delete a WIT

- Enable/disable a WIT
- Modify a default pick list
- Move the field within the layout

- Remove a field from a form
- Restrict modification
- Set Required/Default options

On-premises XML process model

You customize work item types using the On-premises XML process model. For more customization options, see On-premises XML process customization.

- Add a checkbox (Boolean) field (TFS 2017.2)
- Add a custom field
- Add a custom pick list
- Add or modify a field
- Add rules to a field
- Add a custom control field
- Add fields that integrate with test, build, and version control
- Add/remove State or Reason fields (customize workflow)

- Change a field attribute
- Change a field label
- Define Area Paths
- Define Iteration Paths
- Define global lists
- Delete a field
- Modify a predefined pick list
- Remove a field from a form

Related articles

- Query quick reference
- Work item field index
- Quick guide to default permissions and access
- Best tools for adding, updating, and linking work items

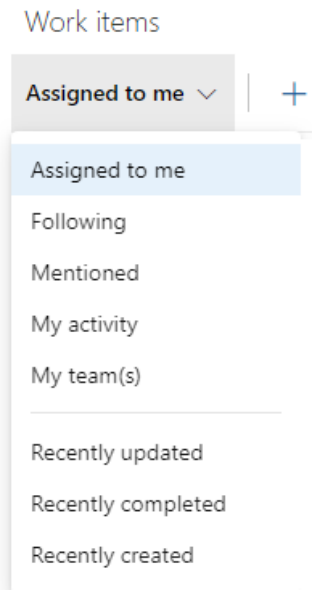
View and add work items using the Work Items page

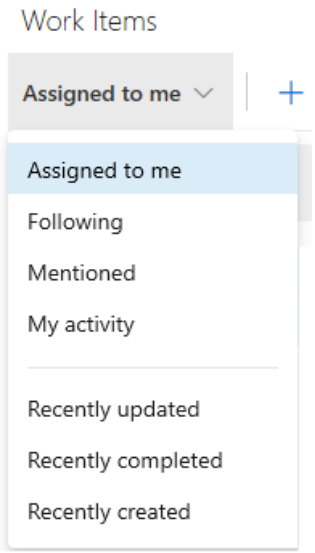
10/4/2022 • 12 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Visual Studio 2019 | Visual Studio 2022

View work items that you created or are assigned to you. The **Work Items** page provides several personalized pivots, as shown in the following image, and interactive filter functions to streamline listing work items. Use this page to quickly find work items defined across teams within a project.





NOTE

The **Work Items** page is available from Azure DevOps Services, Azure DevOps Server 2019 and later versions, and Visual Studio 2019 RC1.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

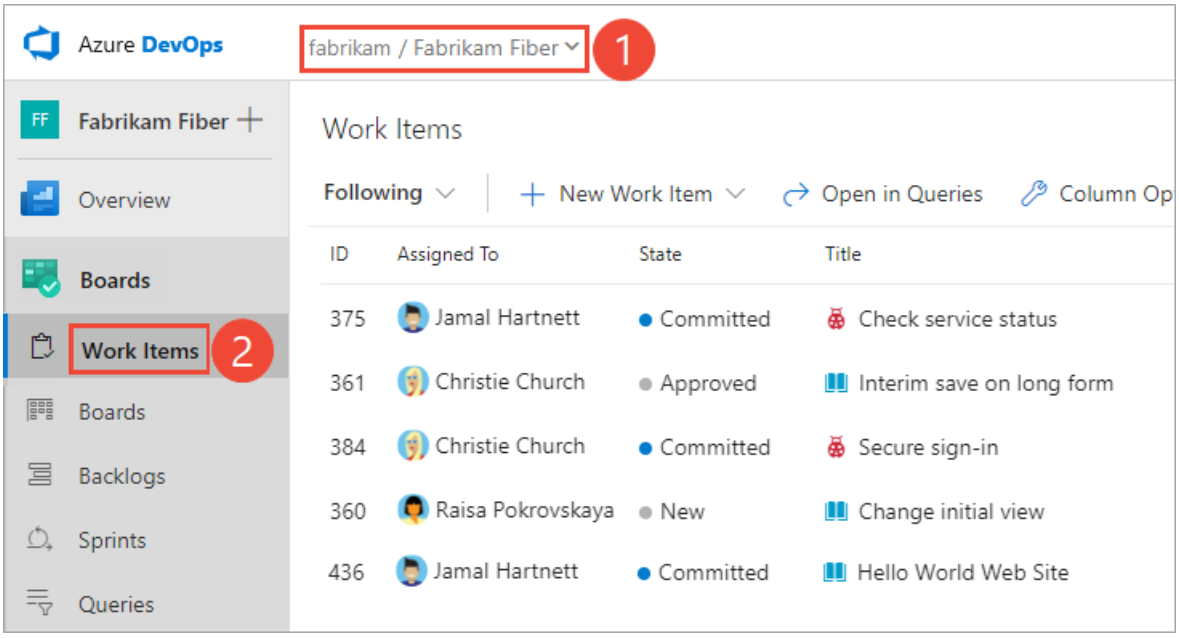
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

Open Work Items

You can start viewing and adding work items once you connect to a project.

- Web portal
- Visual Studio 2019
- Azure DevOps CLI

(1) Check that you've selected the right project, then (2) choose **Boards>Work Items**.



NOTE

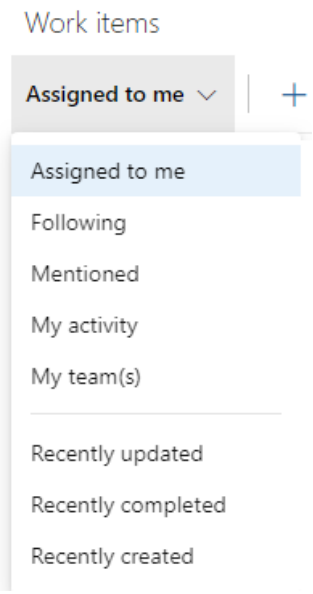
Depending on the process chosen when the project was created—Agile, Basic, Scrum, or CMMI—the types of work items you can create differ. For example, backlog items may be called user stories (Agile), issues (Basic), product backlog items (Scrum), or requirements (CMMI). All three are similar: they describe the customer value to deliver and the work to be performed.

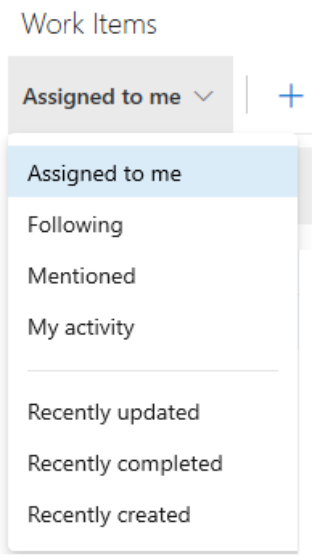
For an overview of these processes, see Choose a process.

View work items

Using the drop-down menu, you can focus on relevant items inside a project using one of seven pivots. Additionally, you can filter and sort each pivot view. You can also use an Azure DevOps CLI command to view details about a work item.

- Web portal
- Visual Studio 2019
- Azure DevOps CLI





- **Assigned to me**: lists all work items assigned to you in the project in the order they were last updated. Doesn't include items moved to the **Removed** category state. To open or update a work item, simply click its title.

- **Following**: lists work items that you're following.
- **Mentioned**: lists work items in which you've been mentioned in the last 30 days.
- **My activity**: lists work items that you've recently viewed or updated.
- **My team(s)**: lists work items that your team members have recently viewed or updated.

- **Assigned to me**: lists all work items assigned to you in the project in the order they were last updated. To open or update a work item, simply click its title.
- **Following**: lists work items that you're following.
- **Mentioned**: lists work items in which you've been mentioned in the last 30 days.
- **My activity**: lists work items that you've recently viewed or updated.
- **My team(s)**: lists work items that your team members have recently viewed or updated.

- **Assigned to me**: lists all work items assigned to you in the project in the order they were last updated. To open or update a work item, simply click its title.
- **Following**: lists work items that you're following.
- **Mentioned**: lists work items in which you've been mentioned in the last 30 days.
- **My activity**: lists work items that you've recently viewed or updated.

- **Recently updated**: lists work items recently updated in the project.
- **Recently completed**: lists work items completed or closed in the project.
- **Recently created**: lists work items created within the last 30 days in the project.

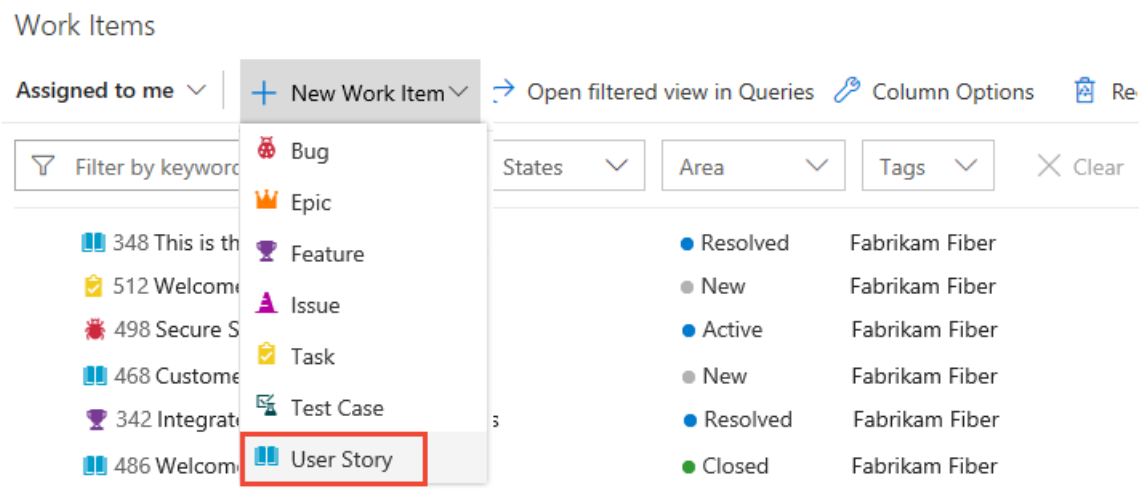
Add a work item

Adding a work item is just one click away. Choose the work item type from the **New Work Item** drop down menu. You can also use an Azure DevOps CLI command to add a new work item.

> **NOTE**
> New work items are assigned the last **Area Path** and **Iteration Path** selected by the user.

- Web portal
- Visual Studio 2019
- Azure DevOps CLI

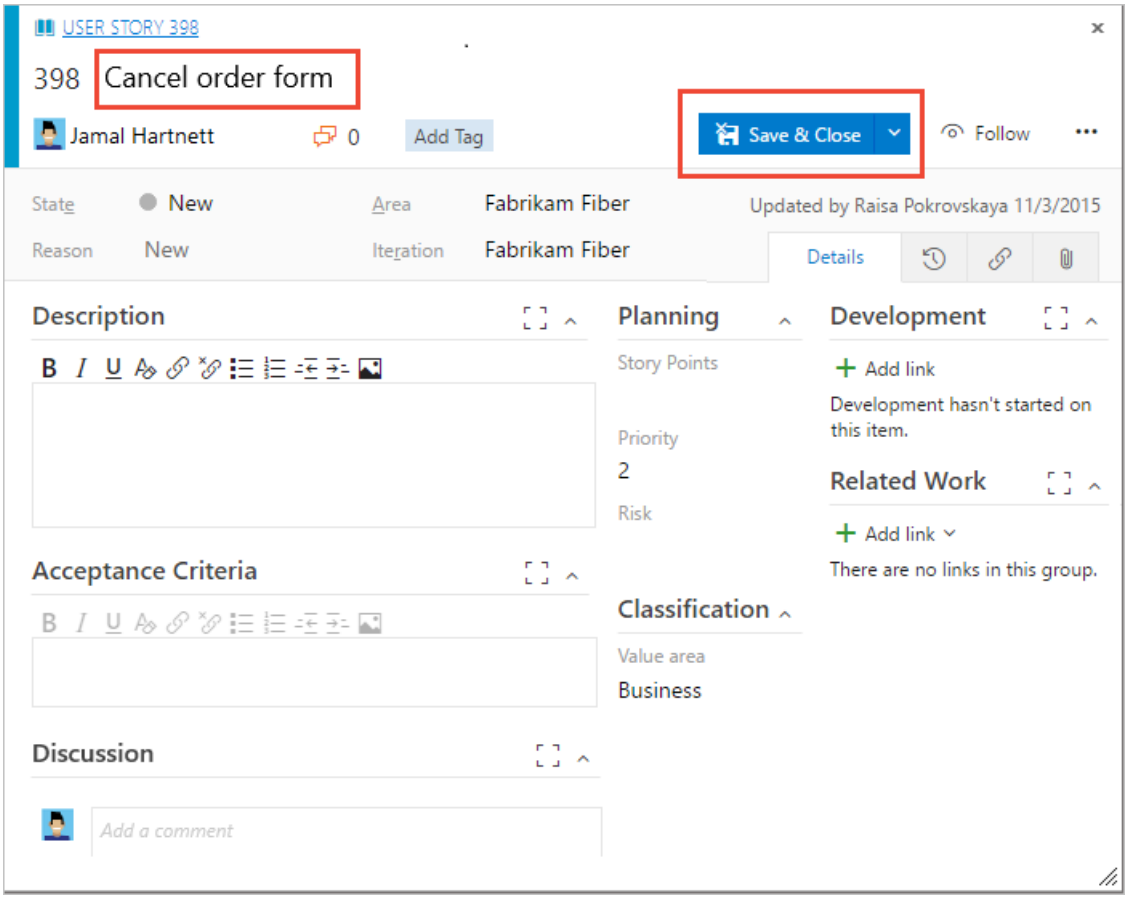
For example, here we choose User Story.



TIP

Work items you add are automatically scoped to the currently selected team's area and iteration paths. To change the team context, see Switch project or team focus.

Enter a title and then save the work item. Before you can change the State from its initial default, you must save it.

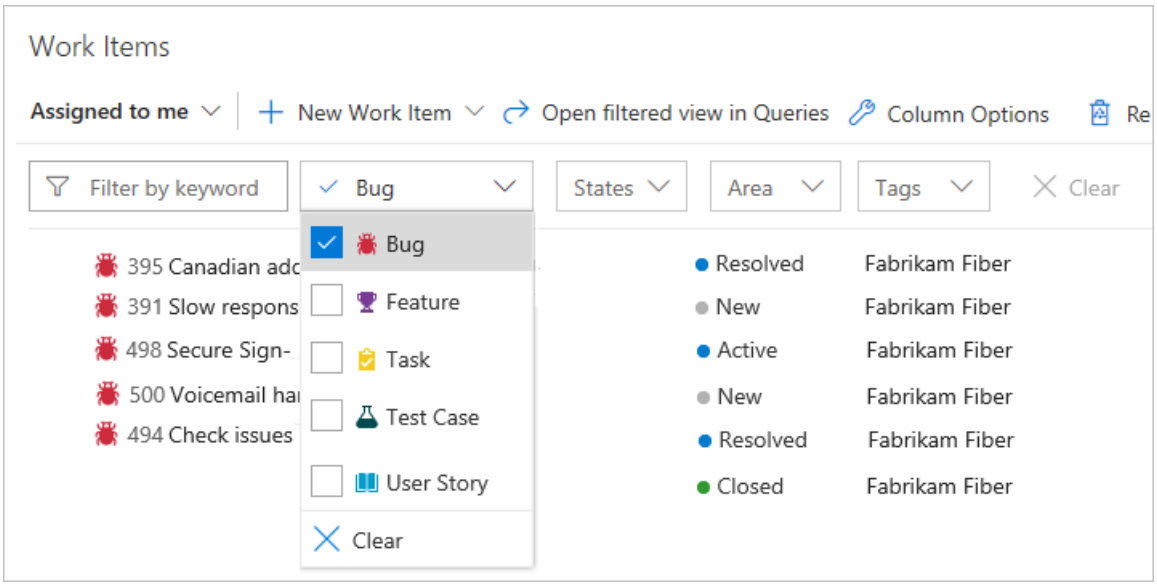


You can add tags to any work item to filter backlogs, queries, and work item lists. Users with **Basic** access can create new tags by default, users with **Stakeholder** access can only add existing tags.

Filter to create personal views

You can filter each work item pivot view by typing a keyword or using one or more of the fields provided, such as work item type (Types), State, Area Path, and Tags. The page remembers the filters you set for each pivot, supporting personalized views across all pivots. To learn more about filtering, see Filter backlogs, boards, queries, and plans.

- Web portal
- Visual Studio 2019
- Azure DevOps CLI

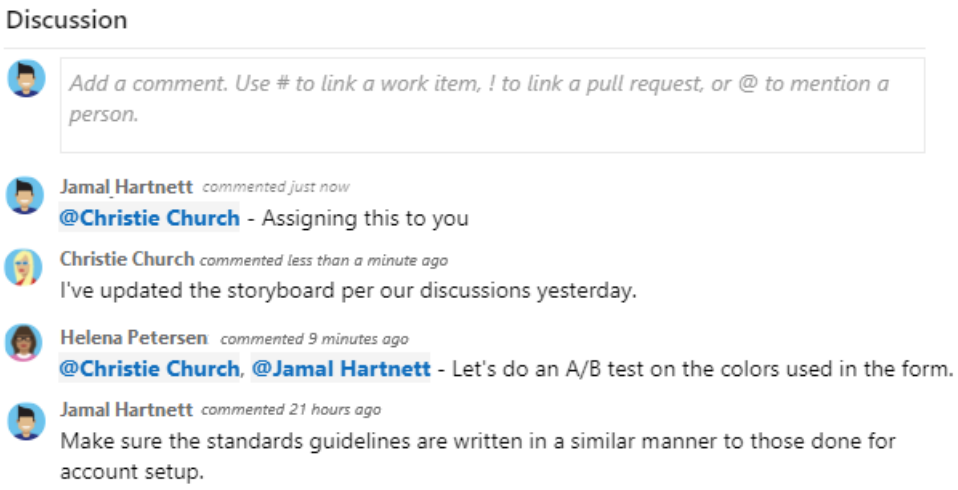


Add columns and sort by a column

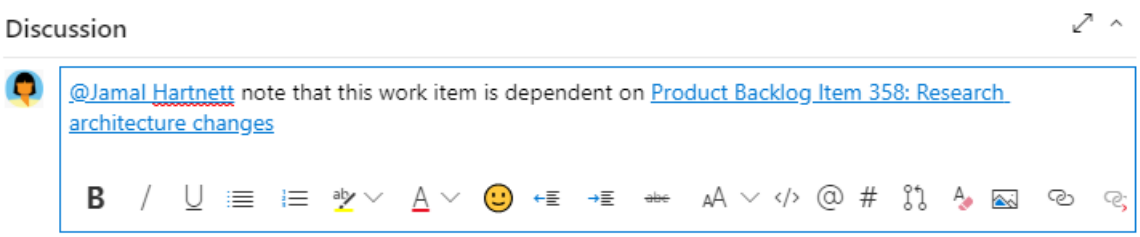
From the web portal, you can sort your view by one of the column fields that you select from the **Column Options** dialog. For details, see Change column options.

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



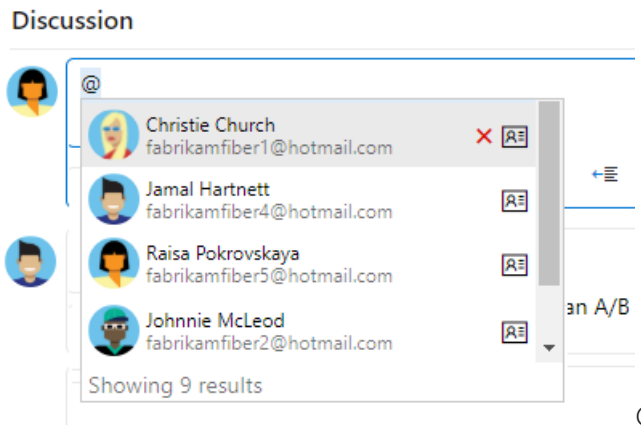
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the History field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons — @ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.



@mention drop-down menu people-picker." />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon and then choose **Delete**.

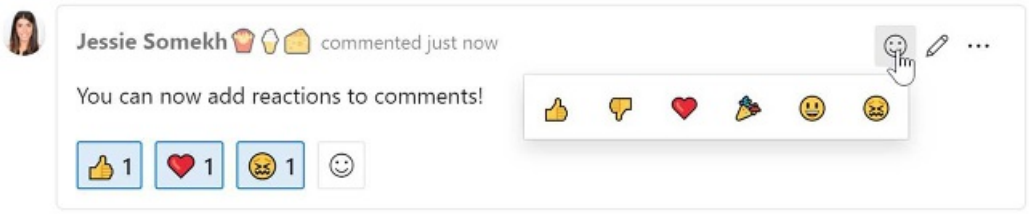


> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it. A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

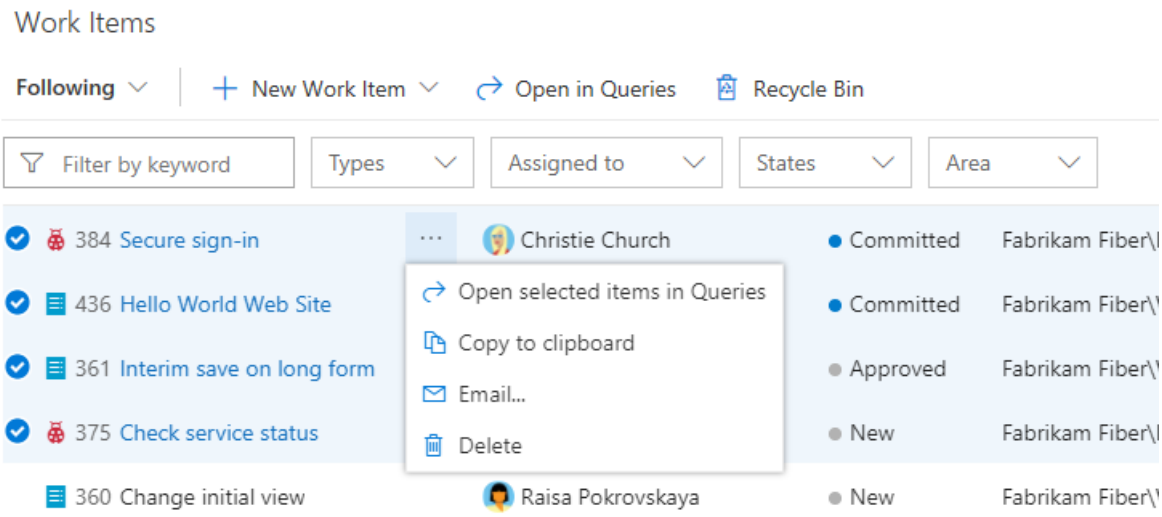
Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



Copy selected items to the clipboard or email them

To select several items in a sequence, hold down the shift key from a web portal page. To select several non-sequential items, use the **Ctrl** key. Then, you can use **Ctrl+c** to copy the selected items to a clipboard. Or, you can open the context menu for the selected work items, click (··· actions icon), and then select an option from the menu.



Open a view as a query

From the web portal, you can open any view, filtered view, or selected set of work items as a query. Choose **Open in Queries** or the **Open selected items in Queries** option from the context menu.

Queries provide other features that you can use, including:

- Edit one or more fields of several work items
- Add or remove tags from several work items
- Change the work item type
- Delete work items
- Apply work item templates
- And more

For details, see Bulk modify work items. To learn more about queries, see Use the query editor to list and manage queries and Query fields, operators, and macros.

Work Items page controls

Use the following three controls to manage your views in the web portal.

CONTROL	FUNCTION
	View/hide completed items
	Turn filtering On/Off
	Enter or exit full screen mode

Related articles

- Azure Boards FAQs
- Best tool to add, update, and link work items
- Move, change, or delete work items (Recycle Bin)
- Manage or enable features
- Use work item form controls
- Keyboard shortcuts
- Work across projects

Add and update a work item

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You add work items to plan and manage your project. Different types of work items track different types of work—such as user stories or product backlog items, tasks, bugs, or issues. Use work items to describe the work to be done, assign work, track status, and coordinate efforts within your team.

> **NOTE**
> This article shows how to add any type of work item. However, the recommended tool for adding backlog or portfolio items—such as, user stories, product backlog items, features, or epics— is to use the backlog or Kanban board to add new items. To learn more, see Create your backlog, Define features and epics and Start using your Kanban board. To create test cases and link them to user stories, see Add, run, and update inline tests and Create test plans and test suites.

For other clients that you can use to add and update work items, see Best tools for adding, updating, and linking work items.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

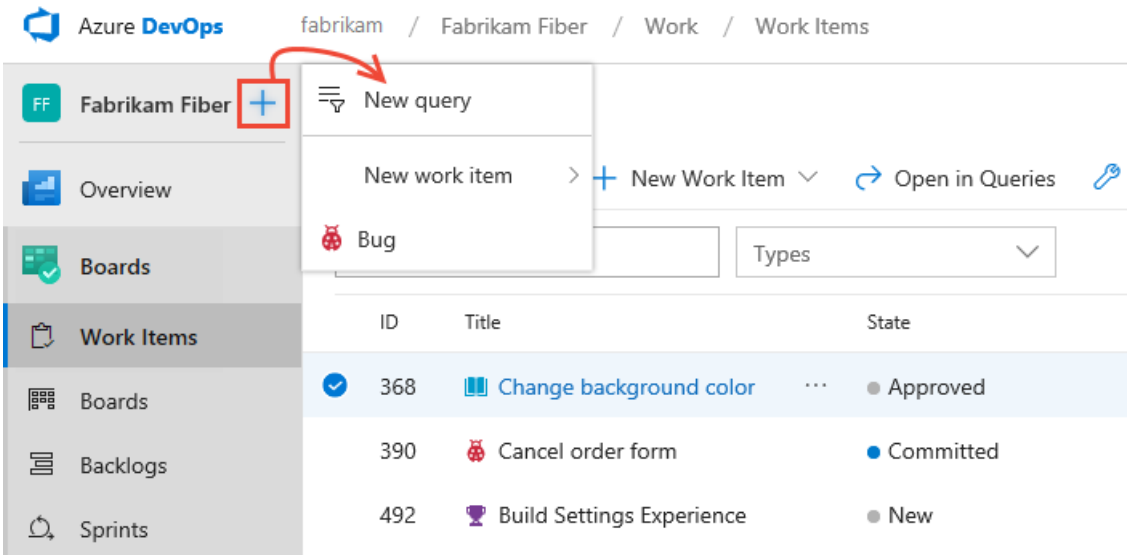
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

Add a work item

You can start adding work items once you connect to a project.

- Browser
- Visual Studio 2019
- Azure DevOps CLI

Choose a **Boards** page—such as **Work Items**, **Boards**, or **Backlogs**. Then choose the + plus icon and select from the menu of options.

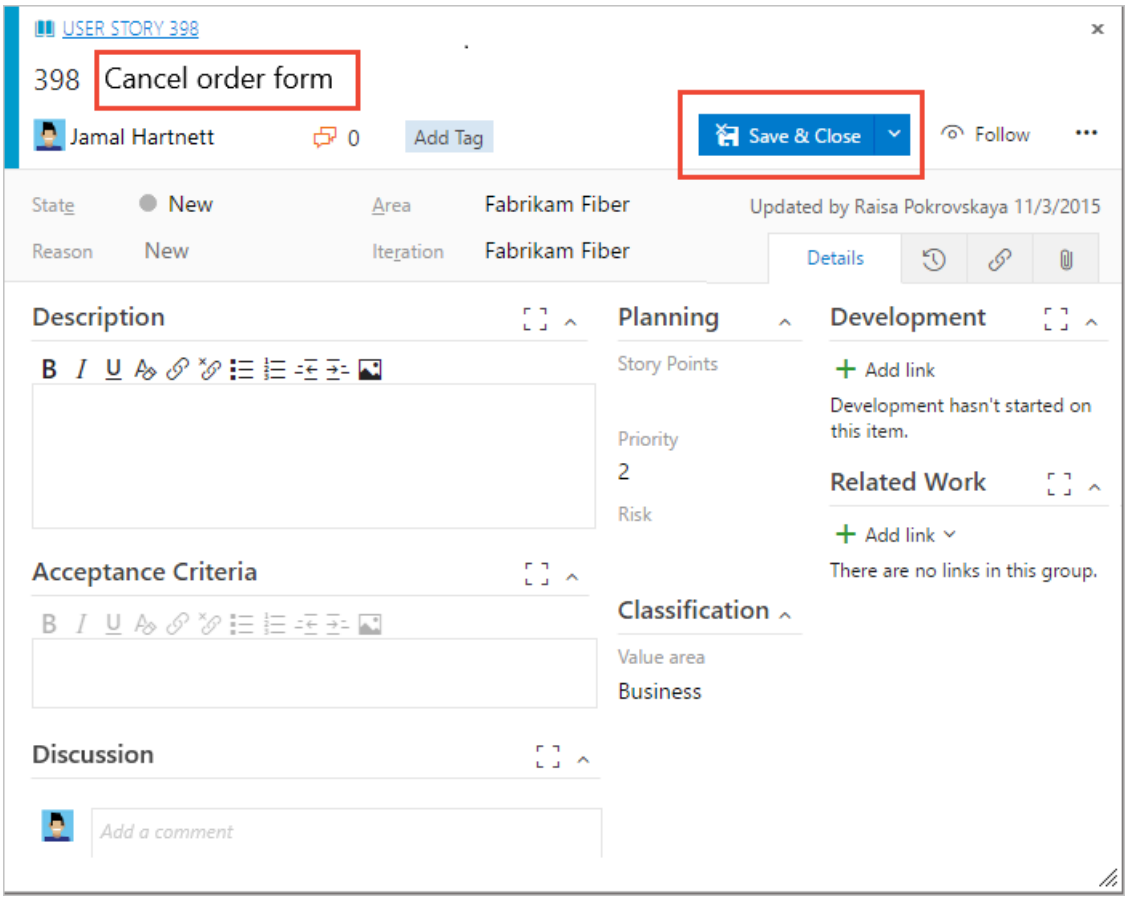


NOTE

Depending on the process chosen when the project was created—Agile, Basic, Scrum, or CMMI—the types of work items you can create will differ. For example, backlog items may be called user stories (Agile), issues (Basic) product backlog items (Scrum), or requirements (CMMI). All four are similar: they describe the customer value to deliver and the work to be performed.

For an overview of all default processes, see Choose a process. The Basic process requires Azure DevOps Server 2019.1 or later version.

Enter a title and then save the work item. Before you can change the State from its initial default, you must save it.



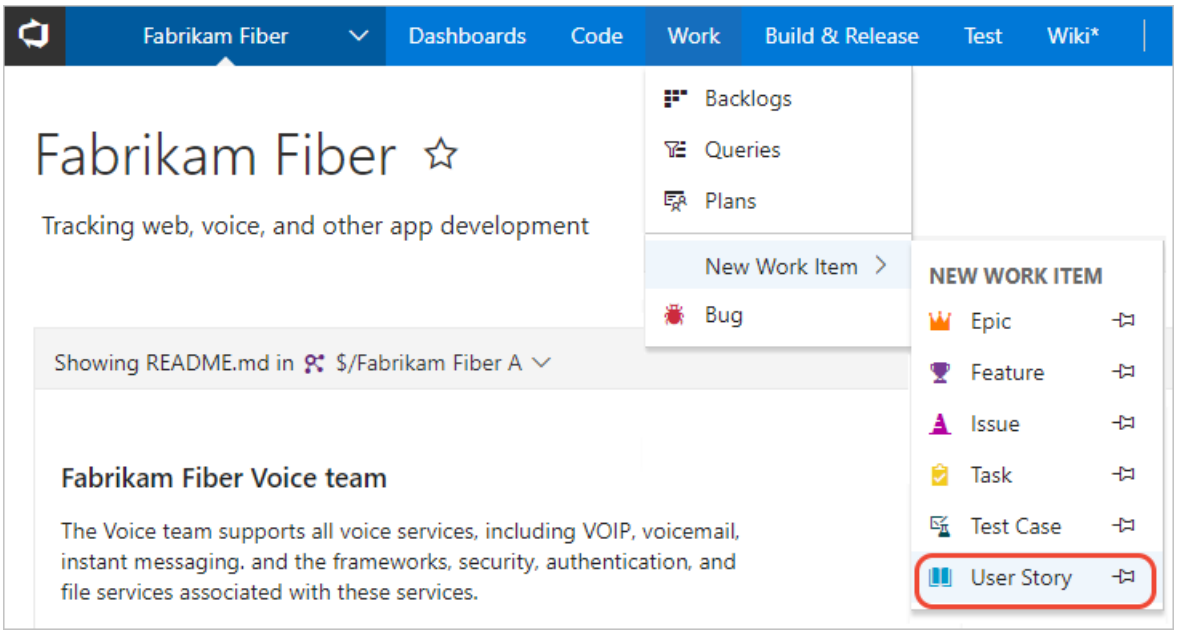
You can add tags to any work item to filter backlogs and queries.

Work items you add are automatically scoped to your team's default area path and iteration path. To change the team context, see Switch project or team focus.

That's it!

Create as many work items as you need of the type you need to track the work you want to manage.

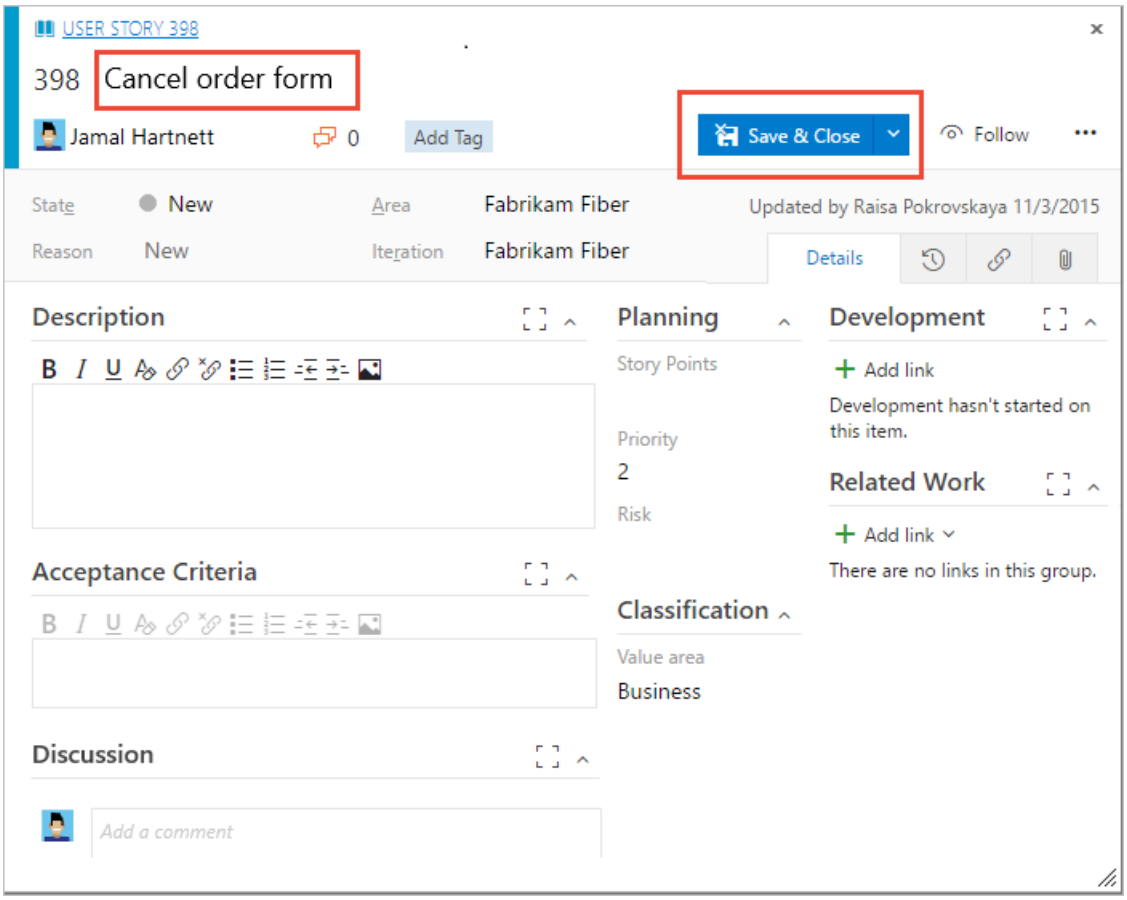
1. From **Work**, choose the work item type from the New Work Item list of options. Here, we choose to create a User Story.



> **NOTE**
>
> Depending on the process chosen when the project was created—Agile, Scrum, or CMMI—the types of work items you can create will differ. For example, backlog items may be called user stories (Agile), product backlog items (Scrum), or requirements (CMMI). All three are similar: they describe the customer value to deliver and the work to be performed.
>
> For an overview of all three processes, see Choose a process.

Choose the pin icon to have it show up within **Work** drop down menu.

2. Enter a title and then save the work item. Before you can change the State from its initial default, you must save it.



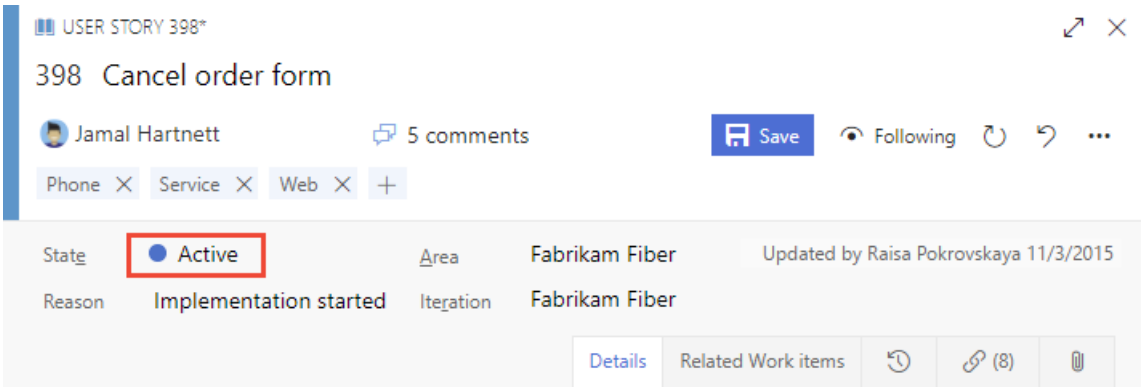
You can add tags to any work item to filter backlogs and queries.

Work items you add are automatically scoped to your team's default area path and iteration path. To change the team context, see Switch project or team focus.

Update work items as work progresses

As work progresses, team members can update the state and reassign it as needed. While the workflow states differ for different work item types, they usually follow a progression from New or Active to Completed or Done.

- Browser
- Visual Studio 2019
- Azure DevOps CLI



The following image shows the workflow states for a user story. If you want to discard a work item, change the state to Removed, or you can delete it. For details, see Move, change, or remove a work item.

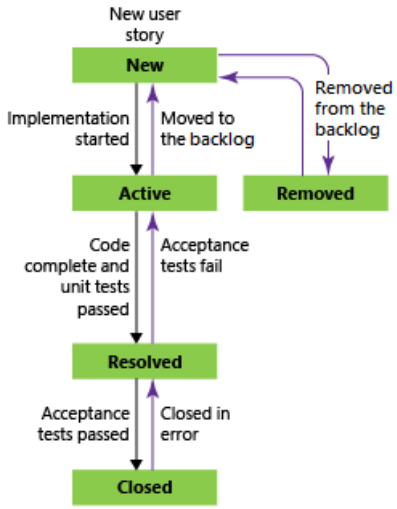
The following image shows the work flow states for a user story. If you want to discard a work item, change the state to Removed, or you can delete it. For details, see Remove or delete a work item.

Typical workflow progression:

- The product owner creates a user story in the **New** state with the default reason, **New user story**
- The team updates the status to **Active** when they decide to complete the work during the sprint
- A user story is moved to **Resolved** when the team has completed all its associated tasks and unit tests for the story pass.
- A user story is moved to the **Closed** state when the product owner agrees that the story has been implemented according to the Acceptance Criteria and acceptance tests pass.

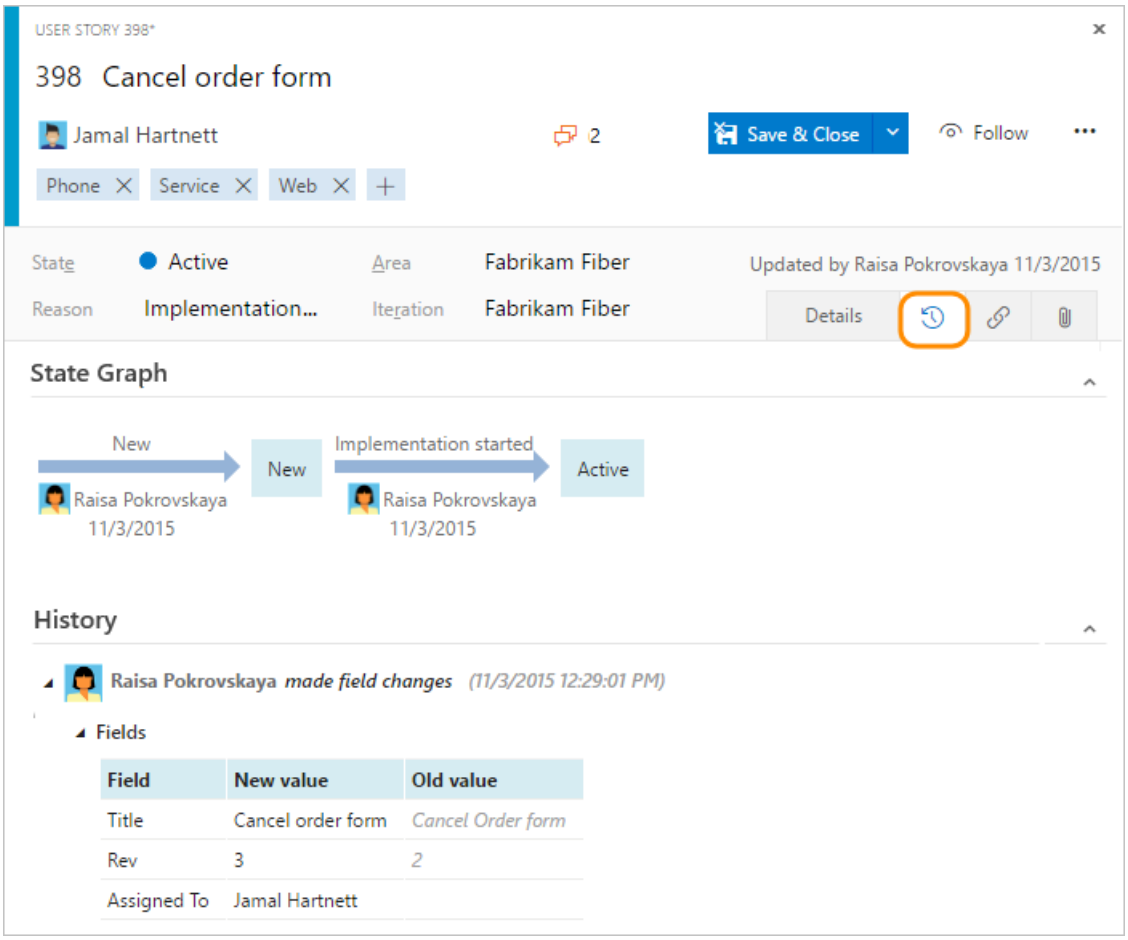
Atypical transitions:

- Change the State from **Active** to **New**.
- Change the State from **Resolved** to **Active**.
- Change the State from **Resolved** to **New**.
- Change the State from **Closed** to **Active**.
- Change the State from **New** to **Removed**.
- Change the State from **Removed** to **New**.



Removed work items remain in the data store and can be reactivated by changing the State.

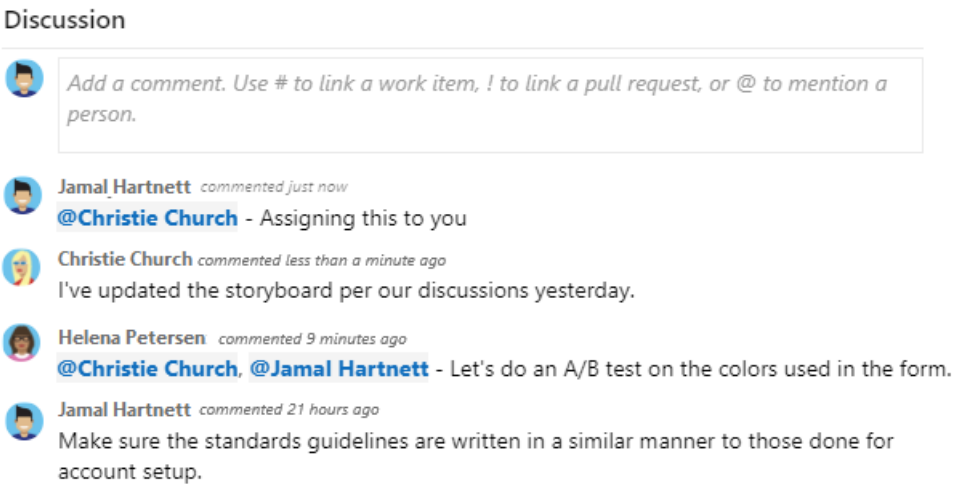
With each update, changes are recorded in the History field, which you can view through the **History** tab.



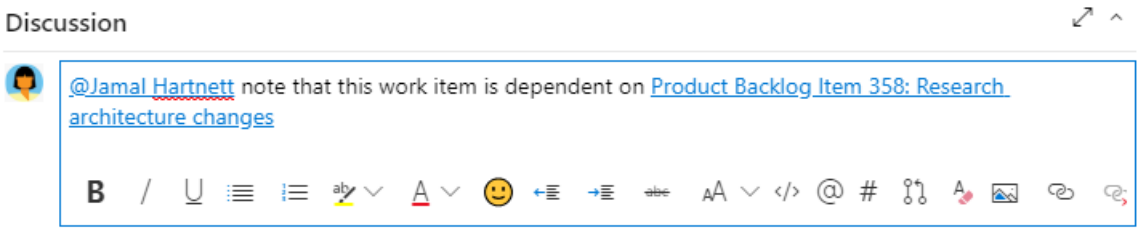
To find work items based on their history, see History & auditing.

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



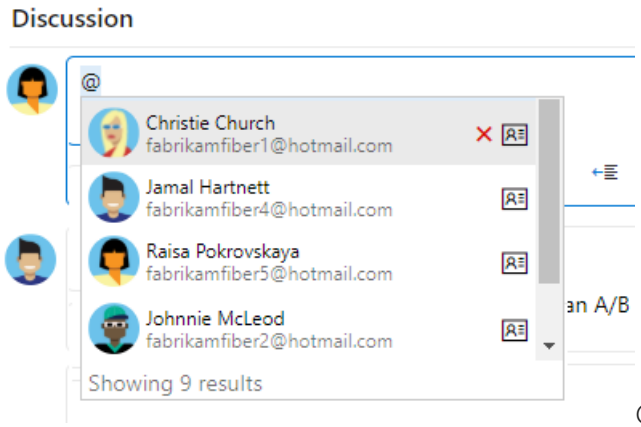
The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.



> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the **History** field. The full content of the text entered into the Discussion text box is added to the History field.

Mention someone, a group, work item, or pull request

Choose one of these icons —@ , # , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type **@**, **#**, or **!**.

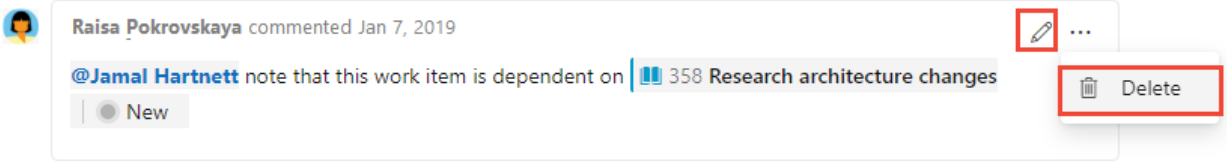


@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the ··· actions icon and then choose **Delete**.



> **NOTE**
> Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently

referenced will appear from which you can select.

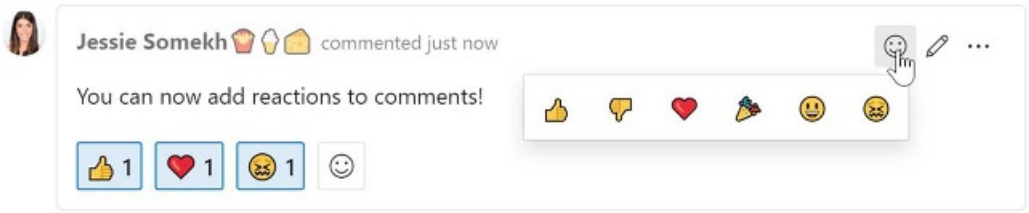
To reference a work item, use the **#ID** control. Type **#** and a list of work items that you've recently referenced will appear from which you can select.

You can't edit or delete comments once you've entered them.

> **IMPORTANT**
> For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

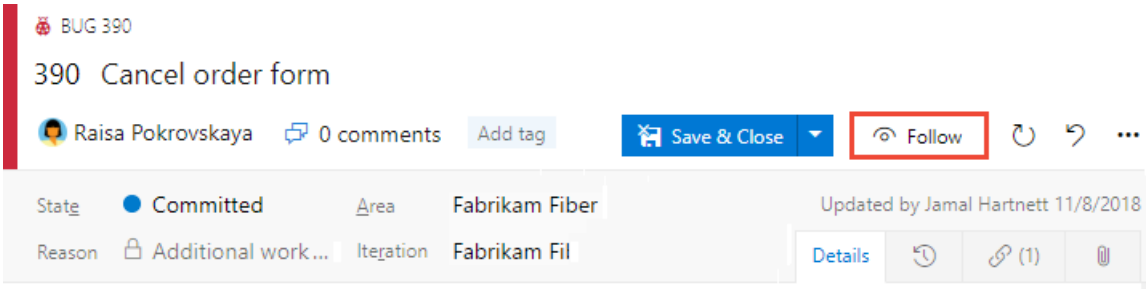
Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



Follow a work item

When you want to track the progress of a single work item, choose the Follow follow icon. This action signals the system to notify you when changes are made to the work item.



You'll only receive notifications when other project members modify the work item, such as adding to the discussion, changing a field value, or adding an attachment.

Notifications are sent to your preferred email address, which you can change from your user profile.

To stop following changes, choose the Following following icon.

> **IMPORTANT**
> To support the follow feature, you must configure an SMTP server in order for team members to receive notifications.

Try this next

To quickly add backlog items, such as user stories, requirements, or bugs, see these articles:

Create your backlog Kanban quickstart

For descriptions of each field and work item form control, see Work item field index and Work item form controls.

Once you've added several work items, you can use additional features to get notified of changes, create queries, define status and trend charts, plus more.

For more clients that you can use to add work items, see Clients that support tracking work items.

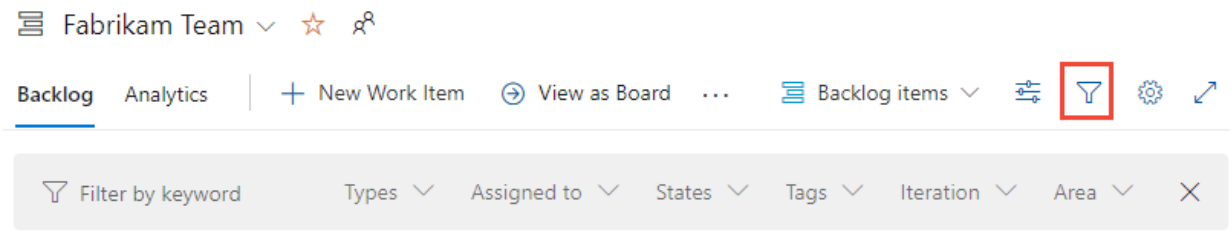
Interactively filter backlogs, boards, queries, and plans in Azure Boards

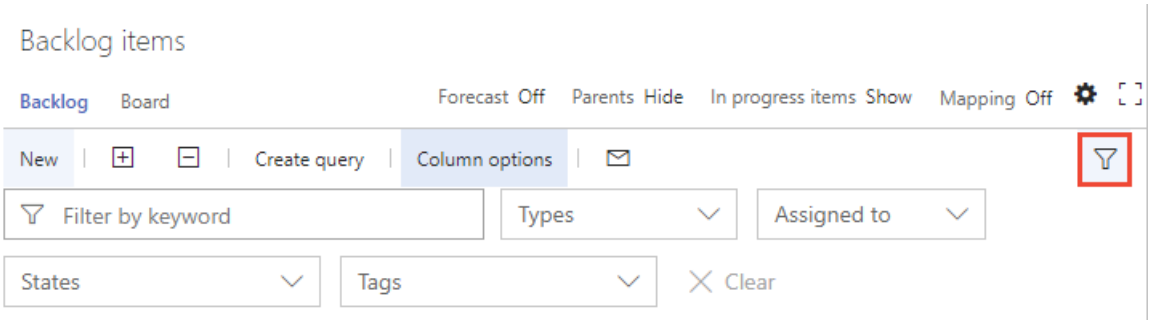
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

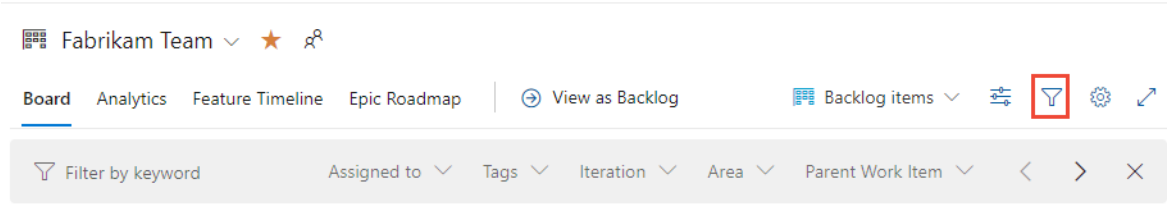
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

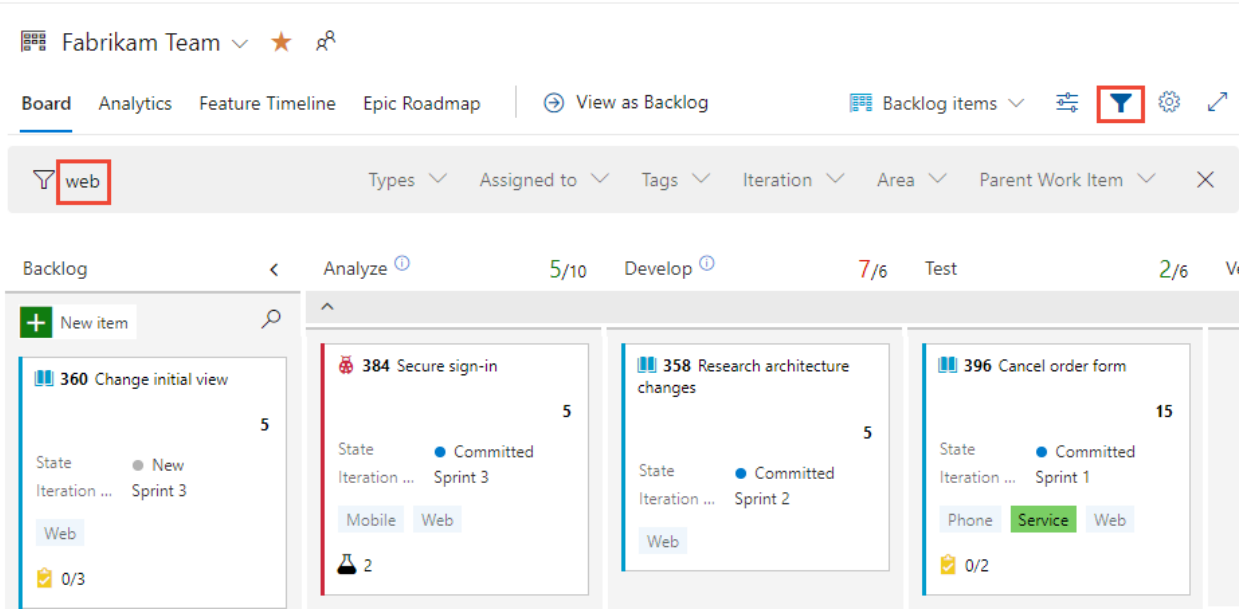
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

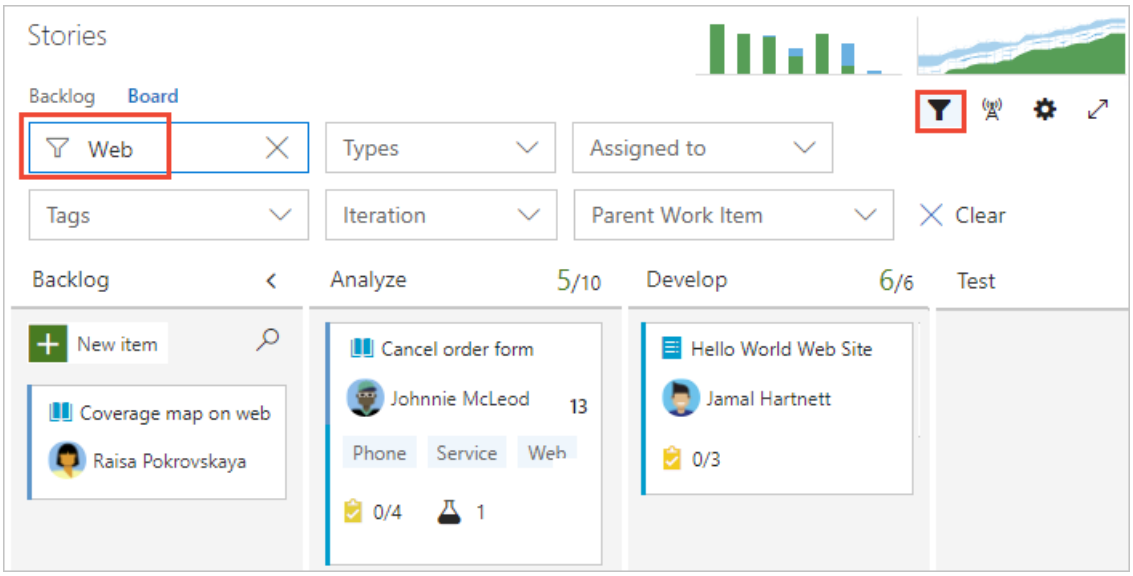
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

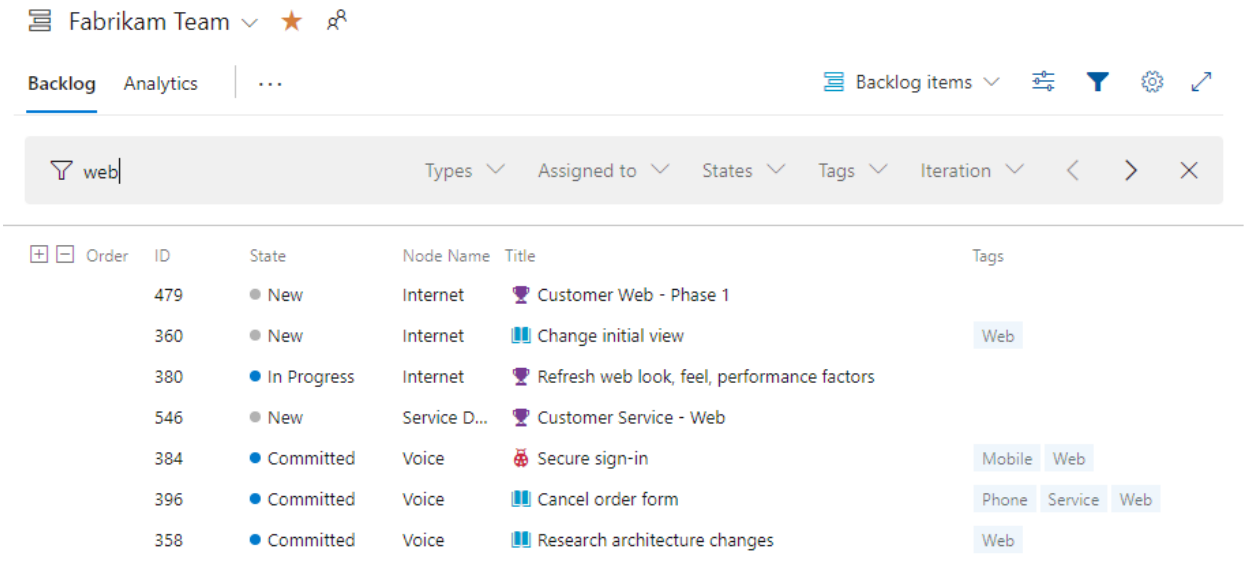


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

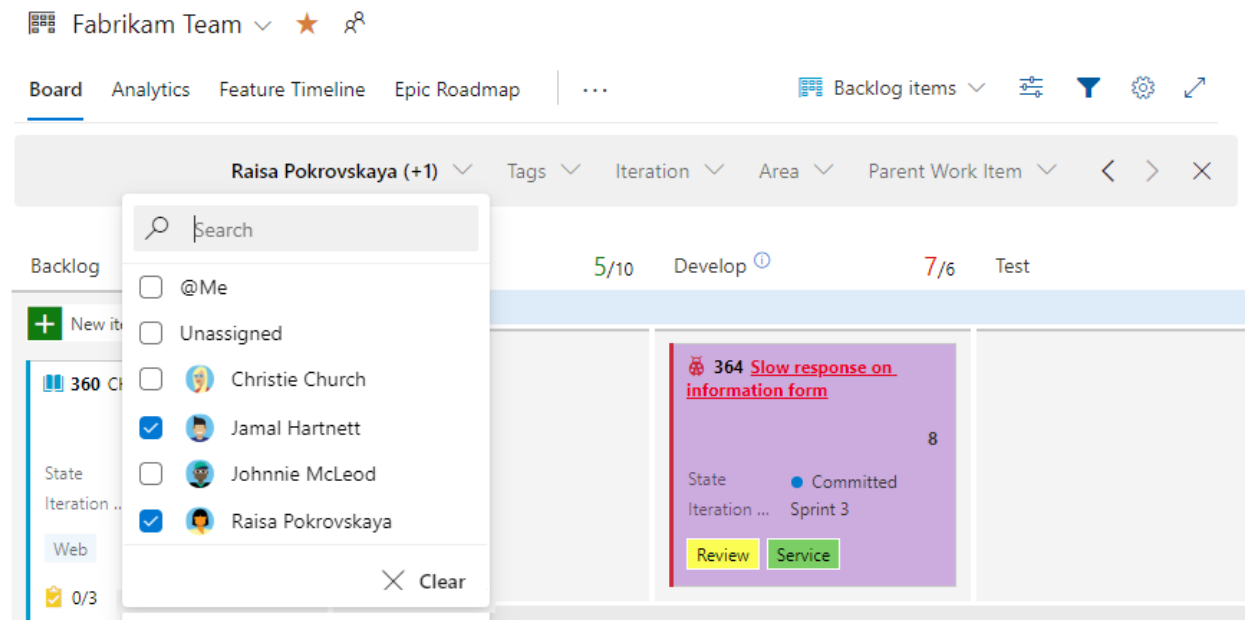
Features board

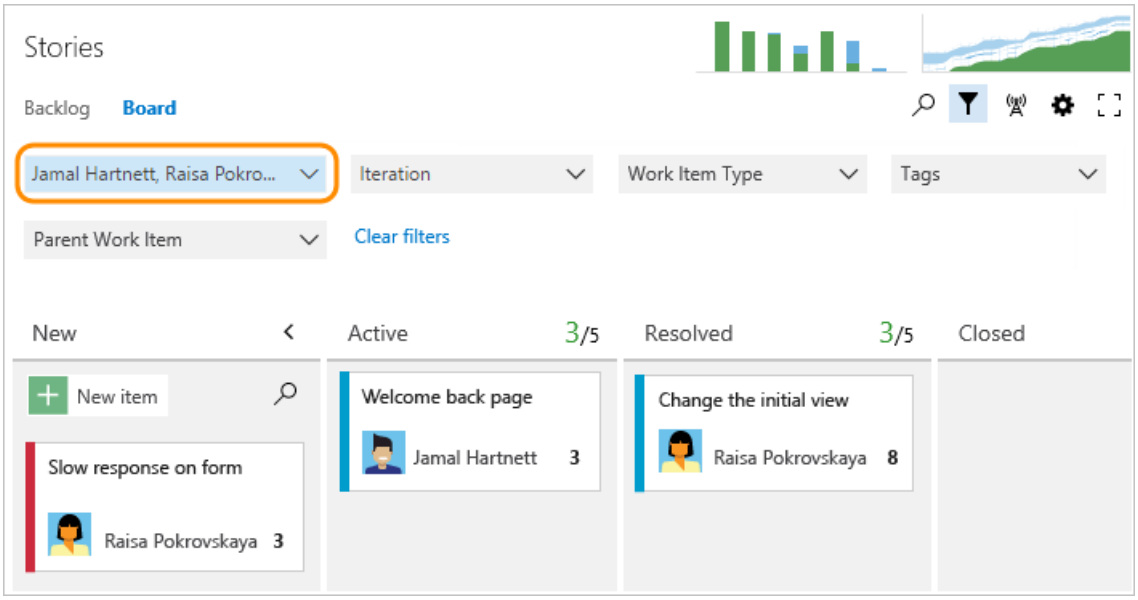
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

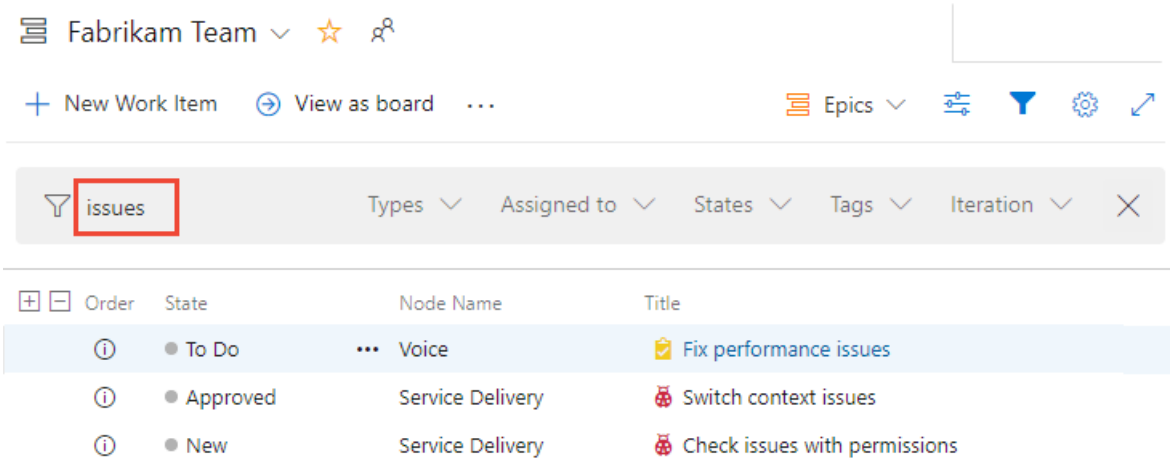
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

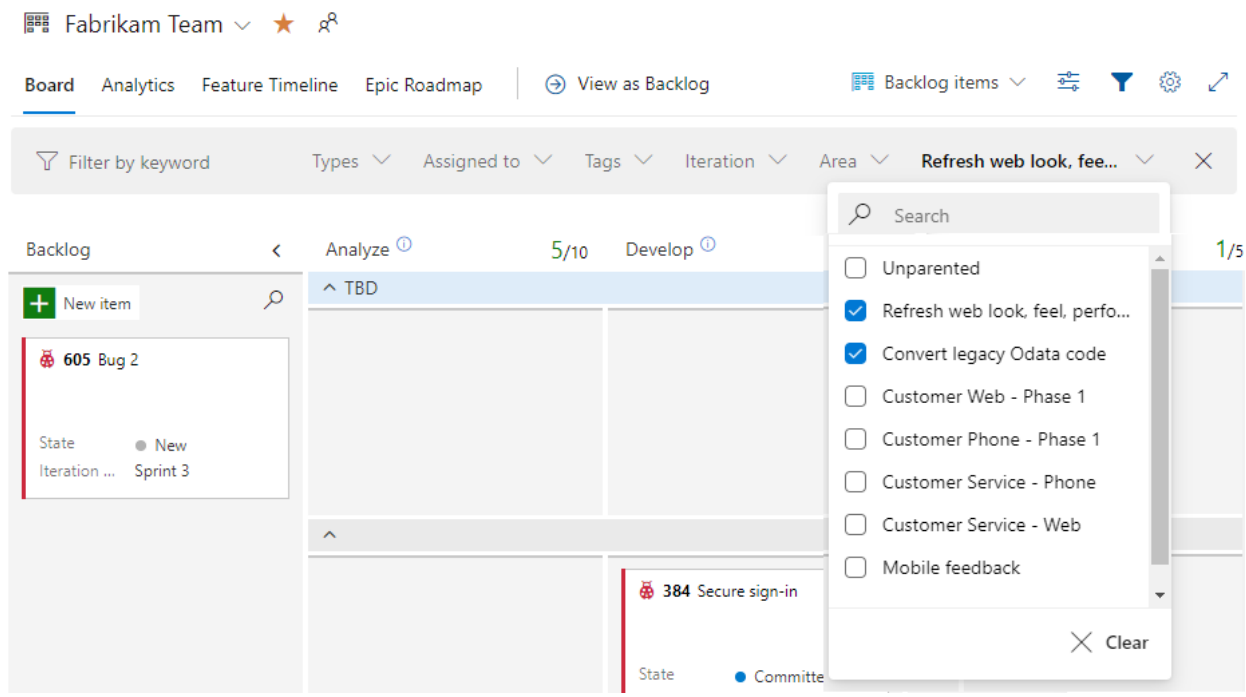
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

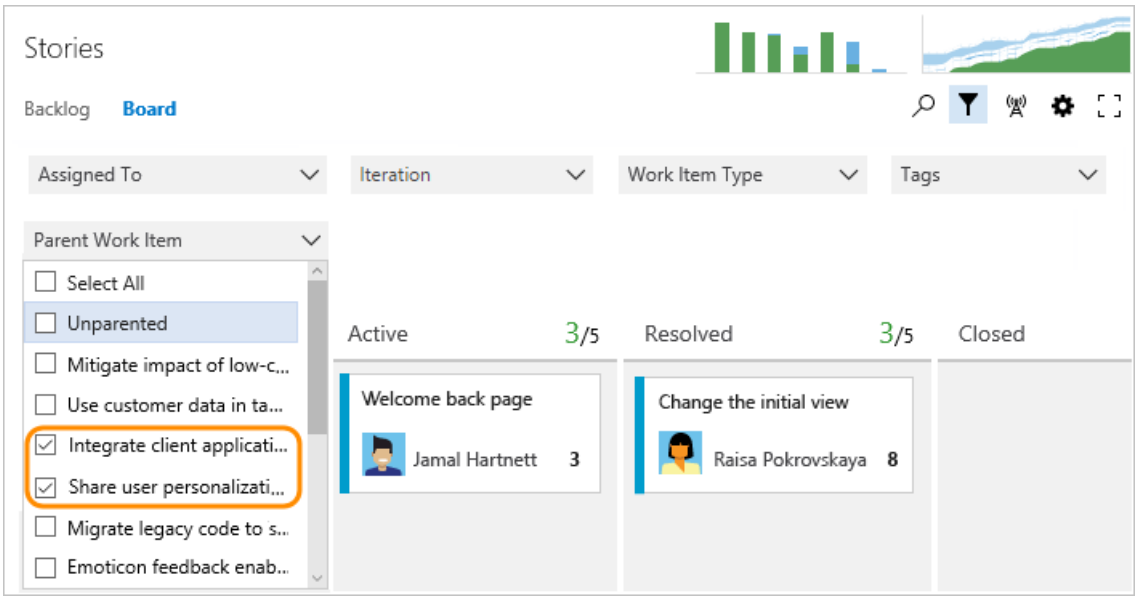
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



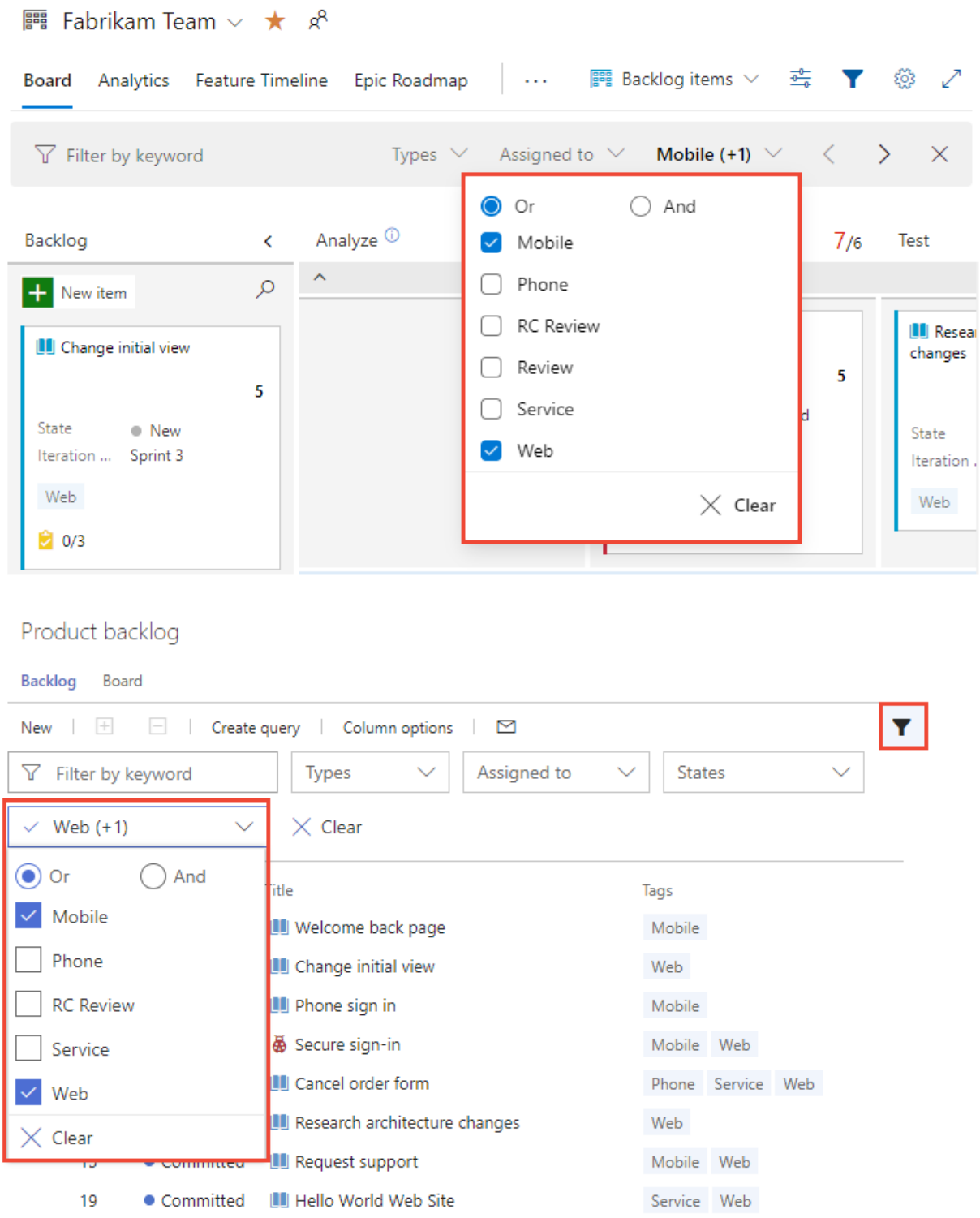


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

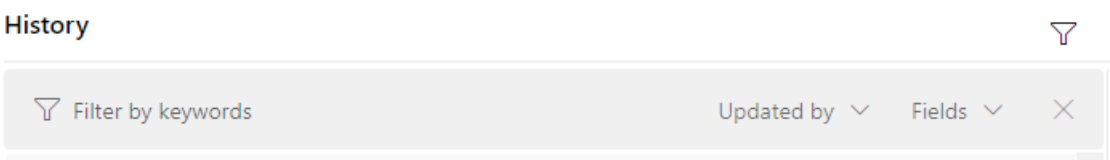


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Manage requirements

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Become familiar with the essential concepts to manage projects using Agile tools. Gain an overview of Azure DevOps tools and features to manage requirements. This article maps Agile requirements management tasks by project managers to the tools Azure DevOps supports. More detailed information is provided under Related articles.

> **NOTE**
> *Requirements management is a continuous process throughout a project lifecycle—encompassing the processes of documenting, analyzing, prioritizing, tracking, and collaborating with stakeholders to agree on work to be performed. A single requirement corresponds to a capability which a project outcome—product, service, architecture, performance—should conform.*

Agile requirements management supports the following scenarios.

- Define and track status of requirements
- Analyze and prioritize requirements
- Assign requirements to timeboxes
- Group and organize requirements
- Manage dependencies
- Determine what can ship and when
- Monitor and report on progress

When managing requirements using Agile methods, you'll also perform within an Agile culture which supports the following principles and methods:

- Alignment within the organization
- Autonomous teams
- Kanban and lean management
- Scrum

Capture requirements

You capture requirements using work items, where each work item is based on a work item type. You have a choice of work item types to use based on the process you select. Or, you can add a custom work item type.

> **NOTE**
> *Requirements* specify expectations of users for a software product. In Azure Boards, requirements are defined by work items that appear on your product backlog. They correspond to User Stories (Agile), Product Backlog Items (Scrum), Issues (Basic), or Requirements (CMMI) based on the process selected for your project. They also belong to the Requirements Category which manages which work item types appear on the product backlog.

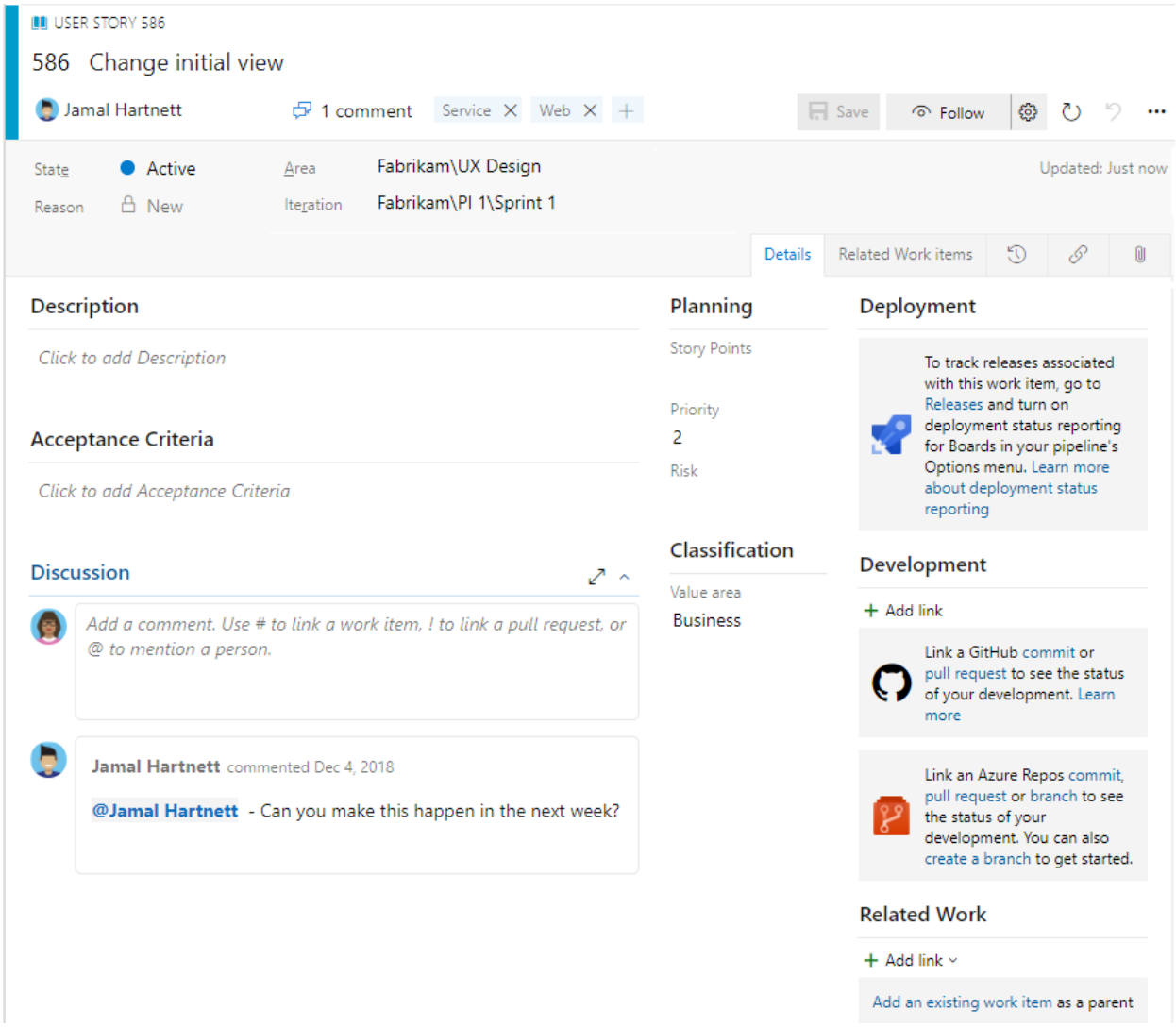
Work item fields and form

You can use work items to track anything you need to track. Each work item represents an object stored in the

work item data store. Each work item is based on a work item type and is assigned a unique identifier.

Each work item supports tracking data contained in work item fields. Also, it captures changes as updates are made within the **History** field and comments made in the **Discussion** section. The following image shows a sample work item form for the User Story work item type.

Example work item form



In a nutshell, you use work items to support these tasks:

- Add information, update status, assign to team members, link work items, and attach files
- Assign work to a timebox or sprint
- Quickly fill in work item fields using work item templates
- Contribute to a queryable discussion thread
- Prioritize work and triage work items.

Other features that support end-to-end traceability are the **Development** and **Deployment** sections. These sections support the following tasks and insights:

- Create a new branch or pull request from a work item
- Complete the pull request
- Perform a squash merge
- Create a branch for several work items
- Link a work item to existing development and build objects
- View the release stages associated with the work item within the work item form in real time
- View the status of releases within those work items that are associated with commits in the build and release

pipelines

Work item types

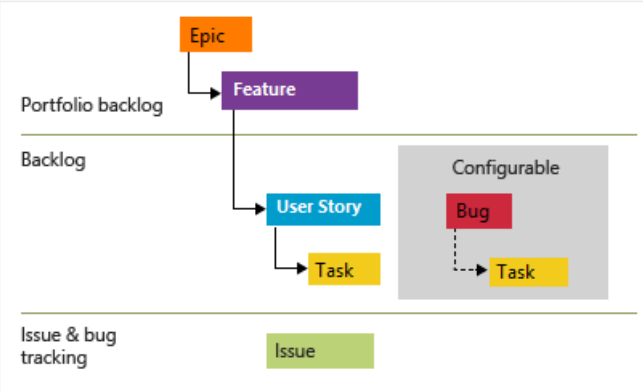
Different work item types support capturing different information and workflow processes. Each work item is based on a work item type. The following images illustrate the default work item types used to capture requirements and code defects. They are based on the four system processes—Agile, Basic, Scrum, or Capability Maturity Model Integration (CMMI).

Each Azure DevOps project is defined based on one of these customizable processes. Each team can determine how they want to track bugs.

Default work item types

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

Customize work item types

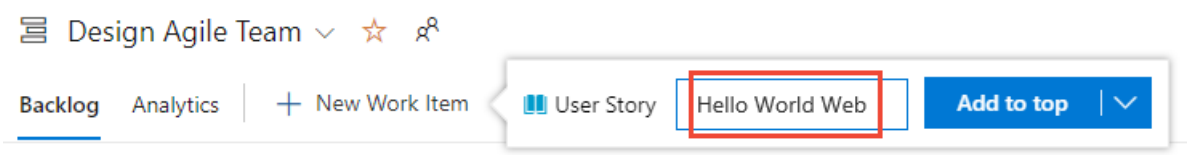
You can add custom work item types or customize the default work item types. Supported customizations include the following additions:

- Add custom fields and workflow states
- Add custom rules to support business workflow processes
- Add custom portfolio backlogs and customize backlogs and boards
- Add custom controls to work item forms to gain enhanced functionality.

Add work items to product backlog or board

Capturing requirements typically starts by adding a Title to a product backlog. You add details later.

Capture requirements on the product backlog



Import and update requirements using Excel

Alternatively, you can import and update requirements you've defined through a .csv file or Excel spreadsheet. These tools support a flat list or a hierarchy of work items. As shown in the following image, a hierarchy of Epics, Features, and User Stories are defined in Excel and then imported to Azure DevOps.

Import requirements from Excel

Project: MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** [None] **List type:** Tree

ID	Work Item Ty	Stat	Title 1	Title 2	Title 3
	Epic	New	Improve User Experience		
	Feature	New		Integrate client application with popular email clients	
	User Story	New			Implement a factory which abstracts the email client
	User Story	New			As a user, I can select a number of support cases and use cases
	User Story	New			Customer request support for GMail
	User Story	New			Customer request support for Yahoo mail
	User Story	New			Customer request support for Roundup
	Feature	New		Integrate client app with IM clients	
	Feature	New		Emoticon feedback enabled in client application	
	User Story	New			As a user, I can select an emoticon and add a short description
	User Story	New			Add animated emoticons
	Epic	New	Framework to port applications to all devices		
	Feature	New		Migrate legacy code to portable frameworks	
	User Story	New			Implement a factory that migrates legacy to portable frameworks
	User Story	New			As a developer, I can analyze a code base to determine compliance with
	Epic	New	Counter the Heartbleed web security bug		
	Feature	New		Scan all software for the Open SLL cryptographic code	
	User Story	New			Scan all code base and identify the affected code
	Feature	New		Update software to resolve the Open SLL cryptographic code	
	User Story	New			Update and retest suite code base affected by the vulnerability
	User Story	New			Update and re-test service code based affected by the vulnerability
	Epic	New	Shared Personalization and state		
	Epic	New	Service interfaces to support REST API		
	Epic	New	Convert legacy Odata service interfactes to REST API		
	Feature	New		Convert all services from using experiemental code	
	Feature	New		Convert all client service calls from using experimental code	

Functional and non-functional requirements

Any work that you or a development team need to track can be captured using work items. You can use the same type of work item to capture both functional and non-functional requirements. Non-functional requirements specify criteria associated with system operations rather than specific product or service functionality.

You can differentiate your requirements using tags, the Business Value field, or a custom field.

Maintain requirement specifications

Requirements often require specifications to provide details that aren't readily captured within the work item. You can use Azure DevOps to maintain and version control your requirements under an Azure Repos repository. Or, you can use a project wiki to provide a central source and repository for specifications.

You can then link your specifications or attach them to your requirements.

Analyze and prioritize requirements

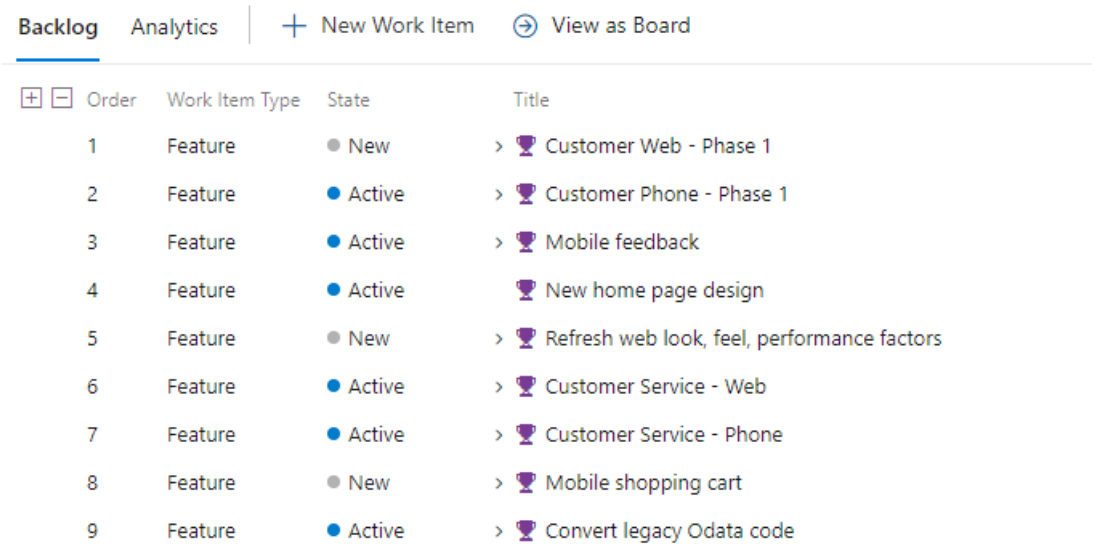
Once you have a working backlog, you'll want to get it in priority order. You'll want to review and refine your requirements and make sure the acceptance criteria is well-defined. These tasks are supported through the following Azure Boards tools:

- **Product backlog**: Supports drag-and-drop of work items to get them in priority order. Supports bulk-edit of work items to change assignments or update fields.
- **Query Results, Triage mode**: Supports review of a list of work items and their forms so that you can

quickly update work items and add details.

Here the features backlog shows the sequence of features to ship.

Prioritize feature backlog



Group and organize requirements

The product backlog starts out as a flat list. However, often you want to group requirements that support specific features or business objectives. Azure Boards supports this by providing portfolio work item types, portfolio backlogs and boards, and a Mapping tool to quickly link requirements to a portfolio work item.

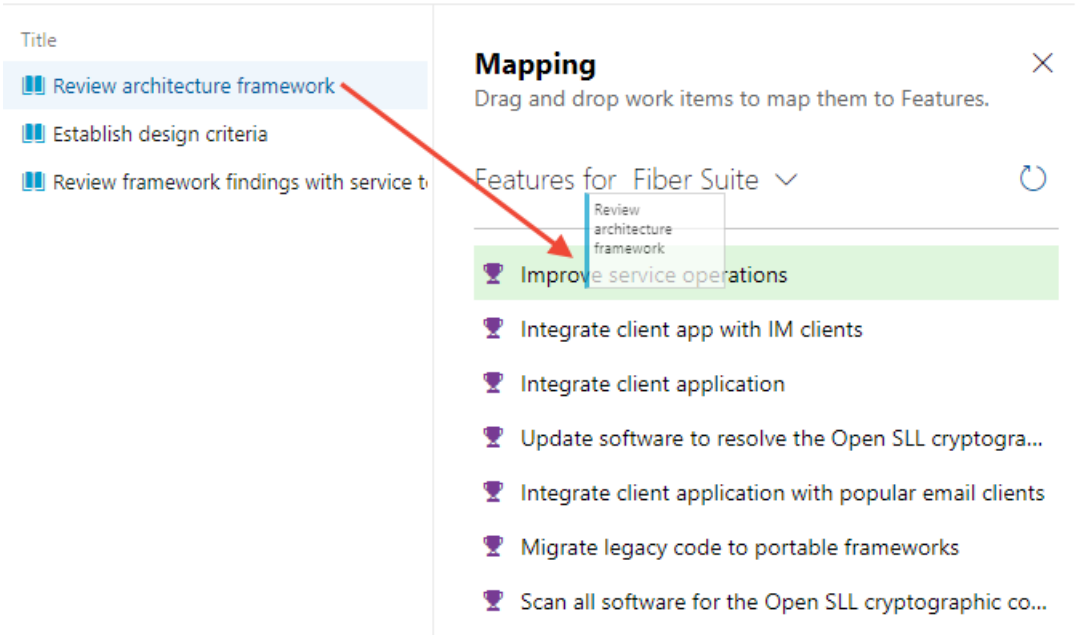
Work item tags are another way you can group requirements.

Epics, features, and portfolio backlogs

You group requirements under Features, and Features under Epics, using parent-child hierarchical links. This type of grouping is recommended for organizations with several teams that want to view rollups associated with multiple teams and to take advantage of all portfolio planning tools.

By grouping work within a hierarchy, you can manage a portfolio of features that are supported by different development and management teams. Also, you can view a rollup of estimates, progress bars, and more on product backlogs.

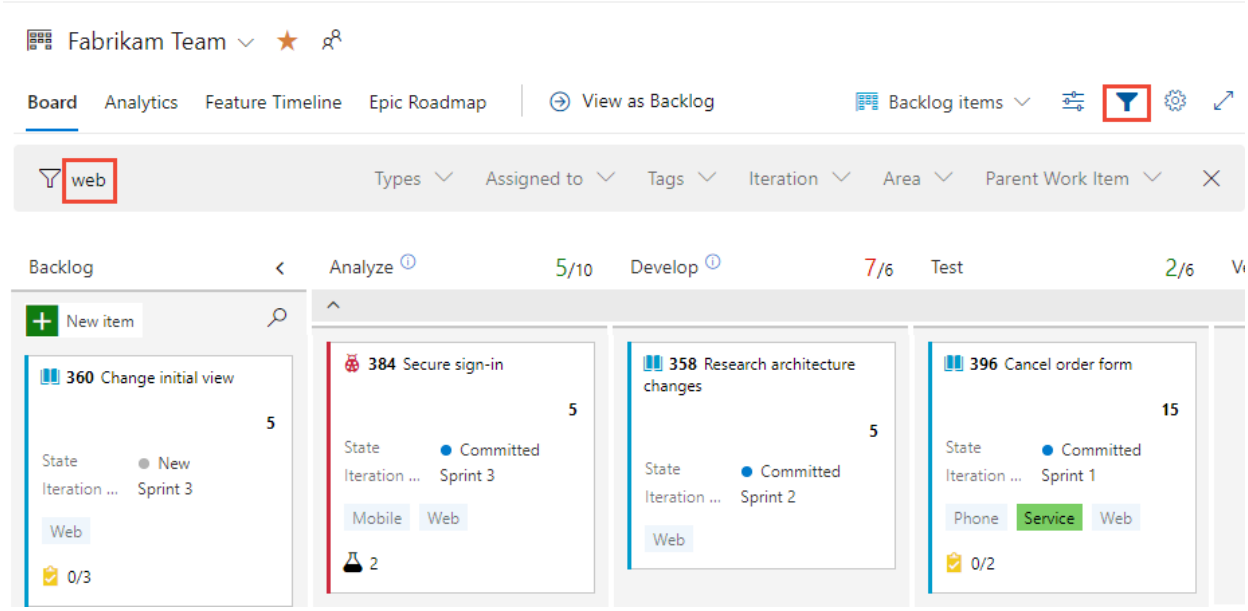
Group User Stories under Features using Mapping



Use tags to group work items

With work item tags, team members can assign ad-hoc tags to work items. You can use these tags to filter backlogs and boards as well as query on work items. For example, the following image illustrates a Kanban board filtered on the *web* keyword which displays cards with the *Web* tag.

Filter backlogs and boards based on tags



Implement Kanban or Scrum

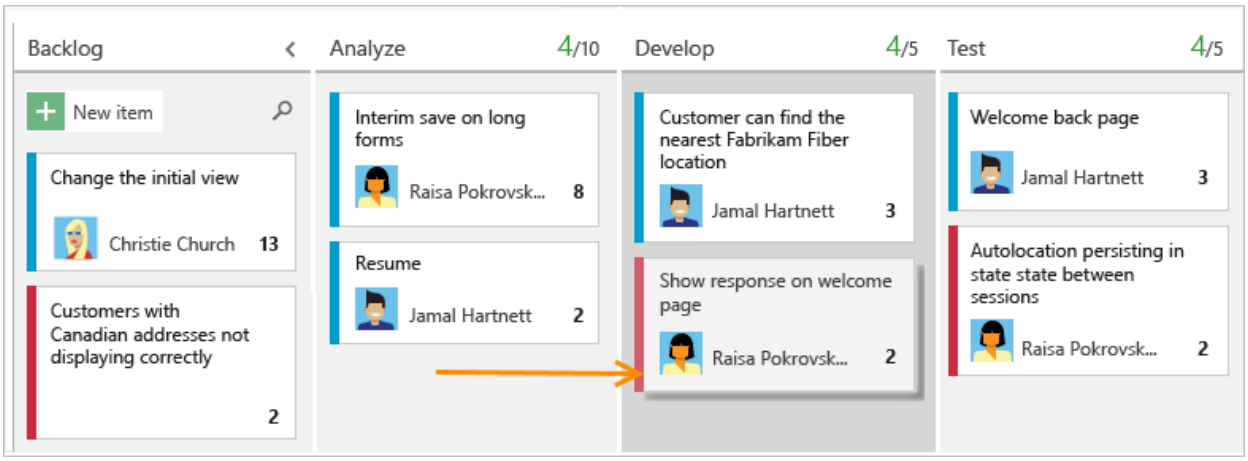
Two of the major Agile methods are Kanban and Scrum. Azure Boards supports both methods. Or, teams can adapt them to use a combination of methods such as Scrumban.

Implement Kanban

Each product and portfolio backlog is associated with a corresponding Kanban board. Both backlogs and boards are associated with a team, and display work items based on the area and iteration paths selected by the team.

Each board supports many Kanban practices such as defining columns and swimlanes, setting Work-in-Progress (WIP) limits, defining the Definition of Done, and more. As work completes in one stage, you update the status of an item by dragging it to a downstream stage.

Example Kanban board

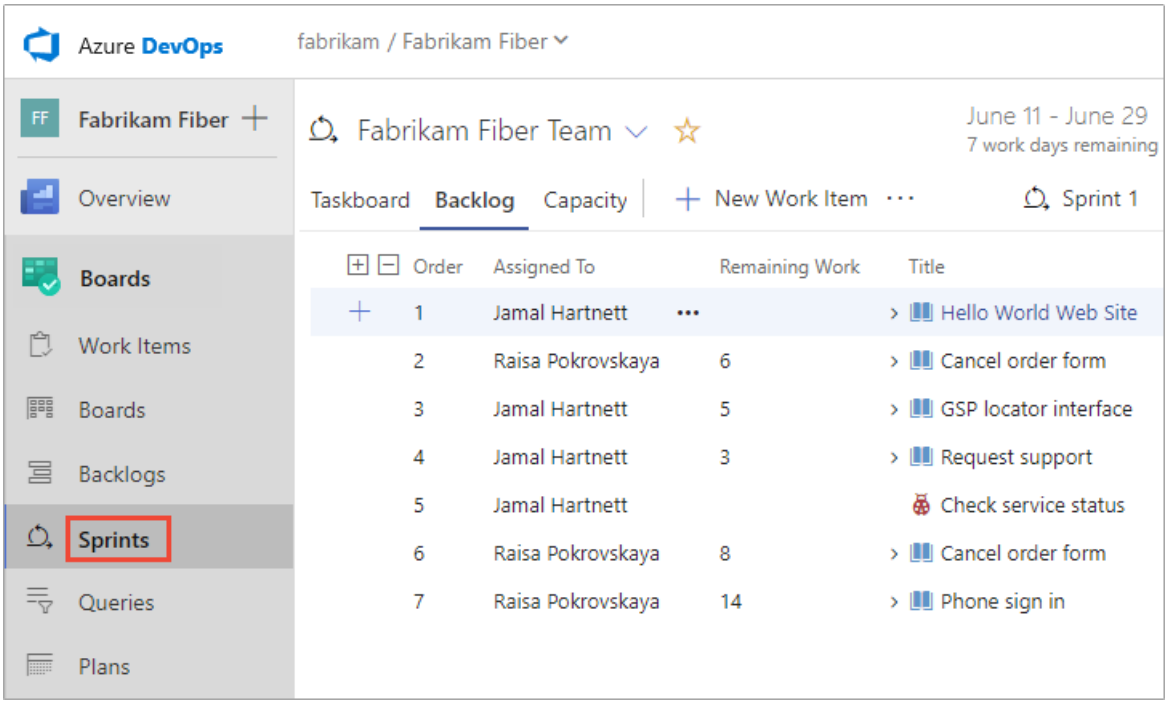


Each team can quickly configure their board and the cards to support their business needs.

Implement Scrum

Sprint backlogs and Taskboards provide a filtered view of work items a team has assigned to a specific iteration path, or sprint. From your requirements backlog, you can drag-and-drop work items onto an iteration path, and then view that work in a separate **Sprint Backlog**.

Example Sprint Backlog



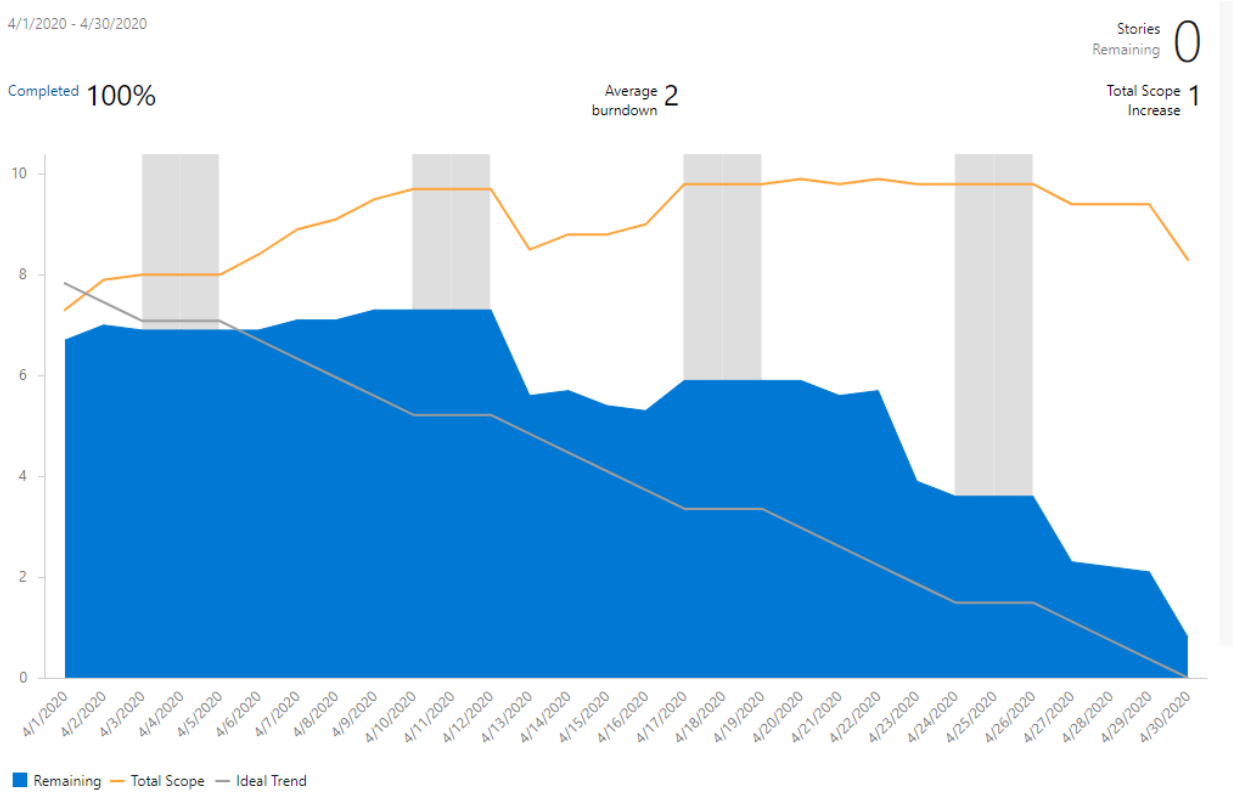
Supported Scrum practices include the following tasks:

- Assign requirements to a sprint
- Add tasks to requirements
- Set sprint capacity for team members
- Adjust work to fit sprint capacity
- Share your sprint plan
- Filter, update tasks, and update task status
- Monitor sprint burndown

Sprint burndown chart

By updating the status of work daily throughout a sprint, you can easily track sprint progress with the Sprint burndown chart, as shown in the following image.

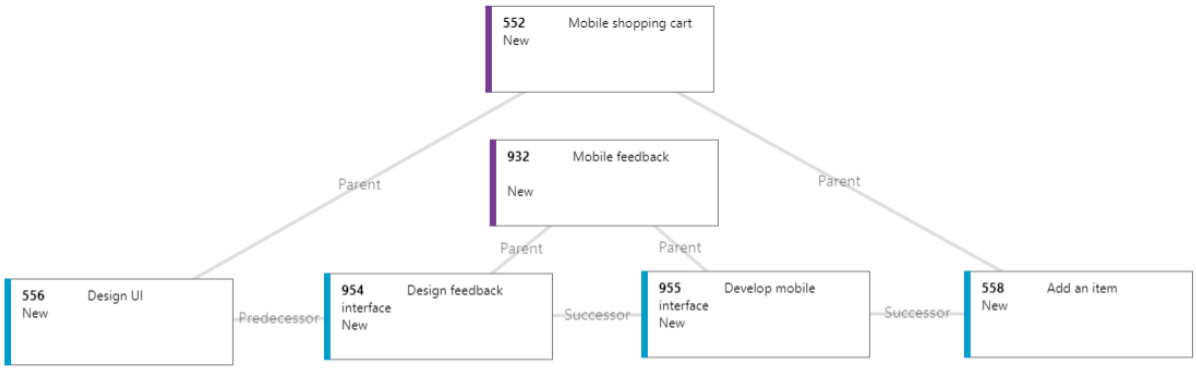
Example Sprint burndown chart



Manage dependencies

In Microsoft Project, you manage tasks that depend on the completion of other tasks by linking them. To manage dependencies in Azure Boards, you can link work items using the Predecessor/Successor link type. Once you've linked work items, you can view link relationships using the Work Item Visualization Marketplace extension. The following image illustrates link relationships among several work items.

To see the full image, click the image to expand. Choose the close icon to close.



Minimum Viable Product versus Critical Path Management

Azure Boards doesn't provide a native view of the critical path. In part, as Agile methodologies favor a Minimum Viable Product (MVP) over Critical Path Management (CPM). By using MVP, you identify the shortest path and dependencies by prioritizing epics, features, stories, and tasks.

Perform milestone planning

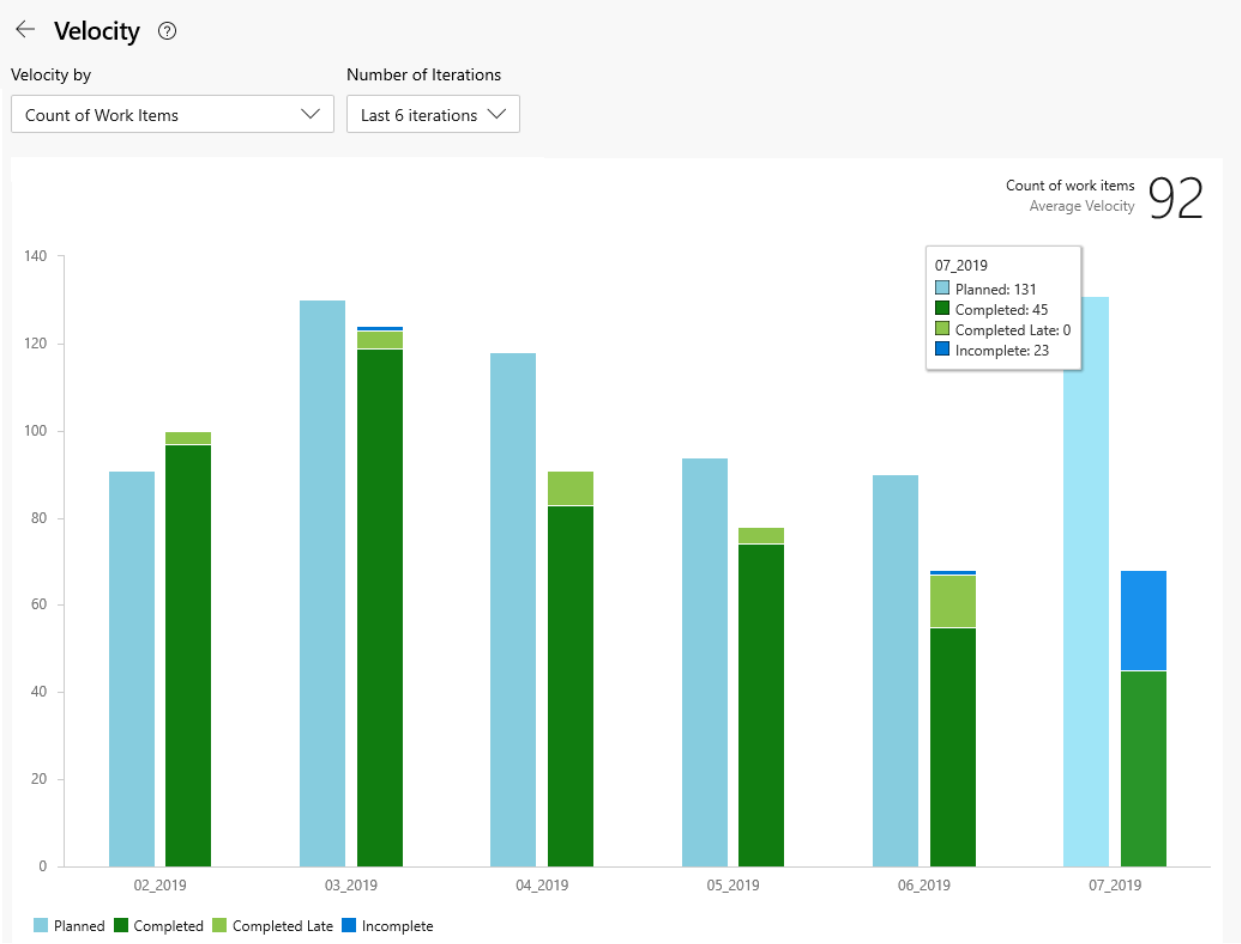
Two tools that support planning what can ship when are team velocity and forecasting.

Team velocity

One of the advantages of working in sprints is that you gain insight into team velocity. Velocity provides an

indication of how much work a team can complete during a sprint based either on a count of completed work items or the sum of requirement estimates.

Example team Velocity chart



Forecast requirements

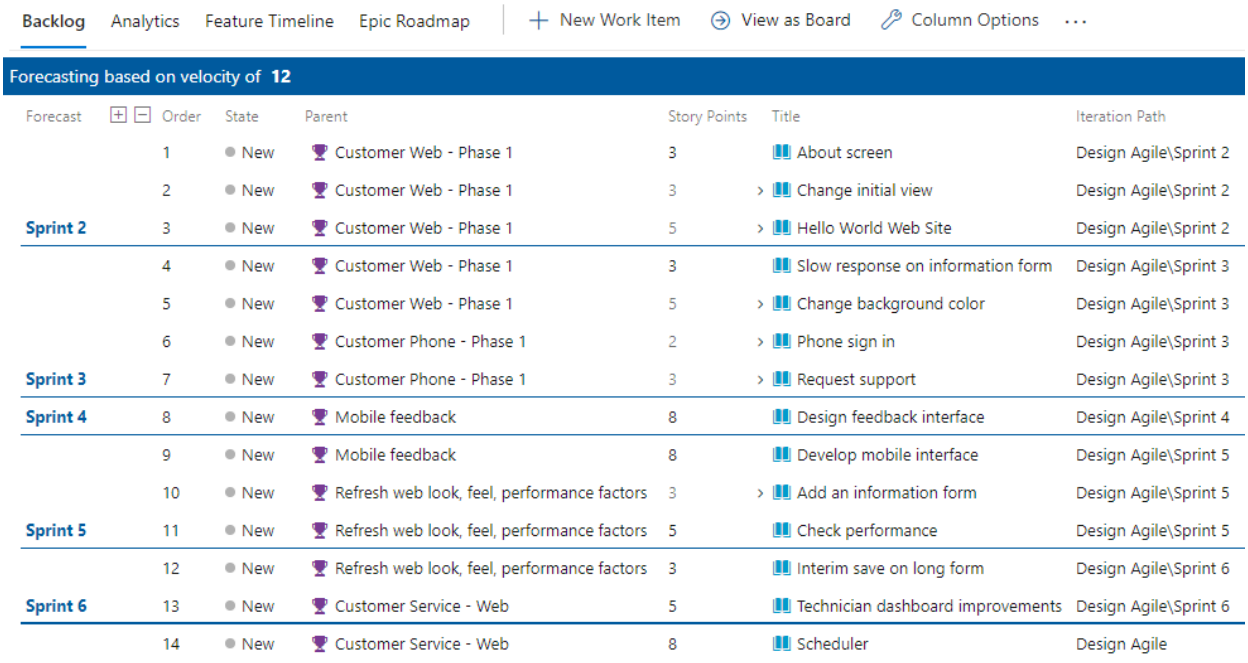
The **Forecast** tool requires that you provide estimates to the Story Points, Effort, or Size field for each requirement.

With estimates assigned to each requirement, you can set a team velocity. In the example below, we specify 12 for the velocity, equivalent to stating that on average the team can complete 12 Story Points per sprint. The Forecast tool shows which requirements and features the team can complete within the next six sprints. Using the Planning tool, you can quickly assign requirements to the forecasted sprints.

Example Forecast of requirements backlog

To see the full image, click the image to expand. Choose the close icon to close.

[



]../boards/media/best-practices/forecast-product-backlog-ordered-parent.png#lightbox)

If you want to integrate your requirements planning with Microsoft Project tools, you may do so via a Marketplace extension.

Milestone markers

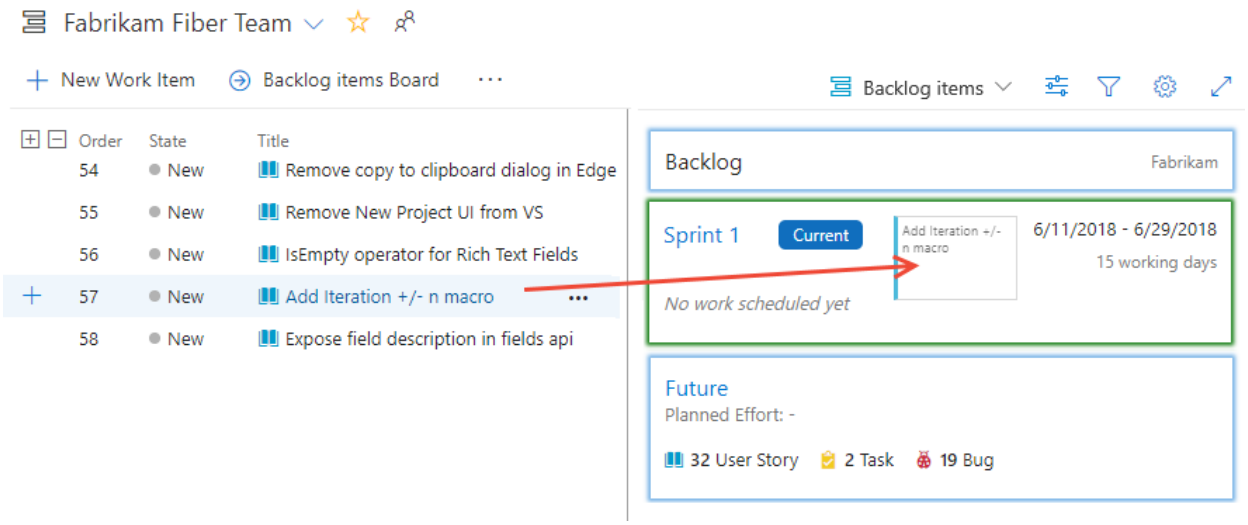
Milestone markers aren't used in Azure Boards work tracking, except for Delivery Plans. Delivery Plans provide a calendar view and allow you to define a milestone marker. However, you can use one or more of the following options to mark a work item as a milestone:

- Simply prepend or append the word **Milestone** in the title of your work item
- Add a work item tag labeled **Milestone**
- Add a custom field labeled **Milestone** and populate it with a pick list of milestones
- Link work items using the Predecessor/Successor or Related link type to a milestone work item
- Assign a milestone work item to the sprint in which it's targeted for completion.

Assign requirements to timeboxes

You can quickly assign work items to a sprint through drag-and-drop from the product backlog to the sprint listed within the Planning pane.

Example assign requirements to sprints



Monitor and report on progress

The three main tools you'll want to use to review progress and deliverables are:

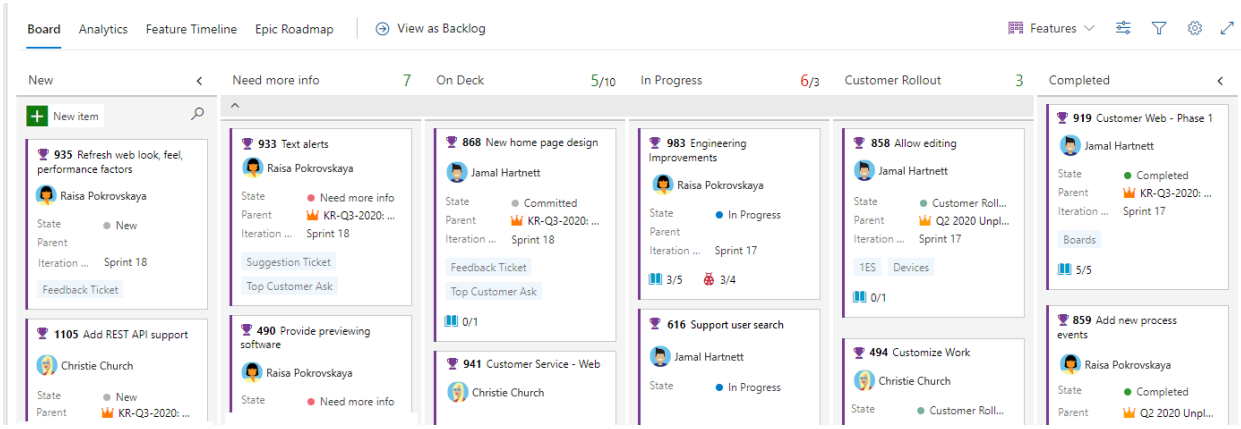
- Features Kanban board
- Features backlog with rollup columns
- Delivery plans

Features Kanban board

Your Features board is another place to review progress and ensure the continuous flow of deliverables. The following image illustrates a customized Features board. In progress columns have been added such as *Need more info*, *Spec Complete*, *In Progress*, and *Customer Rollout*. These column labels provide a more natural set of states as Features get proposed, researched, designed, developed, and then deployed to production.

Example of Features board with customized columns

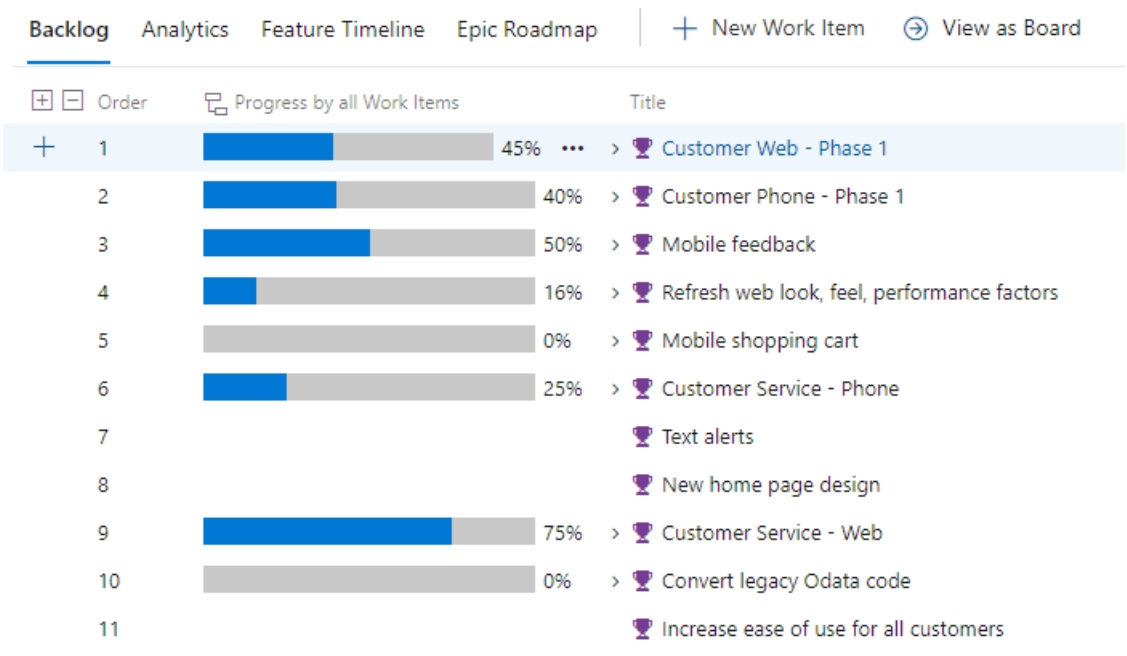
To see the full image, click the image to expand. Choose the ✕ *close icon to close.*



Rollup

One quick and visual way to monitor progress is from the Features backlog. By adding the rollup progress bar column, you can see what percentage of work items are completed for each feature, as shown in the following image.

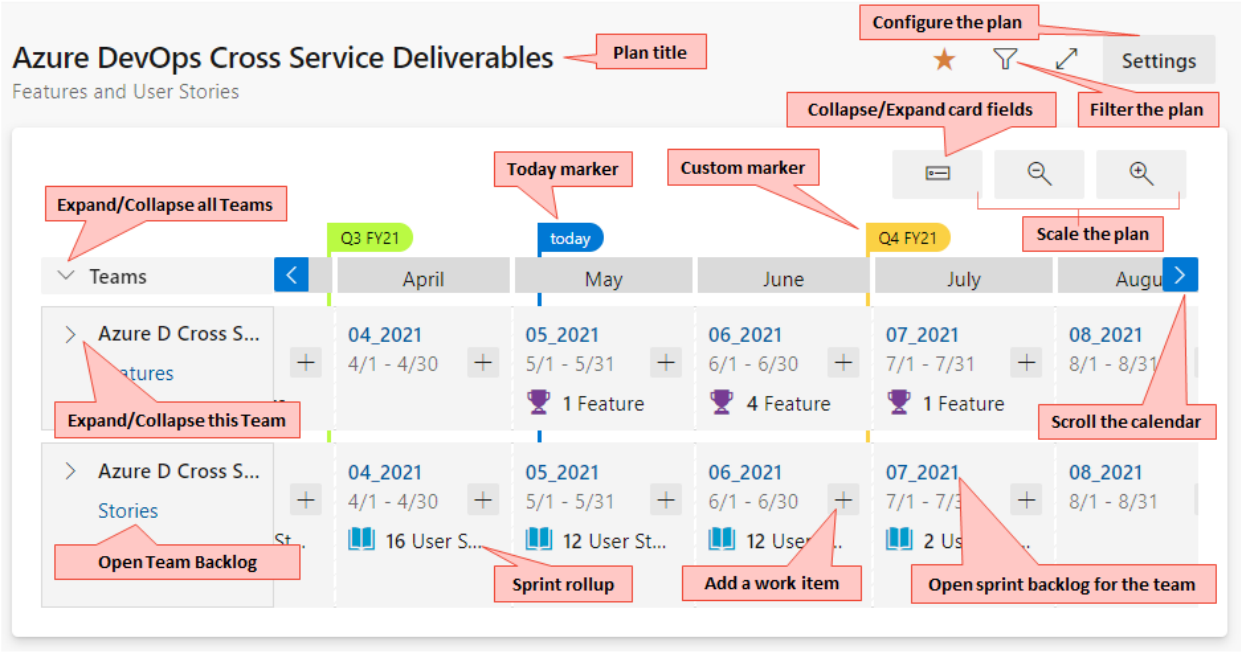
Example of Requirements backlog showing progress rollup

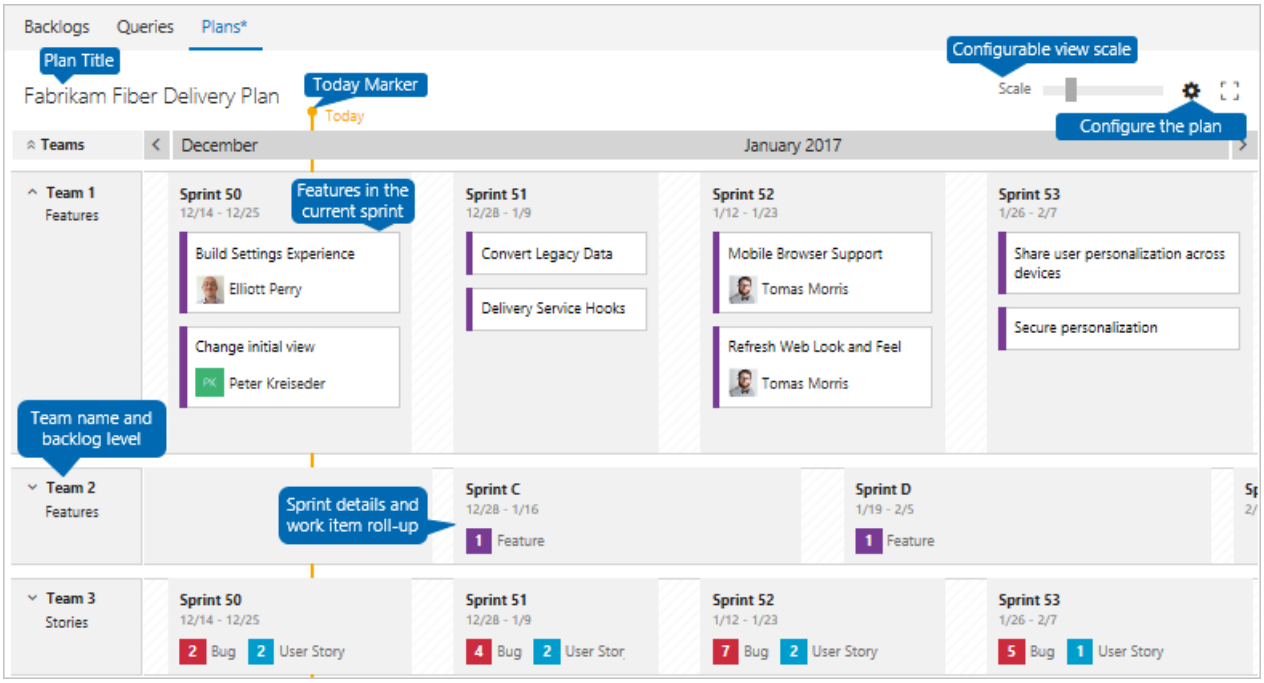


Delivery plans and multiple team deliverables

To review features delivered across several teams, configure a delivery plan. Delivery plans provide an interactive board to review a calendar schedule of stories or features several teams plan to deliver.

Example of multi-team Delivery Plan





Get notified of changes

Azure DevOps provides a robust alert system, allowing project members to set alerts for themselves, a team, or a project.

As changes occur to work items, code reviews, source control files, and builds, you can receive email notifications. For example, you can set an alert to be notified whenever a bug that you opened is resolved or a work item is assigned to you. You can set personal alerts, team, project, or organization alerts.

Related articles

To learn more about any of the concepts introduced in this article, refer to the following articles.

Agile and Agile culture

- What is Agile?
- Agile culture
- Best practices for "light-weight" Agile project management
- Scaling Agile - Practices that scale

Work items, work item types, and process models

- About work items
- Add work item tags to categorize and filter lists and boards
- Choose a process
- About process customization and inherited processes
- Bulk add or modify work items with Excel
- About Area and Iteration Paths (sprints)
- Work tracking, process, and project limits

Backlogs and boards

- Create your backlog
- Organize your backlog
- Define features and epics
- Refine your backlog
- About teams and Agile tools
- Tasks supported by Backlogs, Boards, Taskboards, and Plans
- Configure and customize Azure Boards

Kanban

- Start using your Kanban board
- Add columns to your Kanban board
- Customize cards
- Filter your Kanban board
- Kanban best practices

Scrum

- Assign backlog items to a sprint
- Configure and monitor sprint burndown
- Scrum and best practices

Dependency management

- Link user stories, issues, bugs, and other work items
- Triage work items

Milestone planning

- View or configure team velocity
- Forecast your product backlog
- The Critical Path on Agile Projects
- Running a lean startup on Azure DevOps

Monitor and report on progress

- Display rollup progress or totals
- Review team Delivery Plans

Maintain specifications and share information

- About Wikis, READMEs, and Markdown
- Share information within work items and social tools

Notifications

- Default and supported notifications
- Manage personal notifications
- Manage notifications for a team or group

Define, capture, triage, and manage software bugs in Azure Boards

10/4/2022 • 22 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

How do you track and manage defects in your code? How do you make sure software problems and customer feedback get addressed quickly to support high-quality software deployments? And, how do you make good progress on new features and address your technical debt?

At a minimum, you need a way to capture your software issues, prioritize them, assign them to a team member, and track progress. And, you want to manage your code defects in ways that align with your Agile practices.

To support these scenarios, Azure Boards provides a specific work item type to track code defects named Bug. Bug work items share all the standard features of other work item types with a few more. For an overview of standard features, see Track work with user stories, issues, bugs, features, and epics.

Bugs also provide the following additional features:

- Options for each team to choose how they want to track bugs
- Test tools to capture bugs
- Built-in integration across Azure DevOps to track bugs linked to builds, releases, and tests

> **NOTE**
> Bug work item types aren't available with the Basic process. The Basic process tracks bugs as Issues and is available when you create a new project from Azure DevOps Services or Azure DevOps Server 2019.1 or later versions.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this

permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.

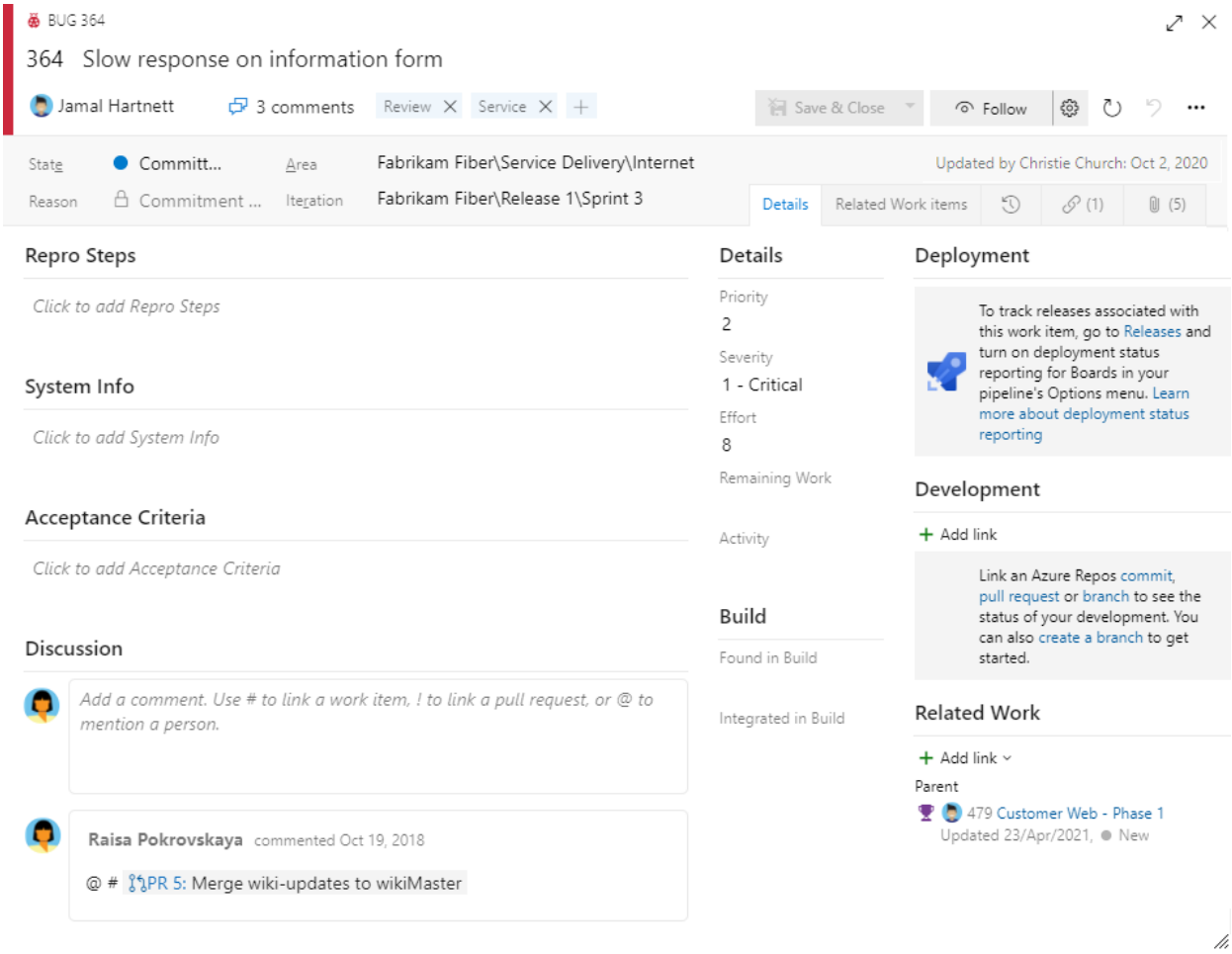
- All project members, even those who belong to the **Readers** group, can email work items.

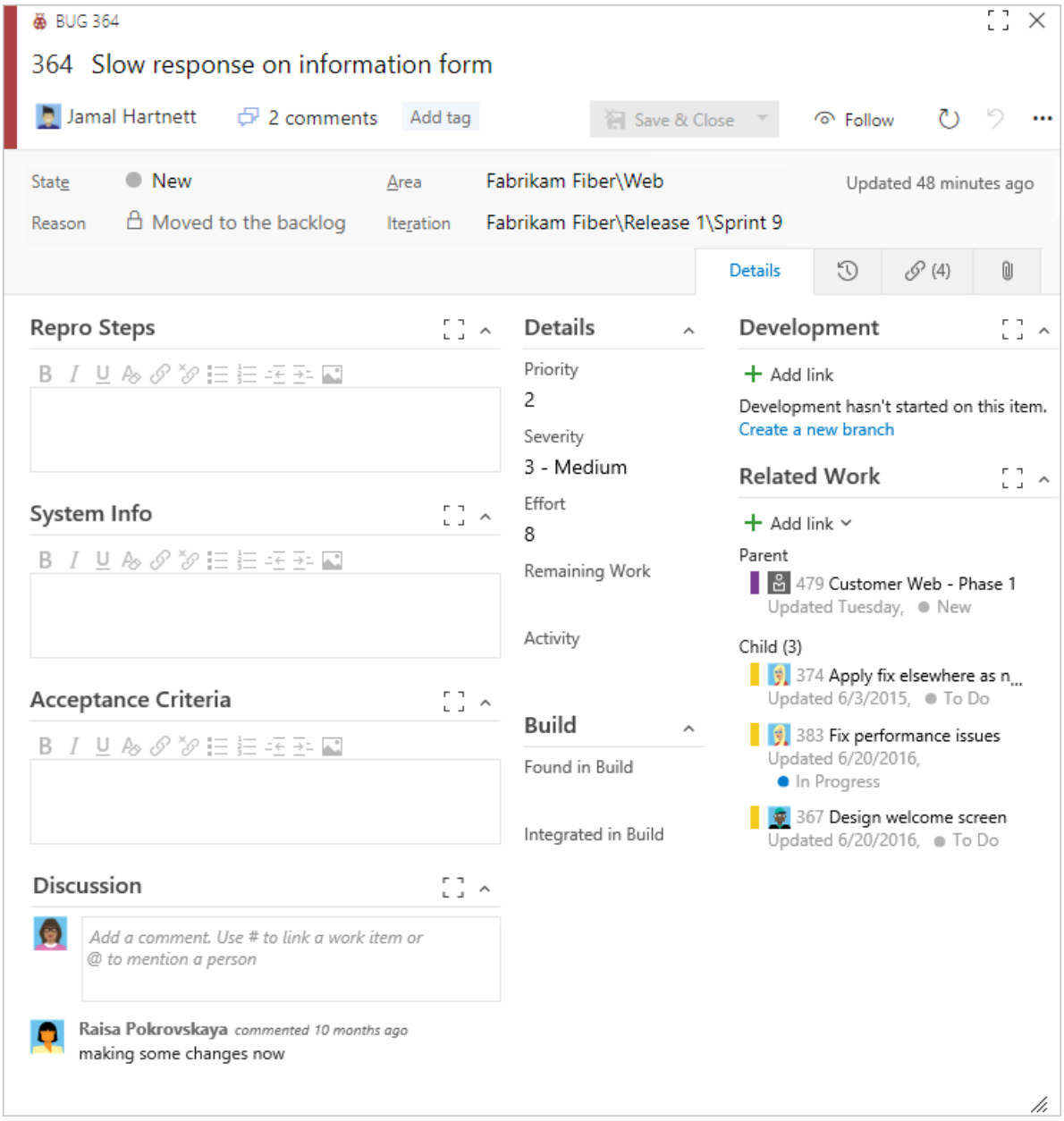
> **TIP**
> To report a bug, a user must have at a minimum, **Stakeholder** access and **Edit work items in this node** permission set to **Allow** for the **Area Path** where they will add the bug. To learn more, see Set permissions and access for work tracking

Bug work item type

The following image shows the Bug work item type for the Scrum process. The Bug work item type for Agile and CMMI processes tracks similar information. It's designed to appear on the product backlog along with requirements or on the Taskboard along with tasks.

> **NOTE**
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your web app, options that you or your admin have enabled, and which process was chosen when creating your project—Agile, Basic, Scrum, or CMMI. The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.





Fields specific to bugs

The Bug work item type uses some bug-specific fields. To capture both the initial issue and ongoing discoveries, use the fields described in the following table. For information about fields specific to the Bug defined for the Capability Maturity Model Integration (CMMI) process, see Bugs, issues, and risks field reference. For information about all other fields, see Work item field index.

Field, Group, or Tab

Usage

Steps to Reproduce
(friendly name=Repro Steps)

Use to capture enough information so that other team members can fully understand the code defect. Include actions taken to find or reproduce the bug and expected behavior.

System Info
Found In Build

Information about the software and system configuration that is relevant to the bug and tests to apply. The

System Info and **Found in Build** fields are automatically filled in when you create a bug through a testing tool. These fields specify information about the software environment and build where the bug occurred. To learn more, see Test different configurations.

Acceptance Criteria

Provide the criteria to meet before the bug can be closed. Before work begins, describe the customer acceptance criteria as clearly as possible. Teams should use this criteria as the basis for acceptance tests, and to evaluate whether an item has been satisfactorily completed.

Integrated in Build

Specifies the name of the build that incorporates the code that fixes the bug. This field should be specified when you resolve the bug. For on-premises Azure DevOps, to access a drop-down menu of all builds that have been run, you can update the `FIELD` definitions for **Found in Build** and **Integrated in Build** to reference a global list. The global list is automatically updated with each build that is run. To learn more, see Query based on build and test integration fields.
For information about how to define build numbers, see build number format options.

Priority[1]

- **1**: Product requires successful resolution of the work item before it ships and addressed soon.
- **2**: Product requires successful resolution of the work item before it ships, but doesn't need to be addressed immediately.
- **3**: Resolution of the work item is optional based on resources, time, and risk.

Severity[1]

A subjective rating of the impact of a bug or work item on the project or software system. For example: If a remote link within the user interface—a rare event—causes an application or web page to crash—a severe customer experience, you might specify **Severity = 2 - High** and **Priority = 3**. Allowed values and suggested guidelines are:

- **1 - Critical**: Must fix. A defect that causes termination of one or more system components or the complete system, or causes extensive data corruption. And, there are no acceptable alternative methods to achieve required results.
- **2 - High**: Consider fix. A defect that causes termination of one or more system components or the complete system, or causes extensive data corruption. However, an acceptable alternative method exists to achieve required results.
- **3 - Medium**: (Default) A defect that causes the system to produce incorrect, incomplete, or inconsistent results.
- **4 - Low**: A minor or cosmetic defect that has acceptable workarounds to achieve required results.

Deployment

The **Deployment** control supports links to and display of releases that contain work items. To use the control, you must enable settings for the release. To learn more, see Link work items to releases later in this article.

Development

The **Development** control supports links to and display of links made to development objects. These objects include Git commits and pull requests, or TFVC changesets and versioned items. You can define links from the work item or from the commits, pull requests, or other development objects. To learn more, see Link work items to development later in this article.

Notes:

[1] To change the menu selection or picklist, see Customize the work tracking experience. The customization method depends on the process model used by your project.

Choose how your team tracks bugs

Your team can track bugs as requirements or as tasks. To support the team choice, consider the following factors.

- Size of your team. Smaller teams can maintain a lightweight footprint by tracking bugs as requirements.
- Organization requirements to track work. If your team is required to track hours, then choose to track bugs as tasks.
- How your team organizes work. If your team relies on the product backlog to prioritize work and add bugs, track bugs as requirements.
- Tools your team wants to use such as the Planning pane, velocity chart, forecast, rollup, and delivery plans. Tracking bugs as tasks prevents use of several of these tools.

The following table summarizes the three options teams have to track bugs. To learn more and to set the option for your team, see Show bugs on backlogs and boards.

Option	Choose when you want to...
Track bugs as Requirements	- Prioritize (stack rank) bugs along with requirements - Estimate Bug effort for forecasting - Update bug status on Kanban board - Include Bugs in Velocity charts and Cumulative Flow Diagrams - Can use the **Forecast** tool to support sprint planning - Can drag and drop bugs onto **Planning** pane to assign bugs to a sprint - Can view Bugs on Delivery Plans **NOTE** - Bugs are assigned to the Requirements Category
Track bugs as Tasks	- Estimate work for bugs similar to tasks - Update bug status on sprint Taskboards - Link bugs to requirements as child items - Can drag and drop bugs onto Planning pane to assign bugs to a sprint **NOTE** - Bugs are assigned to the Task Category - User Stories (Agile), Product Backlog Items (Scrum), or Requirements (CMMI) are the natural parent work item type for Bugs - Bugs won't be visible on Delivery Plans

Bugs don't appear on backlogs or boards

- Manage bugs using queries

> **NOTE**
> - Bugs are associated with the Bugs Category and won't appear on either backlogs or boards
> - Bugs won't be visible on Backlogs, Boards, Sprint Backlogs, Taskboards, or Delivery Plans
> - Can't drag and drop bugs onto Planning pane to assign bugs to a sprint

Customize work item type

You can customize the Bug and other work item types. Or, create custom types to track software issues or customer feedback. With all work item types, you can customize the following elements:

- Add or remove custom fields
- Add custom controls or custom tabs within the work item form
- Customize the workflow states
- Add conditional rules
- Choose the backlog level in which work items appear

Before you customize your process, we recommend you review Configure and customize Azure Boards.

To customize your particular process, see Customize an inheritance process.

To customize your particular process, see Customize an inheritance process or Customize the On-premises XML process model.

To customize your particular process, see Customize the On-premises XML process model.

Add or capture bugs

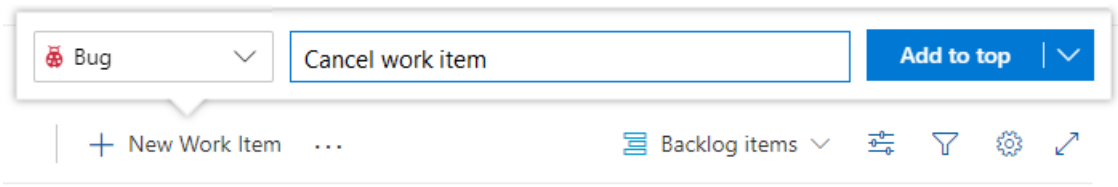
You can define bugs from several different Azure DevOps tools. These include backlogs and boards and testing tools.

> **TIP**
> By default, the **Title** field is the only required field when creating a bug. You can quickly add bugs in the same way you add user stories or product backlog items using Azure Boards. If you want to make some fields required, do that by adding conditional rules based on a state change. To learn more, see Add a rule to a work item type (Inheritance process).

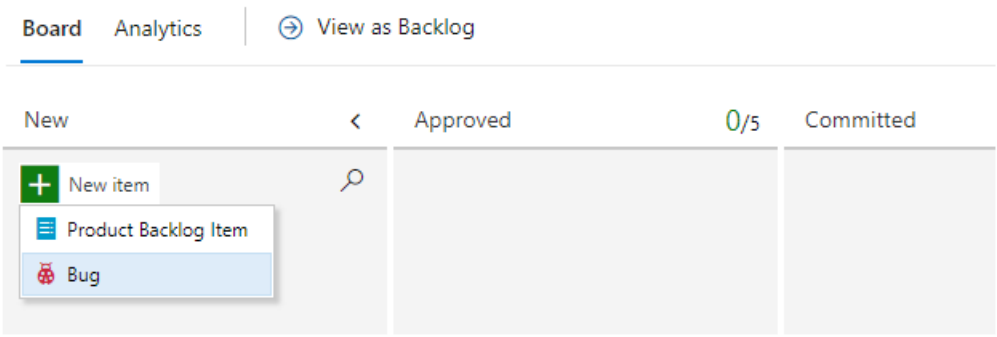
Add a bug from your backlog or board

If your team chose to *manage bugs with requirements*, you can define bugs from your product backlog or Kanban board. To learn more, see Create your product backlog or Start using your Kanban board.

- **Add a bug from the product backlog**



- **Add a bug from the product backlog**



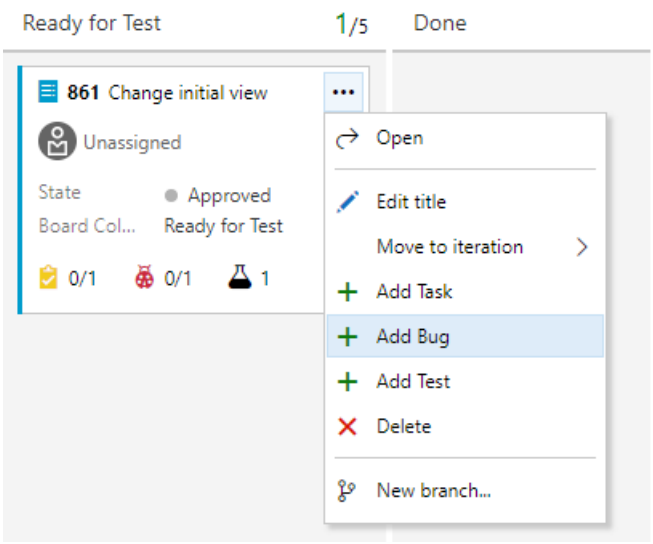
TIP

When you add a bug from your product backlog or Kanban board, the bug is automatically assigned the default Area Path and Iteration Path defined for the team. To learn more, see Team defaults referenced by backlogs and boards.

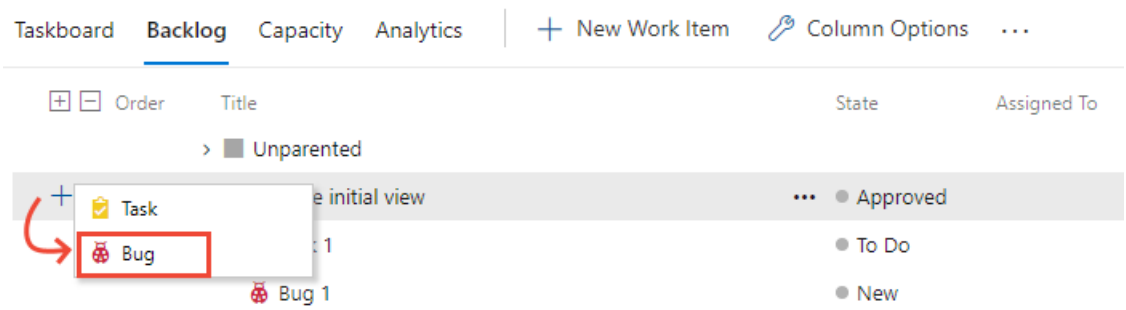
Add a bug from your sprint backlog or Taskboard

If your team chose to *manage bugs with tasks*, you can define bugs from your Kanban board, product backlog, Sprint backlog, or Sprint Taskboard. You add a bug as a child to a product backlog work item.

- **Add a linked child bug from the Kanban board**
  You add a bug in the same way you add a task to a backlog item. To learn more, see Add tasks or child items as checklists.



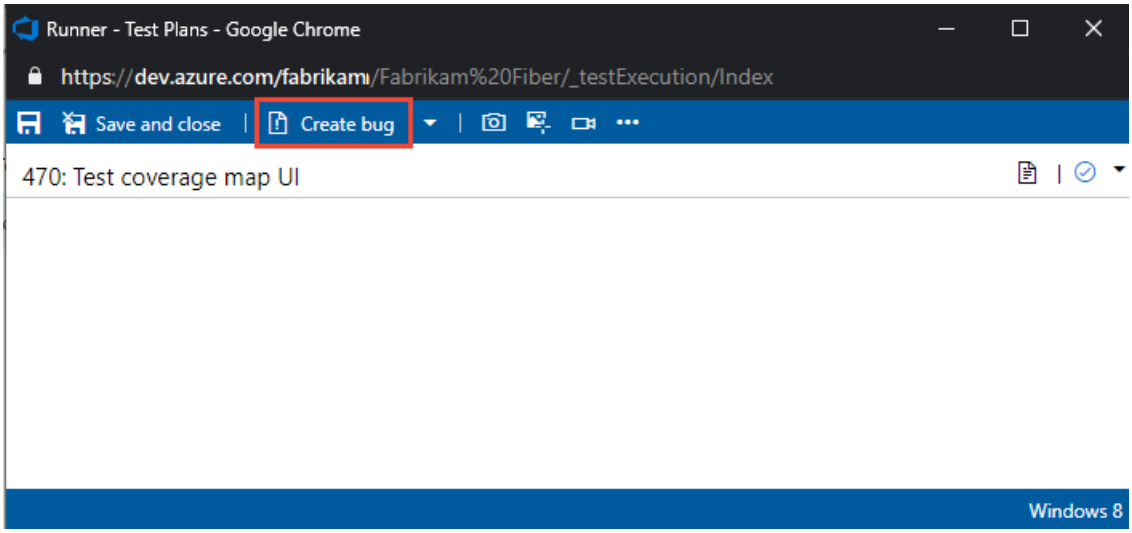
- **Add a linked child bug from the Sprint Backlog**
  You add a bug in the same way you add a task to a Sprint backlog. To learn more, see Add tasks to backlog items.



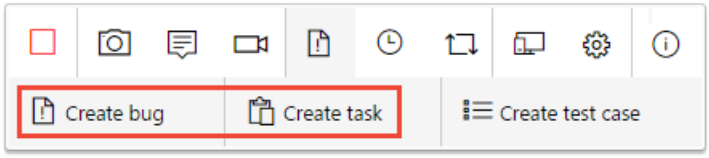
Create a bug from a testing tool

The two testing tools you can use to add bugs while testing include the web portal Test Runner and the Test & Feedback extension.

- **Test Runner**: When running manual tests, you can choose to **Create bug**. To learn more, see Run manual tests.

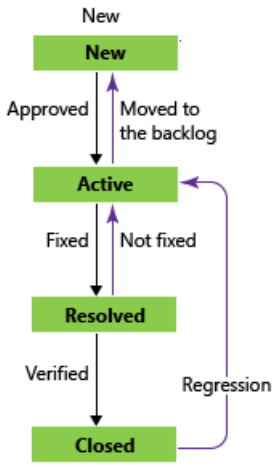
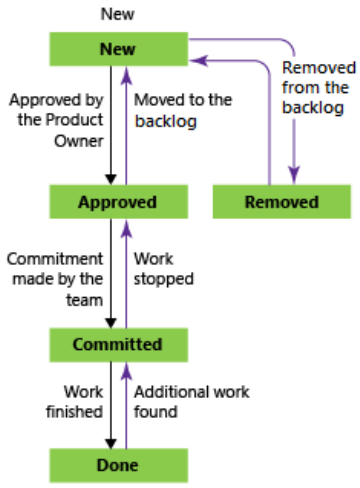
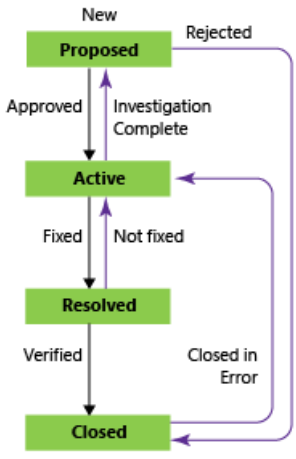


- **Test & Feedback extension**: When running exploratory tests, you can choose to **Create bug** or **Create task**. To learn more, see Exploratory testing with the Test & Feedback extension



Bug lifecycle and workflow states

As with all other work item types, the Bug work item type has a well-defined workflow. Each workflow consists of three or more **States** and a **Reason**. Reasons specify why the item transitioned from one State to another. The following images illustrate the default bug workflow defined for the Agile, Scrum, and CMMI processes.

AGILE	SCRUM	CMMI
		

For Scrum bugs, you change the **State** from *Committed* (similar to *Active*) to *Done*. For Agile and CMMI, you first resolve the bug and select a reason that indicates the bug is fixed. Typically, the person who created the bug then verifies the fix and updates the **State** from *Resolved* to *Closed*. If more work has been found after a bug

has been resolved or closed, you can reactivate it by setting the **State** to *Committed* or *Active*.

> **NOTE**
> The Agile process bug work item type previously had a rule which reassigned the bug to the person who created it. This rule has been removed from the default system process. You can reinstate this automation by adding a rule. For an Inheritance process, see Apply rules to workflow states, Automate reassignment based on state change.

Verify a fix

To verify a fix, a developer or tester attempts to reproduce the bug and look for more unexpected behavior. If necessary, they should reactivate the bug.

When verifying a bug fix, you might find that the bug wasn't fixed or you might disagree with the resolution. In this case, discuss the bug with the person who resolved it, come to an agreement, and possibly reactivate the bug. If you reactivate a bug, include the reasons for reactivating the bug in the bug description.

Close a bug

You close a bug once it's verified as fixed. However, you might also close a bug for one of the following reasons. Reasons available to select depend on the project process and the bug transition states.

Agile process:

- **Deferred**: Defer bug fix until the next product release.
- **Fixed**: Bug is verified as fixed.
- **Duplicate**: Bug tracks another bug currently defined. You can link each bug with the **Duplicate/Duplicate of** link type and close one of the bugs.
- **As Designed**: Feature works as designed.
- **Cannot Reproduce**: Tests prove that the bug can't be reproduced.
- **Obsolete**: The bug's feature is no longer in the product.
- **Copied to Backlog**: A user story has been opened to track the bug.

Scrum process:

- **Not a Bug**: Bug is verified that it isn't a bug.
- **Duplicate**: Bug tracks another bug currently defined. You can link each bug with the **Duplicate/Duplicate of** link type and close one of the bugs.
- **Removed from the backlog**: Bug is verified that it isn't a bug. Remove the bug from the backlog.
- **Work finished**: Bug has been verified as fixed.

CMMI process:

- **Deferred**: Defer bug fix until the next product release.
- **Duplicate**: Bug tracks another bug currently defined. You can link each bug with the **Duplicate/Duplicate of** link type and close one of the bugs.
- **Rejected**: Bug is verified that it isn't a bug.
- **Verified**: Bug is verified as fixed.

> **TIP**
> Once a bug has been closed and the fix is actively released in deployments, recommended practice is to never reopen it due to regression. Instead, you should consider opening a new bug and link to the older, closed bug.

It's always a good idea to describe any more details for closing a bug in the **Discussion** field to avoid future

confusion as to why the bug was closed.

Automate bug closure when merging pull requests

If your team uses a Git repository, you can set the State in linked bugs and other work items to close upon successful merging of pull requests. For more information, see Set work item state in pull request later in this article.

List and triage bugs

Most teams, whatever option they chose to track bugs, define one or more bug queries. With queries, you can list active bugs, unassigned bugs, stale bugs, bug trends, and more. You can then add queries and query charts to your team dashboards to monitor bug status and progress.

Bug queries

Open a shared query or use the query editor to create useful bug queries, such as the following options:

- Active bugs by priority (`State <> Done` or `State <> Closed`)
- In Progress bugs (`State = Committed` or `State = Active`)
- Bugs to fix for a target release (`Tags Contains RTM`)
- Recent bugs - bugs opened within the last three weeks (`Created Date > @Today-21`)

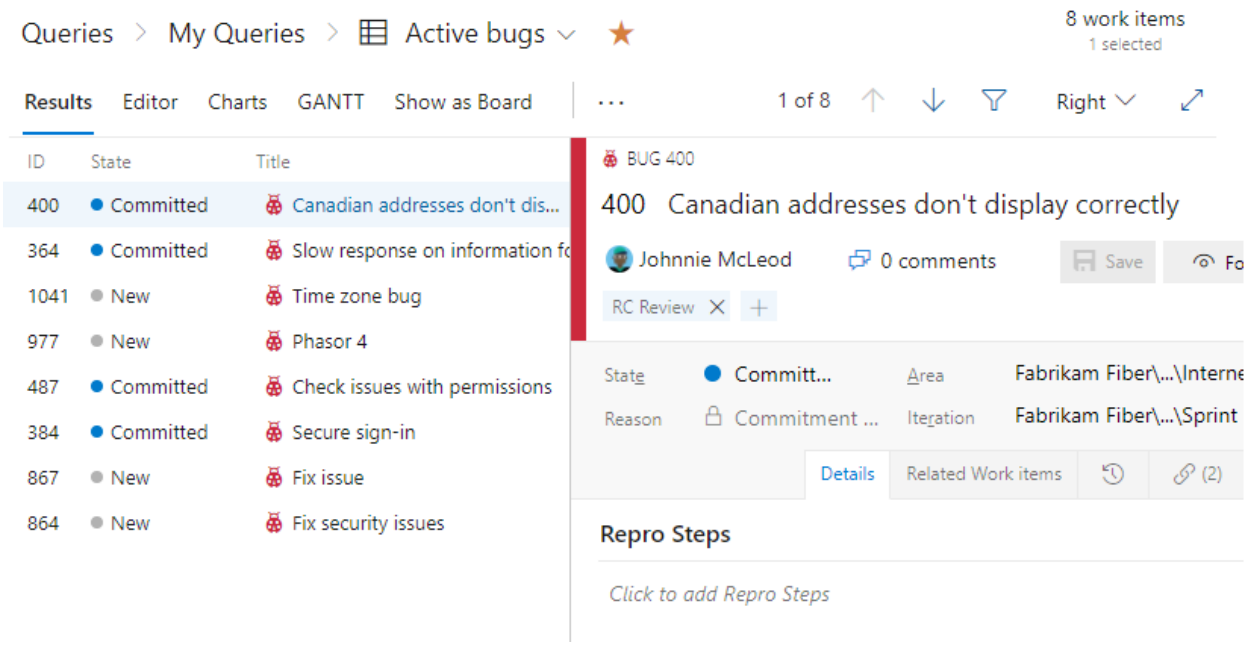
Once you have the queries of interest to your team, you can create status or trend charts. You can also add the chart you create to a dashboard.

Triage mode in query results

Once you've started coding and testing, you'll want to hold periodic triage meetings to review and rank your bugs. Typically, the project owner runs the bug triage meetings. Team leads, business analysts, and other stakeholders who can speak about specific project risks attend the triage meetings.

The project owner can define a shared query for new and reopened bugs to list bugs to be triaged.

From the query results page, you can quickly move up and down within the list of bug work items using the up and down arrows. As you review each bug, you can assign it, add details, or set priority. To learn more, see Triage work items.



Organize and assign bugs to a sprint

If your team *tracks bugs as requirements*, view the list of active bugs from your backlog. With the filter function,

you can focus solely on bugs. From the product backlog, you can also do the following tasks:

- Organize bugs on your backlog, stack rank against other items (stack ranking is disabled when filtering is enabled)
- Assign bugs to a sprint from your backlog using the **Planning** pane
- Parent bugs to Features or other portfolio backlog items using the **Mapping** pane
- View rollup of work to portfolio backlog items.

If your team *tracks bugs as tasks*, use managed queries to list and triage bugs. Then, within each sprint, you'll see the bugs assigned to the sprint from the Sprint backlog or Taskboard.

Taskboard items versus query list items

You might notice and wonder why the items shown on a sprint Taskboard can differ from a query list created in a corresponding sprint backlog.

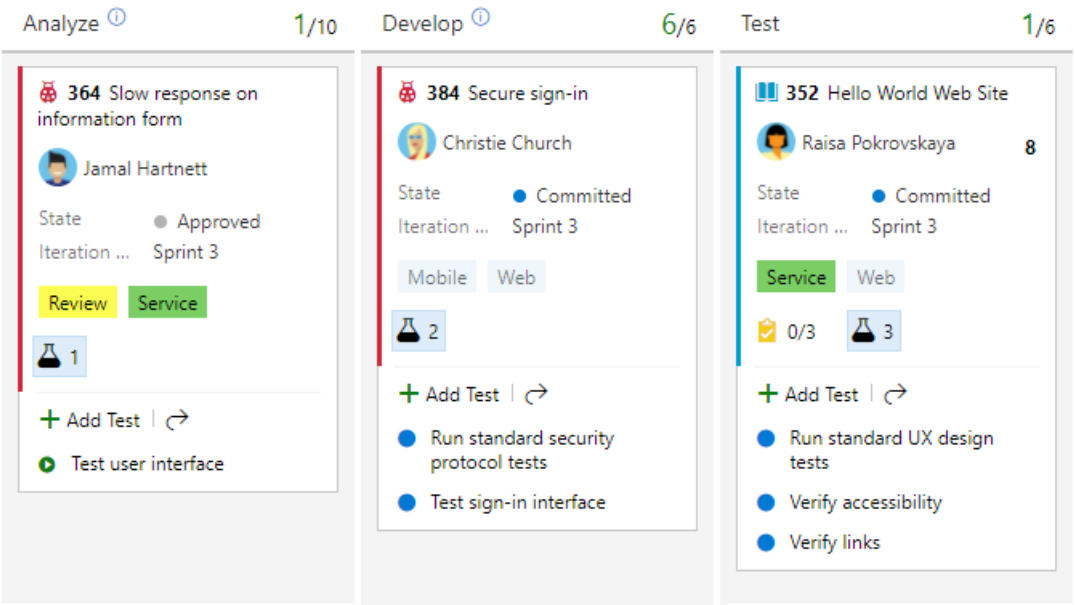
It's possible to assign tasks or bugs to an iteration but not have them linked to a parent backlog item. These items will show up in the created query, but might not show up on the Taskboard itself. The system runs the query and then applies a few background processes before displaying Taskboard items.

These reasons can cause work items that belong to the Task Category to not appear on a sprint backlog or Taskboard:

- The task or bug hasn't been linked to a parent backlog item. Only bugs and tasks you've linked to a parent product backlog item (Scrum), user story (Agile), or requirement (CMMI) with an iteration path set to the sprint appears on the sprint backlog page.
- The task or bug is a parent of another task or bug, or the user story is a parent of another user story. If you've created a hierarchy of tasks, bugs, or user stories, only the child-level tasks or the child-level stories at the bottom of the hierarchy appear.
- The task's or bug's linked parent corresponds to a backlog item defined for another team. Or, the area path of the task's or bug's parent backlog item differs from the task's or bug's area path.

Create inline tests linked to bugs

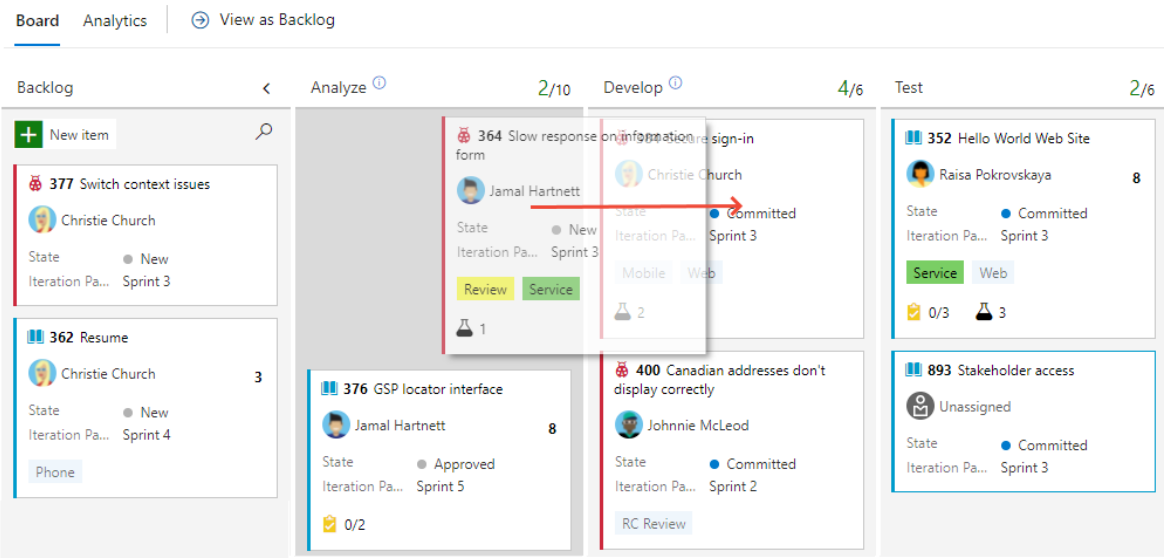
When your team *tracks bugs as requirements*, you can use the Kanban board to add tests to verify bug fixes.



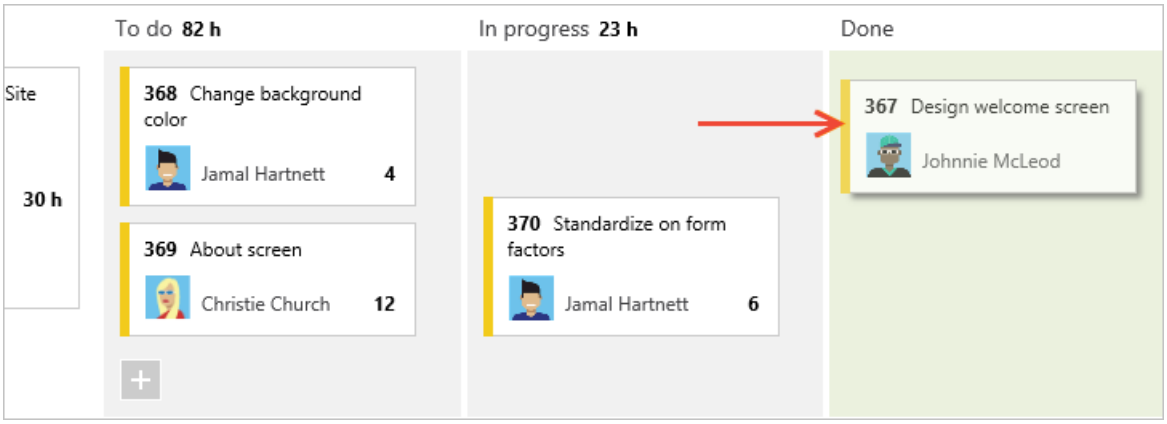
Update bug status

You can update the bug status by dragging and dropping bugs to a new column on a board.

- If your team *tracks bugs as requirements*, you use the Kanban board as shown in the following image. To learn more, see Get started with your Kanban board.



- If your team *tracks bugs as tasks*, you use the Taskboard. To learn more, see Update and monitor your Taskboard.



Customize your board to track intermediate states

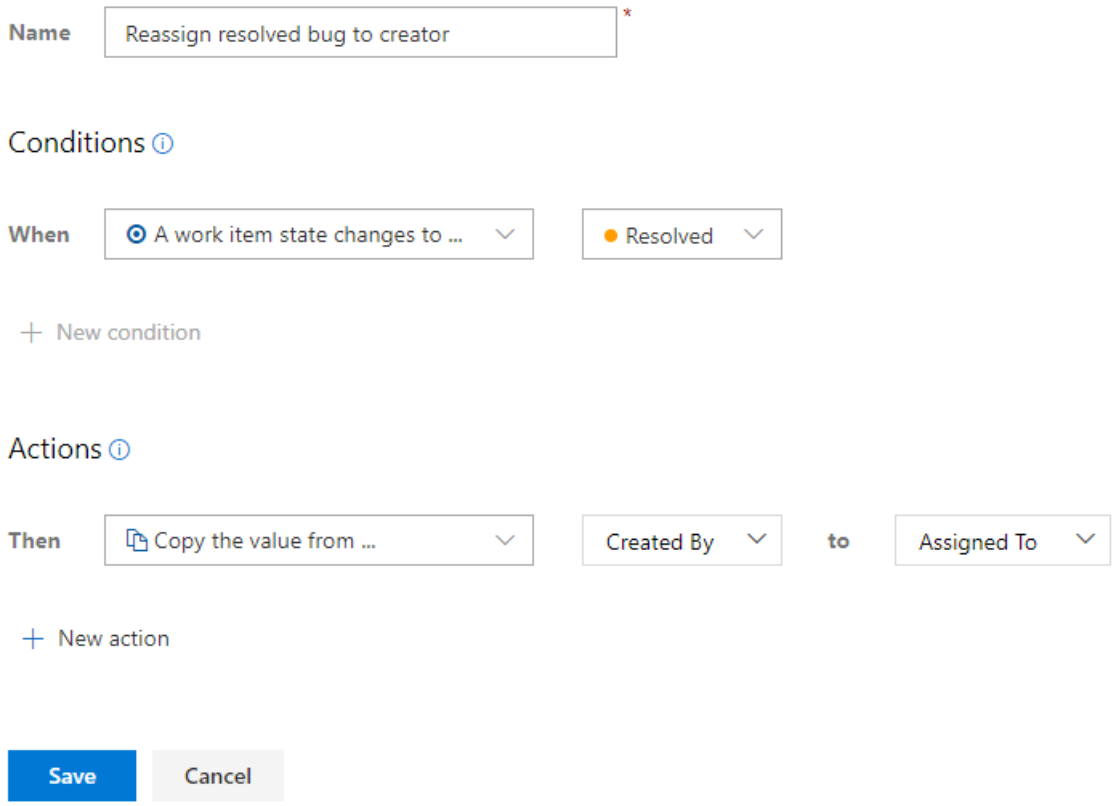
You can add intermediate columns to track your bug status on the board. You can also define queries that filter based on the status of a Board Column. To learn more, see the following articles:

- Add columns to your Kanban board
- Customize a sprint Taskboard
- Kanban board change queries

- Add columns to your Kanban board
- Kanban board change queries

Automate bug reassignment based on workflow state

To automate select actions, add custom rules to your Bug work item type. For example, add a rule as shown in the following image. This rule specifies to reassign a bug to the person who opened the bug once it's resolved. Typically, that person verifies that the bug is fixed and closes the bug. To learn more, see Apply rules to workflow states (Inheritance process).



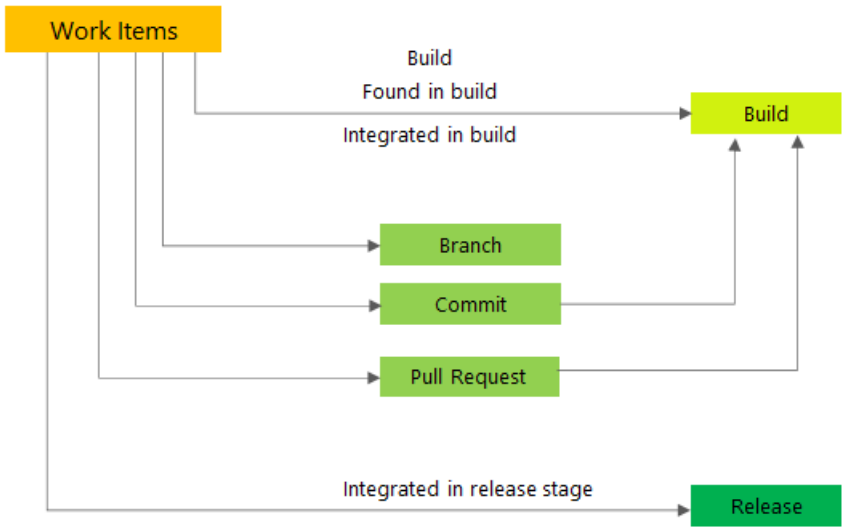
Set work item state in pull request

When you create a pull request, you can set the *state* value of the linked work items in the description. Follow the syntax: `{state value}: #ID`. When you merge the pull request, the system reads the description and updates the work item state. In the following example, we set work items #300 and #301 to Resolved, and #323 and #324 to Closed.

> **NOTE**
> This feature requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

Integration across Azure DevOps

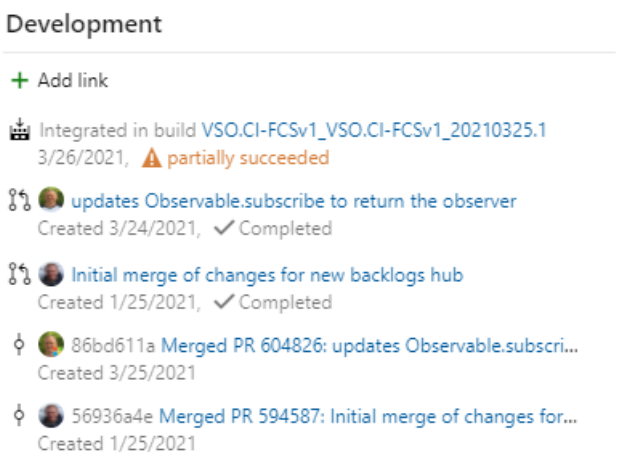
One of the methods used by Azure DevOps to support integration is to link objects to other objects. Along with linking work items to work items, you can also link work items to other objects. Link to objects such as builds, releases, branches, commits, and pull requests as illustrated in the following image.



You can add a link from the work item or from the build and release objects.

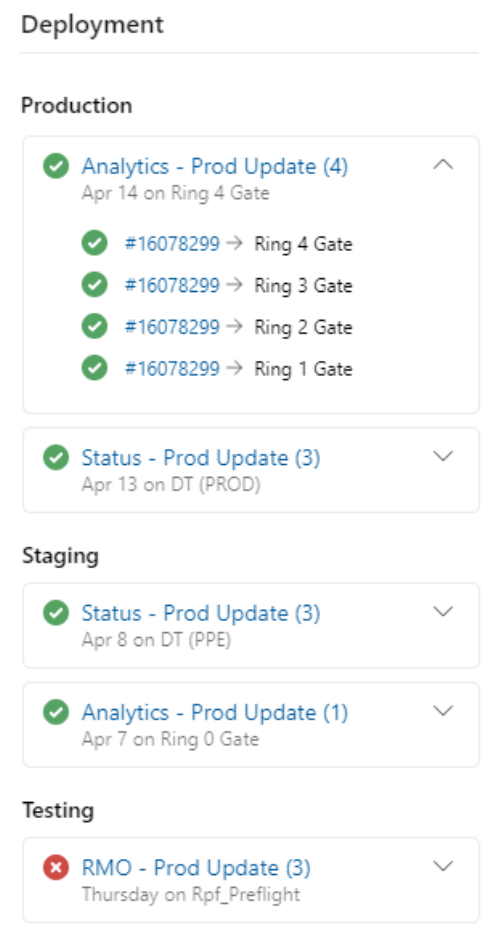
Link work items to development

The **Development** control supports linking to and displaying links made to builds, Git commits and pull requests. Or, when a TFVC repository is used, it supports links to changesets and versioned items. Choosing the link opens the corresponding item in a new browser tab. To learn more, see Drive Git development from a work item.



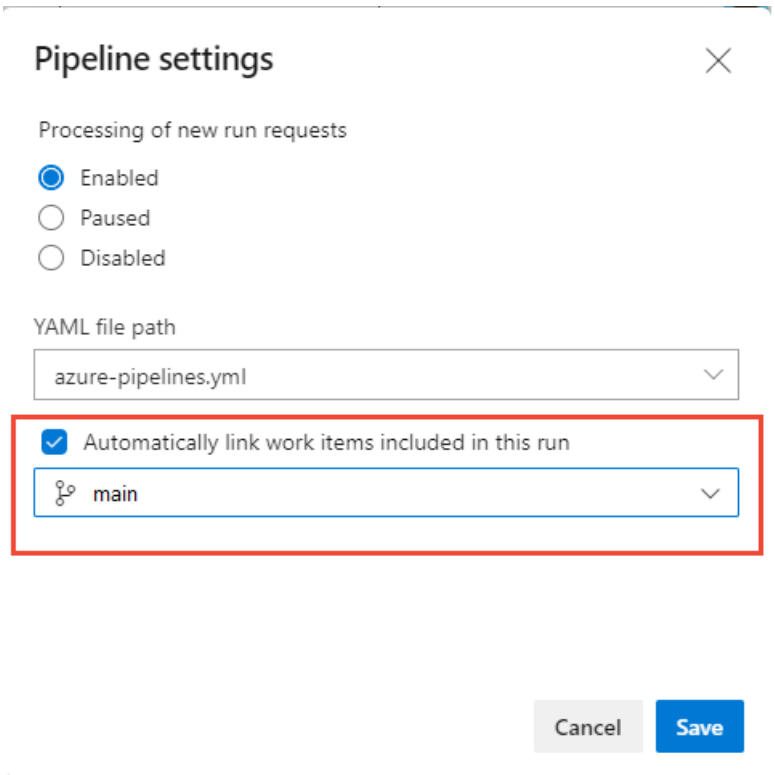
Link work items to releases

The **Deployment** control supports links to and display of releases that contain the work items. For example, the following image shows several releases that contain links to the current work item. You can expand each release to see details about each stage. You can choose the link for each release and stage to open the corresponding release or stage. To learn more, see Link work items to deployments.



Link work items to pipeline runs

Pipelines are often defined to automatically run when a new commit occurs to a Git repository. Work items associated with the commit pipelines will show up as part of the pipeline run if you customize your pipeline settings. To learn more, see Customize your pipeline.



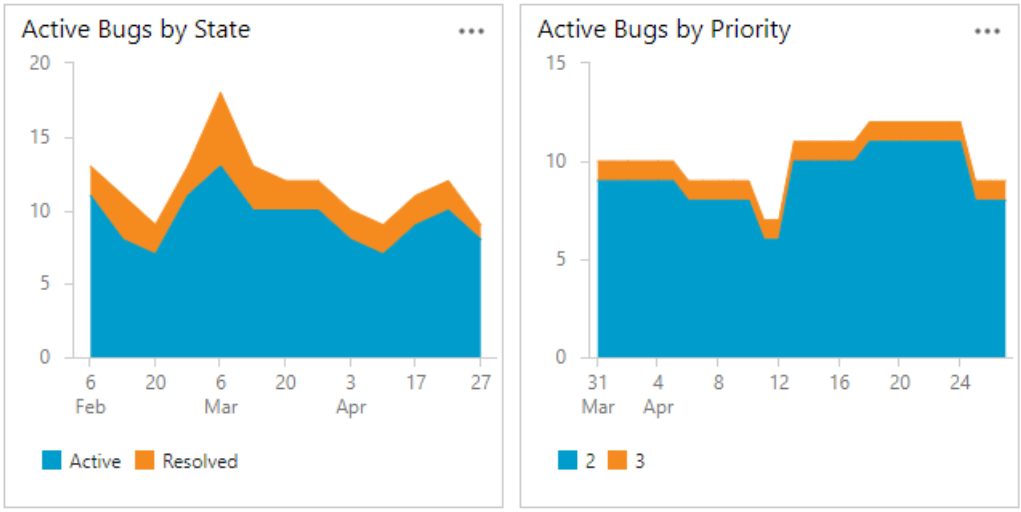
Create or edit a work item upon a build failure

If you use classic pipelines (not YAML), you can create work items on a build failure. For details, see Build

options, Create a work item on failure.

Monitor bug status, assignments, and trends

You can track the bug status, assignments, and trends using queries that you can then chart and add to a dashboard. For example, here are two examples showing active bug trends by State and Active Bugs by Priority over time.



To learn more about queries, charts, and dashboards; see About managed queries and Charts, and Dashboards.

Use Analytics views and the Analytics service to create bug reports

The Analytics service is the reporting platform for Azure DevOps, replacing the previous platform based on SQL Server Reporting Services.

Analytics views provide pre-built filters to view work items. Four Analytic views are supported for bug reporting. You can use these views as defined or further edit them to create a custom, filtered view.

- Bugs - All history by month
- Bugs - Last 26 weeks
- Bugs - Last 30 days
- Bugs - Today

To learn more about using Analytic views, see What are Analytics views and Create an active bugs report in Power BI based on a custom Analytics view.

You can use Power BI to create more complex reports than what you can get from a query. To learn more, see Connect with Power BI Data Connector.

Pre-defined SQL Server bug reports

The following reports are supported for Agile and CMMI processes.

- Bug Status
- Bug Trends
- Reactivations

These reports require you have SQL Server Analysis Services and SQL Server Reporting Services configured for your project. To learn how to add SQL Server reports for a project, see Add reports to a project.

Marketplace extensions

There are multiple bug-related Marketplace extensions. See Marketplace for Azure DevOps.

For more information on extensions, see Azure Boards extensions developed by Microsoft.

Try this next

Triage work items

Related articles

- Use templates to add and update work items
- Move, change type, or delete work items
- Copy or clone a work item

Product backlog and Kanban board

- Backlogs, portfolios, and Agile project management
- Create your backlog
- Define features and epics
- Organize your backlog, map child work items to parents
- Interactively filter backlogs, boards, queries, and plans
- Forecast your product backlog
- Refine your backlog

Kanban board

- About Boards and Kanban
- Kanban board quickstart
- Reorder cards
- Add tasks or child items as checklists
- Kanban best practices

Sprint backlog and Taskboard

- Scrum and working with sprints best practices
- Assign backlog items to a sprint
- Add tasks
- Update the Taskboard

Integration within Azure DevOps

- Link user stories, issues, bugs, and other work items
- Follow a work item or pull request
- Configure run or build numbers

Industry resources

- Good and Bad Technical Debt (and how TDD helps) by Henrik Kniberg
- Managing Technical Debt posted by Sven Johann & Eberhard Wolff

Manage issues or impediments in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

If you have known issues you want to track, you can do so by defining an impediment (Scrum) or issue (Agile or CMMI). Impediments and issues represent unplanned activities. Resolving them requires more work beyond what's tracked for actual requirements. Use the impediment work item type to help you track and manage these issues until you can resolve and close them.

Don't confuse impediments with bugs. You track impediments that may cause problems with delivering one or more requirements. For example, you may have to fix feature ambiguity, personnel or resource issues, problems with environments, or other risks that influence scope, quality, or schedule. Other issues that deserve tracking are decisions that require several stakeholders or product teams to weigh in on.

> **IMPORTANT**
> Issues and Impediments discussed in this article are defined for projects created with the Agile, Scrum, or CMMI process. By default, these work item types don't appear on the product backlog or taskboard.
>
> If your project was created using the Basic process, which tracks work using Epics, Issues, and Tasks, then you track Issues using the product backlog. To learn more, see Track issues and tasks.

> **IMPORTANT**
> Issues and Impediments discussed in this article are defined for projects created with the Agile, Scrum, or CMMI process. By default, these work item types don't appear on the product backlog or taskboard.

In this article you'll learn:

- When to use issues versus tasks
- How to capture issues or impediments as a work item
- Add issues or impediments to your product backlog

- When to use issues versus tasks
- How to capture issues or impediments as a work item

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.

- All project members, even those who belong to the **Readers** group, can email work items.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

> **NOTE**
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your web app, options that you or your admin have enabled, and which process was chosen when creating your project—Agile, Basic, Scrum, or CMMI. The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.

Define a task

You use issues or impediments to track items that may block work from getting done. In general, you link these items to user stories or other work items using a Related link type.

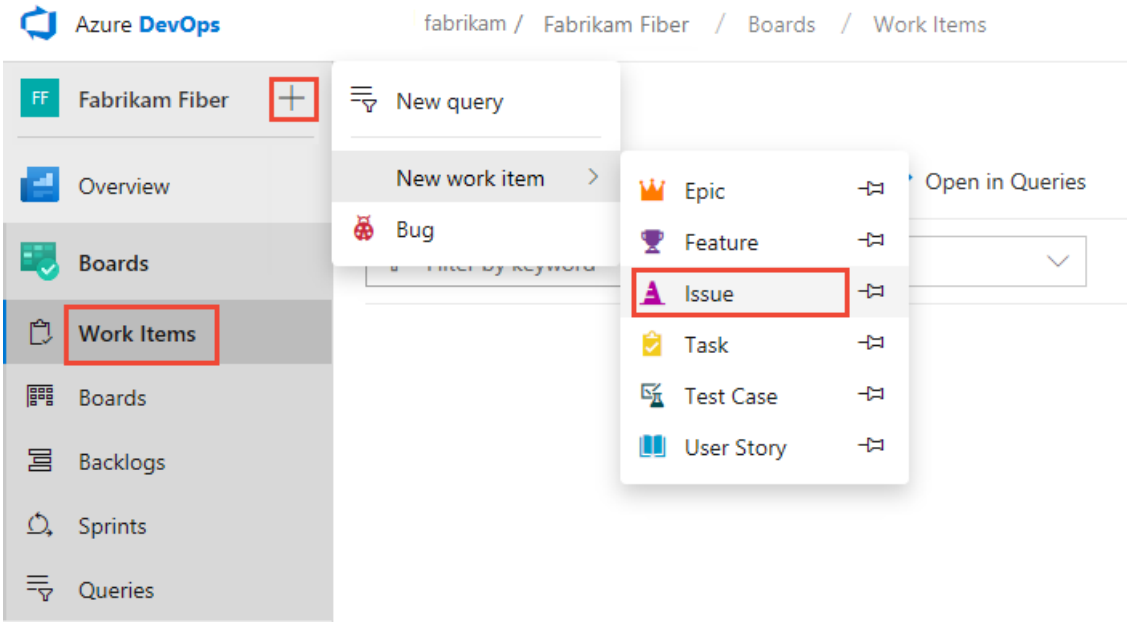
Define tasks when you want to create a checklist of tasks. You can also define tasks if you use Scrum methods and track work using the Remaining Work field. By linking requirement work item types to tasks using the Parent-Child link type, the tasks appear on the taskboard for each linked user story.

> **NOTE**
> If your project collection uses the On-premises XML process model to customize work tracking, you can enable work item types that you add to the Task Category to appear as a checklist on your product Kanban board. To learn how, see Set up your backlogs and boards, Customize your Kanban Board checklist items.

If you want to add these work item types to a backlog, see Customize your backlogs or boards.

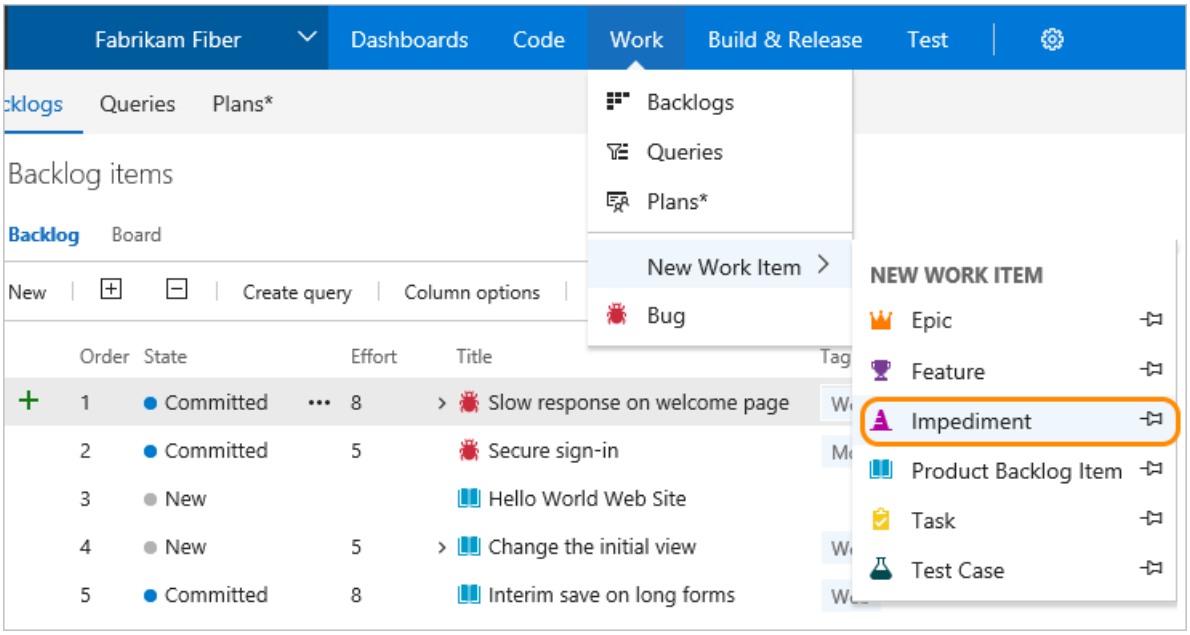
Add an issue or impediment

Open **Boards>Work Items**, and choose the + plus icon, and then select from the **New work item** menu of options.



Choose the pin icon to have it show up within the add drop down menu.

From **Work**, choose **Impediment** from the **New Work Item** list of options. Choose the pin icon to have it show up within **Work** drop down menu.



Customize issue tracking

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize an inheritance process or Customize the On-premises XML process model depending on the process model used by your project.

You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For details, see Customize the On-premises XML process model.

Issues and impediments don't appear on your backlog by default. Instead, you track them using queries. To track them on a backlog, see the next section, Add issues or impediments to your product backlog.

Impediments and issues don't appear on your backlog. Instead, you track them using queries. They only appear on your backlog if your project is customized using the On-premises XML process model. To learn more, see Customize the On-premises XML process model.

Impediments and issues don't appear on your backlog. Instead, you track them using queries. If you want them to appear on your backlog, or you want to track other work item types on your backlog, see Add a work item type to a backlog and board.

Add issues or impediments to your product backlog

If you want to track issues or impediments along with your requirements or a portfolio backlog, you can track them by adding them to your custom Inherited process. For details, see Customize your backlogs or boards (Inheritance process).

If you want to track issues or impediments along with your requirements or a portfolio backlog, you can track them by customizing your project's process. For details, see the following:

- **For the Inherited process**: Customize your backlogs or boards (Inheritance process).
- **For the On-premise XML process**: Process configuration XML element reference).

If you want to track issues or impediments along with your requirements or a portfolio backlog, you can track them by adding them to your On-premise XML process. For details, see Process configuration XML element reference).

Related articles

- Add work items
- Work item form controls
- Manage bugs or code defects
- Create your backlog

Link user stories, issues, bugs, and other work items in Azure Boards

10/4/2022 • 15 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Link work items to other work items to manage dependencies and see relationships within work. You can link work items within your project or to work items within another project in your organization. Use different link types to support different business goals. You can also link work items to other objects such as builds, commits, versioned items, or network resources.

Link work items to other work items to manage dependencies and see relationships within work. You can link work items within your project or to work items within another project in your collection. Use different link types to support different business goals. You can also link work items to other objects such as builds, commits, versioned items, or network resources.

In general, you use the following link types when linking work items to one another:

- Use **Parent/Child** links to link work items that you want to group within a hierarchy
- Use **Predecessor/Successor** or **Affects/Affected by** links to link work items that have dependencies
- Use **Duplicate/Duplicate of** links to link work items that track the same code defect or work
- Use **Related** to link work items with some level of relationship, but not that strong

You can add a link to a work item from within the work item form or from a backlog or query results list. From a backlog or query results list, you can select multiple work items and then link them to a new or existing work item. In general, use the bulk edit to update several work items to link to the same work item, either new or existing.

Use this article to learn how to:

- Link one or more work items to an existing work item
- Link one or more work items to a new work item that you add when linking
- Add a link to a remote work item
- Link several work items to a new git branch
- Link work items to a build artifact
- Find work items that you want to link to
- Bulk modify link relationships

- Link one or more work items to an existing work item
- Link one or more work items to a new work item that you add when linking
- Link several work items to a new git branch
- Link work items to a build artifact
- Find work items that you want to link to
- Bulk modify link relationships

> **NOTE**
> If you want to link work items in a parent-child hierarchy, use the mapping pane as described in Organize your backlog and map child work items to parents. If you want to link test cases to user stories, see Add, run, and update inline tests and Create test plans and test suites.

For an overview of how links are used to support traceability, see End-to-end traceability.

Link guidance

For an overview of which link types to use and link-related capabilities, see Linking, traceability, and managing dependencies. In general, we recommend you follow these guidelines:

- For work items that appear on your backlogs, both product and portfolio, use the **Parent** and **Child** link types to create a hierarchy and group work. To quickly link many backlog work items within a hierarchy, see Organize your backlog, map child work items to parents.
- When linking work items with **Parent** and **Child** link types, avoid nesting work items of the same type. While the system allows you to nest work items of the same type, such as linking bugs to bugs or bugs to user stories when tracking both types on your product backlog, it can cause problems with drag-and-drop of work items on a backlog or display of items on a Kanban board. To learn more, see Fix display, reordering, and nesting issues.
- To track dependencies of work items, use the **Predecessor** and **Successor** link types.
- For all other general tracking purposes, use the **Related** link type.

The following link relationships are restricted:

- You can't assign a work item more than one parent using a **Parent/Child** or other tree-topology link type. To learn more about link types, see Link type reference.
- You can't add links in such a way as to create a circular relationship.
- You can't link more than 1,000 work items to a single work item.

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

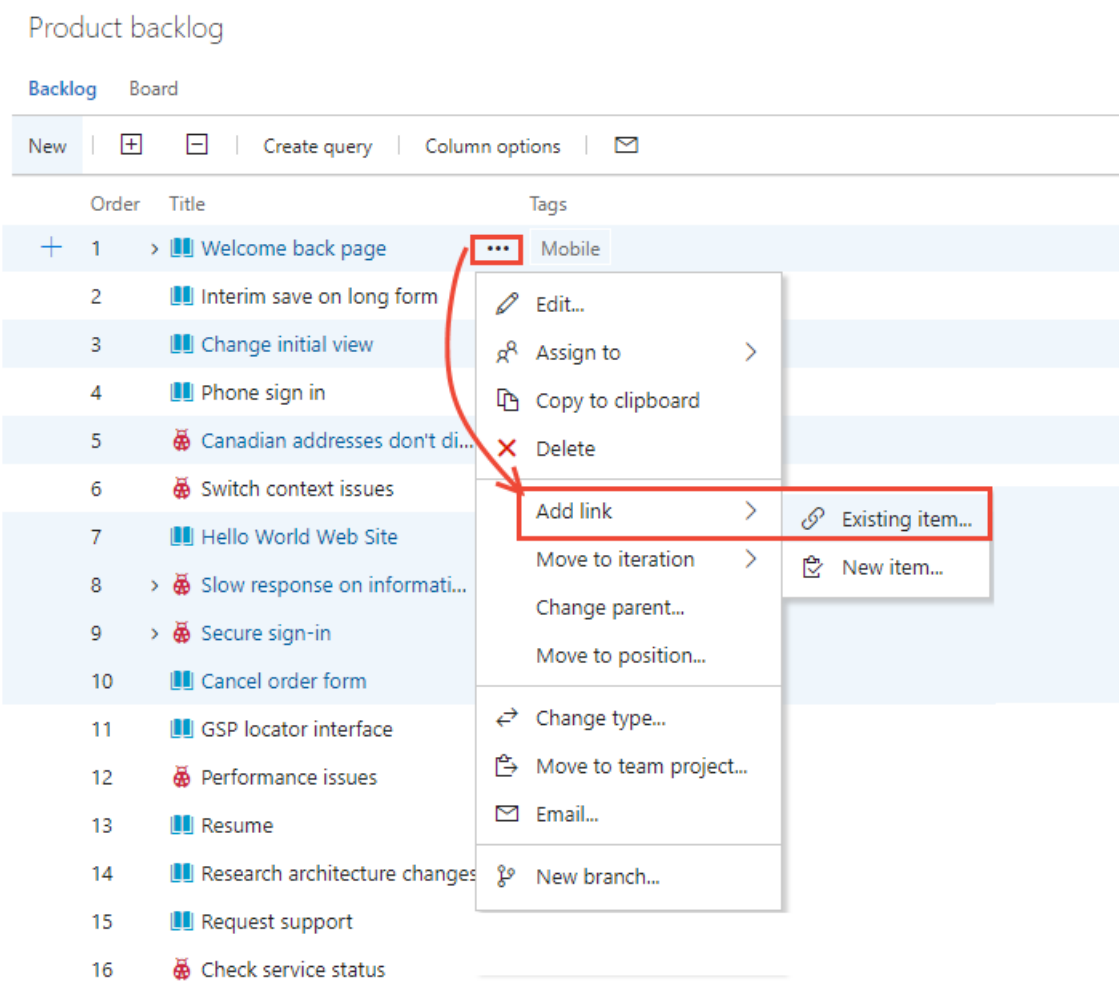
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

Link several work items

1. From the web portal, open a backlog or query results page, and multi-select the work items you want to add a link to.

2. Open the ... context menu of one of the selected work items, choose **Add link**, and then choose **Existing item...** or **New item...**.

   Here we multi-select from the product backlog and choose **Existing item...**.



Link to an existing work item

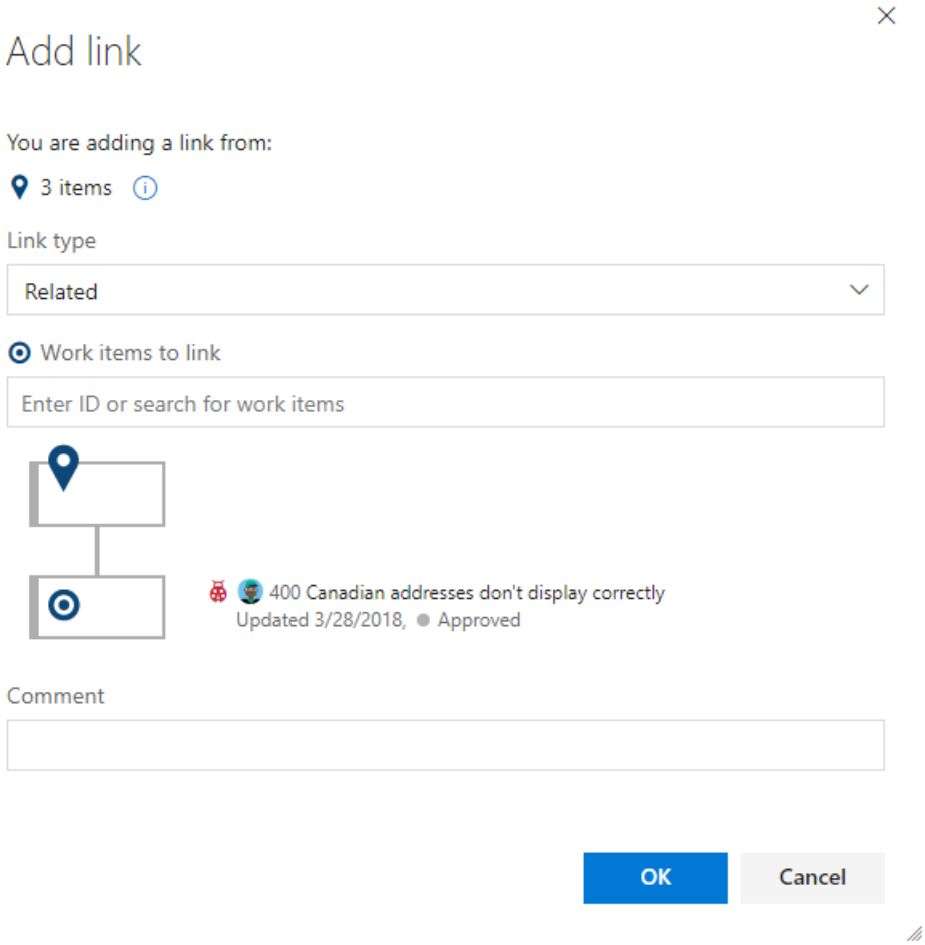
When you link work items, you select a link type. In general, use the following link types as indicated. To learn more about each link type, see Link type reference.

- Use the **Duplicate** link type when two work items have been created that essentially capture the same information; close one of the work items and keep the other one active
- Use the **Parent/Child** link types when you want to break down work items into smaller items—for example, break down features into stories, or stories into tasks
- Use **Predecessor-Successor** link types when you want to track tasks that must be completed before others can be started; this link type is most often used when you plan work using Project
- Use the **Related** link type when the work items being linked are at the same level—such as two user stories that define features that overlap one another—or to link work items that are defined in different projects or managed by different teams.

- Browser
- Visual Studio

From the **Add link** dialog, select the link type, enter a work item ID, and then choose OK.

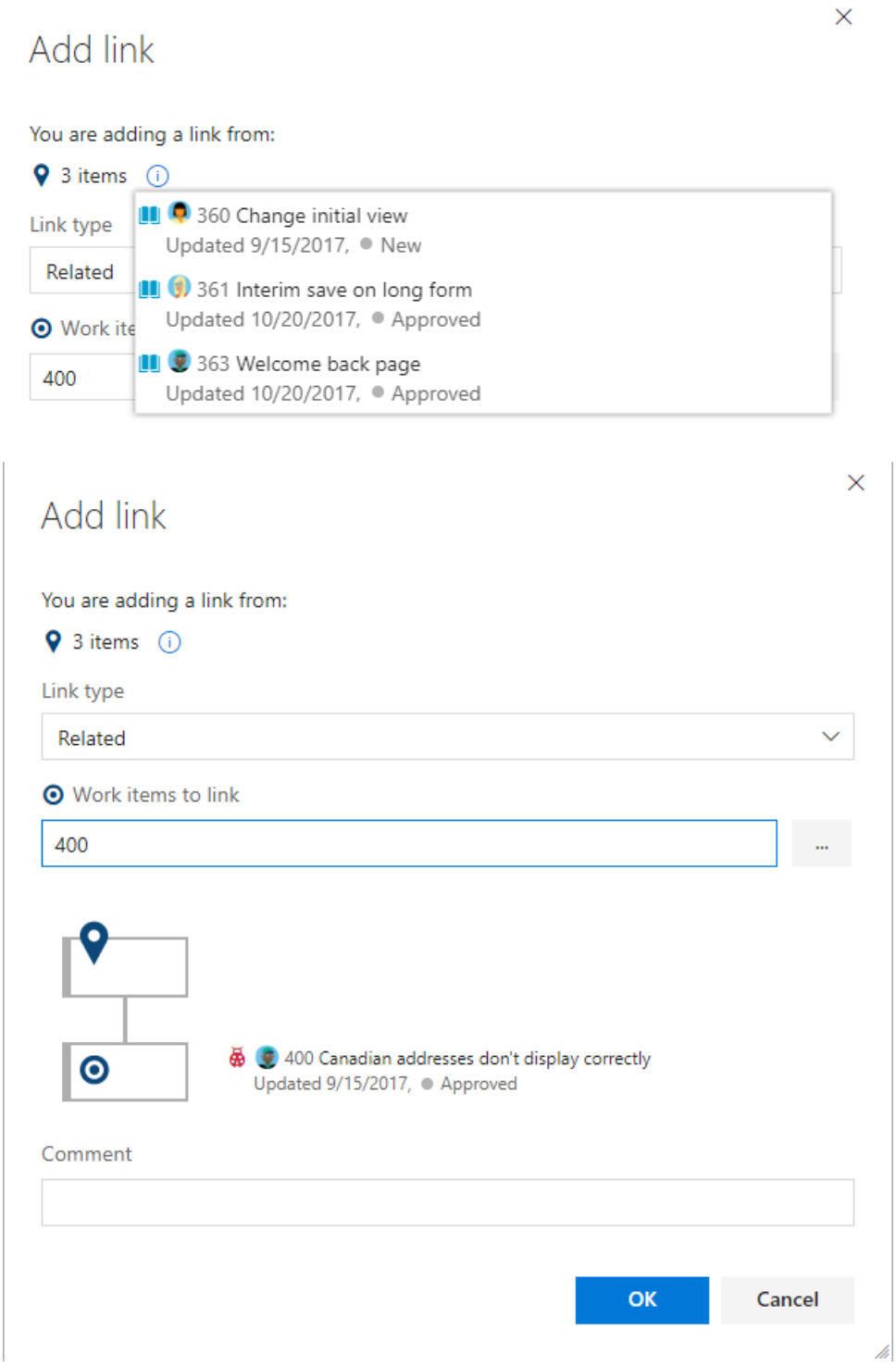
For example, here we use the **Related** link type to link three items to the bug with ID of *400*.



To link to multiple work items, you can use inline add which finds work items based on your recent activity or keyword searches. Select one or more of the work items displayed automatically based on your recent activity, or enter a keyword. Keyword searches will display work items based on work items that include that keyword in their title.

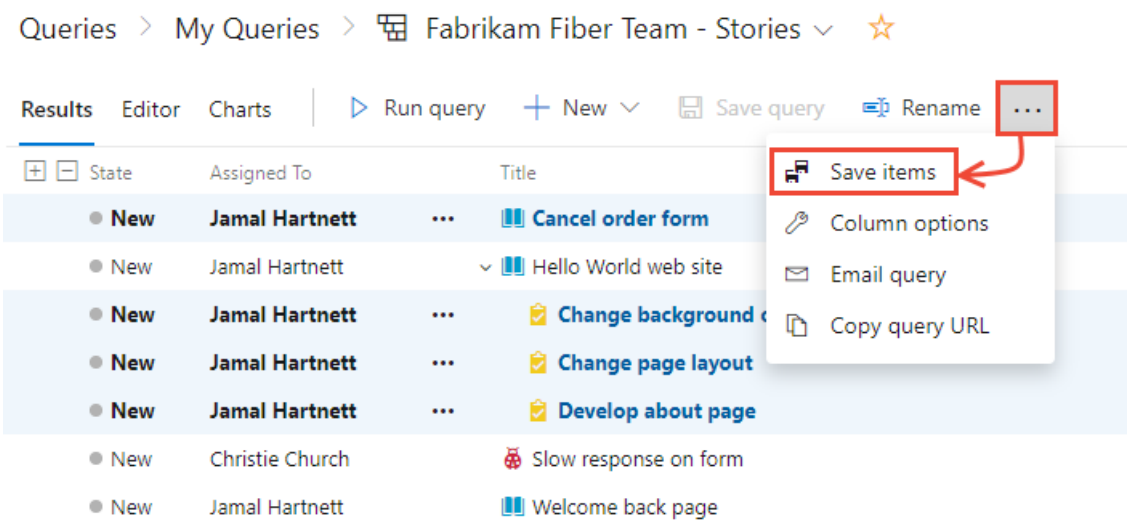
> **NOTE**
> You need to add each link one at a time. (You can no longer enter their IDs separated by commas or spaces.) To quickly find work items of interest, you can also use work item search.

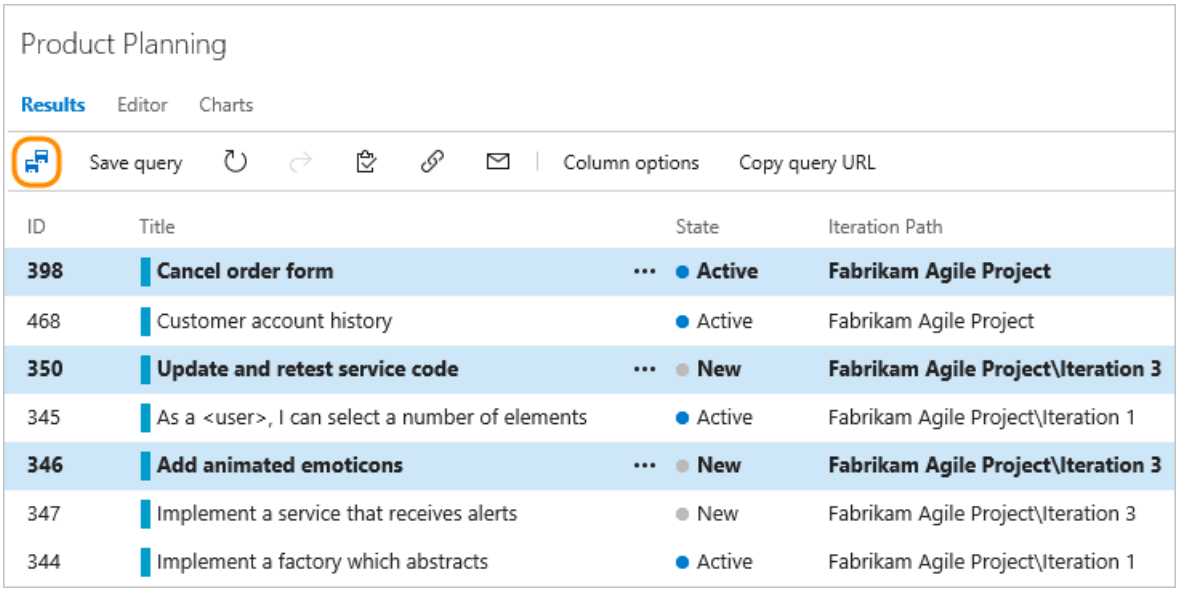
To view the work items selected for linking, you can choose the ⓘ .



To link to multiple work items, enter their IDs separated by commas or spaces. If you don't know the IDs or you want to link to an item in a different project, you can choose the ... context menu to open a dialogue that will support you in choosing work items that are based on IDs, a query, or title keyword.

If you're working from the Query Results page, you'll need to bulk save the work items you've modified. When you work from a backlog, work items are automatically saved.

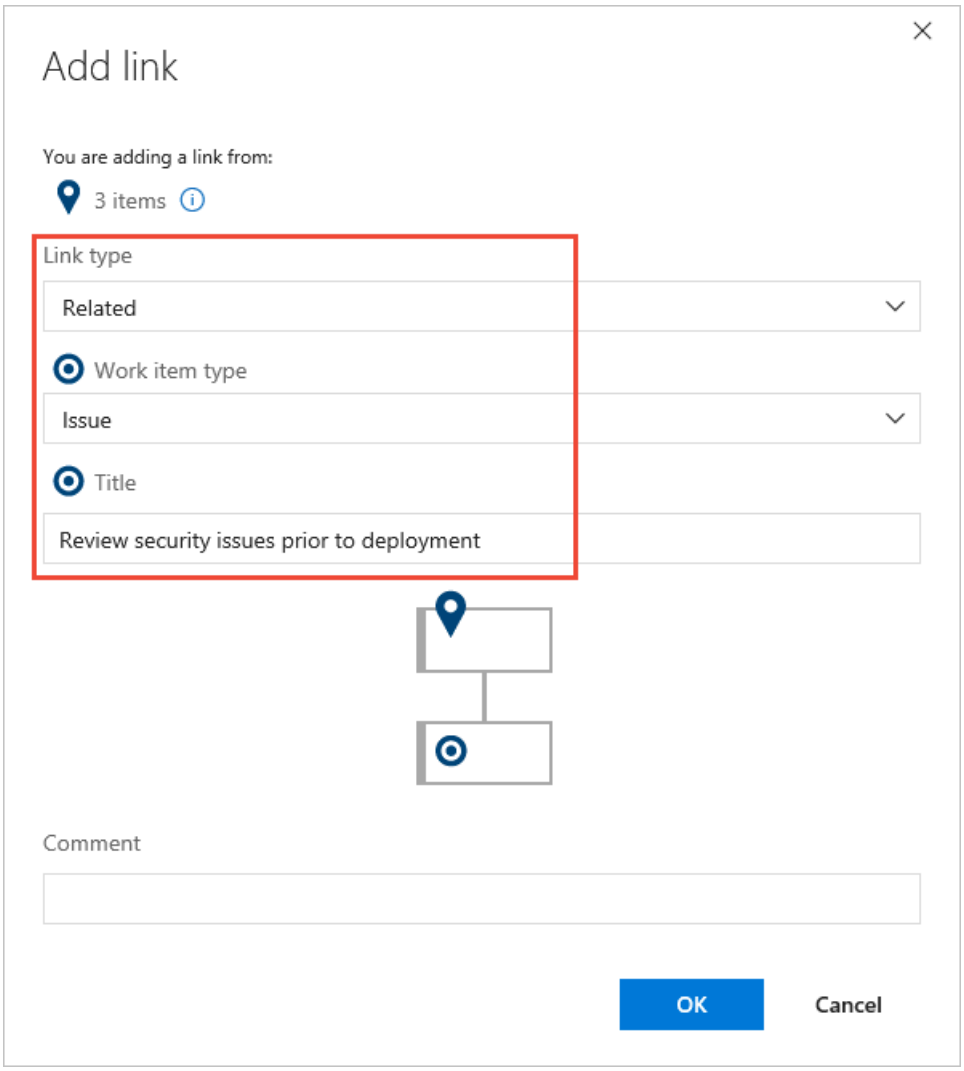




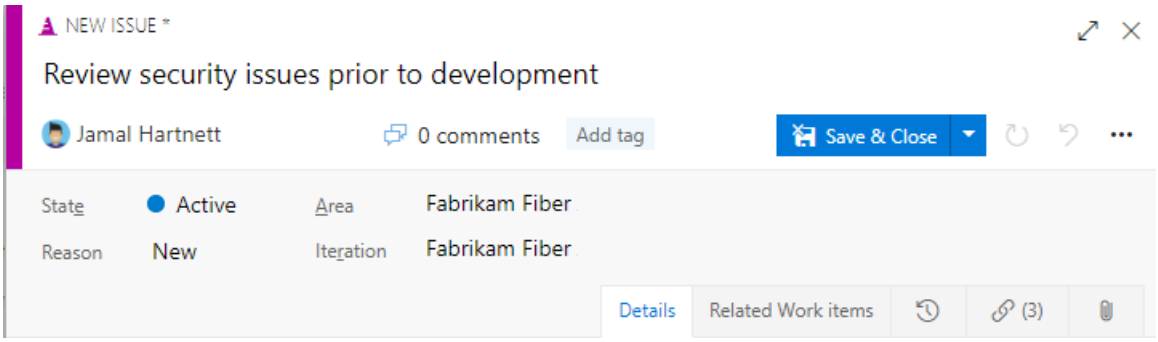
Link to a new work item

Here, we've selected to add a link to the selected work items.

1. Specify the link type, work item type, and title of the new work item. Choose **OK**.



2. A work item of the type selected opens. Enter additional information and save the work item.



3. If you're working from the Query Results page, you'll need to bulk save the work items you've modified as shown in the previous procedure.

Link to a remote work item

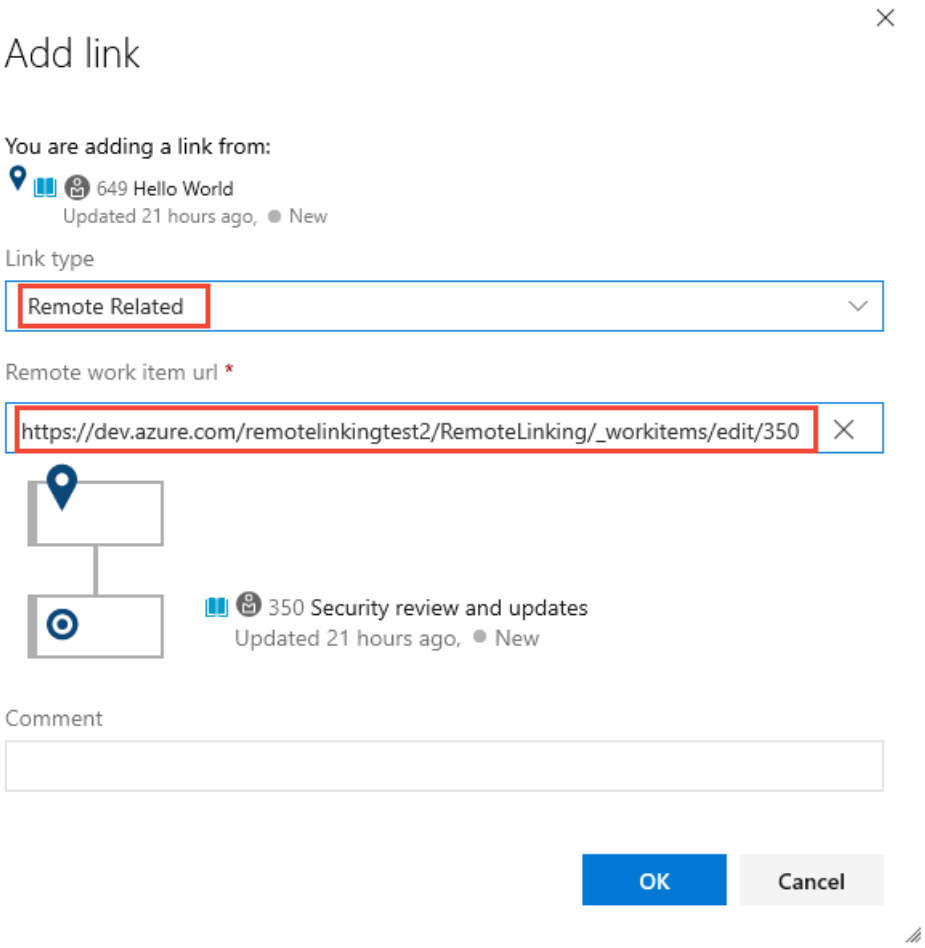
You can link work items to objects defined in other Azure DevOps organizations as long as both organizations use the same Azure Active Directory to manage users.

Choose from one of the following two remote link types supported.

- Use the **Consumes From/Produces For** link types when you want to track dependencies of work items that are defined in different organizations and managed by different teams.
- Use the **Remote Related** link type when the work items being linked are defined in different organizations and managed by different teams, but don't have strong inter-dependencies.

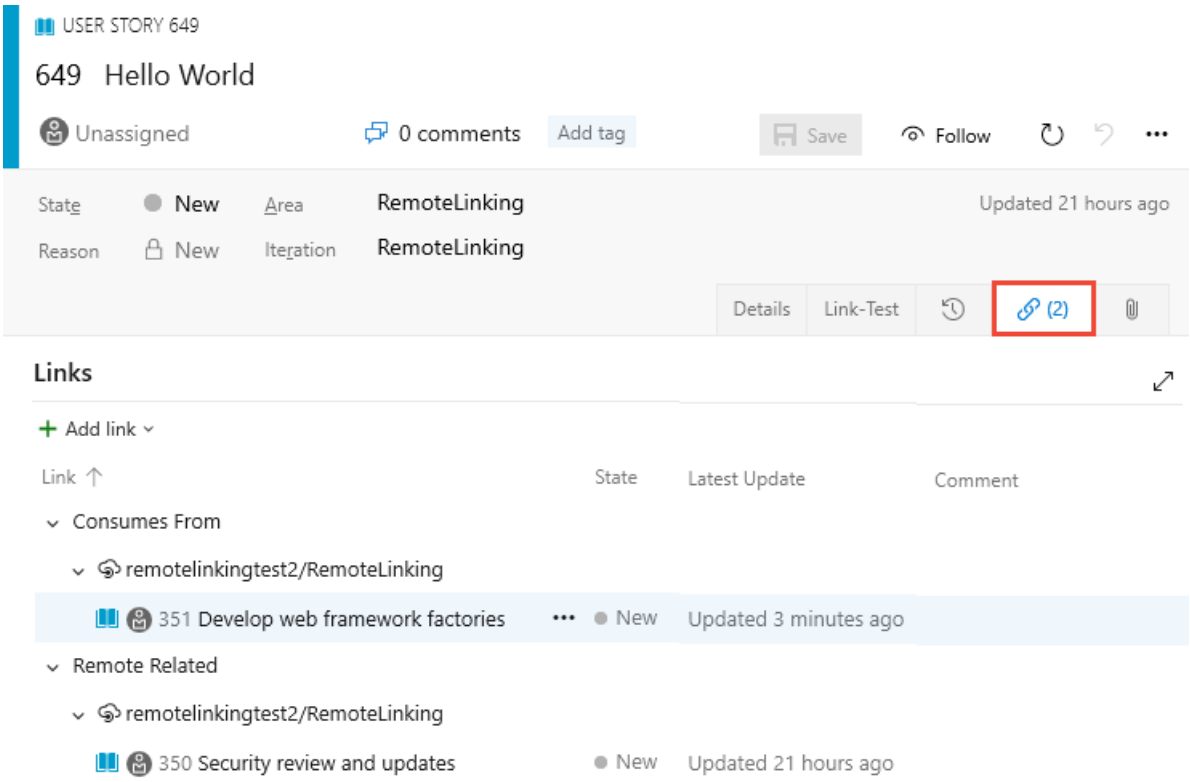
From the Add link dialog, select the link type, enter the URL of the remote work item, and then choose OK.

For example, here we use the **Remote Related** link type to link to work item ID *350* that exists in the *remotelinkingtest2* organization, *RemoteLinking* project.



The link tab maintains a count of all links to the work item. The *Remote Link Count* field maintains a count of the number of links added to a work item that link to a work item defined in another project or organization.

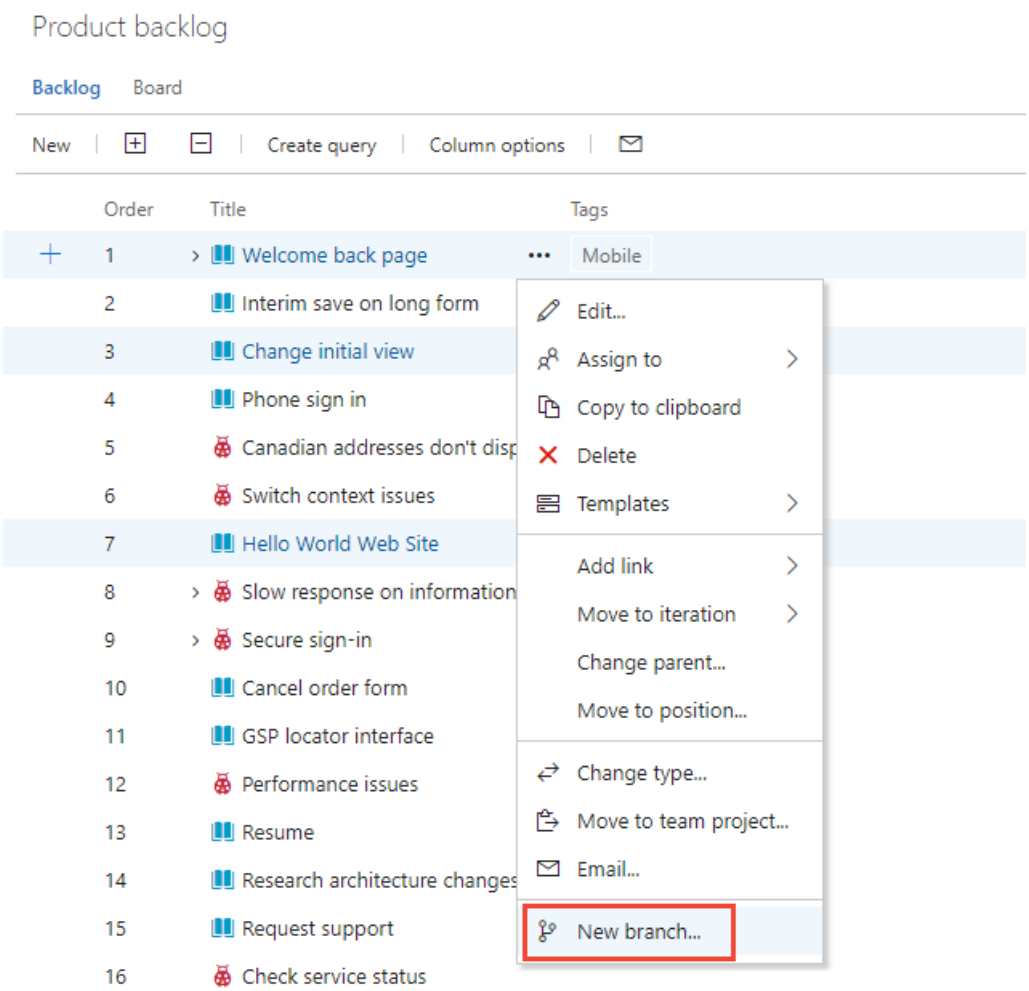
The following image shows an example of two remote links, indicated by the cloud icon, added to a user story.



Link several work items to a new git branch

You can add a new git branch and link them to existing work items at the same time.

From a backlog or query results page, multi-select the work items you want to link to a new git branch, choose the ··· actions icon, and then **New branch...**. To learn more, see Link work items to Git development objects.



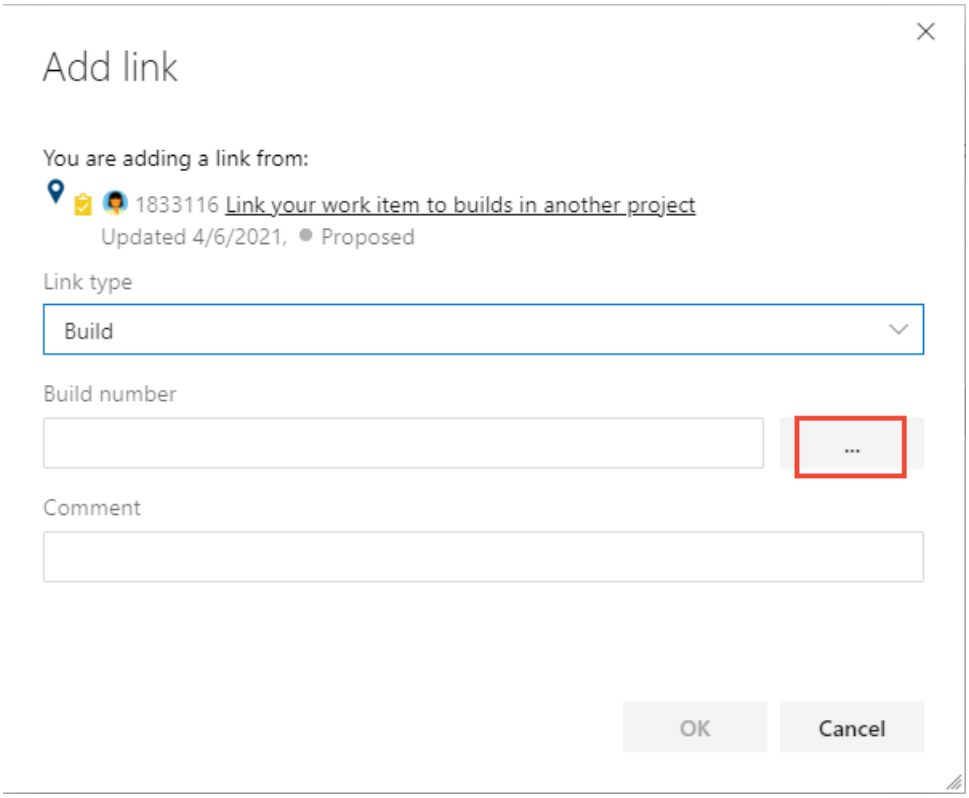
Link work items to a build in same or other projects

You can link work items to existing builds from the **Add link** dialog. These builds can be within your project or to other projects in your organization or collection.

> **NOTE**
> This feature requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

1. From the **Links** tab of a work item, choose **Add link > Existing item**.
2. From the **Add link** dialog, choose one of the build link types—**Build**, **Found in build**, **Integrated in build**— and specify the build number.

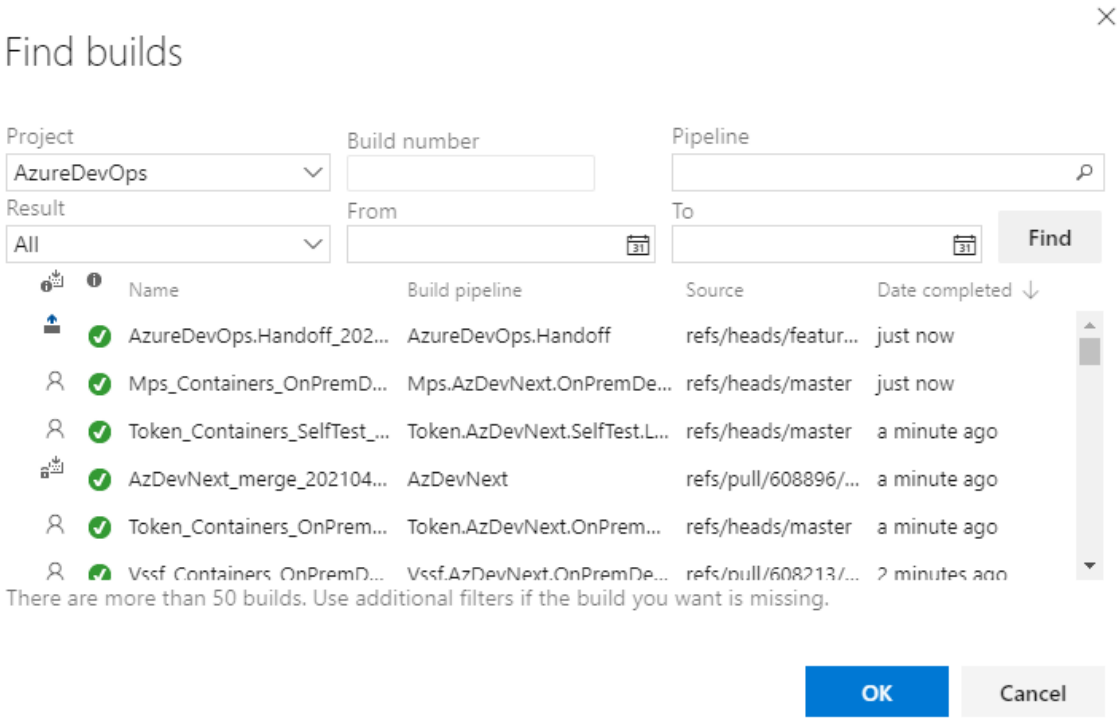
   If you don't know the build number—a combination of the pipeline and build name—you can search for it by choosing the ··· icon.



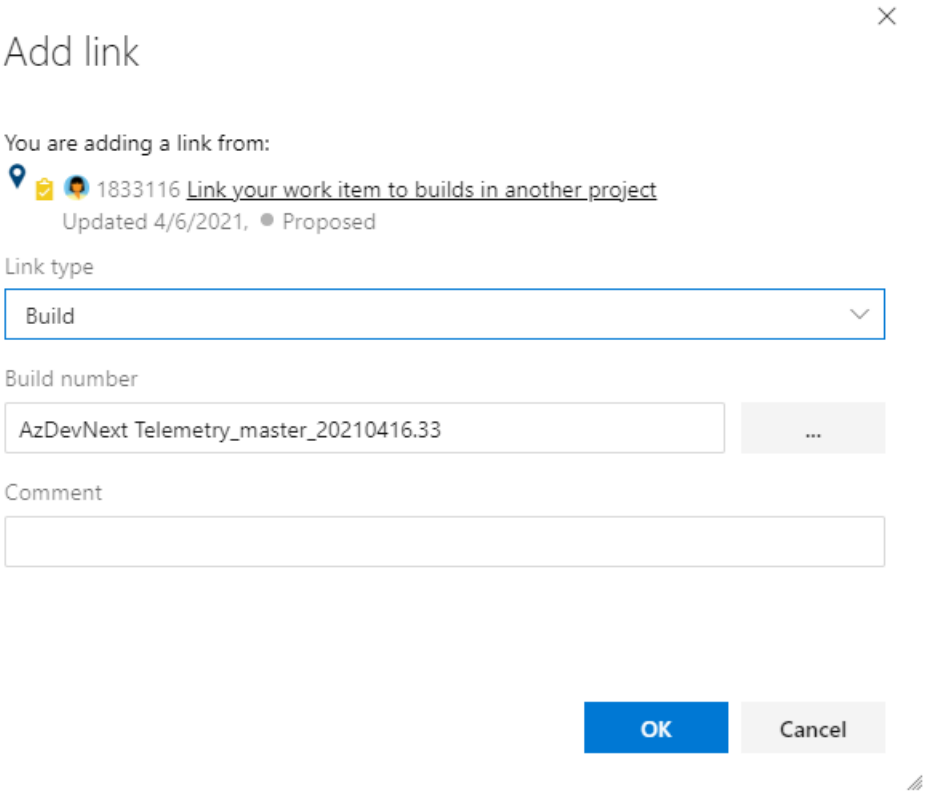
3. From the **Link builds** dialog, choose the parameters to filter your search of builds.

   If linking to a build in a different project, then first choose the **Project** whose build you want to link to.

   For example, you can specify a build number, select a build pipeline, a build result—such as, **All**, **succeeded**, **partially succeeded**, **failed**, or **canceled**. Or, with **All** selected for **Result**, choose **Find** to list the available builds to link to.



4. Choose the build from the list you want to link to and then choose **OK**.
5. From the **Add link** dialog, choose **OK** to complete the operation.

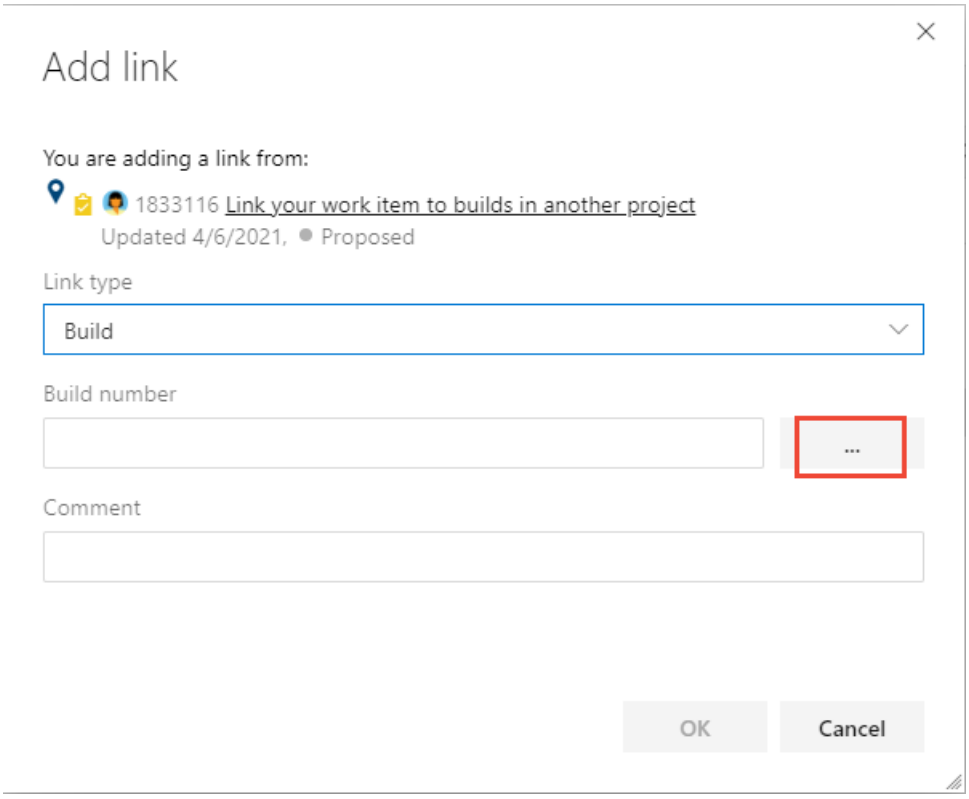


Link work items to a build

You can link work items to existing builds from the **Add link** dialog.

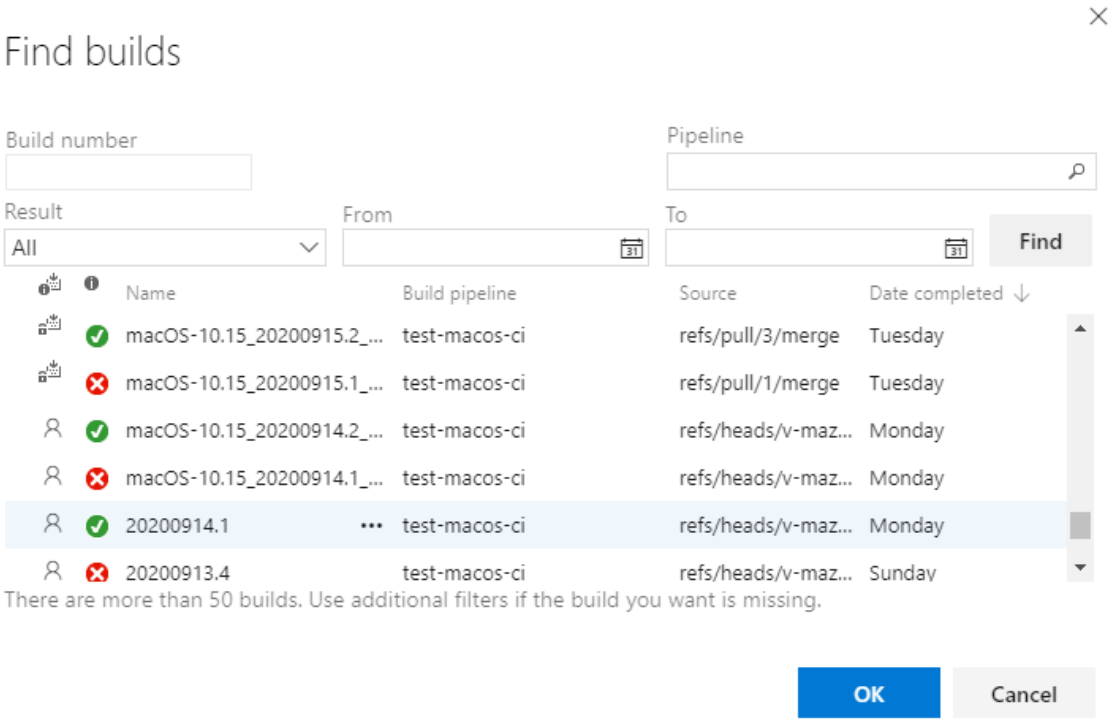
1. From the **Links** tab of a work item, choose **Add link** > **Existing item**.
2. From the **Add link** dialog, choose one of the build link types—**Build**, **Found in build**, **Integrated in build**— and specify the build number.

If you don't know the build number—a combination of the pipeline and build name—you can search for it by choosing the ··· icon.



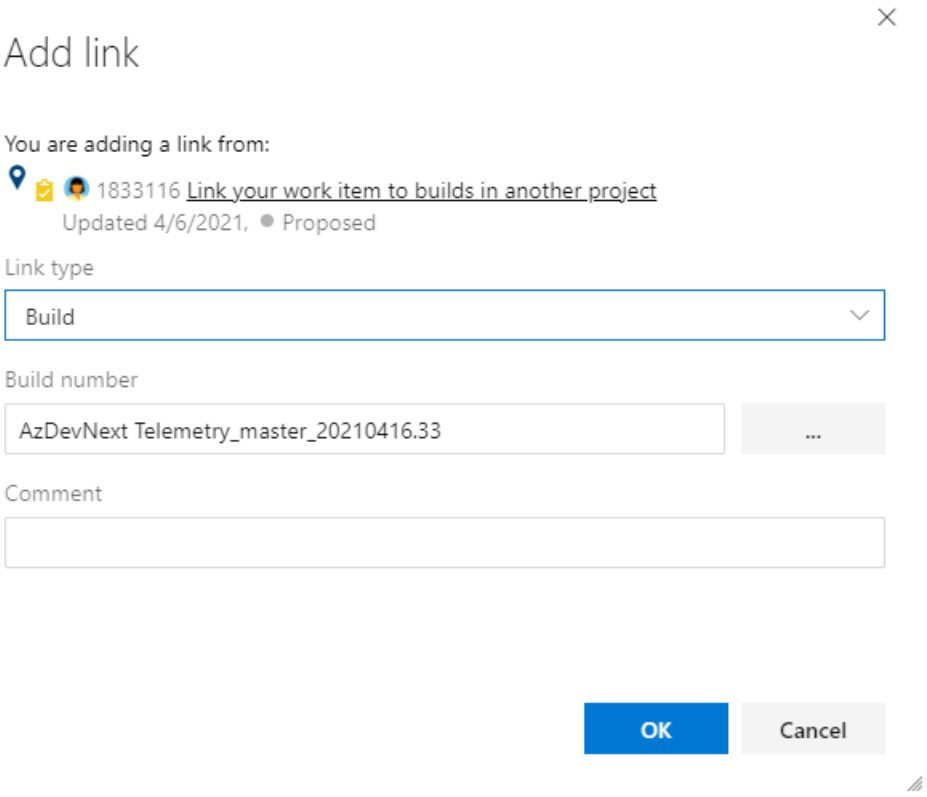
3. From the **Link builds** dialog, choose the parameters to filter your search of builds.

   For example, you can specify a build number, select a build pipeline, a build result—such as, **All**, **succeeded**, **partially succeeded**, **failed**, or **canceled**. Or, with **All** selected for **Result**, choose **Find** to list the available builds to link to.



4. Choose the build from the list you want to link to and then choose **OK**.

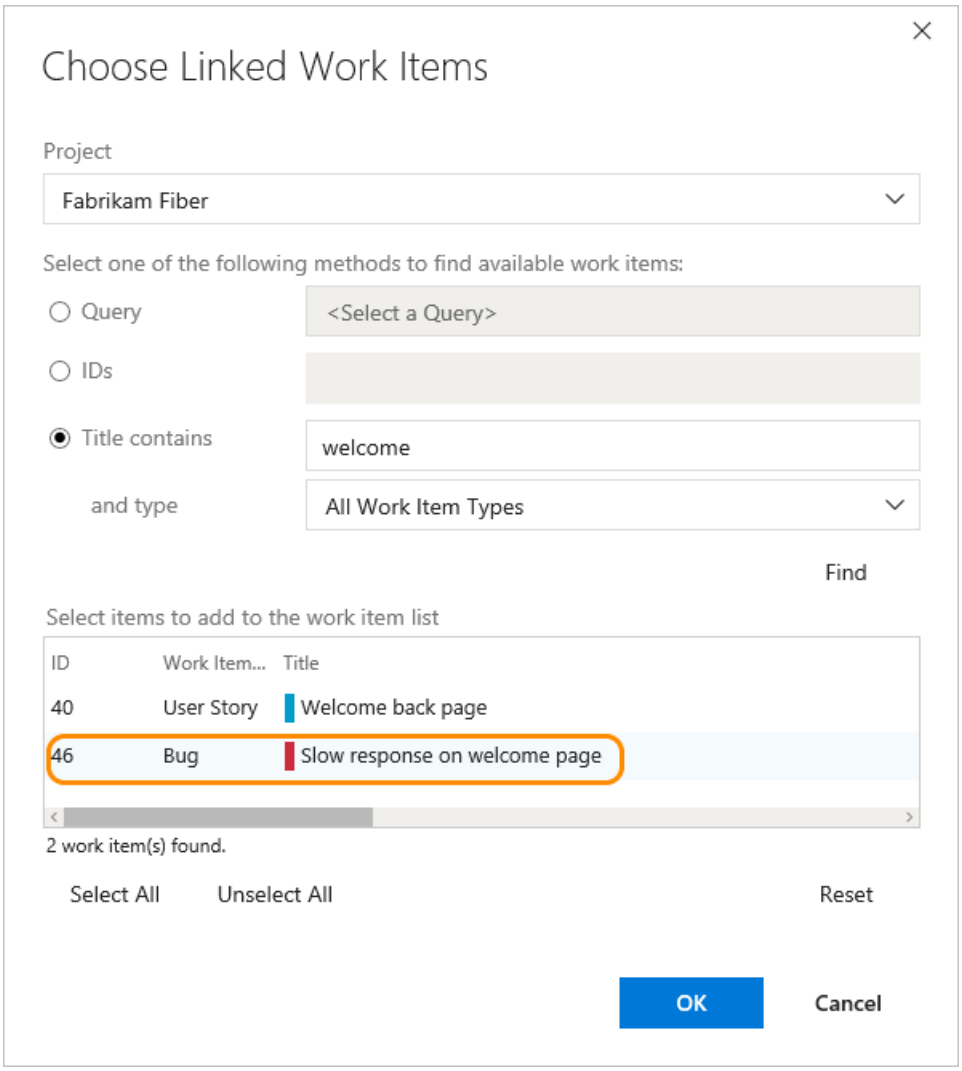
5. From the **Add link** dialog, choose **OK** to complete the operation.



Find work items to link to

From the Add link dialog, you can open a secondary dialog to help you choose one or more work items to link to. If you're going to find and list work items to link to by using a saved query, first define the query that you want to use.

1. From the Add link dialog, choose the ... context menu or **Browse** button (Visual Studio) to open the following dialog.



If the work items are defined in another project, then first select the Project. Then, make your selections:

- **Query**. Use this method when you've defined a query that you know contains the set or superset of the work items that you want.
- **IDs**. Use this method when you know the IDs of the work items that you want to link to. In the **IDs** box, type the IDs of the work items that you want to find, separated by commas or spaces.
- **Title contains**. Use this method to find work items that have a common word or phrase in the title field. In the **and type** list, select the type of work item that you want to retrieve.

> **NOTE**
> To minimize the time required to run the query, narrow the filter criteria of the search.

2. Select the **Find** button.

   Only those work items defined for the selected project and specified work item type are listed. To sort on a column field, choose the column title.

3. In the list of returned work items, select one or more work items.

   - From the web portal: To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the Ctrl key.
   - For Visual Studio, select each work item that should link to the current work item. You can also press the SHIFT key while clicking to select a range of work items, or press the CTRL key while clicking to select multiple work items.

Set work item state in pull request

When you create a pull request, you can set the *state* value of the linked work items in the description. Follow the syntax: `{state value}: #ID`. When you merge the pull request, the system reads the description and updates the work item state. In the following example, we set work items #300 and #301 to Resolved, and #323 and #324 to Closed.

> **NOTE**
> This feature requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

To learn more about this feature, see Link to work items from other objects.

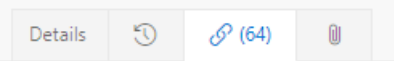
More options for modifying links in bulk

Other features you can use to quickly link or change links that use the parent-child link type (some features are version-dependent, see the linked article for details):

- To quickly link backlog items to portfolio backlog items with parent-child links, use the mapping pane to organize your backlog. Or, you can choose to Show Parents and drag-and-drop items within the tree hierarchy.
- To create and link tasks to backlog items, use the sprint backlog page.
- To indent (), outdent (), and change the tree hierarchy, use a tree query in Visual Studio.
- To add or delete work items or change the link structure, you can use Excel. See Bulk add or modify work items with Excel.

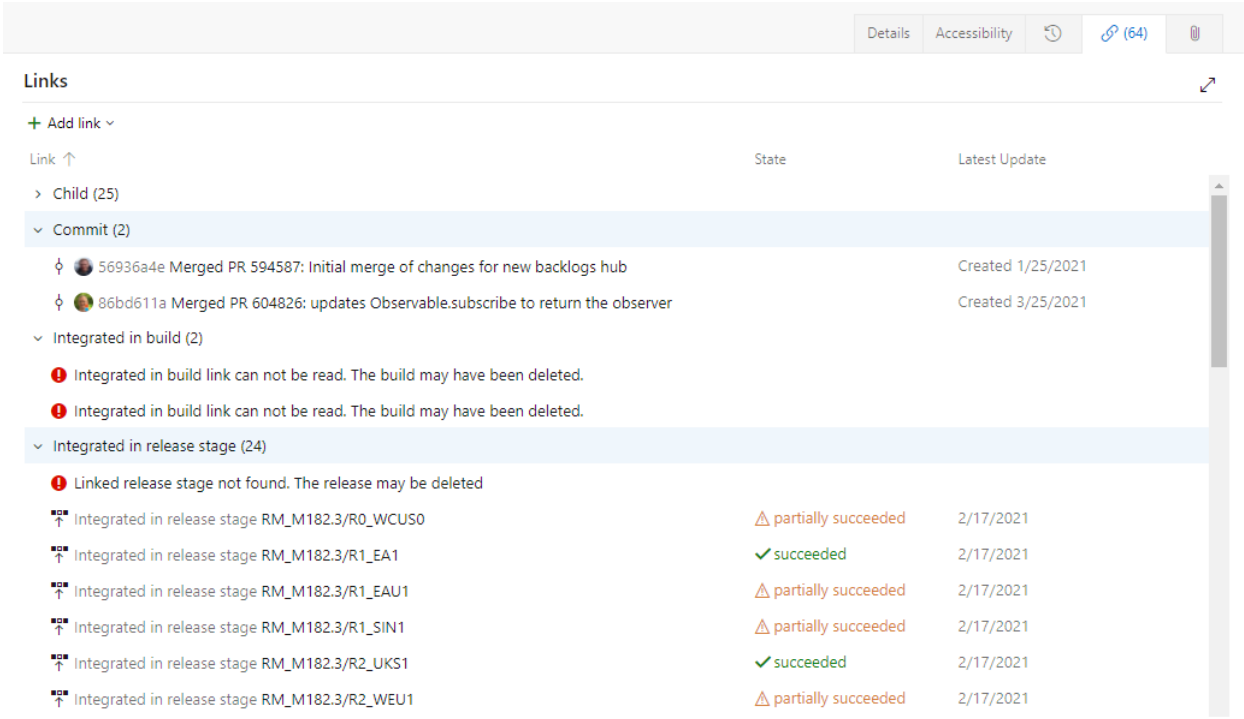
View list of linked objects

To view the list of all objects linked to a work item, open the work item and choose the **Links** tab. The links tab indicates the count of all linked objects.



Linked objects are grouped under their link type, with a count within each group. You can expand or collapse each group, and sort within each group by **State**, **Latest Update**, or **Comment** by choosing the corresponding column title.

For example, the following **Links** tab shows a portion of the 64 linked objects for a work item.



Links prefaced with the red exclamation mark indicate that the build, release, or other object has been deleted. This is usually due to retention policies which automatically delete these objects after a certain time period has passed.

Related articles

- Map backlog items to portfolio backlog items
- Link work items to Git development objects
- Link GitHub commits and pull requests to work items
- Use Excel to edit tree-topology links
- Linking, traceability, and managing dependencies

- Map backlog items to portfolio backlog items
- Link to work items from other objects
- Link work items to Git development objects
- Use Excel to edit parent-child links
- Linking, traceability, and managing dependencies

Copy or clone stories, issues and other work items

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

There are two types of copy functions you can use. The first is to duplicate a single work item, referred to as copy or clone. Also, you can choose to change the project or work item type when copying/cloning a work item.

There are two types of copy functions you can use. The first is to duplicate a single work item, referred to as copy or clone.

The second copy function is to copy a multi-selected list of work items to the clipboard, referred to as copy as HTML or copy to clipboard.

> **TIP**
> You can't copy or clone linked work items at this time. To learn more, see the Azure Boards FAQs.

Use this article to learn how to:

- Copy or clone a work item, essentially making a duplicate of a work item
- Change the work item type, for example from a bug to a user story
- Copy a list of work items to email or share with others
- Copy the URL of a work item

> **NOTE**
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to Azure DevOps Services. However, the basic functionality available to you remains the same unless explicitly mentioned.

> **NOTE**
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your on-premises Azure DevOps. However, the basic functionality available to you remains the same unless explicitly mentioned.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick

reference.

- All project members, even those who belong to the **Readers** group, can email work items.

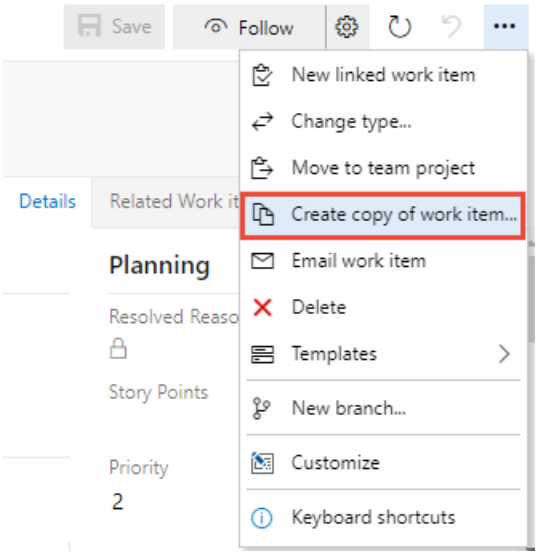
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

Copy or clone a work item

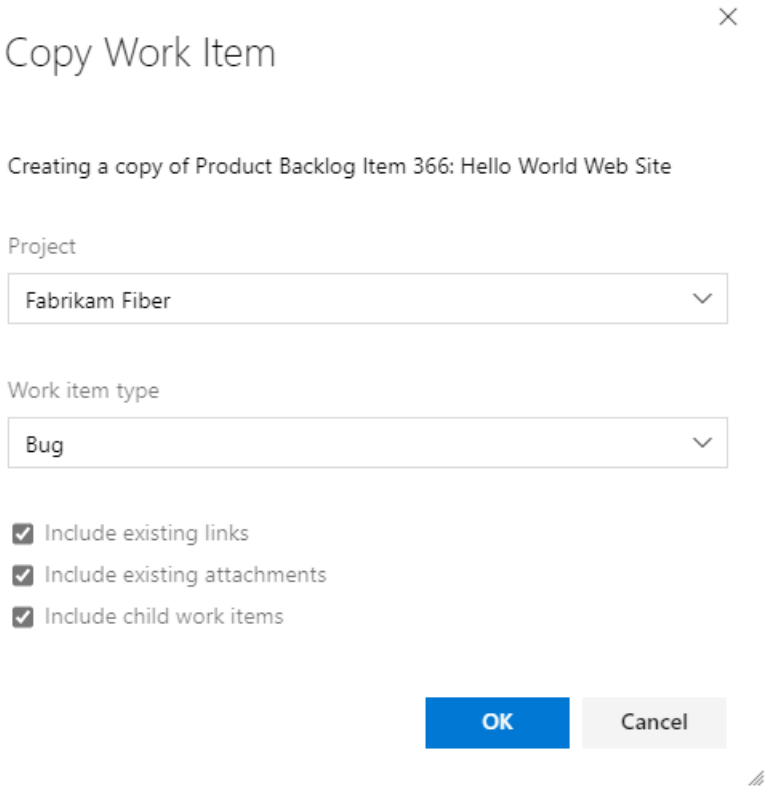
Clone a work item when you want to create another instance of it. This action opens a form with all fields filled out. Copy a work item when you want to create another instance of it and optionally change its work item type. This action opens a form with all fields filled out except for the Title. A related link to the original work item is created. Also any parent link is copied over. No history or attachments are copied over from the original work item.

> **NOTE**
> It is possible that some fields are copied over depending on the on-premise version you are working with and how you have customized your work item types. If the work item type of the work item that you are cloning has no state transition rule that says to clear the *Closed By* field when the *State* is **New** or **Active**, then that field gets copied over. The current system out-of-box templates have this rule defined. It was added to TFS 2018 and later versions.

1. From the web portal, open the work item you want to copy or clone, open the ... context menu, and choose **Create copy of work item**.



2. Choose the project and work item type if different from the copied work item. Optionally change the Title and provide more details.



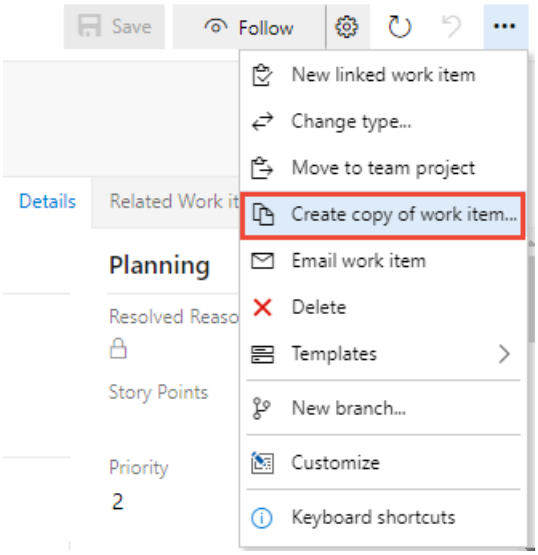
Optionally, check one or more of the boxes:

- **Include existing links**: To include all **Related** and external links in the copied work item. Note that a **Related** link is created automatically for the work item copied, and included in the **Discussion** section. There is no method for disabling this feature.
- **Include existing attachments**: To include attachments in the copied work item
- **Include child work items**: To include existing links to child work items in the copied work item. This feature isn't recursive. Only those work items directly linked as children to the work item being copied are included. This option appears even if there are no child items linked to the work item.

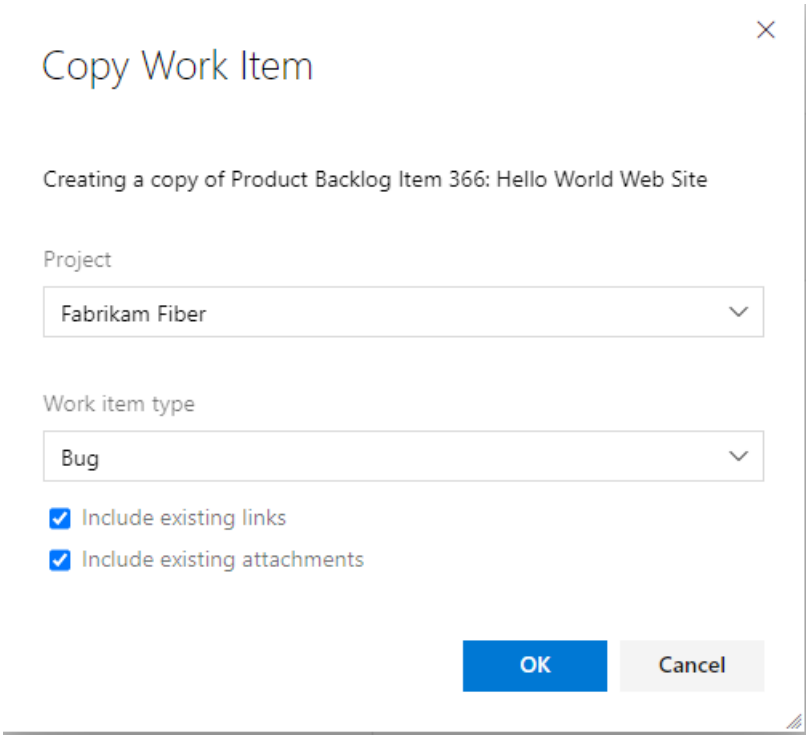
> **NOTE**
>
> - When you copy the work item to a different project, **Include child work items** is disabled.
> - When you copy a work item and choose to **Include child work items**, a copy is made of each child work item and linked to the copied work item through a Parent-Child link.
> - The **Include child work items** feature requires installation of Azure DevOps Server 2020.1 update.

3. In the work item form that opens, update other fields as needed. All work items start in the New state.

1. From the web portal, open the work item you want to copy or clone, open the ... context menu, and choose **Create copy of work item**.



2. Choose the project and work item type if different from the copied work item.



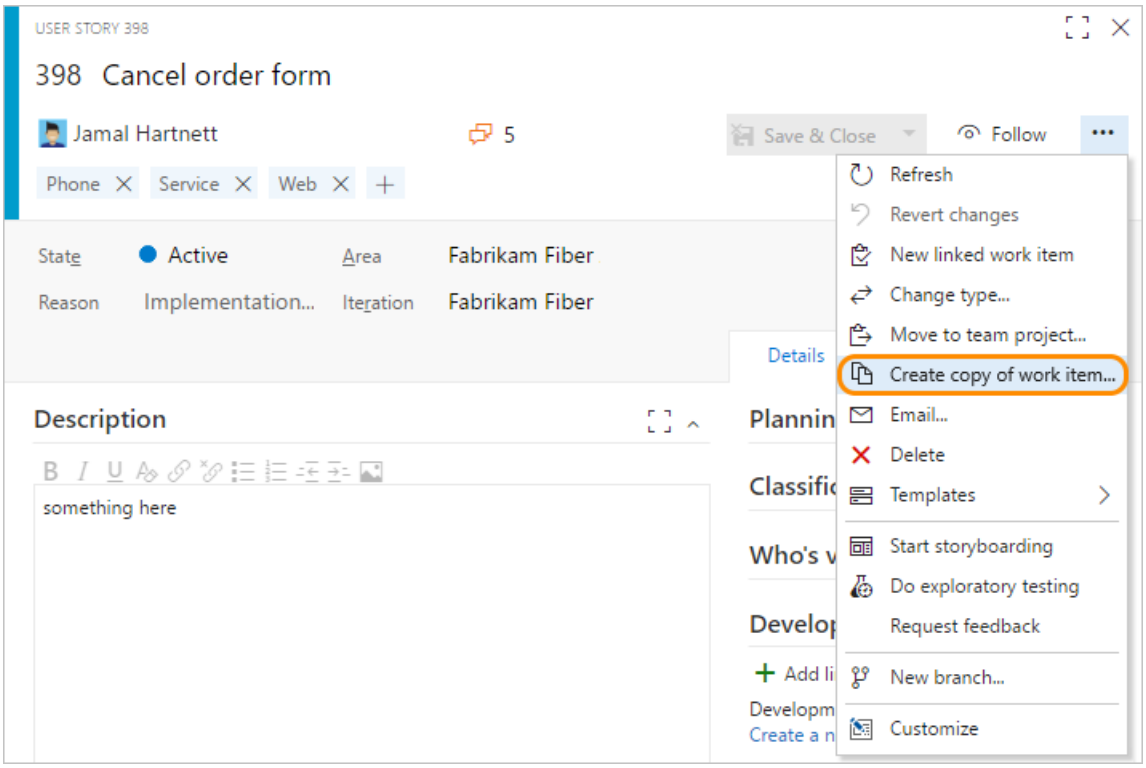
Optionally, check one or more of the boxes:

- **Include existing links**: To link the copied work item as a Related link type and maintain all other related and external links included in the copied work item.
- **Include existing attachments**: To include attachments in the copied work item.

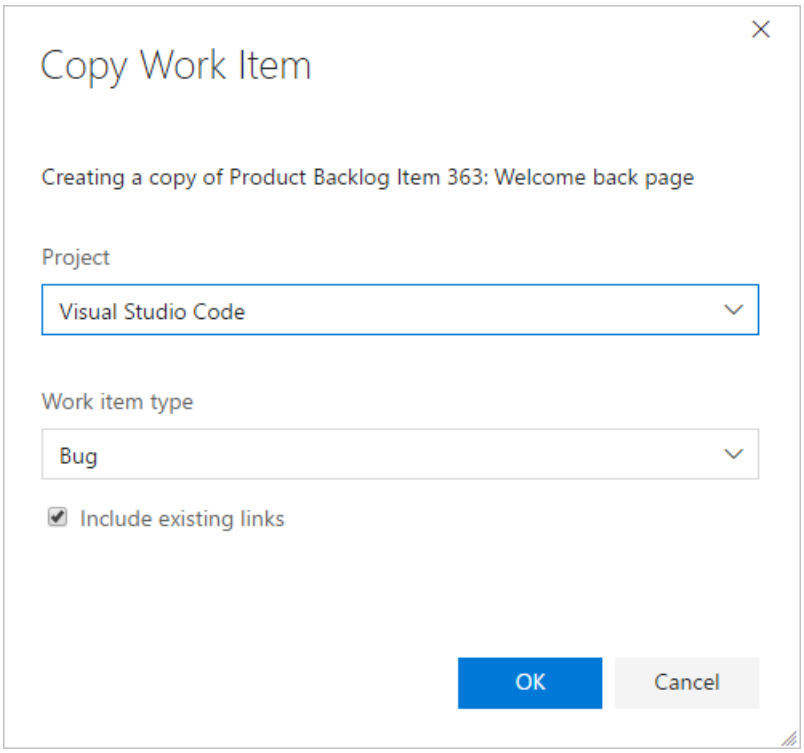
3. Choose **OK**.

4. In the work item form that opens, update other fields as needed. All work items start in the New state.

1. From the web portal, open the work item you want to copy or clone, open the ... context menu, and choose **Create copy of work item**.



2. Choose the project and work item type if different from the copied work item. Optionally change the Title and provide more details. To link the copied work item as a Related link type and maintain all other links (related links and external links) included in the copied work item, check the **Include existing links** checkbox.



3. Choose **OK**.

4. In the work item form that opens, update other fields as needed. All work items start in the New state.

> **TIP**
> Copied or cloned work items always have an ID that is greater than the work items from which they were copied or cloned.

Change the work item type

If you have a large number of work items whose type you want to change, use Change work item type. If the **Change work item type** option isn't available to you, you can export a set of work items using Excel or CSV, copy them to a new list, and re-import them by specifying a different work item type. See Bulk add or modify work items with Excel or Import or update work items in bulk by using CSV files.

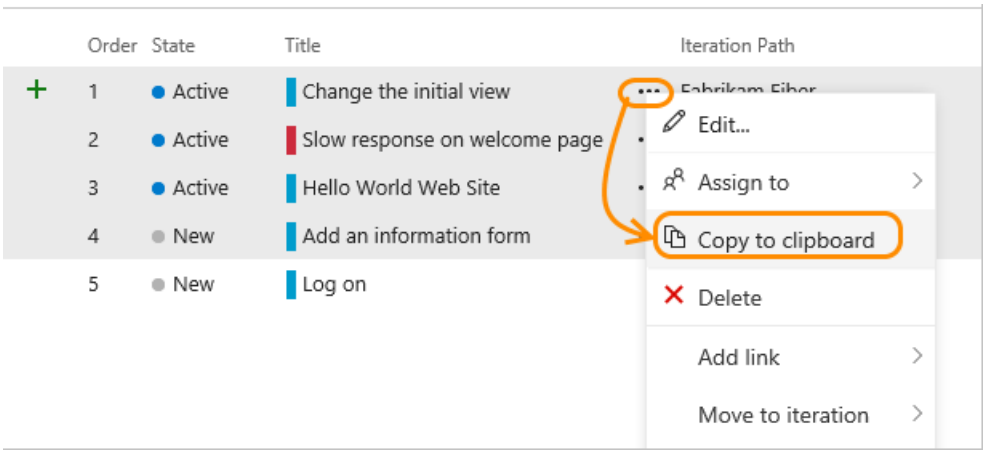
Copy a list of work items

With this option, you can copy an HTML formatted table of selected items from either a backlog page or query results list. You can then email this list using your choice of email client, or paste into a Word document, Excel spreadsheet, or other application.

> **NOTE**
> The data copied with **Copy as HTML** is the same as that copied when you select **Email selected work items**. If you don't have an SMTP server configured, you can work around this by using **Copy as HTML**. For on-premises Azure DevOps, all email actions require an SMTP server to be configured.

1. From the web portal, open a backlog or query results page, and multi-select the work items you want to copy to the clipboard.
2. Open the ... context menu of one of the selected work items, and then choose **Copy to clipboard** or **Copy as HTML**.

   Here we multi-select from the product backlog and choose **Copy to clipboard**.



Paste the contents into your email client

Paste the contents of the clipboard into your email client or other application. To open a linked work item, requires users to have read access to the project or area node for those work items.

The formatted table contains a link to each work item included in your selected results list. A link to a query that will open only those work items selected is also provided.

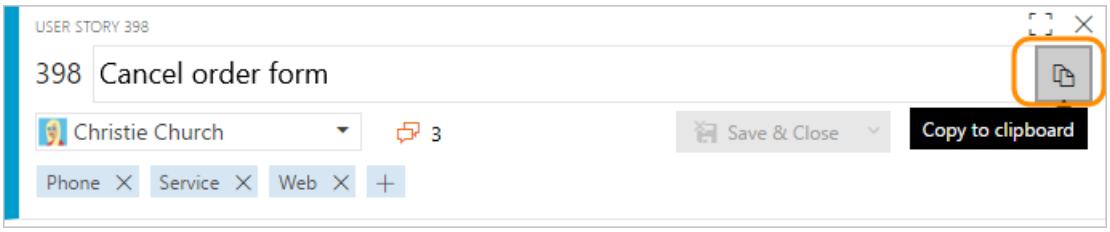
ID	Work Item Type	Title	State
398	User Story	Cancel order form	Active
468	User Story	Customer account history	Active
391	Bug	Slow response on form	Active
395	Bug	Canadian addresses don't display	New
346	User Story	Add animated emoticons	New

Open as query

Copy the URL

- Browser
- Visual Studio

Copy the URL from the web browser address or hover over the title and then select the copy-to-clipboard icon.



Related articles

- Copy or clone test plans, test suites, test cases, and other test items
- Bulk modify work items
- Move work items and change the work item type
- Remove, delete, or restore work items
- Pre-populate fields using work item templates
- Azure Boards FAQs

Tutorial: Follow changes made to a user story, bug, or other work item or pull request

10/4/2022 • 5 minutes to read • Edit Online

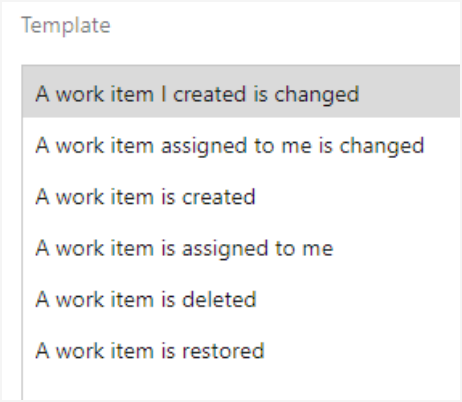
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To get notified of changes made to a specific work item or a pull request, you can choose to follow them. The Follow feature provides an improvised way of getting notified on a case-by-case basis.

If you want to subscribe to receive notifications automatically based on changes that occur based on your targeted set of criteria, see Manage personal notifications. For example, you can create a subscription to automatically get notified whenever a work item that you created or that was assigned to you is modified.

NOTE

Notification subscriptions allow you to personalize the notifications you receive automatically based on additional criteria you specify for yourself, a team, or a project. For example, you can create a subscription and add field criteria to receive changes based on one or more of the following templates.



This article shows you how to:

- Follow a work item
- Follow a pull request
- Manage work items that you're following

Configure an SMTP server in order for team members to receive notifications.

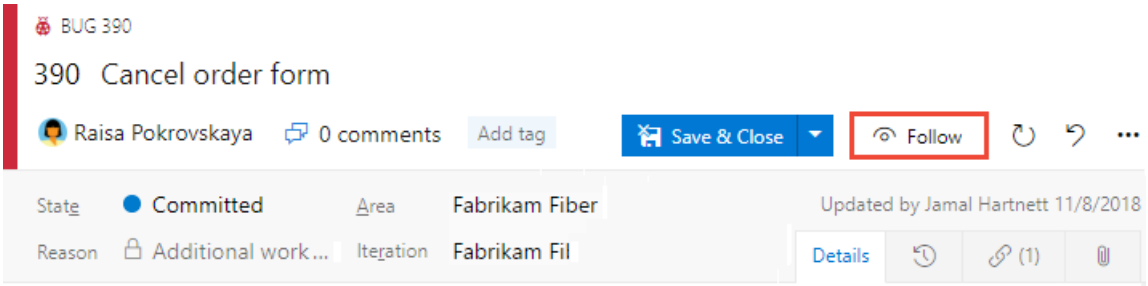
Prerequisites

- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To view or follow work items, you must be granted **Stakeholder** access or higher. For details, see About access levels. Also, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To view or follow pull requests, you must have **Basic** access or higher.

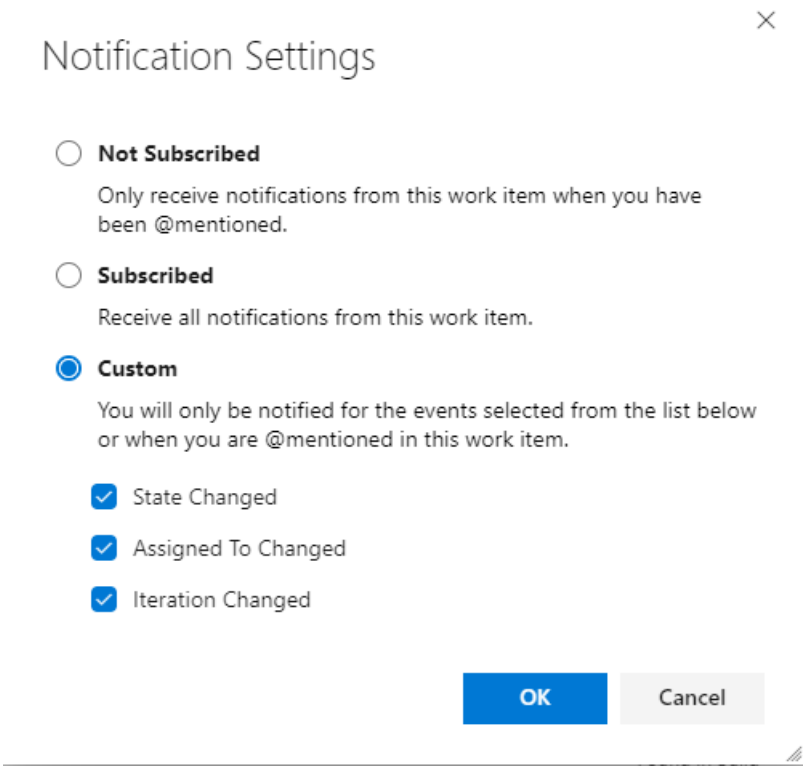
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To view or follow work items, you must be granted **Stakeholder** access or higher. For details, see About access levels. Also, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To view or follow pull requests, you must have **Basic** access or higher.

Follow a work item

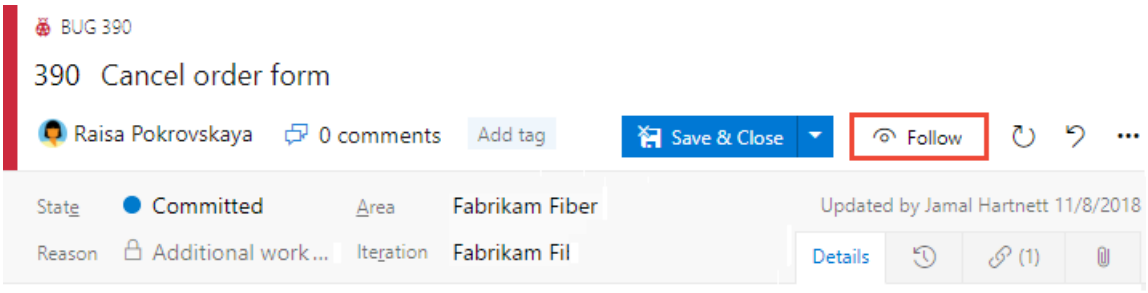
When you want to track the progress of a single work item, choose the Follow follow icon. This signals the system to notify you when changes are made to the work item.



If you want to specify conditions on when you'll get notified of changes, choose the gear icon and choose from the options provided.



By default, you're **Subscribed** to receive a notification when any change is made to the work item. Choose **Not Subscribed** to receive notification only when you're @mentioned. Or choose **Custom** to receive notifications when one of the checked fields changes, **State**, **Assigned To**, or **Iteration Path**.



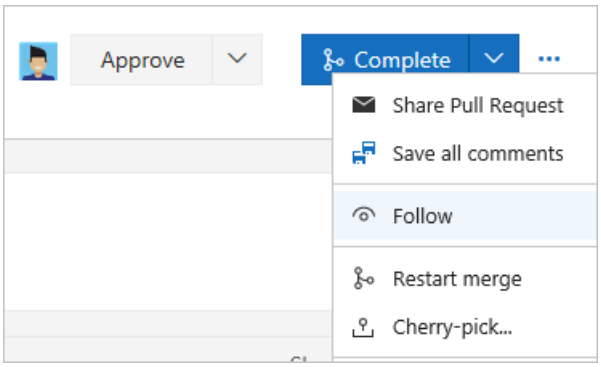
You'll only receive notifications when other members of your team modify the work item, such as adding to the discussion, changing a field value, or adding an attachment.

Notifications are sent to your preferred email address, which you can change from your user profile

To stop following changes, choose the Following following icon.

Follow a pull request

To track the progress of a single pull request, choose the ··· actions icon for the pull request, and select the Follow **Follow** option. This signals the system to notify you when changes are made to the PR.



You'll only receive notifications when other members of your team modify the PR, such as adding to the discussion or adding an attachment.

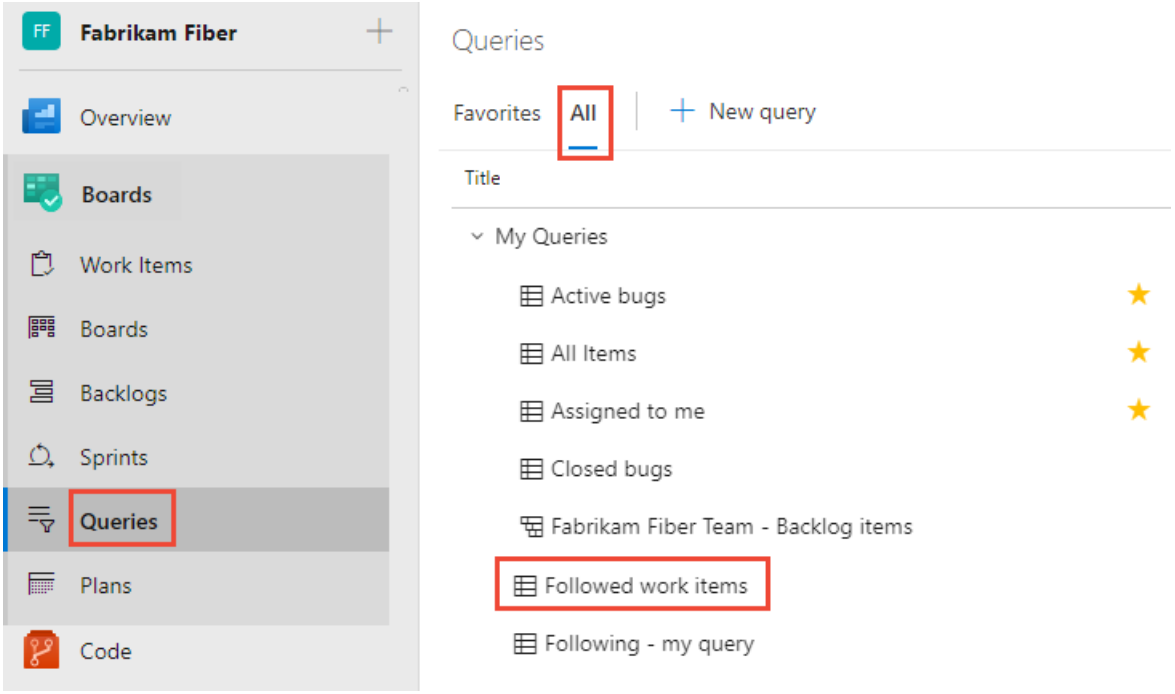
Notifications are sent to your preferred email address, which you can change from your user profile.

To stop following changes, open the PR context menu and choose the Following Following icon.

Manage work items that you're following

You can review and manage all the work items you've selected to follow.

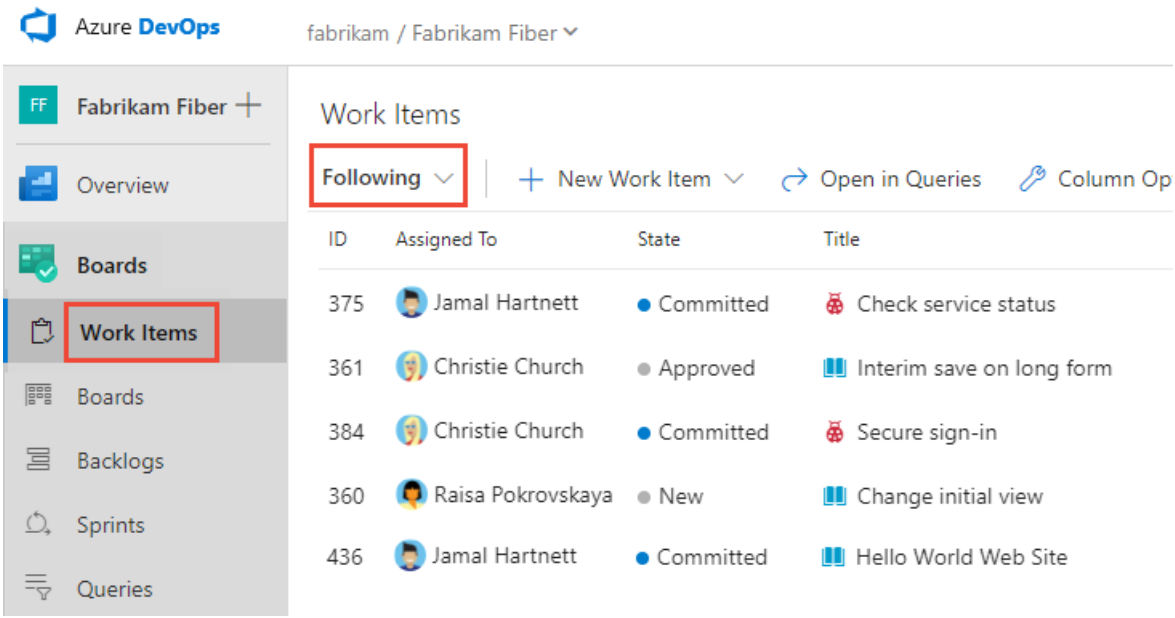
Open **Boards>Queries**, choose **All**, and under **My Queries**, choose **Followed work items**.



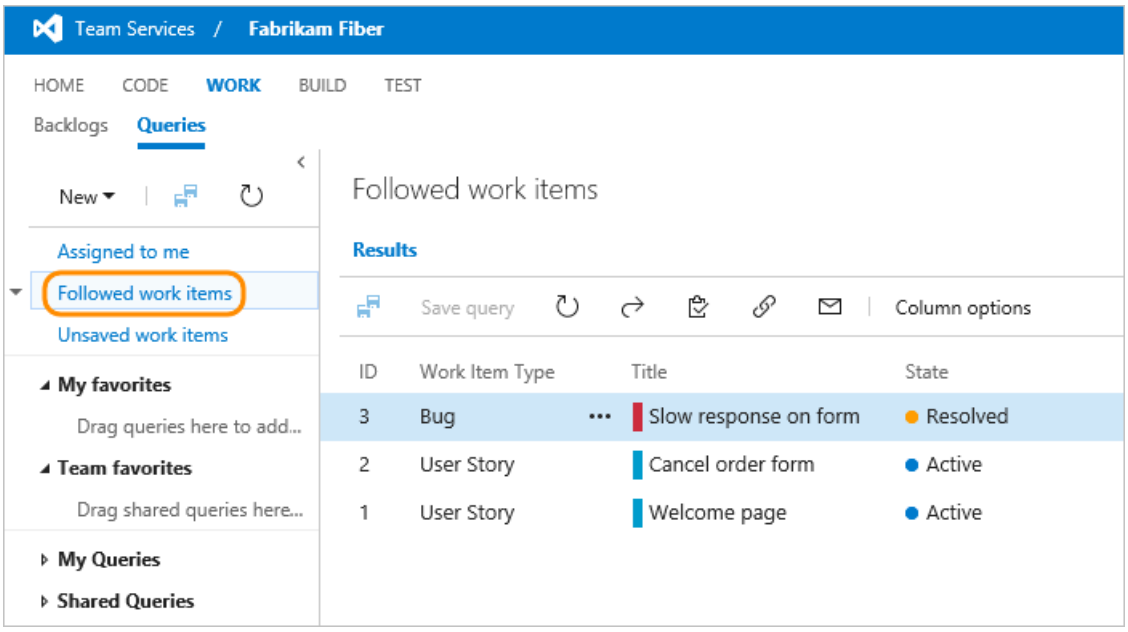
From this view, you can view all items you're following across all projects. Also, you can complete similar actions supported with a query results view, such as:

- Refresh the view
- Add or remove visible columns
- Sort the order of specific columns
- Filter results by text or tags
- Set work item pane
- Enter full screen mode.

You can also view and manage work that you're following from **Boards > Work Items** and pivot to **Following**.



Open **Work > Queries** and choose **Followed work items**.



From this view, you can view all items you're following across all projects. Also, you can complete similar actions supported with a query results view, such as:

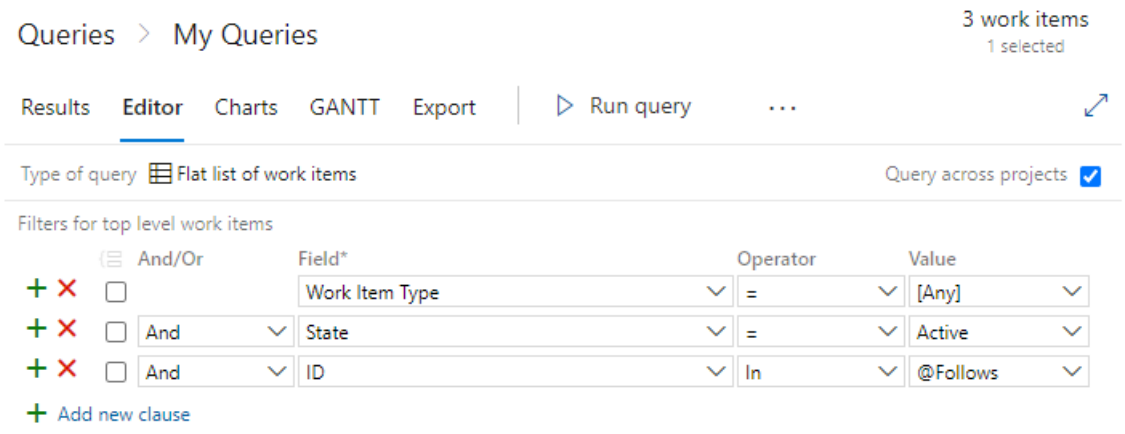
- Refresh the view
- Add or remove visible columns
- Sort the order of specific columns
- Filter results by text or tags
- Set work item pane
- Enter full screen mode.

You can also view and manage work that you're following from your Project pages. To learn more, see Work across projects.

Query work items that you're following

You can use the **@Follows** macro in a work item query to filter a list based on work items you're following along with other query filters.

For example, the following query shows how to query across all projects for active work items that you're following. You use the **ID** field and the **In** operator with the **@Follows** macro.



Try this next

Add, update, and follow a work item

Related articles

- Manage personal notifications
- View and update work items via the mobile work item form

Q: Can I add someone else to follow a work item or PR?

A: No, you can't add another team member to follow a work item or pull request at this time. You can subscribe them to get notified based on select criteria, such as when a work item is create or modified, or a pull request is created. For details, see Manage team notifications.

Use @mentions in work items and pull requests

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The **@mention** control allows you to quickly add a user or group to a work item or pull request discussion. Using the people picker of the **@mention** control, you can select a project member or group from the search list, and they'll receive an email notifying them of your comment.

For organizations that manage their users and groups using Azure Active Directory (Azure AD), people pickers support searching all users and groups added to Azure AD, not only those users and groups added to your project. To limit the set to project members and groups, see Manage your organization, Limit identity search and selection.

> **NOTE**
> You can post an @mention via API. Get the Azure DevOps User Id, and then add the following html code: `<div><a href="#" data-vss-mention="version:2.0,{user id}">@John Doe</a> Testing mentioning</div>` For more information, see the Microsoft Power Automate Community forum post.

The **@mention** control allows you to quickly add a user to a work item or pull request discussion. Using the people picker of the **@mention** control, you can select a project member from the search list, and they'll receive an email notifying them of your comment.

For organizations that manage their users using Active Directory, people pickers provide support for searching all users added to the Active Directory, not only those users added to your project.

Use the **@mention** control to start or continue a discussion within the following areas:

- A work item discussion or any rich-text field
- A pull request discussion
- Commit comments
- Changeset or shelveset comments

- A work item discussion
- A pull request discussion
- Commit comments
- Changeset or shelveset comments

> **NOTE**
> For on-premises Azure DevOps Server, configure an SMTP server for team members to see the **Notifications** option from their organization or user profile menu and to receive notifications.

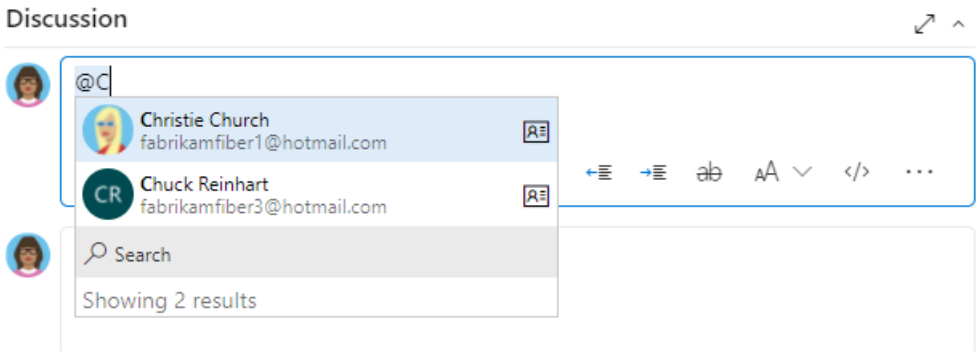
Identity search selection

When you leave a code comment in a pull request, enter **@** to trigger the **@mention** people picker. From the people picker, you see a list of users you've recently mentioned. To do a directory search, choose a name or enter the name of the user you're looking for.

WARNING

If you have permission to invite users to the organization, regardless of whether the **Restrict invitations** policy is disabled, you can **@mention** a user who isn't part of your organization. This action invites that user to your organization. For more information, see Restrict new user invitations from project and team administrators.

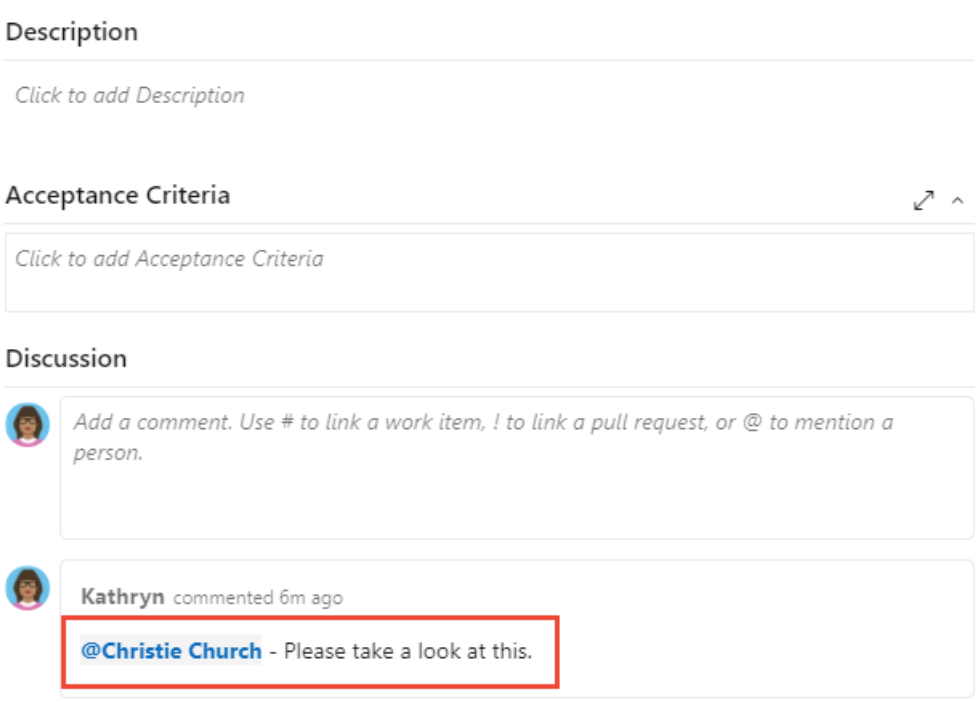
To filter the list, enter the user name or alias until you've found a match.

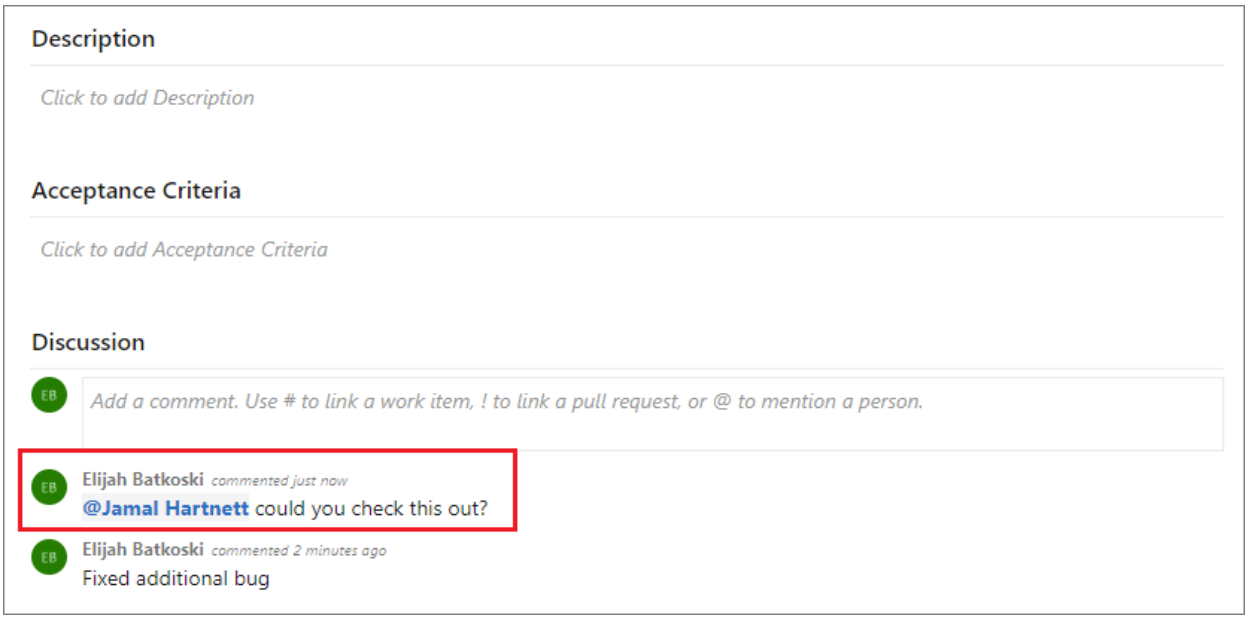


You can also use group mentions. Enter the name of a team or a security group, choose **Search**, and then select from the options listed.

To **@mention** a user you've never selected previously, continue to enter the entire name to do your search against the full directory.

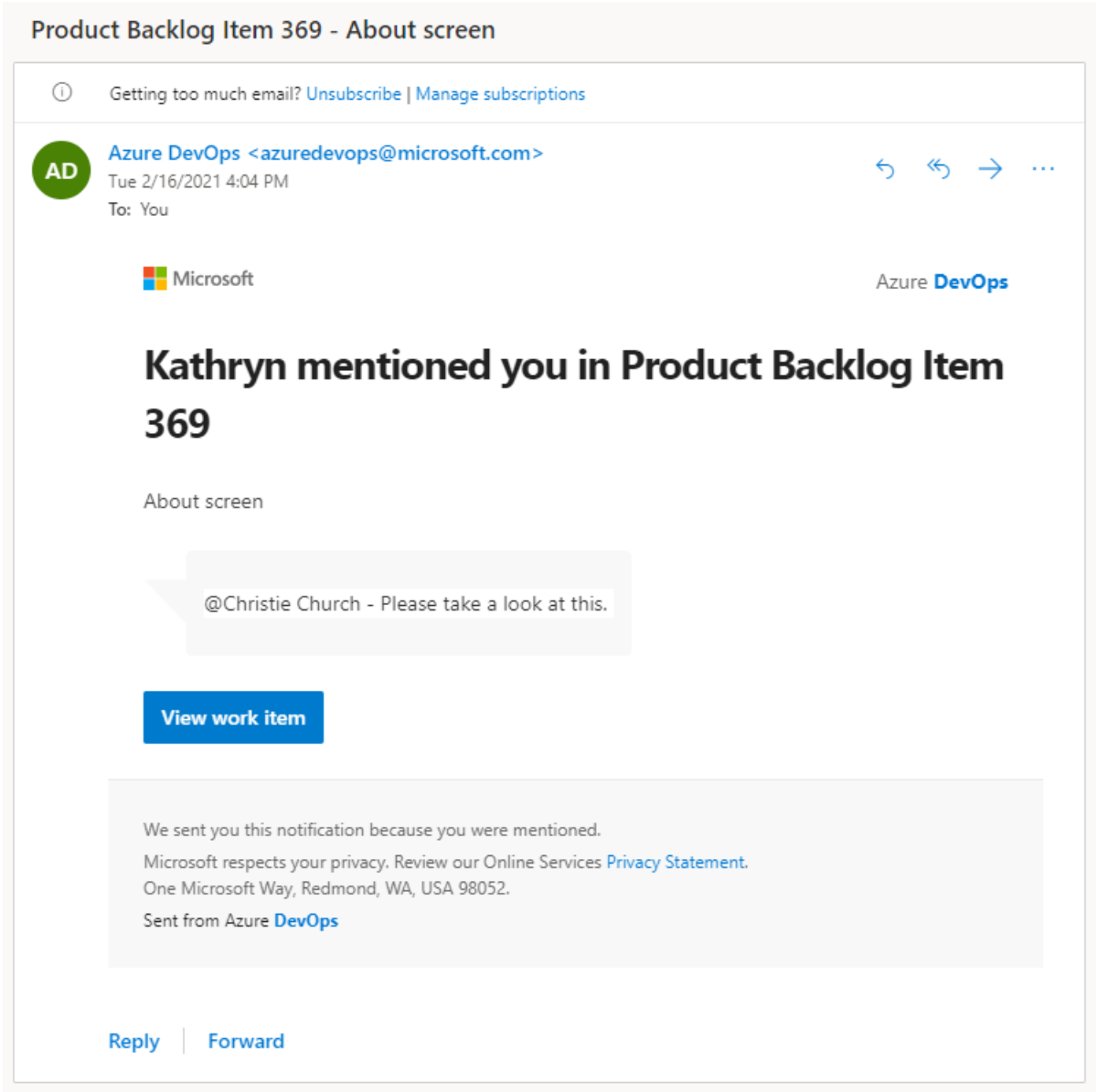
Names of mentioned users appear in blue text. Choose the **@mention link name** to open the user's contact information. The contact information provides more context for why they were added to the conversation.

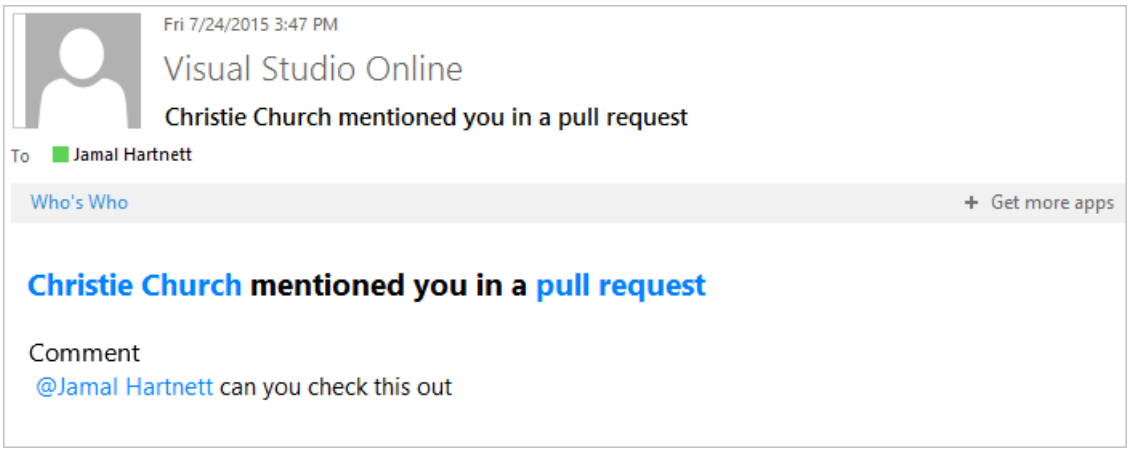




> **NOTE**
> Don't copy/paste **@mention** users from a previous comment. While the resulting formatting looks identical to a properly entered mention, it doesn't register as a true mention nor send an email notification.

Upon completion of your selection and text entry, your **@mention** user receives an email alerting them about the mention.





Use the **@mention** control in pull request discussions, commit comments, changeset comments, and shelveset comments.

Limited identities in search selection

In general, people pickers search and select any user or group added to an organization's Azure AD.

For organizations that manage their users and groups using Azure Active Directory (Azure AD), people pickers provide support for searching users and groups added to the Azure AD. For organizations that want to limit the search and selection to only those users and groups added to a specific project, they can do so by enabling the

Limit user visibility and collaboration to specific projects preview feature for their organization.

When the **Limit user visibility and collaboration to specific projects** preview feature is enabled for an organization, the list of identities you can select from a people picker is limited in one of the following ways:

- Users added to the **Project-Scoped Users** group are only able to select from an identity list that contains users and groups added explicitly to the project they're connected to.
- If all project members are added to the **Project-Scoped Users** group, then people pickers are limited to only those users and groups added to the project. All project members can only select identities that match users and groups added explicitly to the project they're connected to.

To learn how, see Manage your organization, Limit identity search and selection.

Related articles

- Work item form controls
- Pull requests

Add work item tags to categorize and filter lists and boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Tagging work items helps you quickly filter the product backlog or a work item query by categories that you define. A tag corresponds to a one or two keyword phrase that you define and that supports your needs to filter a backlog or query, or define a query.

Tags are a better choice to filter work items than using text strings as described in Guidance to create high-performing queries.

You can add and modify tags from the web portal, from Team Explorer plug-in for Visual Studio. Also, you can open a query in Excel to modify tags in bulk.

> **NOTE**
> Tags are a shared resource, they're associated with a project and not a team. If your project contains multiple teams, all teams will add to and work from the same set of tags.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

> **NOTE**
> Users with **Stakeholder** access for public projects are allowed to add new tags.

Add tags to a work item

Tags should be 400 characters or less and not contain separators such as a `,` (comma), `;` (semicolon), or other formatting character.

> **TIP**
> We recommend that you don't use the `@` character in a tag. Tags that start with the `@` character can't be used in a work item query. The `@` character signifies a macro within a query and therefore the tag isn't recognized as a tag.

From the web portal, open a work item and add a tag. Choose **Add tag** and type your keyword. Or, select from the list of previously assigned tags.



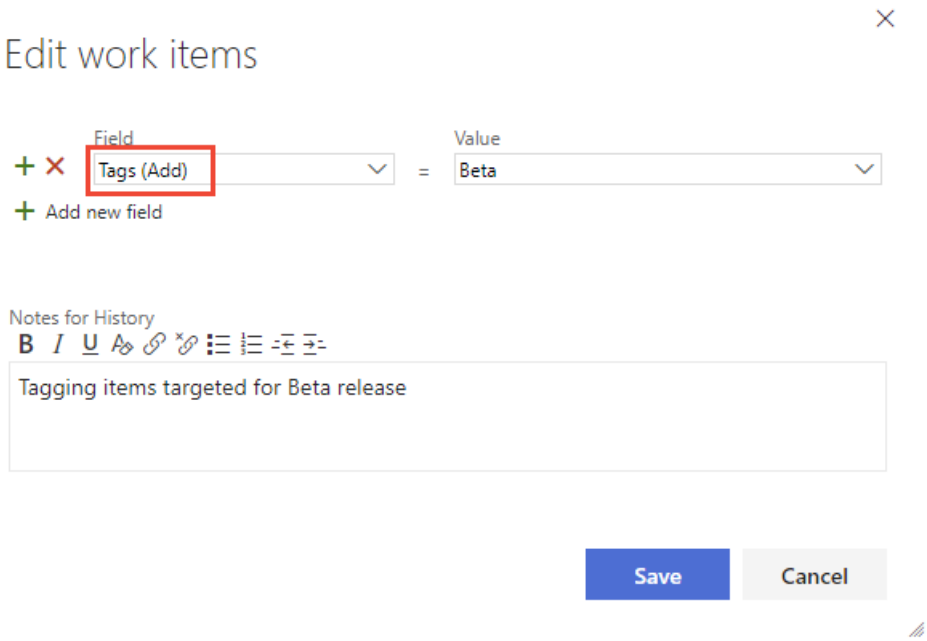
To add several tags at one time, type a comma between tags. Tags are case sensitive.

Tags that appear in the tag bar are already assigned to the work item. To unassign a tag, choose the x on the tag, Web ×.

> **NOTE**
> By default, all Contributors and Stakeholders of public projects are granted permissions to add new and existing tags. Stakeholders in private projects can add tags that are already defined, but not add new tags. To grant or restrict permissions to create new tags, you set the permission **Create tag definition** at the project-level. To learn more, see Change project-level permissions.

Bulk add or remove tags

You can bulk update work items to add or remove tags from the web portal. You bulk modify tags in the same way as you bulk modify other fields using the web portal. Or, you can use Excel to bulk add or remove tags.



NOTE

Bulk modify of tags from the Visual Studio or other supported clients isn't supported.

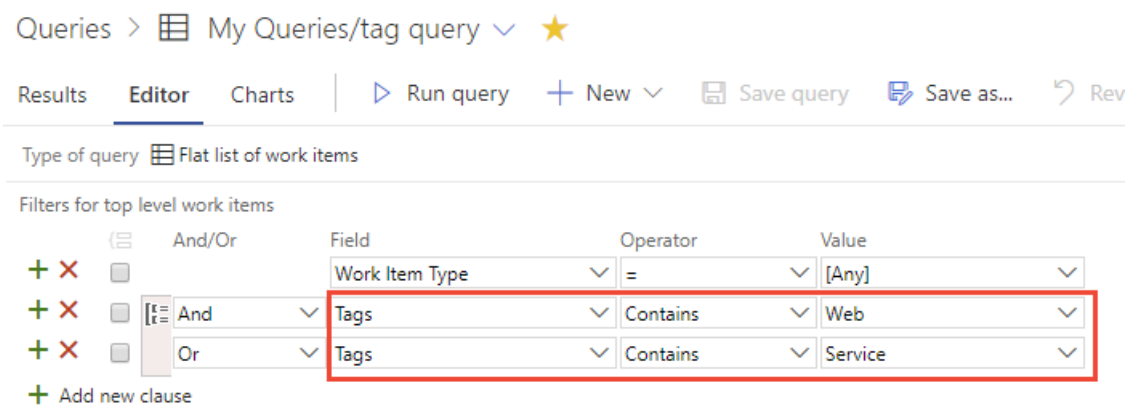
Query for work items based on tags

To query work items based on tags, add a clause for each tag you want to use to support your query.

TIP

You can use the **Contains** or **Does Not Contain** operators. Tags that start with the `@` character can't be used in a work item query as the query editor interprets the `@` character as a macro. To learn more about queries, see Create managed queries.

For example, here we query for all work items that are tagged either `Web` or `Service`.



TIP

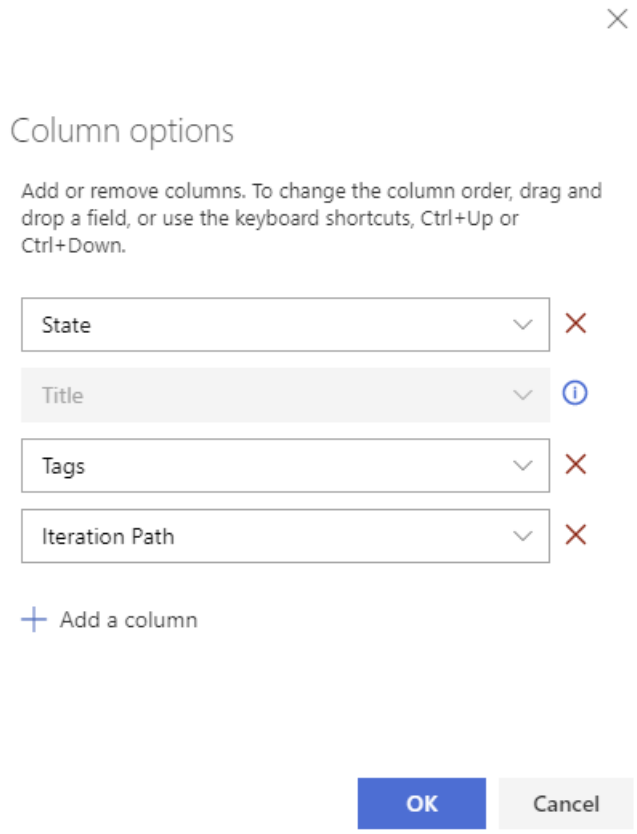
To understand how `AND/OR` clauses are grouped, see Create and save managed queries, Group clauses. To view the WIQL syntax for a query, install the WIQL query editor extension which allows you to see the WIQL version of any query editor entry.

NOTE

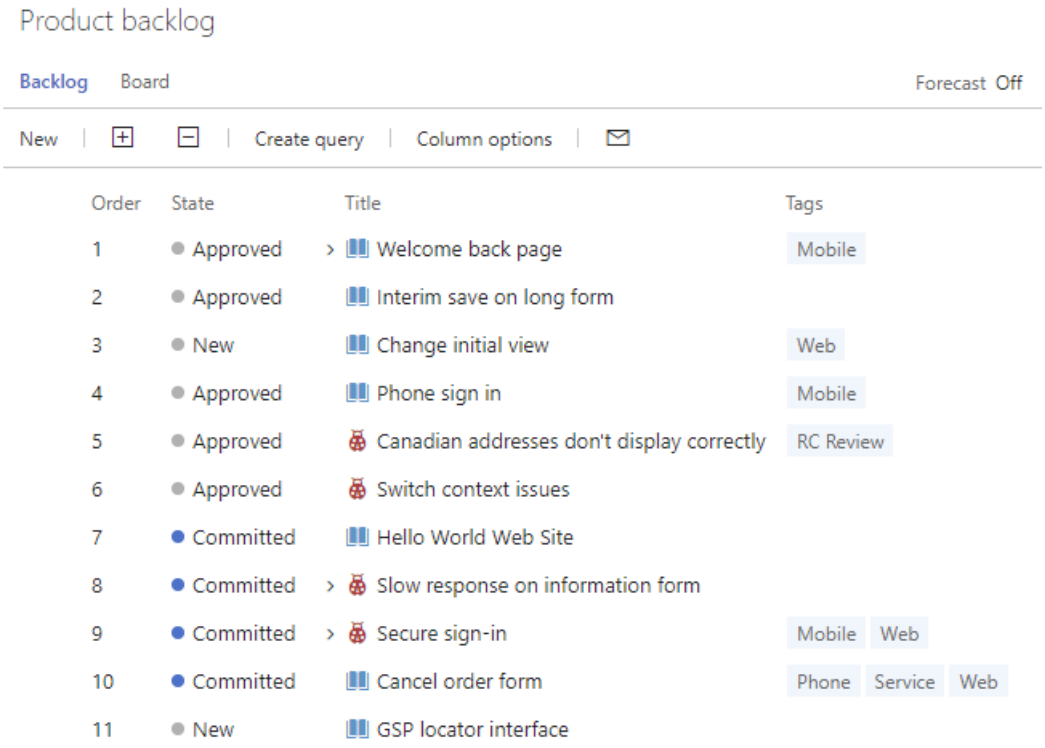
The ability to query for work items that don't have any tags attached to them is not a supported feature. If you'd like to up vote the request to support this feature, you can do so on our Developer Community page, Be able to search for empty tags.

Show tags in your backlog or query results

Choose **Column Options** to add the **Tags** field to the product backlog or a work item query. If the option doesn't appear, choose **More commands** ··· to select it from the menu of options.



All tags that have been added to the listed work items appear.

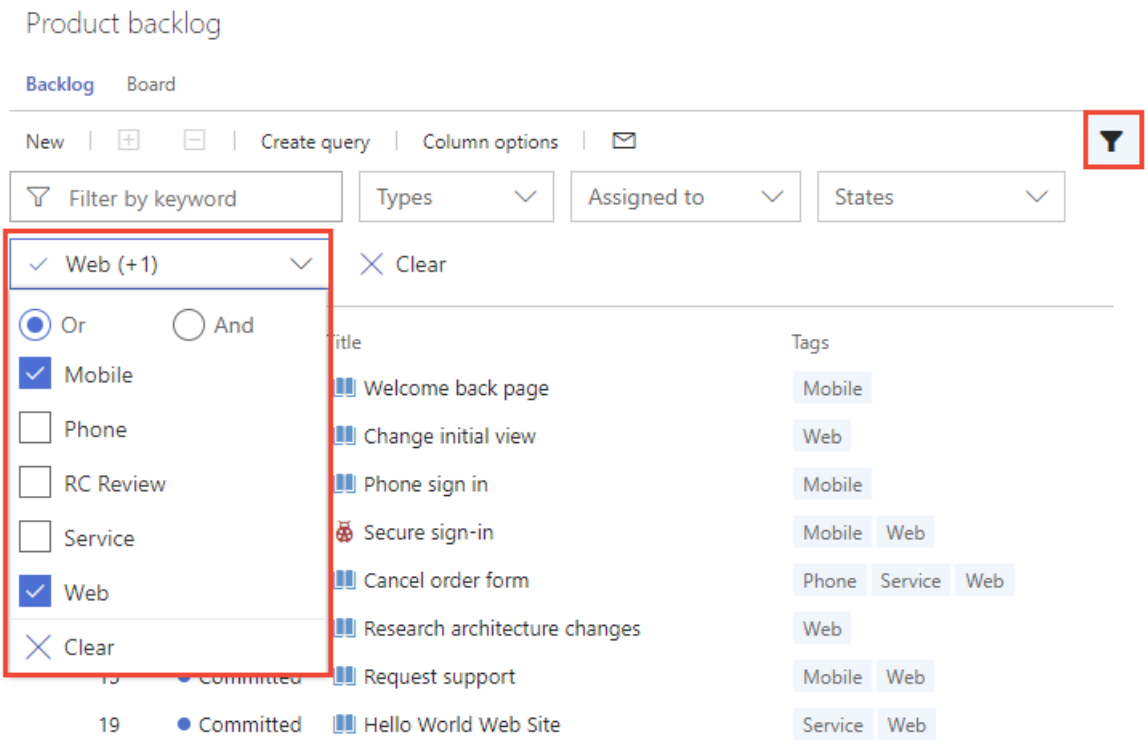


Filter lists using tags

From the web portal, you can filter backlogs, boards, and query results using tags.

Begin by choosing **Filter** .

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to run a logical OR for all the tags you selected. Or, choose the **AND** option to run a logical AND on all the selected tags.



Delete, remove, or manage tags

You can't delete a tag itself. However, if you delete a tag from all work items to which it's currently assigned, the

system will delete the tag. The system automatically deletes unassigned tags after three days of disuse.

If you misspell a tag, don't assign the misspelled tag to any work item and the system will automatically delete it within three days.

Another option is to install the Marketplace Tags Manager, which adds a **Tags** page under **Boards** or **Work** to manage tags.

Color-code tags on boards

You can highlight tags on Kanban board cards by color-coding them. These colors only appear on the Kanban board that you configure. They don't appear on backlogs or taskboards. To learn more, see Customize cards, color-code tags.

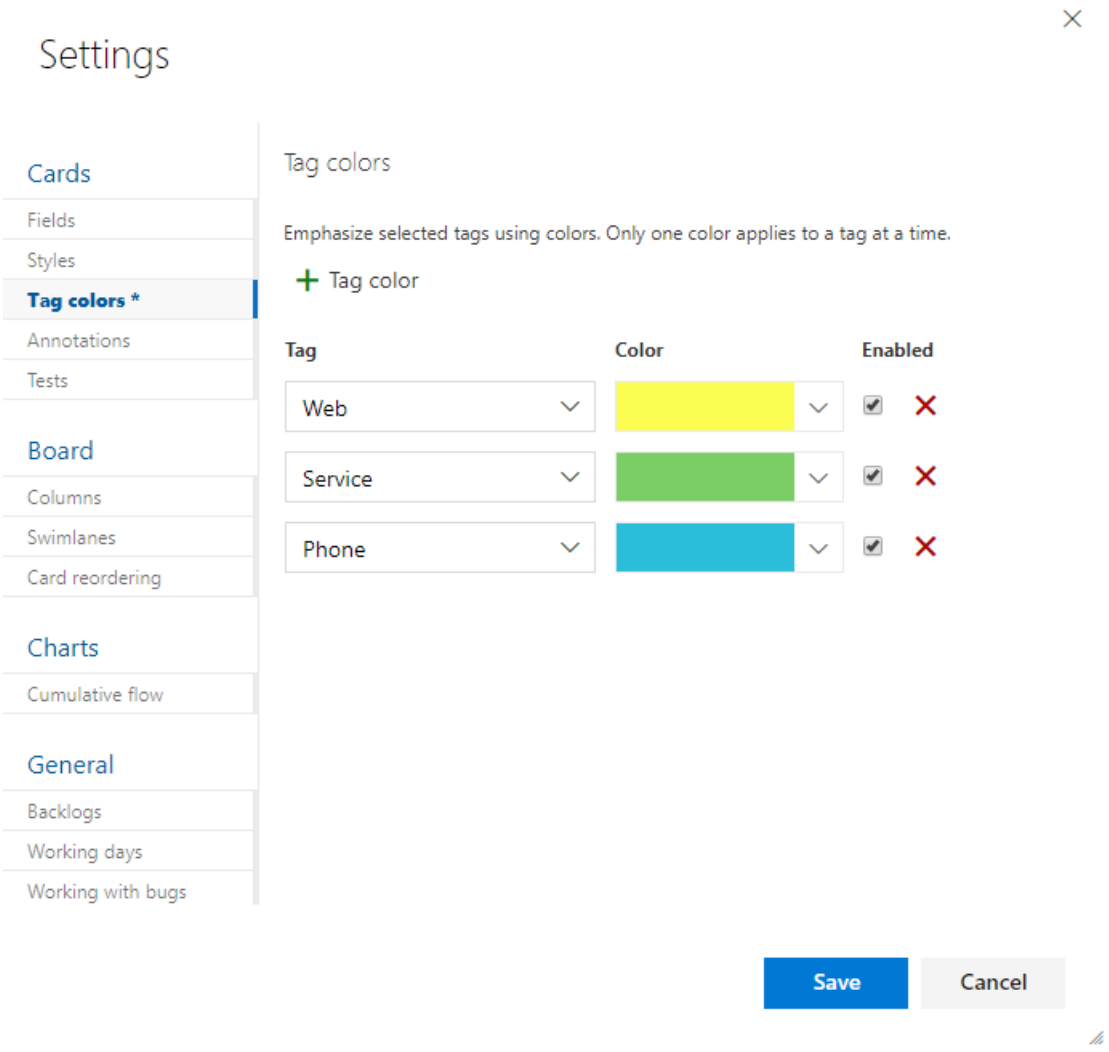
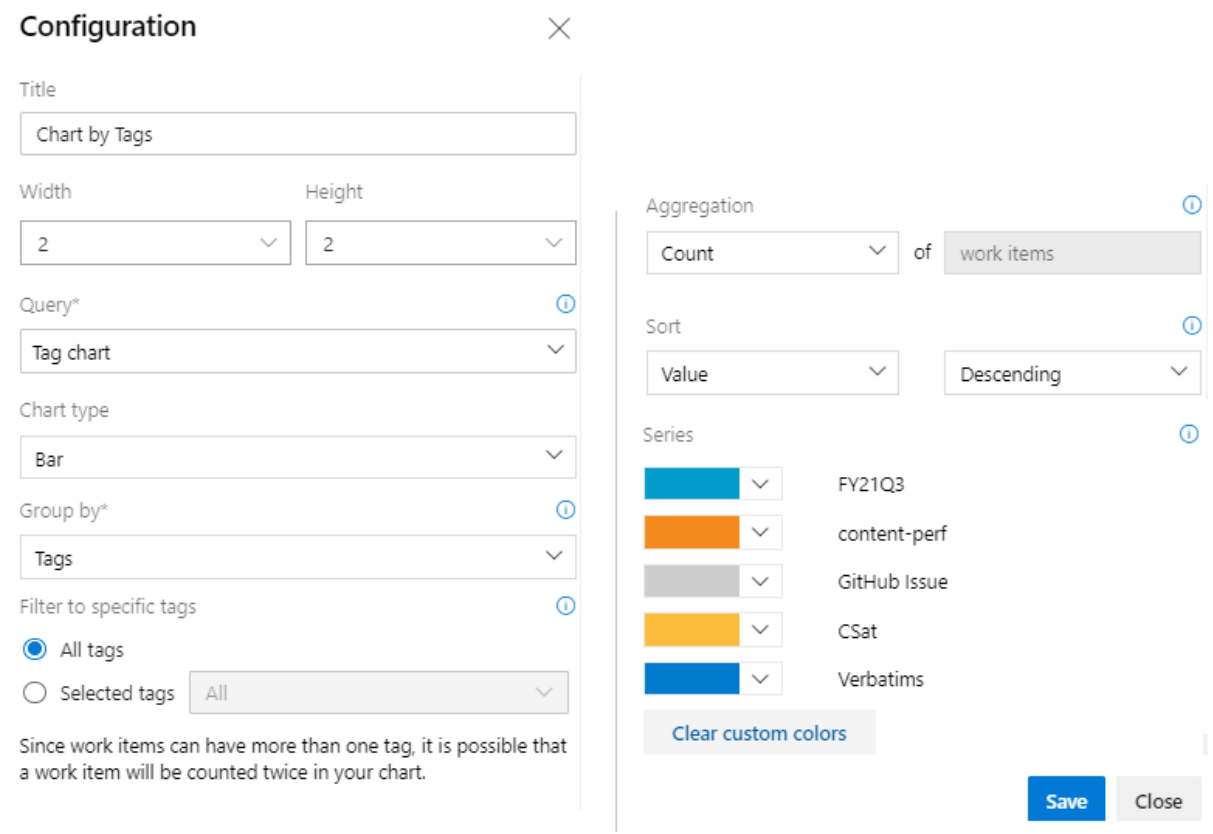


Chart work items and group by tags

> **NOTE**
> You can't group a query-based chart by tags, however, you can group a **Chart for Work Items** widget by tags that you add to a dashboard. This feature is in public preview. To enable it, see Manage or enable features and turn on **Enable group by tags for work item chart widget on dashboard**.

To group a **Chart for Work Items** widget by tags, complete the same steps provided in Track progress with status and trend query-based charts, Add a chart widget to a dashboard. Make sure that your flat-list query contains **Tags** in the query clause or as a column option. Then, choose **Tags** for the **Group by** selection. To filter

the chart to show only some tags, choose the **Selected tags** radio button and then choose the tags you want the chart to display.



Related articles

- Best tool to add, update, and link work items
- Use the query editor to list and manage queries
- Show tags on cards
- Bulk modify work items from the web portal
- Bulk modify work items from Excel

Marketplace extension

- Marketplace Tags Manager

Limits on the number of tags

While no hard limit exists, creating more than 100,000 tags for a project collection can negatively impact performance. Also, the autocomplete dropdown menu for the tag control displays a maximum of 200 tags. When more than 200 tags are defined, begin typing to cause the tag control to display relevant tags.

You can't assign more than 100 tags to a work item or you'll receive the following message:

```
TF401243: Failed to save work item because too many new tags were added to the work item.
```

Save the work item with the tags (100 or less) that you've added, and then you can add more tags.

Limit queries to fewer than 25 tags. More than that amount and the query will likely time out.

Add tags to the default column view on the product backlog

To add the **Tags** field as a column field for the product backlog, you modify the ProcessConfiguration file to include `System.Tags`. To learn how, see the Process configuration XML element reference.

Share information within work items and social tools in Azure Boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Using work items to track your work provides a host of benefits, including the ability to easily share information. You can capture most information within the work item *Description* or other rich-text formatted fields. If you need to maintain the information in a different format, you can easily link to or attach a file.

Other ways to share information include using dashboards, README files, and project Wikis.

Using work items, you can share information in the following ways:

- Add information to the Description or other rich-text field
- Link to a web site or file, or attach files
- Link to a storyboard file

> **TIP**
> If you have stakeholders who don't contribute code but want to contribute to the discussion and review progress, make sure you provide them stakeholder access so that they can view work items and dashboards.

Rich text fields

To convey detailed information, you can format text and insert images inline within any HTML field type. By default, the following fields are HTML fields and many others not listed here for the CMMI process.

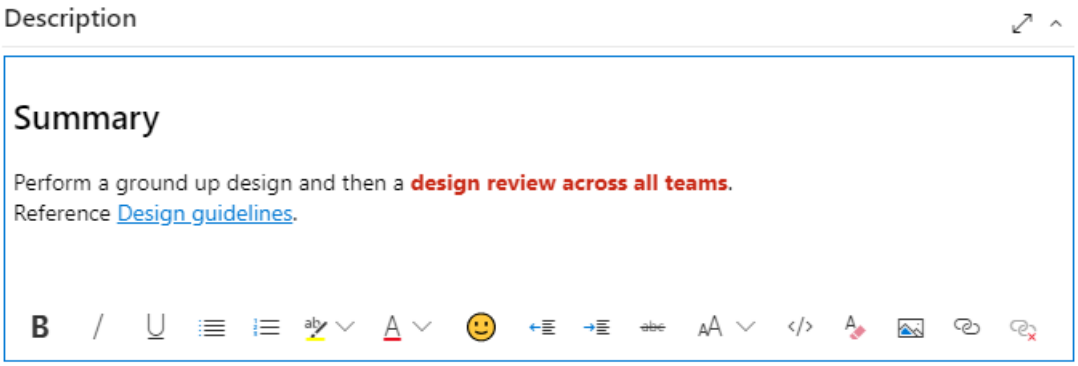
- *Acceptance Criteria*
- *Description*
- *Steps*
- *System Info*

You can add custom HTML fields to support your business processes. To learn more about each field, look it up in the Work item field index

The specific set of formatting features differs depending on the client you use. In all clients, you can bold, italicize, and underline text. You can also add and remove hyperlinks, format text as ordered or unordered lists, and add images.

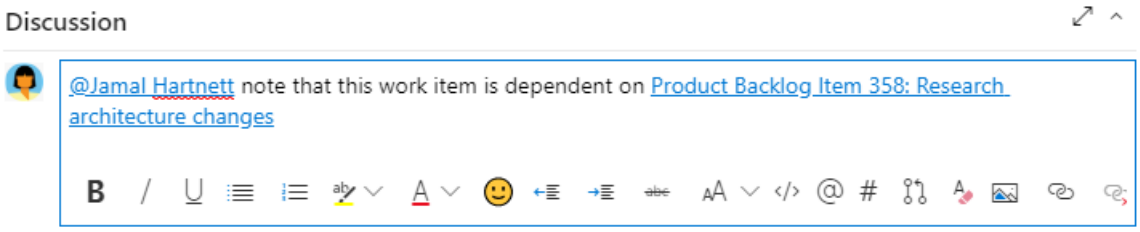
- Browser
- Visual Studio

The editor toolbar appears below each text box that accepts formatted text. It only becomes active when you move your cursor within the text box.



You can use the clear format icon or **CTRL+Spacebar** to remove formatting from highlighted text.

For the **Discussion** section, the tool bar comes with a few extra icons—@ at-mention, # #-work-item-id, and pull-request id—to help bring others into the discussion or link to work items or pull requests. Choose one of these icons and a menu displays with the most recent options that you've worked with.

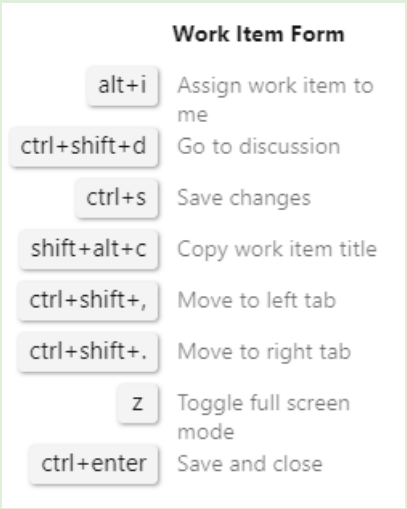


Keyboard shortcuts

You can copy and paste HTML text or an image from another application directly into the text box using **Ctrl+c** and **Ctrl+v** shortcuts. You can also use the keyboard shortcuts listed in Keyboard shortcuts.

TIP

Enter **Shift-?** to view additional Keyboard shortcuts for the work item form.



The rich text formatting toolbar appears above each text box that can be formatted. It only becomes active when you click within the text box.



You can use the following shortcut keys to format your text:

- **Bold**: Ctrl+B
- *Italic*: Ctrl+I
- Underscore: Ctrl+U

You can copy and paste HTML text or an image from another application directly into the text box using **Ctrl+C** and **Ctrl+V** shortcuts. You can also use the icon or **CTRL+Spacebar** to remove formatting from highlighted text.

Link items to work items or objects

To support traceability and add context to work items, you can link items to other work items or objects. From a links control tab, you can choose from various link types to based on the objects you want to link to.

Your link options will differ depending on the client you use. For specific guidance, see these resources:

- Add link to work items
- Link items to support traceability and manage dependencies

Attach files to work items

You and your team can centralize reference materials by attaching files to your work items. That way they're always readily accessible when you need them. For example, you can attach a screen image that illustrates a problem, a line of code in a text file, a log, an e-mail thread, or a product feature's specification.

You can add up to 100 attachments to a work item. Attachments are limited to 60 MB. Attempts to add more result in an error message upon saving the work item.

You can add up to 100 attachments to a work item. Attachments are limited to 60 MB.

By default, the size of work item attachments is limited to 4 MB. You can use the TFS web service to increase the size of files you attach up to 2 GB.

Choose the or **Attachments** tab to attach a file with supplemental information. The following file types support preview as attachments.

- **Image types**: "jpg", "jpeg", "png", "jif", "jfif", "jpx", "fpx", "pcd", "bmp", "img", "eps", "psd", "wmf", "gif"
- **Video Types**: "mp4", "mov", "m4v", "webm
- **Text Types**: "txt", "log"

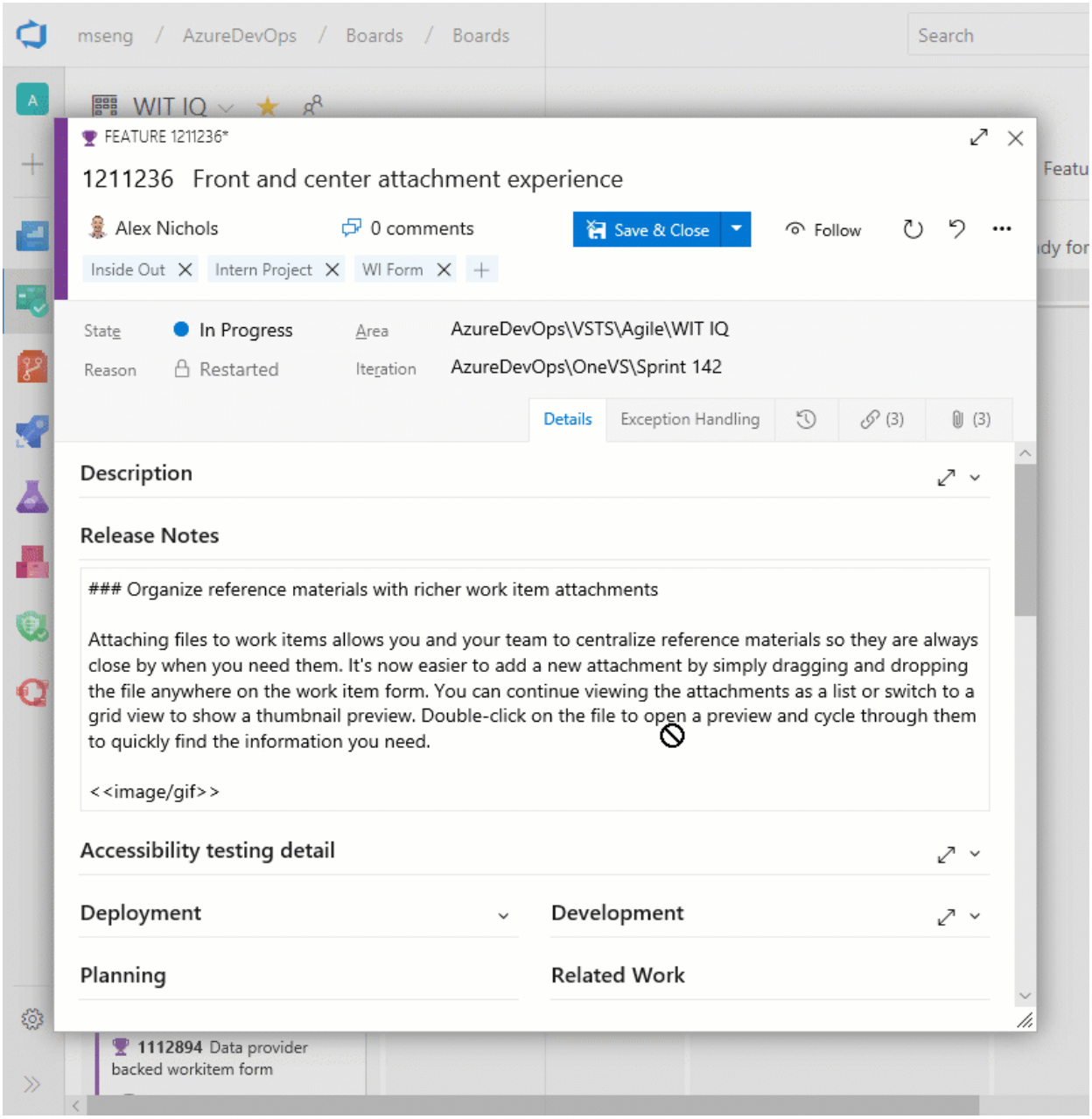
- Browser
- Visual Studio

Choose the Attachment tab icon to attach a file to the work item.

You can drag and drop a file onto the tab or anywhere on the work item form.

> **NOTE**
> Some features require upgrade to Azure DevOps Server 2019.1.

You can continue viewing the attachments as a list or switch to a grid view to show a thumbnail preview. Double-click or right-click on the file to open a preview and cycle through them to quickly find the information you need.

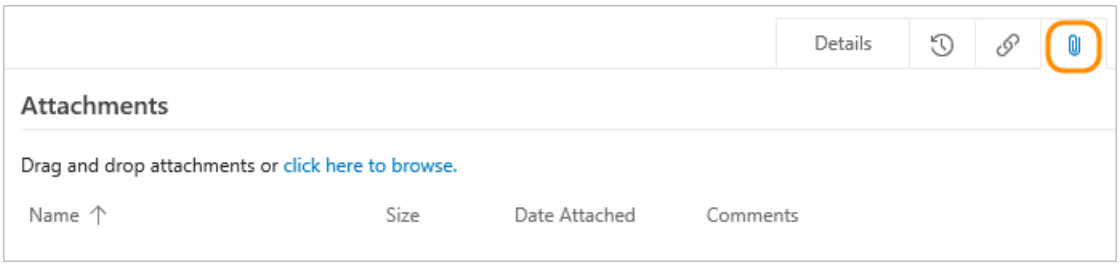


You can drag and drop files into the attachment area. From the browse menu, you can multi-select several files and attach within a single action. You can add attachments to your pull request comments. You can also add attachments in pull request comments by drag-and-drop or by browsing. For details, see Syntax support for Markdown files, widgets, and pull request comments, Attachments.

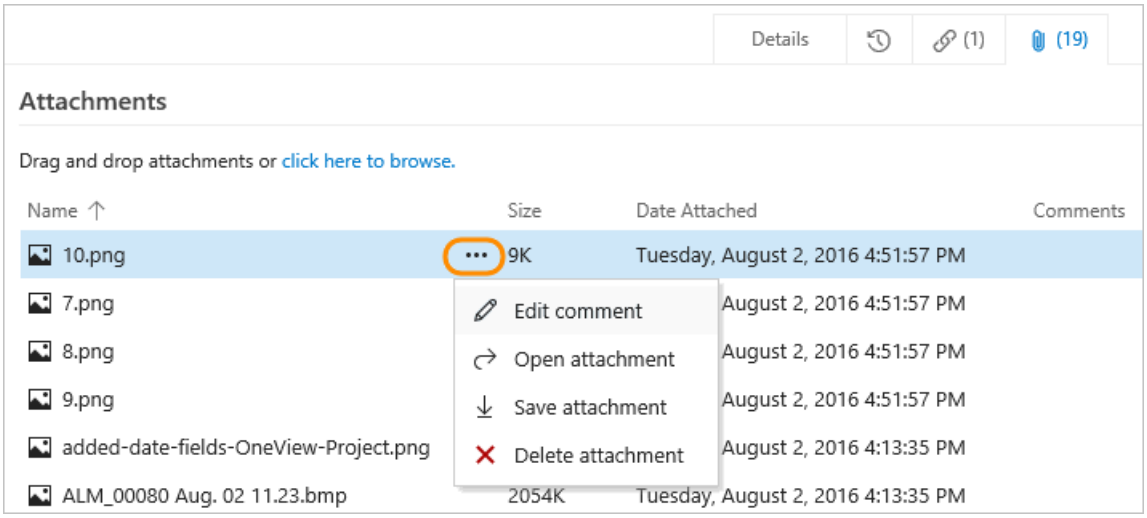
> **TIP**
> To get the URL of an image file you've attached, choose to preview it, right-click the image and choose the copy image address. Paste the address into a text editor and discard everything starting with **&download** to the end.

Choose the Attachment tab icon to attach a file to the work item.

You can drag and drop a file onto the tab or anywhere on the work item form.



You can edit, open, save, or delete an attachment by choosing an attachment and opening its ··· actions menu.



Storyboard your information

Storyboarding your ideas and goals increases visual understanding. With PowerPoint Storyboarding you can bring your ideas to life with storyboard shapes, text, animation, and all the other features that PowerPoint provides.

> **NOTE**
> Starting with Visual Studio 2019, the Team Foundation plug-in for Office deprecated support for Storyboarding with PowerPoint. Also, the Visual Studio Gallery for PowerPoint Storyboarding is deprecated. You can still use the Storyboard link type to link to network share objects to share information, however, you can't link from PowerPoint to Azure DevOps using the link feature.

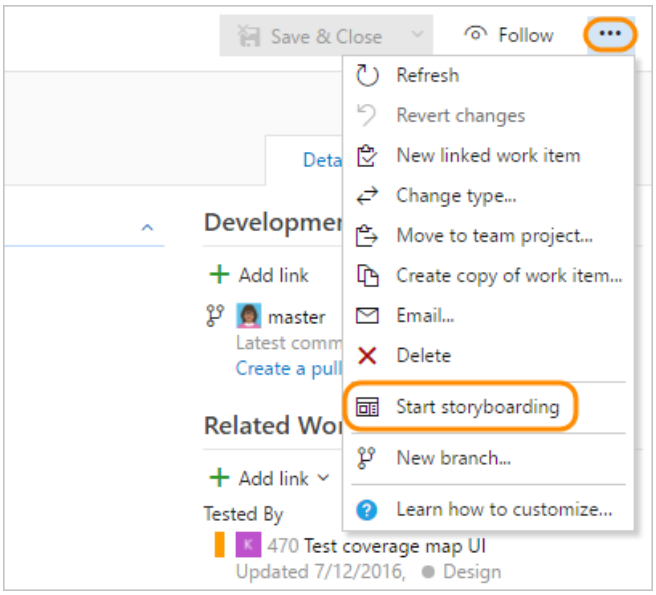
> **NOTE**
> Storyboarding with PowerPoint requires Office PowerPoint 2007 or later and the TFS Storyboarding add-in. You install the TFS Storyboarding add-in for PowerPoint by installing one of the latest editions of Visual Studio or Team Foundation Server Standalone Office Integration.

By linking your storyboard to a work item, you provide your team access to the shared file where they can add their comments. From the 🔗, **Links**, or a **Storyboards** tab, you can link storyboards that you created using PowerPoint Storyboarding or other application. When you make changes to a linked storyboard, the work item continues to link to the file with the latest changes.

To open PowerPoint with storyboarding, see Storyboard your ideas using PowerPoint.

- Browser
- Visual Studio

You can open Storyboarding with PowerPoint from the ··· actions menu within a work item form.



To link to an existing storyboard, click the Links tab and add a storyboard link.

Email a work item query list

A common way teams share information is by sharing a list of work items. You can email a formatted list or share a link to a query.

- Browser
- Visual Studio

You can quickly generate a formatted list using the **Copy as HTML** or **Copy to clipboard** options. See Copy list.

> **IMPORTANT**
> If you use the built-in email feature, you can only send the email to individual address for a project member that is recognized by the system. Adding a team group or security group to the to line isn't supported. If you add an email account that the system doesn't recognize, you receive a message that one or more recipients of your email don't have permissions to read the mailed work items.

Share data with team dashboards

You can share progress and status with your team using configurable team dashboards. Dashboards provide easy-to-read, easy access, real-time information. You can add widgets to provide Markdown information, query charts, and more.

For details, see Manage dashboards.

Share information with a project wiki

You can use your project wiki to share information with other team members. Each wiki corresponds to its own git repository and supports collaborative editing of its content and structure.

To learn more, see Create a wiki for your project.

Marketplace extensions

You may find more ways to share information by exporting work items to other applications such as Microsoft Word. To learn more, review the Marketplace extensions that support Microsoft Word.

Related articles

As you can see, there are many ways to share information using work items alone. See these other tools and features to support planning, tracking, and sharing information with your team.

- Dashboards
- Add and edit a wiki
- Work tracking, process, and project limits

Copy a list of stories, issues, or other work items

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

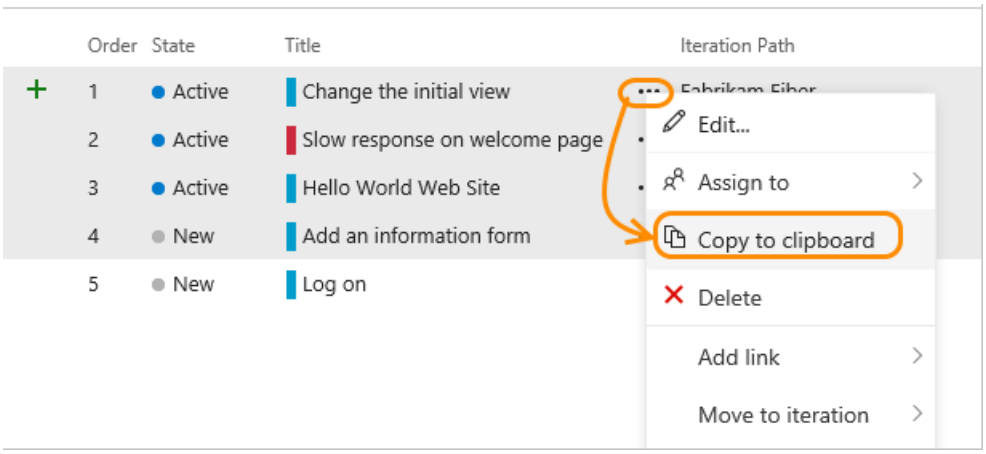
You can copy an HTML formatted table of selected items from either a backlog page or query results list. You can then email this list using your choice of email client, or paste into a Word document, Excel spreadsheet, or other application.

> **TIP**
> The data copied with **Copy to clipboard** is the same as that copied when you select **Email ...**.

> **TIP**
> The data copied with **Copy as HTML** is the same as that copied when you select **Email selected work items**. If you don't have an SMTP server configured, you can work around this by using **Copy as HTML**. For on-premises Azure DevOps, all email actions require an SMTP server to be configured.

1. From the web portal, open a backlog or query results page. Then, multi-select the work items you want to copy to the clipboard.
2. Open the ... context menu of one of the selected work items, and then choose **Copy to clipboard** or **Copy as HTML**.

   Here we multi-select from the product backlog and choose **Copy to clipboard**.



Paste the contents into your email client

Once you've copied your list, you can paste the contents of the clipboard into your email client or other application. To open a linked work item, requires users to have read access to the project or area node for those work items.

The formatted table contains a link to each work item included in your selected results list. A link to a query that will open only those work items selected is also provided.

ID	Work Item Type	Title	State
398	User Story	Cancel order form	Active
468	User Story	Customer account history	Active
391	Bug	Slow response on form	Active
395	Bug	Canadian addresses don't display	New
346	User Story	Add animated emoticons	New

Open as query

Related articles

- Email or print work items
- Share information within work items and social tools

Email or print user stories, bugs, and other work items

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

** Visual Studio 2019 - Visual Studio 2015 | Team Explorer Everywhere**

Using work items to track your work provides a host of benefits, including the ability to easily share information. You can capture most information within the work item Description or other rich-text formatted field. If you need to maintain the information in a different format, you can easily link to or attach a file.

Supported tasks

Emailing lists of work items is a common way to share work tracking information. The following table indicates which tasks or features are supported from the web portal, Visual Studio, or the Eclipse plug-in, Team Explorer Everywhere (TEE).

Task/feature	Web portal	Visual Studio 2019-2015	TEE (Eclipse)
Email summary list with links to work item(s)	✔	✔	✔
Print work item(s)	✔		
Email link to a work item query	✔	✔	
Email query results	✔	✔	
Export query result as CSV	✔		
Visual Studio 2019/Team Explorer			

The tasks/features listed in the table aren't available when you're connected to a GitHub or third-party Git repository. Also, they aren't available from Visual Studio 2019 under the following conditions:

- If you're set to use the default Landing page experience as described in Set the Work Items experience in Visual Studio 2019.
- If you're set to use the new Git Tool as described in Git experience in Visual Studio.

> **NOTE**
> For the email feature to work, your administrator for Azure DevOps Server must configure a Simple Mail Transfer Protocol (SMTP) server.

Make sure you provide members of your organization Stakeholder access who want to contribute to the discussion and review progress. These are typically members who don't contribute to code, but want to view work items, backlogs, Kanban boards, and dashboards.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

Email a single item

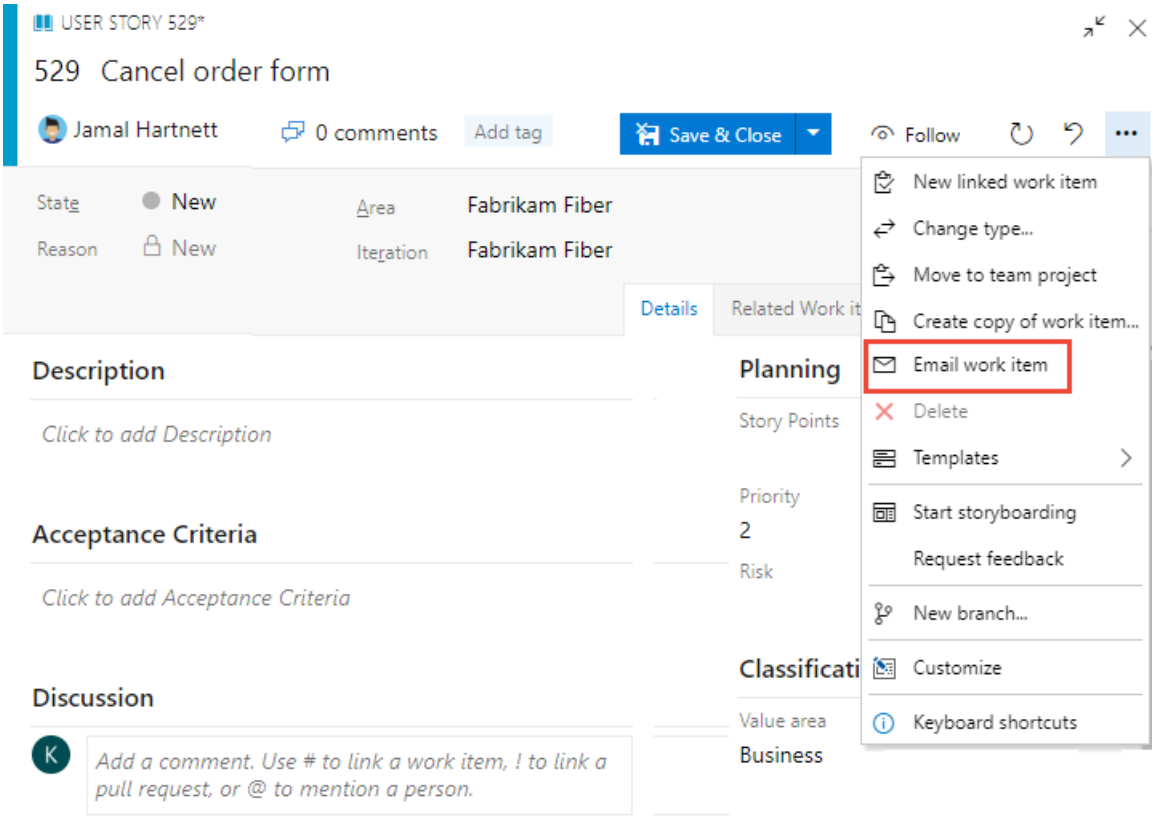
You can quickly email a summary of one or more work items. Summaries include the values assigned to these fields: work item ID, title, work item type, assigned to, state, and tags.

IMPORTANT

If you use the built-in email feature, you can only send the email to individual address for a project member that is recognized by the system. Adding a team group or security group to the to line isn't supported. If you add an email account that the system doesn't recognize, you receive a message that one or more recipients of your email don't have permissions to read the mailed work items.

- Web portal
- Visual Studio
- Team Explorer Everywhere

From the web portal, open the work item, choose the ··· actions icon, and select the **Email work item** option. The first 200 items in the list will appear in a formatted table.



NOTE

If you connect to an on-premises Azure DevOps Server, your server administrator must have configured an SMTP server for the email feature to work.

Email summary lists with links to items

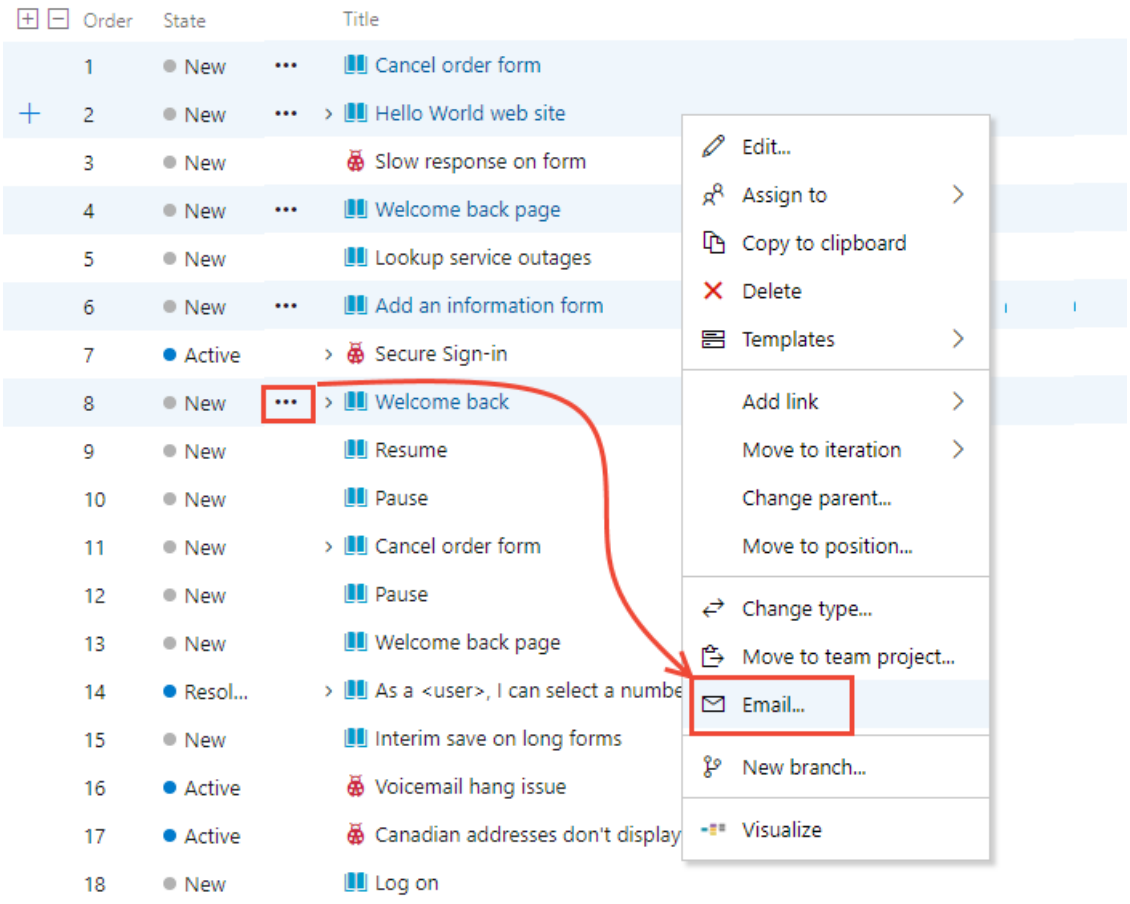
Another way to share items is by emailing summary lists, such as a sprint summary plan or active bugs list. You can share items from a backlog or query results list.

Depending on the option and client you choose, summary lists may or may not include a hyperlink to the work item ID.

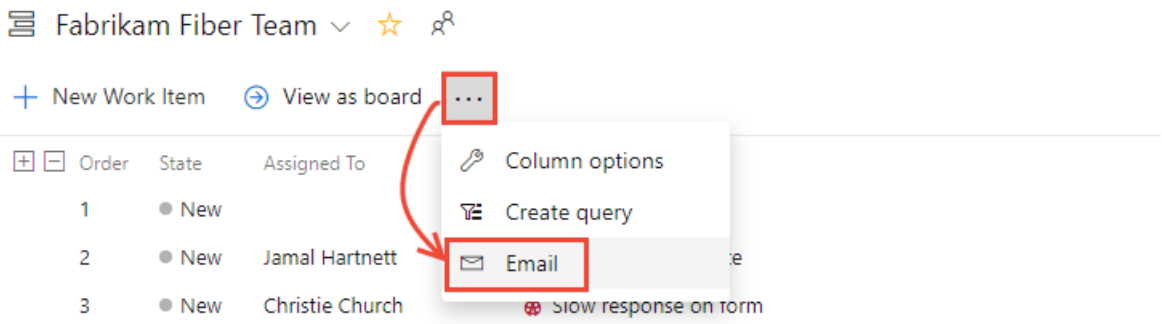
- Web portal
- Visual Studio

- Team Explorer Everywhere

To email items from the web portal: Open a backlog or query and highlight the items from the list. Open the context menu for one of the selected items and select to email them.



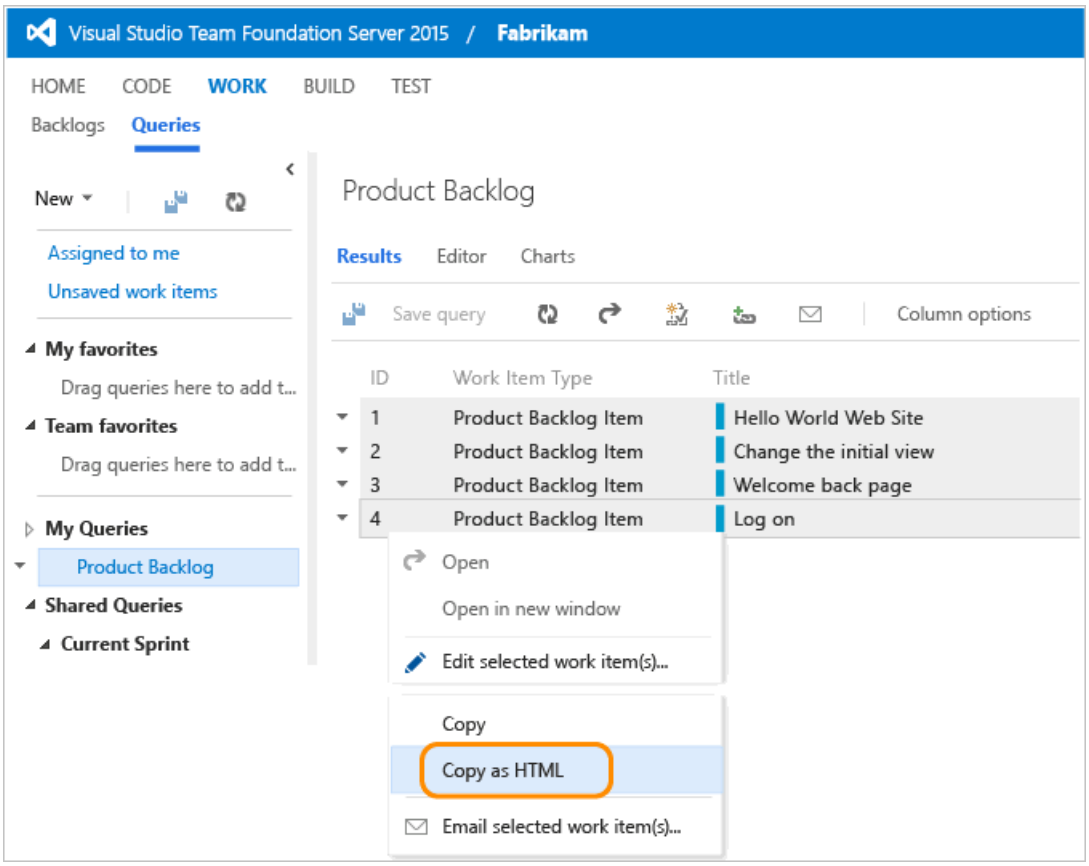
If you want to mail a list of all items in the backlog or query, choose the ··· actions icon, and select the **Email** option.



Copy formatted list of work items

With this option, you can copy an HTML formatted table of selected items. You can then email this list using your choice of email client.

1. From the web portal, open a backlog or a list of query results.
2. Select the work items you want to copy.



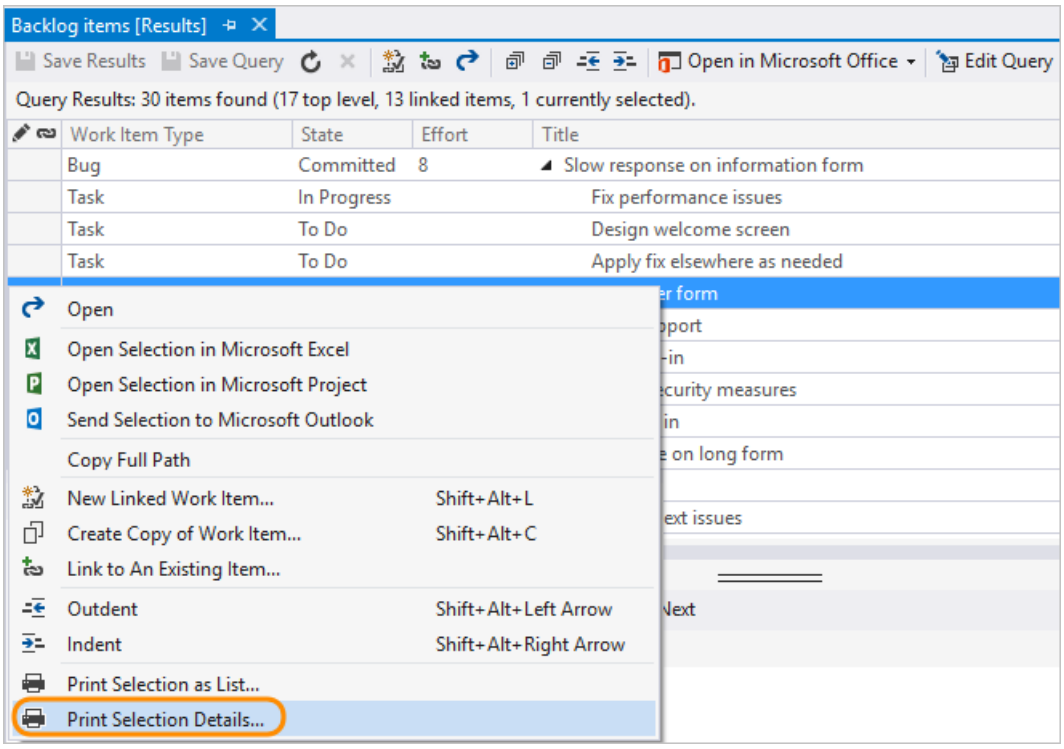
The formatted table contains a link to each work item included in your selected results list.

3. Paste the contents of the clipboard into your email client or other application. To open a linked work item, requires users to have read access to the project or area node for those work items.

Print items

To print work item details, open a query in Visual Studio that contains the work item(s) you want to print, and select or highlight those items that you want to print. Then, choose the **Print** option from the context menu.

> **IMPORTANT**
> To print work items in Visual Studio 2019, you need to Set the Work Items experience to the legacy option.



Print work items as cards

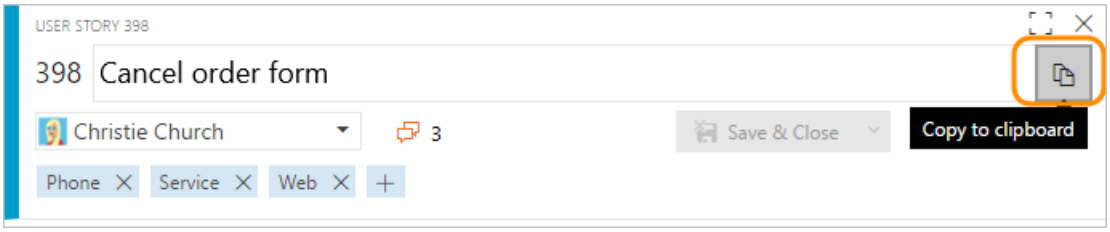
Some teams want to work with physical cards when planning or updating their physical Kanban or Scrum task boards. There's no native support for printing work items as cards. However, you may find a solution from the Azure DevOps Marketplace.

Copy the URL to a single work item

> **NOTE**
> All URLs you copy, regardless of the client you use to copy them, opens the work item in the web portal.

- Web portal
- Visual Studio
- Team Explorer Everywhere

From the web portal, copy the URL from the web browser address or hover over the title and then select the copy-to-clipboard icon.

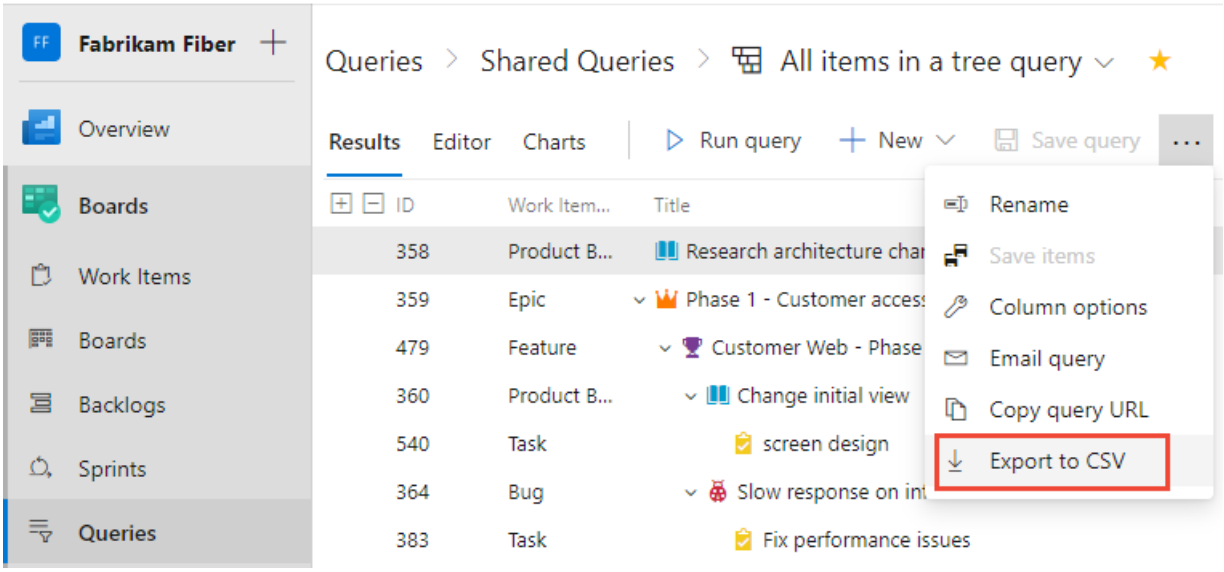


Export list as CSV

From any query, you can export a list of work items as a comma-delimited list. Open the query, choose the ··· actions icon, and choose **Export to CSV**. To learn more, see Bulk import or update work items using CSV files.

NOTE

Requires Azure DevOps Server 2019 Update 1 or later version.



Marketplace extensions

You may find other ways to share information by exporting work items to other applications such as Microsoft Word. To learn more, review the Marketplace extensions that support Microsoft Word.

Related articles

- Use templates to add and update work items
- Share information in work items and social tools
- Define the hyperlink for a work item

- Use templates to add and update work items
- Share information in work items and social tools
- Define the hyperlink for a work item

- Configure an SMTP server

Move work items from one team to another team

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you add a team or your teams undergo reorganization, you may need to move work items assigned to one team to new **Area Paths** owned by another team. All work items are assigned to an **Area Path**, even if it is at the top of the hierarchy for the project.

Work items that belong to the Requirements category appear on a team's backlog based on their assignment to the **Area Path(s)** owned by a team. Assigning other work items to a team's **Area Path(s)** support queries based on team ownership.

To learn how to add a team, see Create or add a team.

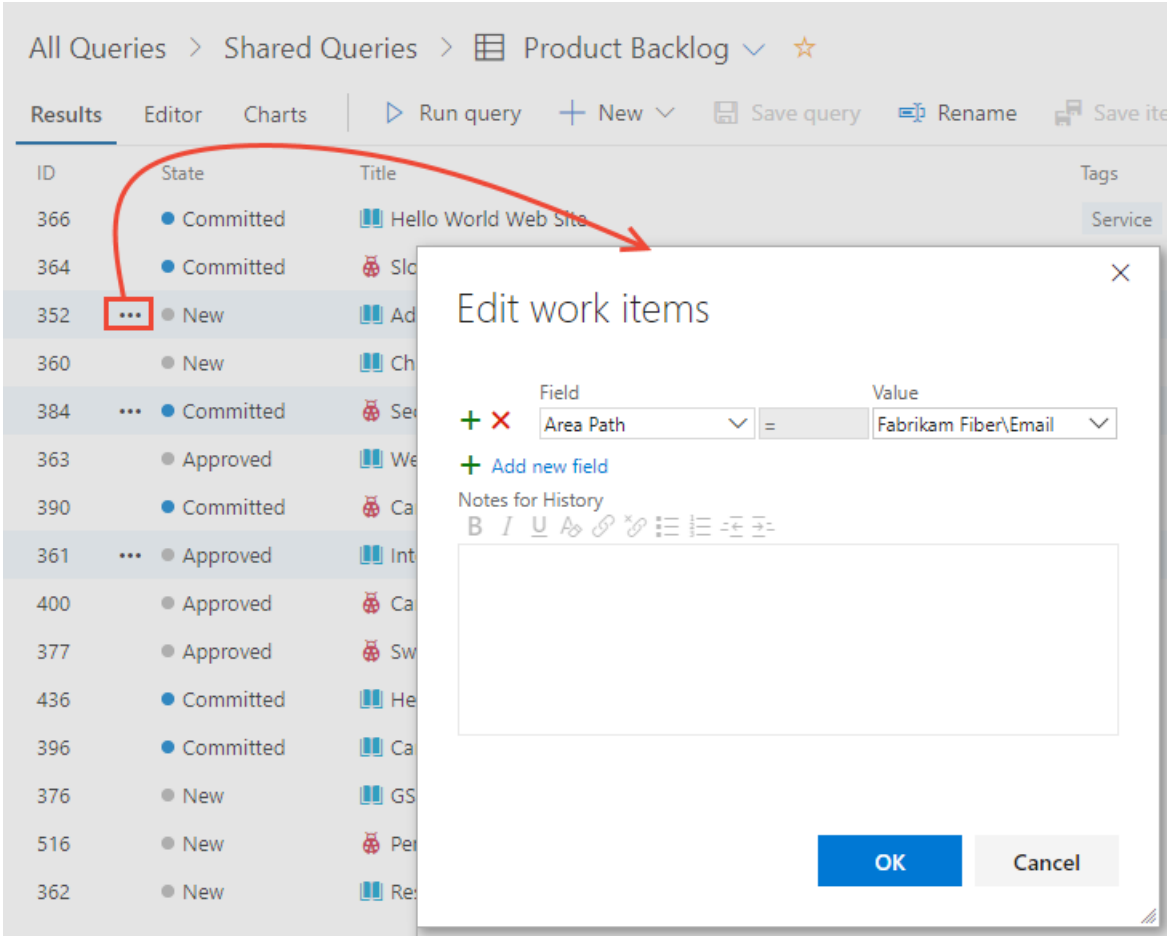
Prerequisites

- To change the **Area Paths** of work items, you must be a project member and have permissions to view and edit work items under the **Area Path** nodes. To learn about these permissions, see Set work tracking permissions, Create child nodes, modify work items under an area or iteration path.
- To use Azure CLI commands, you must first install Azure CLI as described in Get started with Azure DevOps CLI.
- To change the **Area Paths** of work items, you must be a project member and have permissions to view and edit work items under the **Area Path** nodes. To learn about these permissions, see Set work tracking permissions, Create child nodes, modify work items under an area or iteration path.

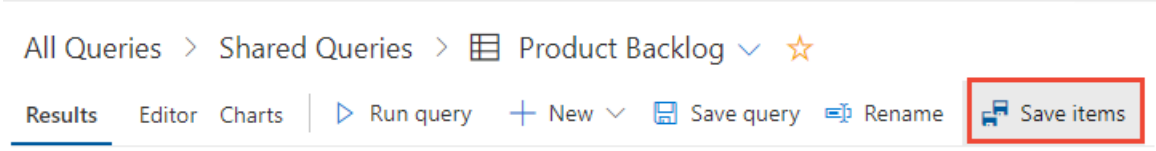
Move work items under teams

From the web portal, you can perform bulk updates of the **Area Path**.

1. Create a query of all work items you want to reassign.
2. Multi-select those items, which belong to each team, and bulk edit the area path.

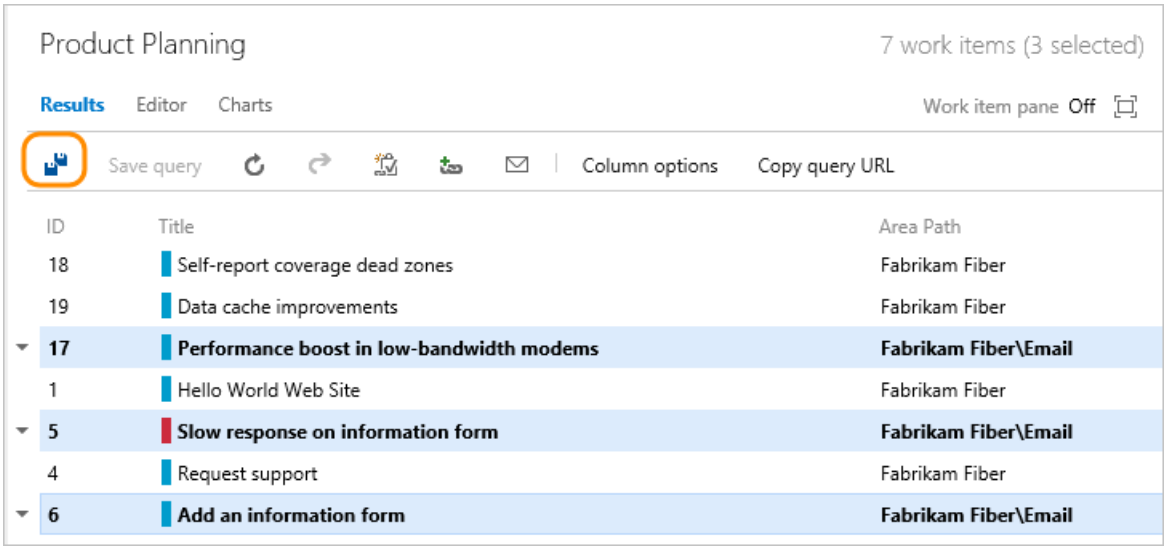


3. After you bulk modify, do a bulk save.



1. Create a query of all work items you want to reassign. Multi-select those items belonging to each team, and bulk edit the area path.

2. After you bulk modify, do a bulk save.



Move a work item using Azure CLI

You can use the az boards work-item update to move a single work item by updating it's Area Path.

```
az boards work-item update --id
                           [--area]
                           [--assigned-to]
                           [--description]
                           [--discussion]
                           [--fields]
                           [--iteration]
                           [--open]
                           [--reason]
                           [--state]
                           [--title]
```

Parameters

- **id**: Required. The ID of the work item to update.
- **area**: Optional. Absolute path of an area. Example: --path \ProjectName\Area\AreaName.
- **assigned-to**: Optional. Name of the person the work item is assigned to *Jamal*.
- **description**: Optional. Description of the work item.
- **discussion**: Optional. Comment to add to a discussion in a work item.
- **fields**: Optional. Space separated "field=value" pairs for custom fields you want to set.
- **iteration**: Optional. Absolute path of an iteration. Example: \ProjectName\Iteration\IterationName.
- **open**: Optional. Open the work item in the default web browser.
- **reason**: Optional. Reason for the state of the work item.
- **state**: Optional. State of the work item, for example, *Active*.
- **title**: Optional. Title of the work item.

Example

You can only move one work item at a time using Azure DevOps CLI. In this example, we move work item *ID=148* under the *Fabrikam Fiber\Production Planning* area path.

```
az boards work-item update --id 148 --area "Fabrikam Fiber\Production Planning" --output yaml
```

The YAML output listed below provides information on each of the fields defined for the work item.

```
fields:
  Microsoft.VSTS.Common.Priority: 2
  Microsoft.VSTS.Common.StackRank: 1500000001.0
  Microsoft.VSTS.Common.StateChangeDate: '2021-11-23T22:26:28.27Z'
  Microsoft.VSTS.Common.ValueArea: Business
  System.AreaPath: Fabrikam Fiber\Production Planning
  System.AssignedTo:
    _links:
      avatar:
        href:
https://fabrikamprime.visualstudio.com/_apis/GraphProfile/MemberAvatars/aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEt
NGZlYmU2NGE1NTM4
    descriptor: aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEtNGZlYmU2NGE1NTM4
    displayName: Jamal Hartnett
    id: d291b0c4-a05c-4ea6-8df1-4b41d5f39eff
    imageUrl:
https://fabrikamprime.visualstudio.com/_apis/GraphProfile/MemberAvatars/aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEt
NGZlYmU2NGE1NTM4
    uniqueName: fabrikamfiber4@hotmail.com
    url: https://spsprodeus27.vssps.visualstudio.com/A5d5b8da6-3db7-4829-baf9-
1e500c21cc12/_apis/Identities/d291b0c4-a05c-4ea6-8df1-4b41d5f39eff
  System.BoardColumn: Backlog
  System.ChangedBy:
   _links:
      avatar:
        href:
```

```
https://fabrikamprime.visualstudio.com/_apis/GraphProfile/MemberAvatars/aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEt
NGZlYmU2NGE1NTM4
    descriptor: aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEtNGZlYmU2NGE1NTM4
    displayName: Jamal Hartnett
    id: d291b0c4-a05c-4ea6-8df1-4b41d5f39eff
    imageUrl:
https://fabrikamprime.visualstudio.com/_apis/GraphProfile/MemberAvatars/aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEt
NGZlYmU2NGE1NTM4
    uniqueName: fabrikamfiber4@hotmail.com
    url: https://spsprodeus27.vssps.visualstudio.com/A5d5b8da6-3db7-4829-baf9-
1e500c21cc12/_apis/Identities/d291b0c4-a05c-4ea6-8df1-4b41d5f39eff
  System.ChangedDate: '2022-05-19T22:58:52.93Z'
  System.CommentCount: 0
  System.CreatedBy:
   _links:
      avatar:
        href:
https://fabrikamprime.visualstudio.com/_apis/GraphProfile/MemberAvatars/aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEt
NGZlYmU2NGE1NTM4
    descriptor: aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEtNGZlYmU2NGE1NTM4
    displayName: Jamal Hartnett
    id: d291b0c4-a05c-4ea6-8df1-4b41d5f39eff
    imageUrl:
https://fabrikamprime.visualstudio.com/_apis/GraphProfile/MemberAvatars/aad.NDEwY2FkMDQtOWQyOS03NDFlLTk2MmEt
NGZlYmU2NGE1NTM4
    uniqueName: fabrikamfiber4@hotmail.com
    url: https://spsprodeus27.vssps.visualstudio.com/A5d5b8da6-3db7-4829-baf9-
1e500c21cc12/_apis/Identities/d291b0c4-a05c-4ea6-8df1-4b41d5f39eff
  System.CreatedDate: '2021-11-23T22:26:28.27Z'
  System.Description: <div>This user story is for documentation purposes.  </div>
  System.IterationPath: Fabrikam Fiber\Release 2\Sprint 1
  System.Reason: New
  System.State: New
  System.TeamProject: Fabrikam Fiber
  System.Title: Test the Request feedback functionality
  System.WorkItemType: User Story
  WEF_10182DA5BCCD4CE2A43629FFBD290EF2_Kanban.Column: Backlog
id: 148
relations:
- attributes:
    isLocked: false
    name: Child
  rel: System.LinkTypes.Hierarchy-Forward
  url: https://fabrikamprime.visualstudio.com/854a3f67-9962-43d1-a968-2e5f2eb66c99/_apis/wit/workItems/152
- attributes:
    isLocked: false
    name: Child
  rel: System.LinkTypes.Hierarchy-Forward
  url: https://fabrikamprime.visualstudio.com/854a3f67-9962-43d1-a968-2e5f2eb66c99/_apis/wit/workItems/153
- attributes:
    isLocked: false
    name: Child
  rel: System.LinkTypes.Hierarchy-Forward
  url: https://fabrikamprime.visualstudio.com/854a3f67-9962-43d1-a968-2e5f2eb66c99/_apis/wit/workItems/151
- attributes:
    isLocked: false
    name: Child
  rel: System.LinkTypes.Hierarchy-Forward
  url: https://fabrikamprime.visualstudio.com/854a3f67-9962-43d1-a968-2e5f2eb66c99/_apis/wit/workItems/149
rev: 5
url: https://fabrikamprime.visualstudio.com/854a3f67-9962-43d1-a968-2e5f2eb66c99/_apis/wit/workItems/148
```


Verify your team backlog

After you move work items from one team to another team, check your team backlog to ensure the items

appear. See Create your backlog.

If you encounter any problems, review Set up your project's backlogs and boards.

Related articles

- Create or add a team
- About teams and Agile tools

Modify work items in bulk in Azure Boards

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use bulk modify when you need to quickly make the same change to many work items. For example, you might want to change the priority of several bugs or reassign several tasks to the same team member. Use the web portal to quickly modify one or more fields for work items that will contain the same value.

> **TIP**
> To add work items in bulk or update multiple fields with different values, use Excel. You can't complete a bulk add of work items through the web portal.

With bulk modify, you may edit fields and add or remove tags. You can also reassign work or move work to a specific sprint. You can also use bulk modify to change the work item type or move work items to other projects. The options available to you depend on the platform you work from and the permissions you've been granted.

In this article you'll learn:

- How to multi-select work items from a list and open the context menu
- Edit one or more fields of several work items
- Assign work from a backlog to a sprint using drag-and-drop
- Add or remove tags from several work items

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.

- All project members, even those who belong to the **Readers** group, can email work items.

Supported tasks

All of the following actions can be completed by team members that belong to the Contributors group. Members provided with Stakeholder access can run multi-select, bulk edit, change type, email, and copy as HTML/copy to clipboard actions. For details, see Work as a stakeholder.

Area

Task

Multi-select work items

- Multi-select-query results
- Multi-select-backlog

Link work items

- Link to a new item
- Link to an existing item
- New branch[1]

Bulk edit/update/delete

- Edit field(s)
- Assign to
- Move to iteration
- Change position
- Change parent
- Add/remove tags
- Update from template[1]
- Delete [1]

Copy, clone, change type, move, or email work items

- Clone or copy a single item [2]
- Copy as HTML/Copy to clipboard
- Email selected item(s)
- Change work item type[1]
- Move items to another project[1, 3]

> **NOTE**
>
> 1. You can't perform certain functions on work items whose WITs belong to the Hidden Types Category. This includes all work items that track tests—such as test cases, shared steps, and shared parameters—code review requests and responses, and feedback requests and responses.
> 2. You can choose to copy or clone a single work item from a query results list or from the Actions menu of the work item form. You can only perform a clone or copy action for a single work item. Choose Copy work item when you want to create a copy of a work item and change its work item type. Choose Clone when you want to create another instance of the work item without changes to its work item type.
> 3. You must be a member of the Project Administrators group or be granted explicit permissions to **Move work items**.

Area

Task

Multi-select work items

- Multi-select-query results
- Multi-select-backlog

Link work items

- Link to a new item
- Link to an existing item
- New branch[1]

Bulk edit/update/delete

- Edit field(s)
- Assign to
- Move to iteration
- Change position
- Change parent
- Add/remove tags
- Update from template[1]
- Delete [1]

Copy, clone, or email work items

- Clone or copy a single item [2]
- Copy as HTML/Copy to clipboard
- Email selected item(s)

NOTE

1. You can't perform certain functions on work items whose WITs belong to the Hidden Types Category. This includes all work items that track tests—such as test cases, shared steps, and shared parameters—code review requests and responses, and feedback requests and responses.
2. You can choose to copy or clone a single work item from a query results list or from the Actions menu of the work item form. You can only perform a clone or copy action for a single work item. Choose Copy work item when you want to create a copy of a work item and change its work item type. Choose Clone when you want to create another instance of the work item without changes to its work item type.
3. For on-premises Azure DevOps, you must have an SMTP server configured for your deployment.

Edit multiple work items in bulk

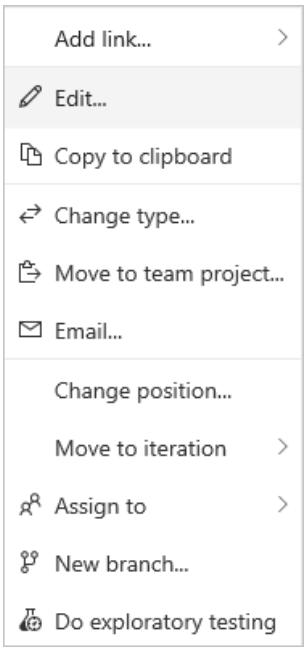
To start a bulk edit, begin by multi-selecting the work items you want to modify, either from the query results or the backlog. You can craft your query using the query editor or search box.

Multi-select of work items on the backlog and sprint backlogs works in the same way as multi-select works within query results.

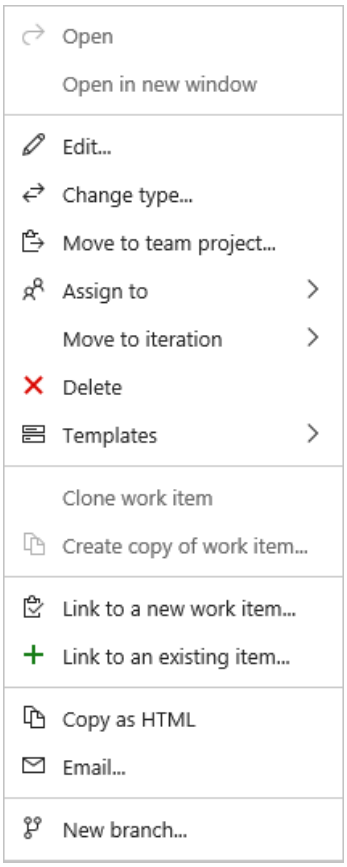
You can use bulk modify by selecting work items from the backlog page or query results list. From the backlog page context menu, change the backlog priority of several items (Change position or Move to position). Then, assign them to a team member, move them to a different sprint, or map them to a feature.

The menu options available to you change depending on the platform you work from and whether you work from a backlog page or query results list.

Backlog menu



Query results multi-select menu

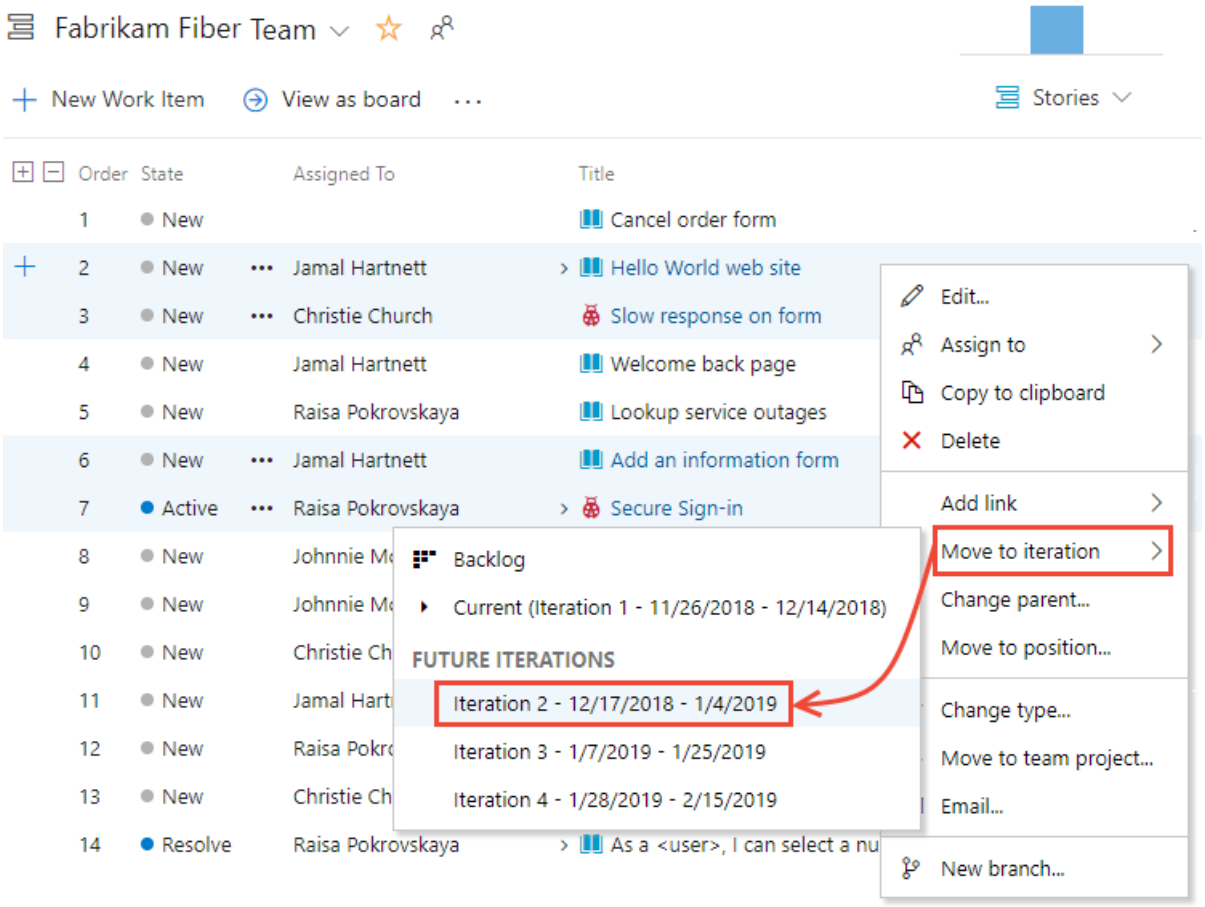


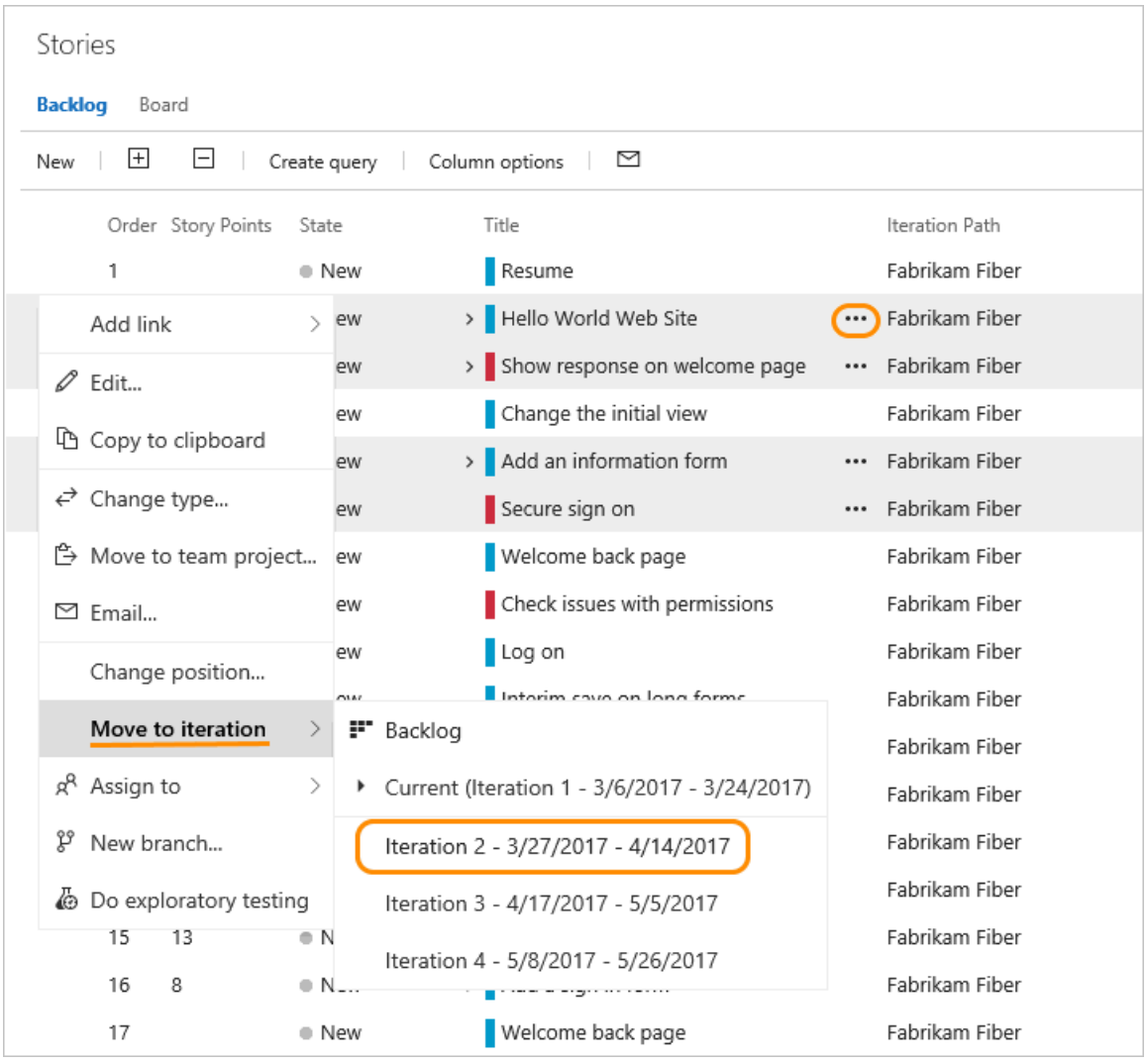
Select multiple items and open the context menu

To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the Ctrl key. Then, you can either drag the selected items to a new position within the backlog, to a different sprint.

To open the context menu, select (•••) or (▾), and then choose the option from the menu.

Here, we use the context menu to move several non-sequential items to the current sprint.


Fabrikam Fiber Team
New Work Item
View as board
Stories
Order
State
Assigned To
Title
1 New Cancel order form
2 New Jamal Hartnett Hello World web site
3 New Christie Church Slow response on form
4 New Jamal Hartnett Welcome back page
5 New Raisa Pokrovskaya Lookup service outages
6 New Jamal Hartnett Add an information form
7 Active Raisa Pokrovskaya Secure Sign-in
8 New
9 New
10 New
11 New
12 New
13 New
14 Resolve Raisa Pokrovskaya As a <user>, I can select a nu
Edit...
Assign to
Copy to clipboard
Delete
Add link
Move to iteration
Change parent...
Move to position...
Change type...
Move to team project...
Email...
New branch...
Backlog
Current (Iteration 1 - 11/26/2018 - 12/14/2018)
FUTURE ITERATIONS
Iteration 2 - 12/17/2018 - 1/4/2019
Iteration 3 - 1/7/2019 - 1/25/2019
Iteration 4 - 1/28/2019 - 2/15/2019

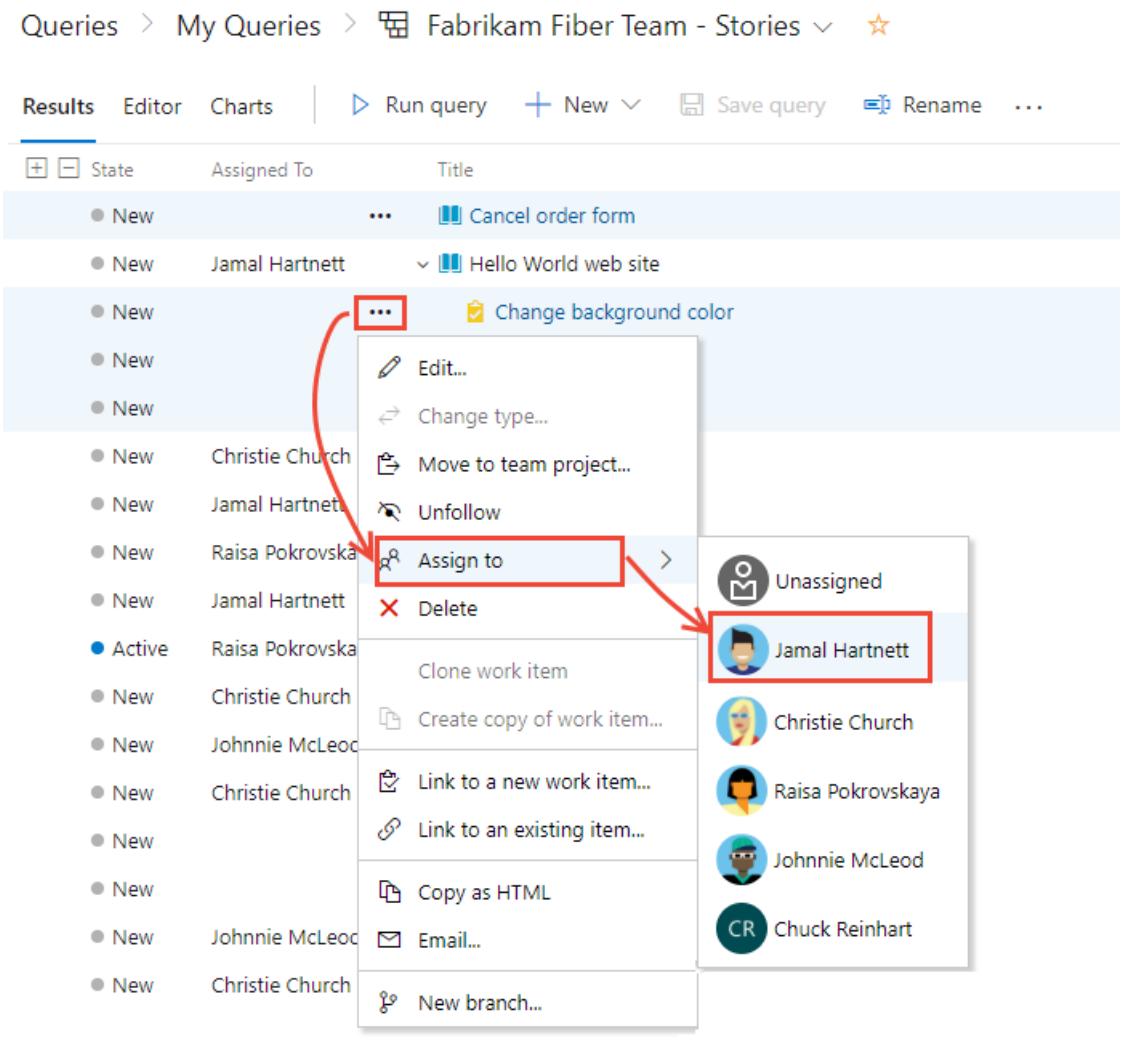

Stories
Backlog
Board
New
Create query
Column options
Order
Story Points
State
Title
Iteration Path
1 New Resume Fabrikam Fiber
Hello World Web Site Fabrikam Fiber
Show response on welcome page Fabrikam Fiber
Change the initial view Fabrikam Fiber
Add an information form Fabrikam Fiber
Secure sign on Fabrikam Fiber
Welcome back page Fabrikam Fiber
Check issues with permissions Fabrikam Fiber
Log on Fabrikam Fiber
Add link
Edit...
Copy to clipboard
Change type...
Move to team project...
Email...
Change position...
Move to iteration
Assign to
New branch...
Do exploratory testing
Backlog
Current (Iteration 1 - 3/6/2017 - 3/24/2017)
Iteration 2 - 3/27/2017 - 4/14/2017
Iteration 3 - 4/17/2017 - 5/5/2017
Iteration 4 - 5/8/2017 - 5/26/2017
15 13
16 8
17 New Welcome back page Fabrikam Fiber

TIP

Use the backlog **Create Query** feature to create a query with the backlog items. You can then open the query within the web portal or Excel to perform additional bulk updates.

Reassign work items

With work items selected, open the context menu for any selected item, and reassign all of them. By doing reassigning them, you can quickly assign them to a member of your team or to another sprint or iteration.

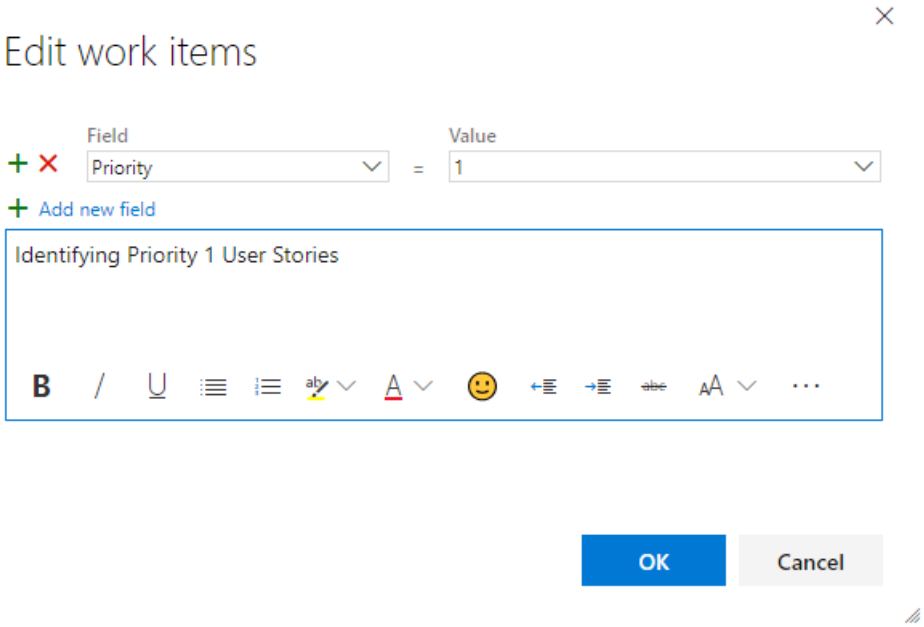


To learn more about the *Assign To* and *Iteration Path* fields, see Query by assignment, workflow, or Kanban board changes and Query by area or iteration path.

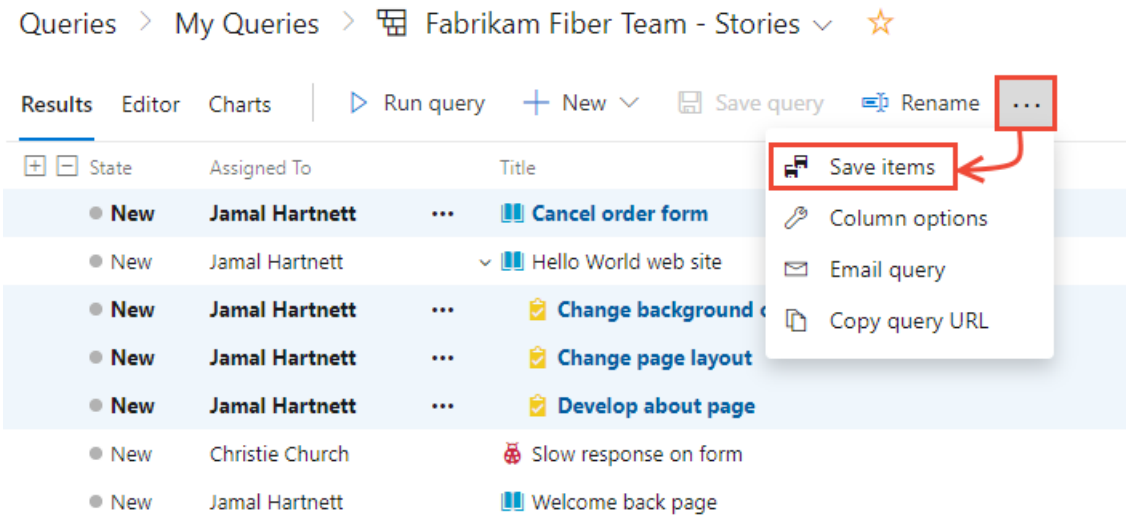
Edit one or more fields

To assign or modify several fields, choose Edit from the context menu of one of the selected work items. Enter a value for each field that you want to update.

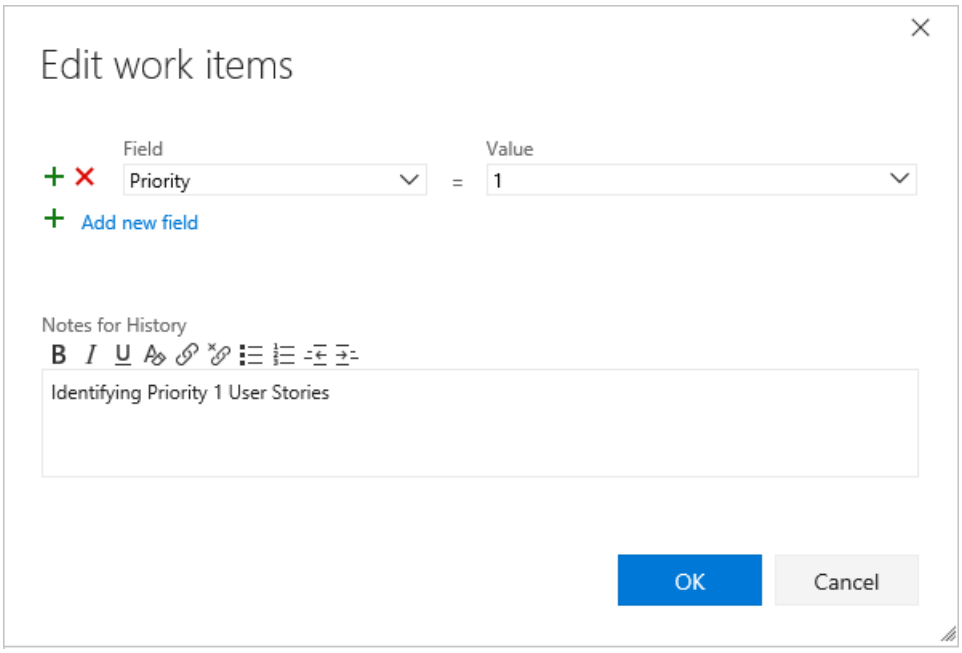
1. For audit purposes, you can type a description for your bulk update task. To learn more about each field, see the Work item field index.



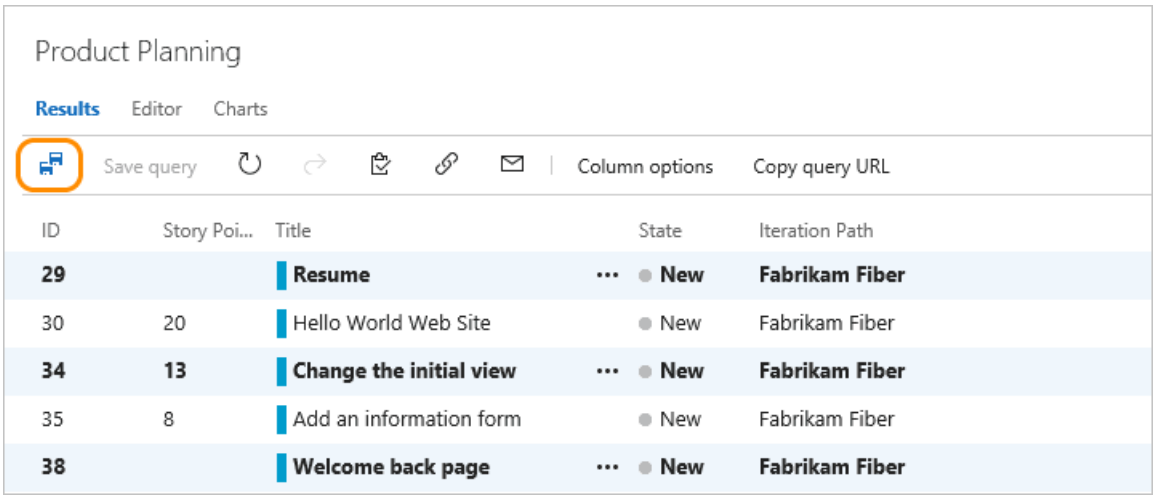
2. From the Query results page, you must save all work items that you bulk-modified. When you bulk modify items from the backlog, they're automatically saved. Work items shown in bold text indicate that local changes haven't yet been saved to the data store. The **Save items** button may be in a different place in the UI than shown in the picture below, depending on the layout of your browser and the specific version in use.



1. For audit purposes, you can type a description for your bulk update task. To learn more about each field, see the Work item field index.



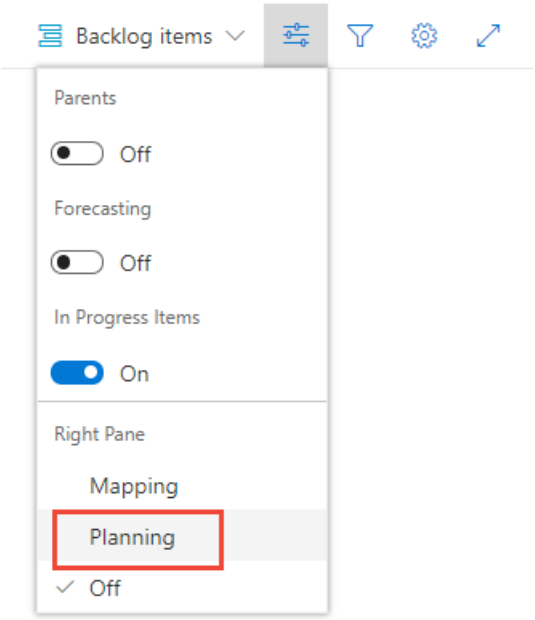
2. From the Query results page, you must save all work items that you bulk-modified. When you bulk modify items from the backlog, they're automatically saved. Work items shown in bold text indicate that local changes haven't yet been saved to the data store.



Move work items to a sprint

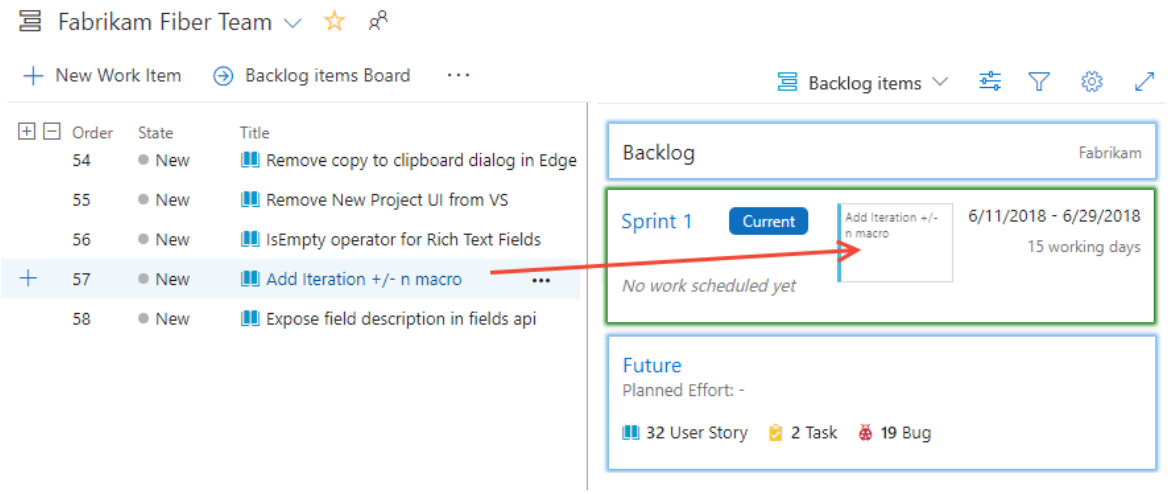
From any product, sprint, or portfolio backlog, you can drag a multi-selected list of work items and drop it onto a sprint in the **Planning** pane to change it's iteration path. (Not supported for users with **Stakeholder** access.)

1. To open the **Planning** pane, choose the view options icon and select **Planning**. You can choose to set **In Progress items** to On or Off.



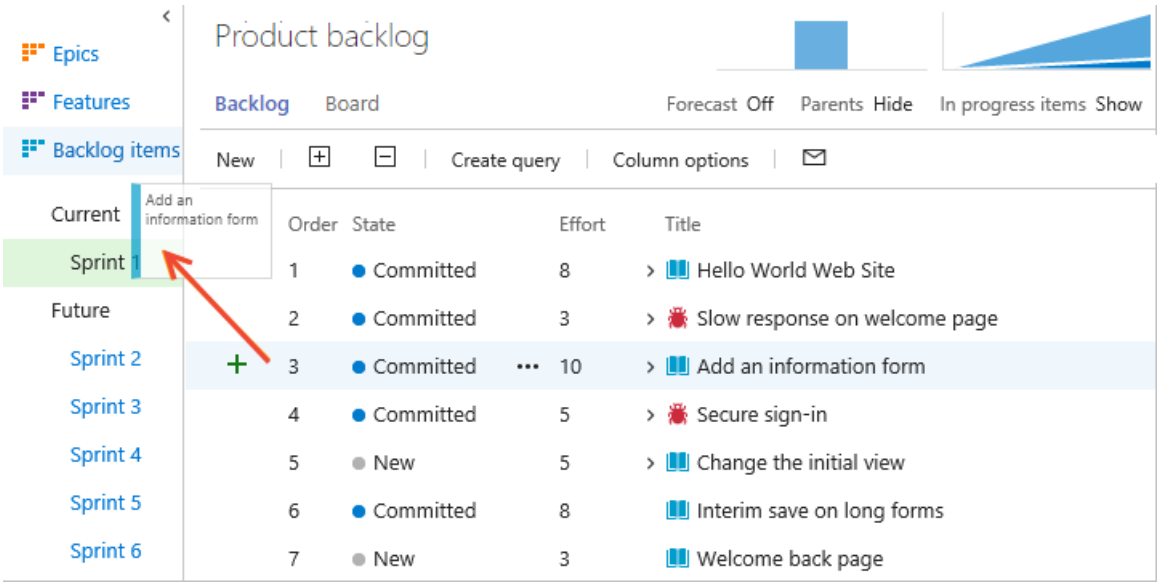
The set of sprints selected for your team appears. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. You can drag and drop items from the **Backlog** onto a sprint.



This action will update the Iteration Path of the backlog items and any of its child tasks to the sprint you selected.

From any backlog or board, you can drag a multi-selected list of work items and drop it onto a sprint to change it's iteration path. From a Kanban or taskboard, you can drag a single work item onto a sprint. (Not supported for users with **Stakeholder** access.)



Child items of the work items whose iteration path you change are also updated with the new iteration path.

Modify rich-text fields in bulk

Rich-text fields support entry of HTML syntax tags to support formatting. Rich-text fields correspond to the **Description**, **Acceptance Criteria**, **Repos Steps**, and others listed in Query samples for select fields.

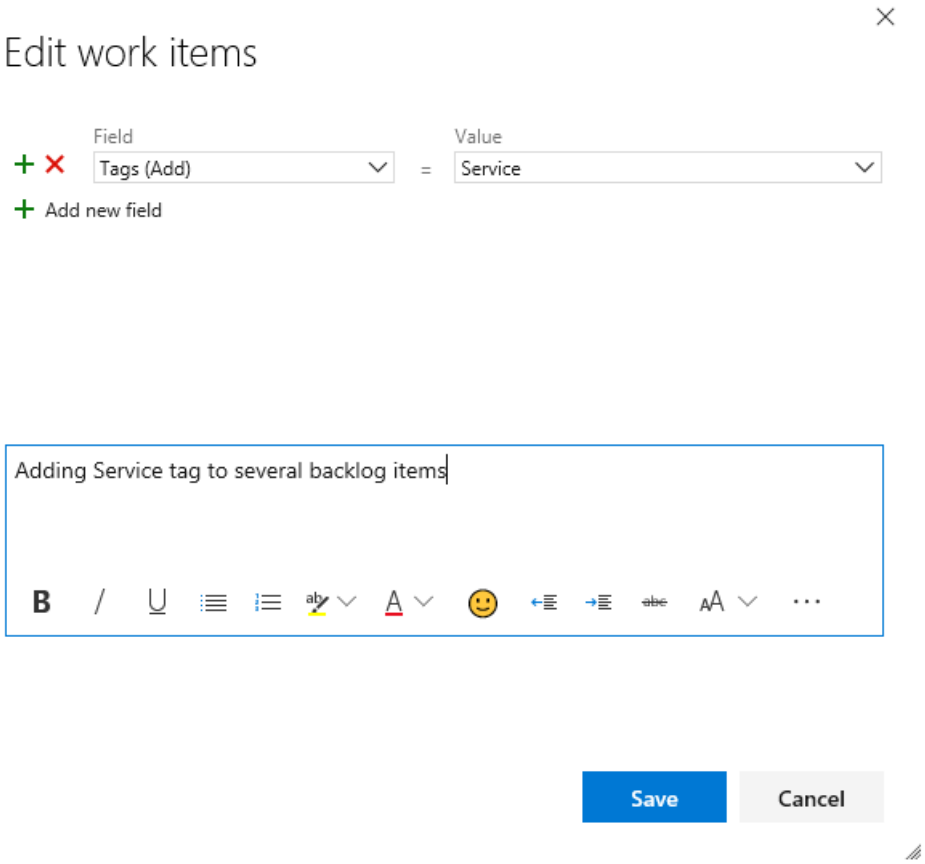
You can bulk update a rich-text field by using the bulk modify tool, selecting the field, and entering the text with syntax in the **Value** field. Or, you can create a work item template with the text you want to use and complete a bulk update by applying the template to the selected work items. For details on using work item templates, see Use templates to add and update work items

For a worked example using templates showing entry of HTML formatted syntax, see Sample work item templates, Add guidance in a rich-text field.

Modify tags in bulk

From the Edit work items dialog, select **Tags (Add)** or **Tags (Remove)**.

Here we choose to add the *Service* tag to the selected work items.



Related articles

To add fields or customize a work item form, see Customize your work tracking experience. The method you use depends on the process model that supports your project.

Migrate or change a large number of work items

For large scale, organizational moves, use the REST API calls for Work item batch operations.

At this time, you can't move work items to a different organization or collection. You can only migrate work item information by exporting and then importing them using Excel.

Add multiple values to a field

If you have implemented a custom control that supports multiple values, you can use Excel to bulk edit the field, but you can't modify it using the web portal. In the web portal, you can only select a single value for the field.

Add or modify Azure Boards work items in bulk with Microsoft Excel

10/4/2022 • 28 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you need to add or modify many work items, using Microsoft Excel can save you time. Excel supports adding work items, updating existing work items, adding links and attachments to multiple work items, and more. You can also use native Excel features to support other actions, such as summing a column, copy-and-paste rows, fill down data into cells, and more.

In this article you'll learn how to complete the following tasks:

- Choose the type of list or query to support your task
- Use select Excel features when connected to Azure Boards
- Import or update work items, either a flat list or hierarchy tree list
- Publish and refresh your work items
- Convert a flat-list to a tree-list, change your list type or query
- Add work items to your worksheet
- Add and remove work item fields from your worksheet
- Select user accounts for identity or person-named fields
- Link work items, find work items to link to, edit links, and more
- Add attachments to one or more work items
- Open a work item from Excel to add additional information (opens in web portal)
- Edit Area and Iteration Paths (opens in web portal)

For information about connecting to Excel, see Connect Azure Boards to an Office client. For answers to specific questions about the integration of Excel and Azure DevOps, see FAQs: Work in Excel connected to Azure Boards .

> **NOTE**
> If you don't have access to Excel, you can still perform bulk import and update using CSV formatted files. To learn more, see Bulk import or update work items using CSV files.

Prerequisites

> **NOTE**
> macOS isn't supported. Even if you've installed Visual Studio for Mac, connection to Azure DevOps from Excel isn't supported.

- Installed Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Installed Azure DevOps Office Integration 2019 (free).

> **NOTE**
> The only way to get the Azure DevOps Office Integration plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Office Integration installer. The plug-in supports connection to Azure Boards and Azure DevOps Server from Excel.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To view or modify work items, you must have these permissions set to **Allow**: **View work items in this node** and **Edit work items in this node**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To use the Select User feature, you need to install Visual Studio (at least VS 2015.1 or later version or Team Foundation Server Office Integration 2015 Update 2 or later version. You can download the free version of Visual Studio Community. Get this feature to avoid data validation errors by misspelling user names and when you must assign user names from a large group of user accounts.

- Installed Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Installed Azure DevOps Office Integration 2019 (free).

> **NOTE**
> The only way to get the plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Standalone Office Integration installer. The Azure DevOps Office Integration 2019 plug-in supports connection to Azure Boards and Azure DevOps from Excel, Project, and the PowerPoint-based storyboarding tool.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To view or modify work items, you must have these permissions set to **Allow**: **View work items in this node** and **Edit work items in this node**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To use the Select User feature, install Visual Studio (at least VS 2015.1 or later version or Team Foundation Server Office Integration 2015 Update 2 or later version. You can download the free version of Visual Studio Community. Get this feature to avoid data validation errors by misspelling user names and when you must assign user names from a large group of user accounts.

To learn more about compatibility requirements, see Compatibility with Azure DevOps Server.

Choose list and query type

When you work in Excel connected to Azure Boards, you're always working with a query type and a list type. Queries correspond to the queries you create using the Query Editor.

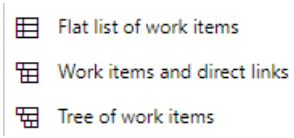
- **Query types**:
  - **None**: Indicates it's an **input list**.
  - **Query title**: Indicates the list of work items is tied to the specified **query**.
- **List types**:
  - **Flat list**: Simple list of work items that shows a single **Title** column. No link management is supported.

- **Tree list**: Hierarchical list of work items that support creating and updating tree topology links, such as Parent-Child links, between work items. These lists include two or more **Title** columns.

You can add, modify, publish, and refresh work items using any query type and list type.

Query types

Azure Boards supports three query types. The icon next to each query indicates the query type. The first two query types, **Flat list of work items** and **Work items and direct links** are imported as flat list queries.

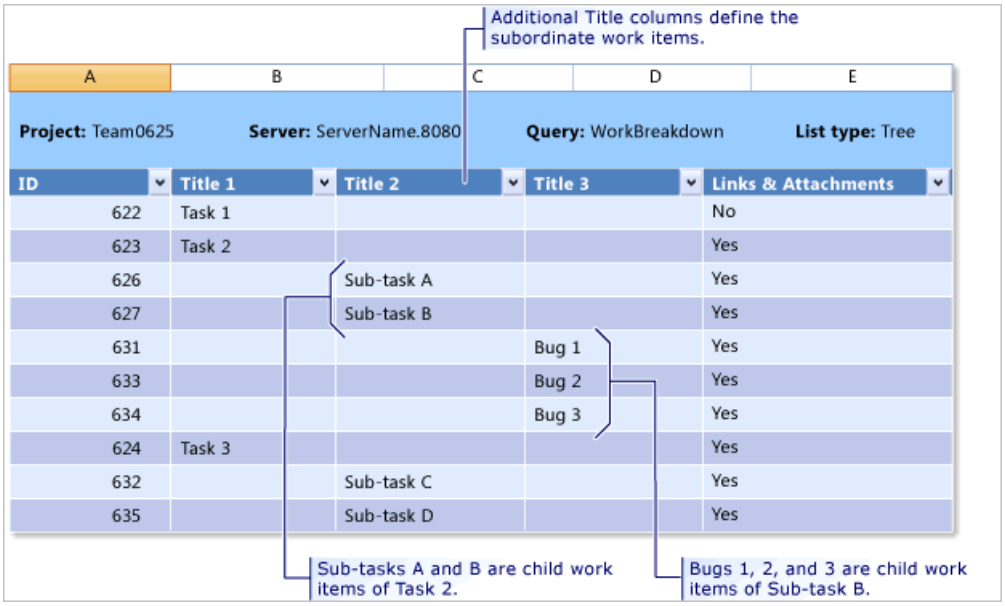


Only the **Tree of work items** queries import as a tree list. Direct links queries are imported as a flat list into Excel as modifying multiple types of links aren't a supported feature in Excel.

Tree lists

You can bulk add a nested list of work items, such as a work breakdown structure or a hierarchical set of user stories and customer experiences. For example, you can add a nested list of tasks, subtasks, and bugs, as shown in the following illustration, or linked tasks to product backlog items.

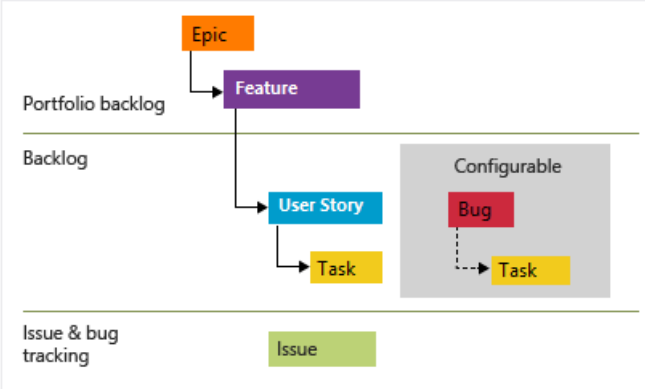
Here's how a three-level nested tree of items appears in Excel.



Parent-child links or other tree topology link types support creating a hierarchical backlog structure. The work item types that participate in the hierarchy differ with different processes and are shown in the following images.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

To import a hierarchical list, see Add or import a hierarchical list of work items later in this article.

My queries versus shared queries

You can open in Excel any query you've defined in Azure Boards. That includes queries defined under My Queries and Shared Queries. However, if you plan to share the workbook with other team members, then you should only work with a Shared Query. Other team members can't use the workbook or worksheet if it's based on a personal query stored under your My Queries folder.

Use list and query types

In general, you Use a flat list to bulk add or modify several types of work items at once, such as backlog items, tasks, bugs, or issues. Use a tree list to bulk add or modify work items and their tree-topology links.

Here's some more guidance:

- Use an **input list, flat list**: To import a list of work items or create new work items, no hierarchy
- Use an **input list, tree list**: To complete top down planning and import hierarchically linked work items
- Use a **query list, tree list**: To view and modify the hierarchy of link relationships of many existing work items.
- Use a **query list, flat list**: To bulk update a list of work items or create new work items, no hierarchy

- Use an input list, flat list: To import a list of work items or create new work items, no hierarchy
- Use an input list, tree list: To complete top down planning and publish parent-child linked work items
- Use a query list, flat list: To create an Excel report based on the query of work items

> **NOTE**
> To create an Excel report, you're project collection must be configured to support Analytics reporting. For more information, see Create Excel reports from a work item query.

- Use a query list, tree list: To view and modify the hierarchy and parent-child link relationships of many existing work items.

Use Excel features

You can use most Excel features when working with a list of work items. For example, you can use the following features:

- Format a cell or apply conditional formatting to a cell or column
- Cut and paste from one cell to other cells
- Cut and paste a single row

- Sum a column or add other formulas
- Fill down cells
- Filter
- Add multiple worksheets to your workbook

Each worksheet in Excel can contain a different input list or query. However, all worksheets within the workbook must connect to the same project within an organization or project collection.

The following features work slightly differently when working with a worksheet connected to Azure Boards.

- Each cell or column of cells corresponds to a work item field. Each field is associated with a data type. Excel won't allow you to enter data into a cell that doesn't meet the data type and requirements for that field.
- You can only insert a single row at a time within the worksheet.
- You can copy and paste multiple rows within the worksheet.
- To move a work item within a hierarchy, cut the entire row and paste it under the work item you want as its parent.
- Use **Outdent** and **Indent** to change the location of a work item within the tree.
- Undo (Ctrl Z) may not work. If you do something that you want to revert, you can refresh the worksheet.

We recommend you publish and refresh your worksheet often to make sure your local work remains in sync with Azure Boards data store.

To learn more about Excel, see Basic Excel tasks .

Sort work items

You can sort work item flat lists using the Excel sort feature.

However, if you're working from a tree list, you don't want to do any type of sort. Doing so changes the tree structure and as such, the links between work items.

If you want to use Excel to manage the order of your work items as they appear in a team backlog, you can do that by using the Stack Rank or Backlog Priority field (Agile or Scrum process). You can set values in these fields, publish your worksheet, and refresh your backlog. Your backlog items should appear reordered based on lowest to highest number. However, the next time the backlog is reordered from the backlog, the values you entered are subject to change.

If you want to maintain a certain order of work items, consider adding a custom field to manage the sort order, and then use that within Excel to sort your flat list of work items. This option, however, won't change the order that appears in your backlog.

Tasks you can and can't do with Excel

You can do the following tasks:

- Add tags and bulk update work items with tags as described in Add work item tags to categorize and filter lists and boards. Add the **Tags** field to your worksheet. Add multiple tags separated by a semicolon (;).
- You can add simple text to a rich-text field, but if you're updating several work items in bulk, you may lose formatting in existing work items.
- You can work offline and then reconnect and publish your changes. For details, see Connect Azure Boards to an Office client, Work offline, and reconnect.

You can't do the following tasks from an Excel worksheet:

- You can't delete work items
- You can't change the work item type of an existing work item
- You can't move work items to another project

- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread
- You can't link to a remote work item.

- You can't delete work items
- You can't change the work item type of an existing work item
- You can't move work items to another project
- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread.

- You can't delete work items
- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread

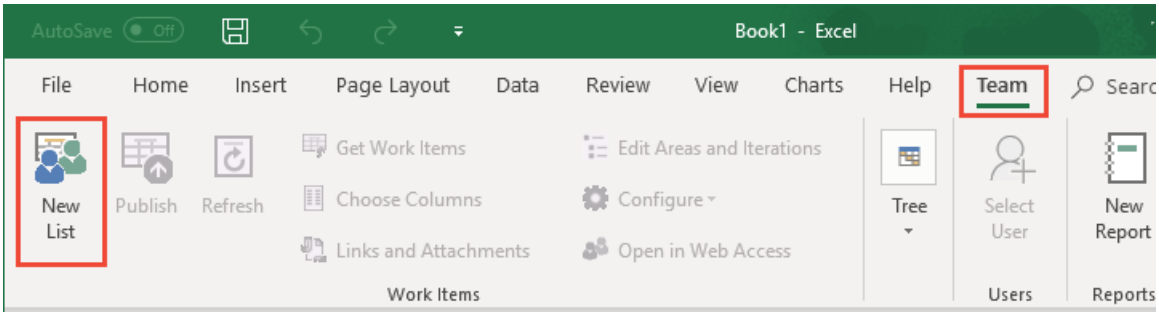
Import work items as a flat list

1. Open Excel and connect to your Azure Boards project. Use one of the four methods provided in Connect Azure DevOps project to Excel.

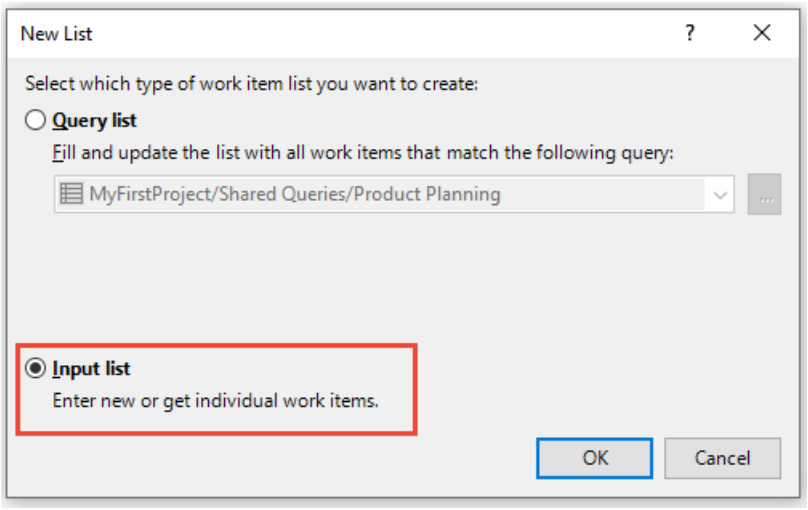
   > **NOTE**
   > When you connect to Azure Boards in the cloud, the **Team Project Collection** is automatically selected as there is only one collection associated with your Azure DevOps Services organization. When you connect to Azure Boards in an on-premises server, you choose the **Team Project Collection** prior to choosing the project.

2. In Excel, start with a blank worksheet. If you don't see the **Team** ribbon (or the **Team** menu if you use Excel 2007), see Azure DevOps Office integration issues.

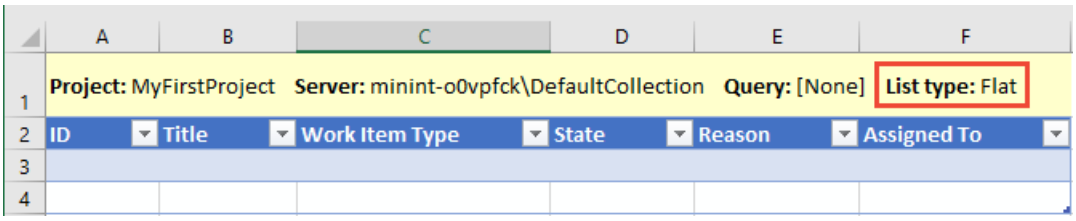
3. Choose **New List** from the Team ribbon.



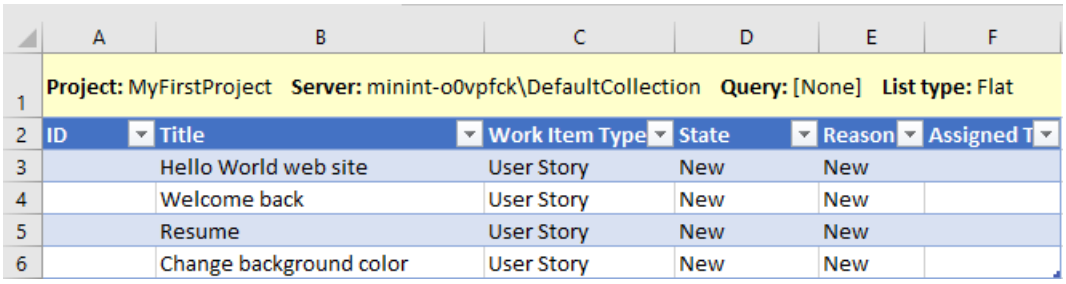
4. From the **New List** dialog, choose **Input list**.



5. Your worksheet is now bound to your project as an input list (Query[None]), flat list.

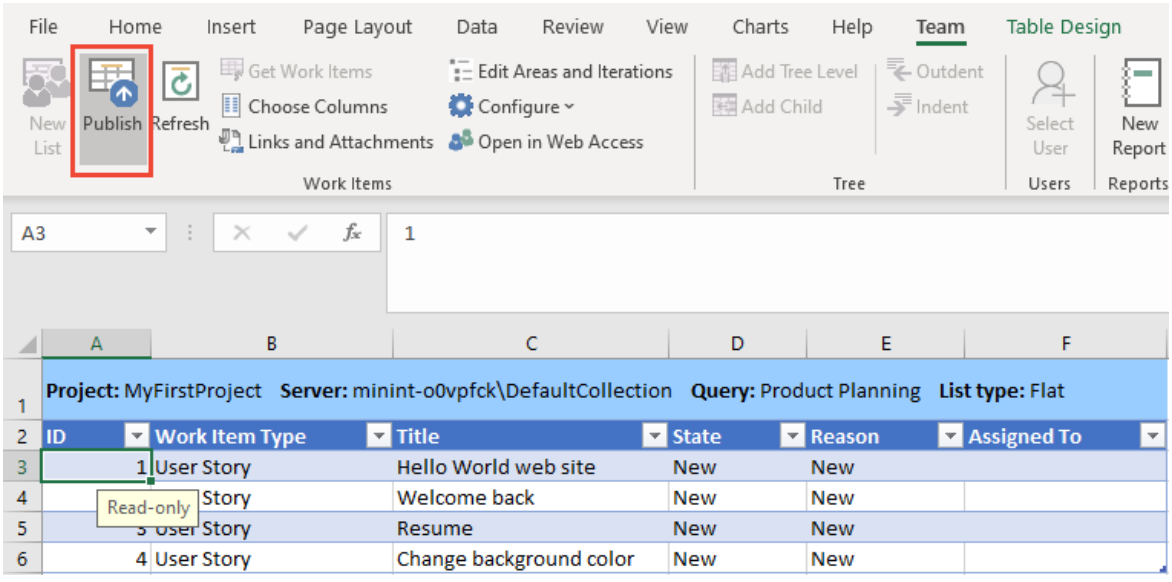


6. Specify the titles of the work items you want to add and their work item type.



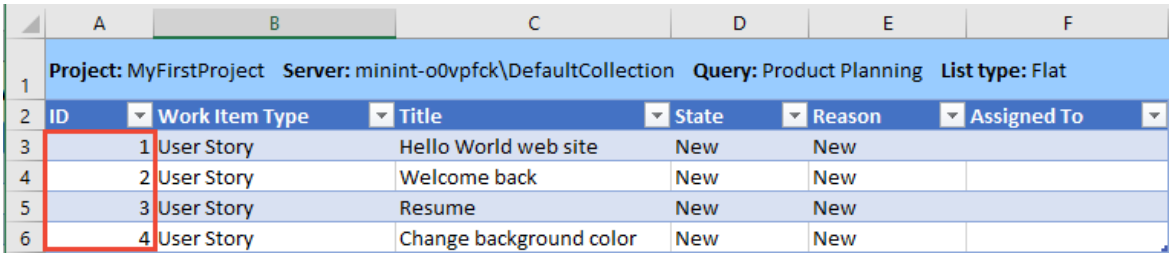
Notice how the **State** and **Reason** fields automatically fill in with default values once your select the work item type.

7. Publish your worksheet.



Make sure your cursor is in a cell that contains data. Otherwise, the **Publish** button might appear disabled.

Notice how IDs are now assigned to your work items.

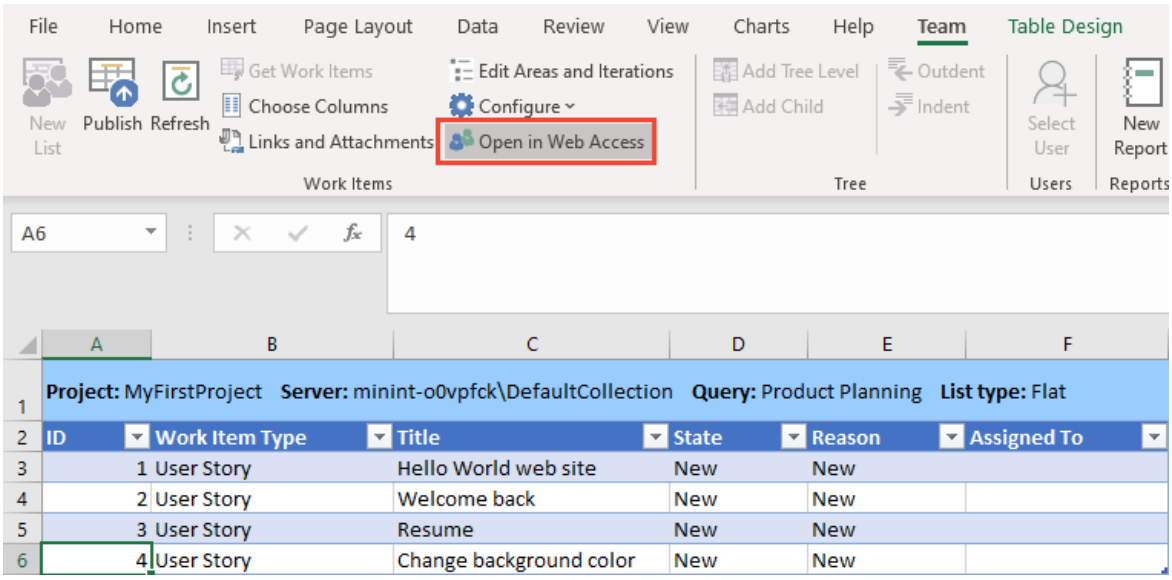


	A	B	C	D	E	F
1	**Project:** MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** Product Planning **List type:** Flat					
2	**ID**	**Work Item Type**	**Title**	**State**	**Reason**	**Assigned To**
3	1	User Story	Hello World web site	New	New	
4	2	User Story	Welcome back	New	New	
5	3	User Story	Resume	New	New	
6	4	User Story	Change background color	New	New	

8. To assign values to other fields, open **Choose Columns**, add the fields, make the assignments, and publish your changes.

> **TIP**
>
> If you're adding work items that you want to appear on a team backlog, make sure that you add and specify the team's Area Path and Iteration Path. If you need to add Area Paths or Iteration Paths, choose the **Edit Areas and Iterations** link. The link opens a web browser to the Project Settings page. To learn more, see Define area paths and assign to a team and Define Iteration Paths and configure team iterations.

9. To open a work item to add more information, Choose the work item you want to open and then choose **Open in Web Access**. Before you do, make sure you publish any changes you've made.



	A	B	C	D	E	F
1	**Project:** MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** Product Planning **List type:** Flat					
2	**ID**	**Work Item Type**	**Title**	**State**	**Reason**	**Assigned To**
3	1	User Story	Hello World web site	New	New	
4	2	User Story	Welcome back	New	New	
5	3	User Story	Resume	New	New	
6	4	User Story	Change background color	New	New	

A web browser opens and displays the work item.

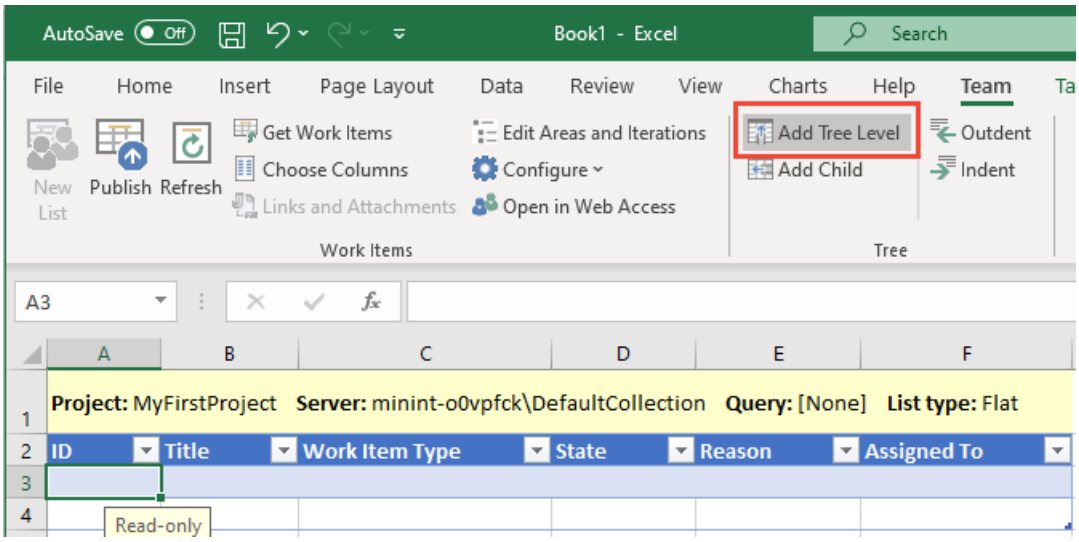
If you make changes to the work item, you should then immediately refresh your worksheet to capture the changes.

Import work items as a tree list

You can add a hierarchy of work items linked using parent-child links or other tree topology link type.

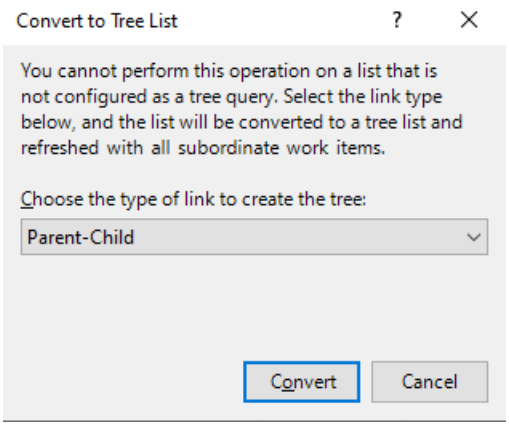
> **IMPORTANT**
>
> Don't sort a tree list. Sorting a tree list can change the hierarchical link relationships.

1. Starting from Step 5 from the previous procedure, convert your flat list, input list into a tree list. Choose a cell within the flat list and then choose **Add Tree Level**.

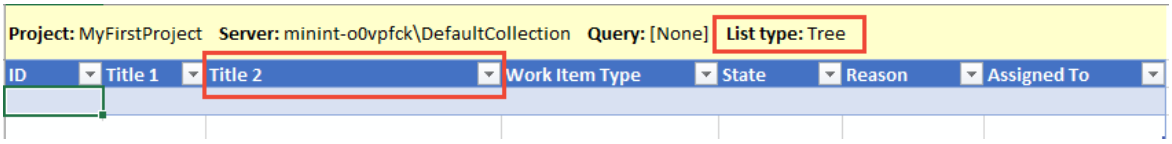


If the **Add Tree Level** is disabled, you're working from a query list. To convert your list to a tree list, you must first reconfigure your list to an input list.

2. Choose the link type to use when adding work items to a hierarchy, and then choose **Convert**. The most usual choice is **Parent-Child**. You can only select from tree topology link types. To learn more, see Link type topologies and restrictions.



Note the List type has changed to **Tree**, and a second **Title** column appears.



3. To add more levels to the hierarchy, choose **Add Tree Level** again. For example, if you want to add a hierarchy of Epics, Features, and User Stories, you'll want to have **Title 1**, **Title 2**, and **Title 3** columns.

   If you want to add tasks, add another tree level to have four title columns. To remove a column, see Remove a tree level.

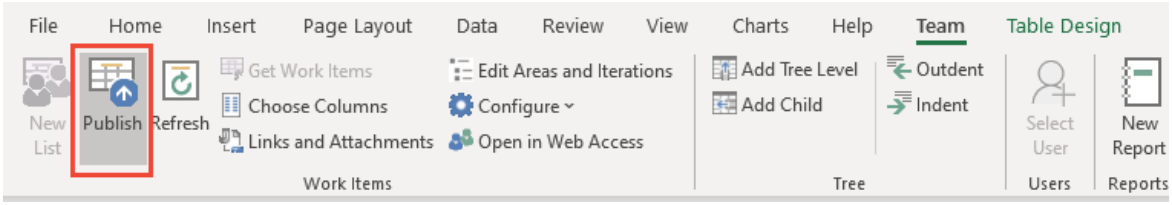
4. Save your Excel file.

5. Enter the **Work Item Type** and **Titles** for the hierarchy you want to import. The **State** fields automatically fill in with default values once you select the work item type.

Project: MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** [None] **List type:** Tree

ID	Work Item Ty	Stat	Title 1	Title 2	Title 3
	Epic	New	Improve User Experience		
	Feature	New		Integrate client application with popular email clients	
	User Story	New			Implement a factory which abstracts the email client
	User Story	New			As a user, I can select a number of support cases and use cases
	User Story	New			Customer request support for GMail
	User Story	New			Customer request support for Yahoo mail
	User Story	New			Customer request support for Roundup
	Feature	New		Integrate client app with IM clients	
	Feature	New		Emoticon feedback enabled in client application	
	User Story	New			As a user, I can select an emoticon and add a short description
	User Story	New			Add animated emoticons
	Epic	New	Framework to port applications to all devices		
	Feature	New		Migrate legacy code to portable frameworks	
	User Story	New			Implement a factory that migrates legacy to portable frameworks
	User Story	New			As a developer, I can analyze a code base to determine compliance with
	Epic	New	Counter the Heartbleed web security bug		
	Feature	New		Scan all software for the Open SLL cryptographic code	
	User Story	New			Scan all code base and identify the affected code
	Feature	New		Update software to resolve the Open SLL cryptographic code	
	User Story	New			Update and retest suite code base affected by the vulnerability
	User Story	New			Update and re-test service code based affected by the vulnerability
	Epic	New	Shared Personalization and state		
	Epic	New	Service interfaces to support REST API		
	Epic	New	Convert legacy Odata service interfactes to REST API		
	Feature	New		Convert all services from using experiemental code	
	Feature	New		Convert all client service calls from using experimental code	

6. Publish your worksheet.

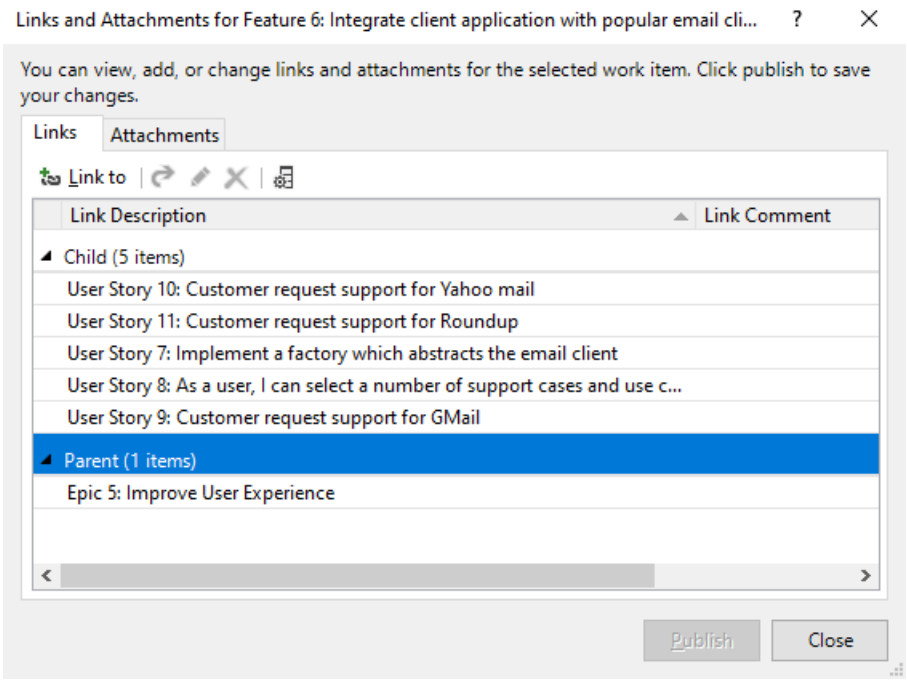


Make sure your cursor is in a cell that contains data. Otherwise, the **Publish** button might appear disabled.

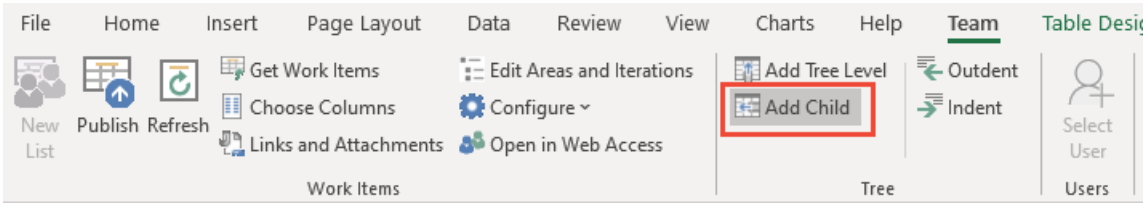
IDs are now assigned to your work items. In the background, the link type you selected is used to link each work item in the hierarchy. Epics are linked to Features, Features are linked to User Stories.

7. To check the links made, choose a work item and choose **Links and Attachments**.

For example, here we show the Child and Parent links created for a Feature that was imported.

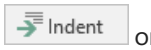


8. To enter a row under a work item where you want to add a child, choose the row and then choose **Add Child**.



9. To assign values to other fields, open Choose Columns, add the fields, make the assignments, and publish your changes.

10. To change the hierarchy, cut and paste the row of a work item to place it under the new parent. Make sure that you select the entire table row. When you publish the change, the old hierarchical links are deleted and the new hierarchical link are created.

    You can use the 

or 

indent/outdent icons to demote or promote a work item within the tree hierarchy. Verify that the column to the left or right of the parent work item's title is a **Title** column. The header at the top of the column should read **Title n**, if it doesn't, add a tree level.

Remove a tree level

1. First, publish changes that you've made to work items before you remove a tree level. Removing a tree level requires a refresh, which overwrites data in the work item list. You'll lose any data you haven't published.

2. Next, delete any content under the tree-level **Title** *number* column you want to remove—the highest numbered column—. This column should be the highest numbered column in the tree.

3. **Refresh** your worksheet. The column containing empty values for the Title is removed.

   You'll receive an error message if you attempt to delete the column.

Useful tips when working with a tree list

- Excel interprets the data in the **Title** columns to determine the pattern of links between work items. When you publish changes, any of the following conditions can result in an error, an invalid link, or a tree link to be created between incorrect work items:
  - A row between the work items is blank within the hierarchy

- The Title of a work item is in the wrong column. Make sure you enter a title for each child work item.
- Within a row, multiple **Title** columns contain data. Enter text in only one of the title columns within each row.
- The list was sorted. Don't sort a tree list. Sorting a tree list can change the hierarchical link relationships. If you do sort a tree list, you can recover from this operation by immediately refreshing.

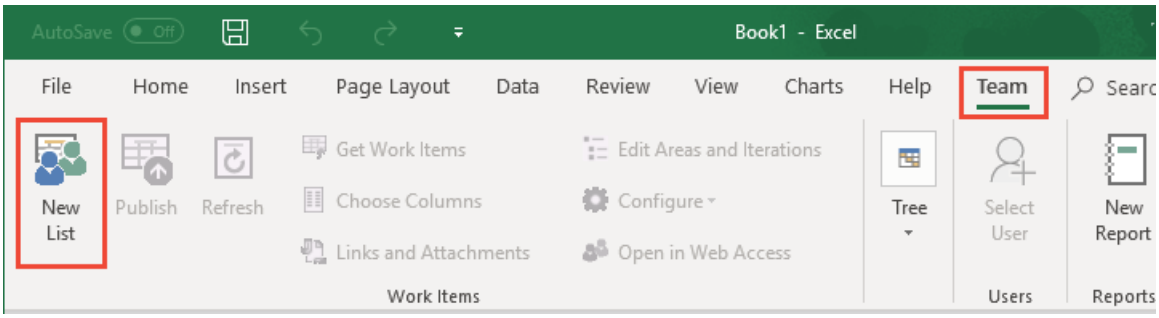
- To resolve an error, see Resolve invalid links.
- A parent-child linked work item can only have one parent. You can't add the same work item task to two backlog items. Instead, you need to define distinct work item tasks.

Update work items in bulk with a query list

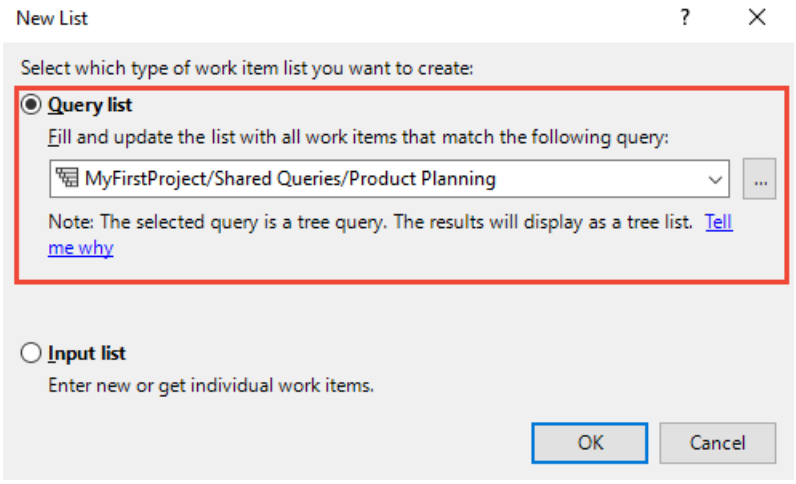
The easiest way to bulk update many work items is to create a query with the work items you want to update, and then open that query in Excel.

> **TIP**
> **Follow these tips to keep your work in sync:**
>
> - When you first open a saved worksheet, use (**Refresh**) to download the latest data from the data store.
> - Enter data for additional fields by adding columns to the worksheet using **Choose Columns**.
> - To avoid data conflicts, publish your additions and modifications often.
> - To prevent loss of data before you publish or refresh, save your workbook periodically.

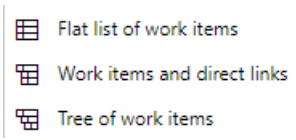
1. From the web portal or Visual Studio, create the work item query that contains the work items you want to update. For details, see Create and save managed queries with the query editor.
2. Open Excel and connect to your Azure Boards project. Use one of the four methods provided in Connect Azure DevOps project to Excel.
3. If you opened your query from the web portal or Visual Studio, you're done. Make any changes you want. Open **Choose Columns**, add fields, make assignments, and publish your changes.
4. If you start from Excel, open a blank worksheet. You can add a worksheet to an existing workbook, as long as you're choosing a query from the same project the workbook is bound to.
5. Choose **New List** from the Team ribbon.



6. From the **New List** dialog, choose **Query list**, and select the query you want from the drop-down menu.



The icon next to each query indicates the query type. The first two query types, **Flat list of work items** and **Work items and direct links** are imported as flat list queries. Only the **Tree of work items** queries import as a tree list.



7. With the work items imported to Excel, make the modifications you want and publish your changes.

   If you're working with a tree list, see also the information provided in Import a hierarchical list of work items.

Enable Tree commands

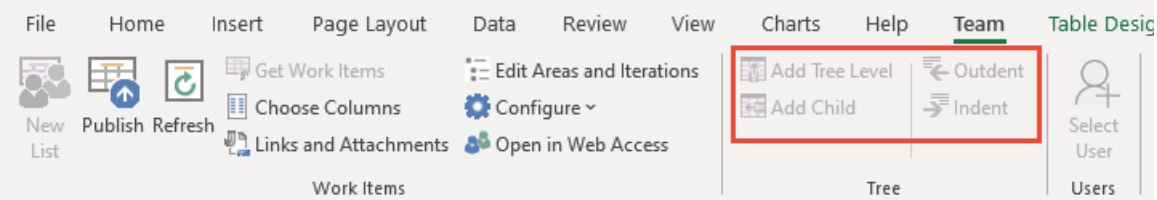
If the **Tree group** commands aren't available, your worksheet is configured as a flat list, query list. Convert the list to either an input list or a list based on a tree query to enable the Tree group commands. To learn how, see the next section on Change your list type or query.

Change your list type or query

You can change the work items listed in your worksheet. Specifically, you can:

- Change your flat list to a tree list
- Change from a query list to an input list
- Change from an input list to a query list
- Change the query your worksheet references

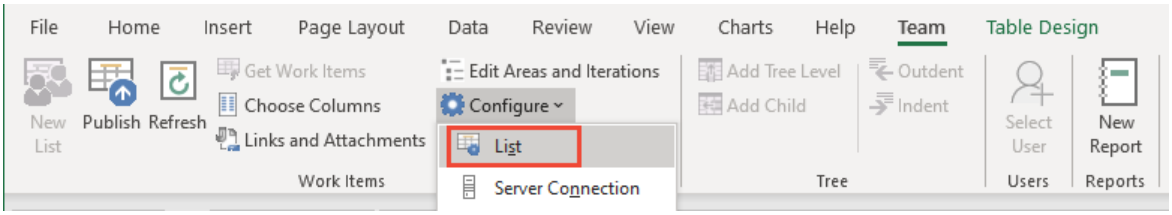
If you want to change your flat list to a tree list, you can. However, if your list is a query list, then you first need to reconfigure the list. You'll know that it's a flat list, query list as the **Tree group** commands are disabled.



To convert your query list to an input list, follow these steps.

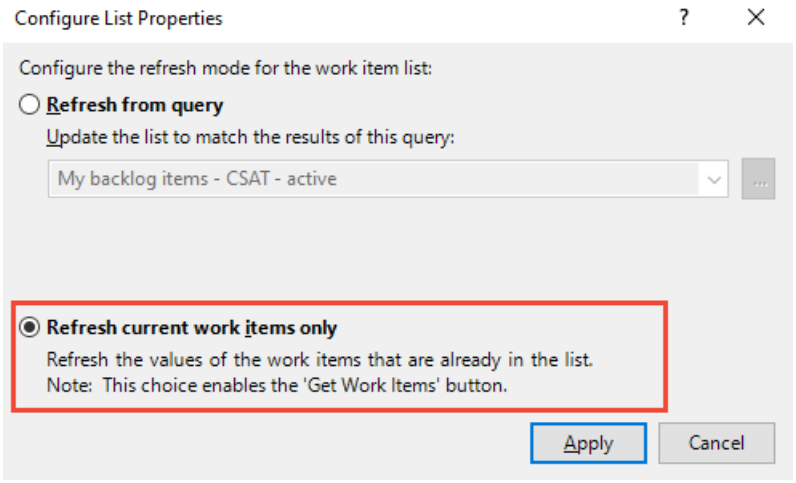
1. First, publish whatever changes you have made.

2. Next, on the **Team** ribbon, choose **Configure**, **List**.

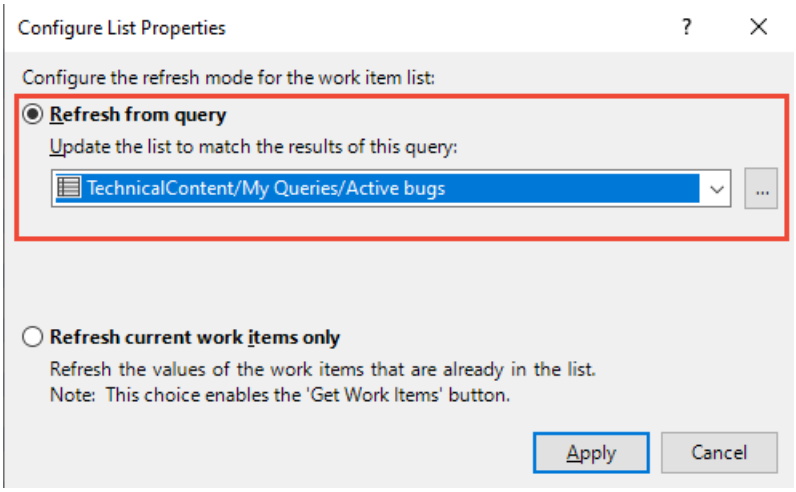


3. Choose **Refresh work items only** and then **Apply**.

   This choice changes the query list to an input list.



4. To convert from an input list to a query list, choose **Refresh from query**, select the query, and then **Apply**.

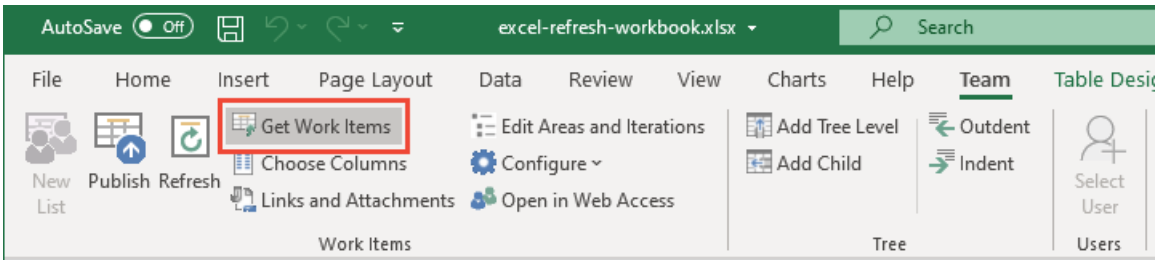


Add existing work items to your worksheet

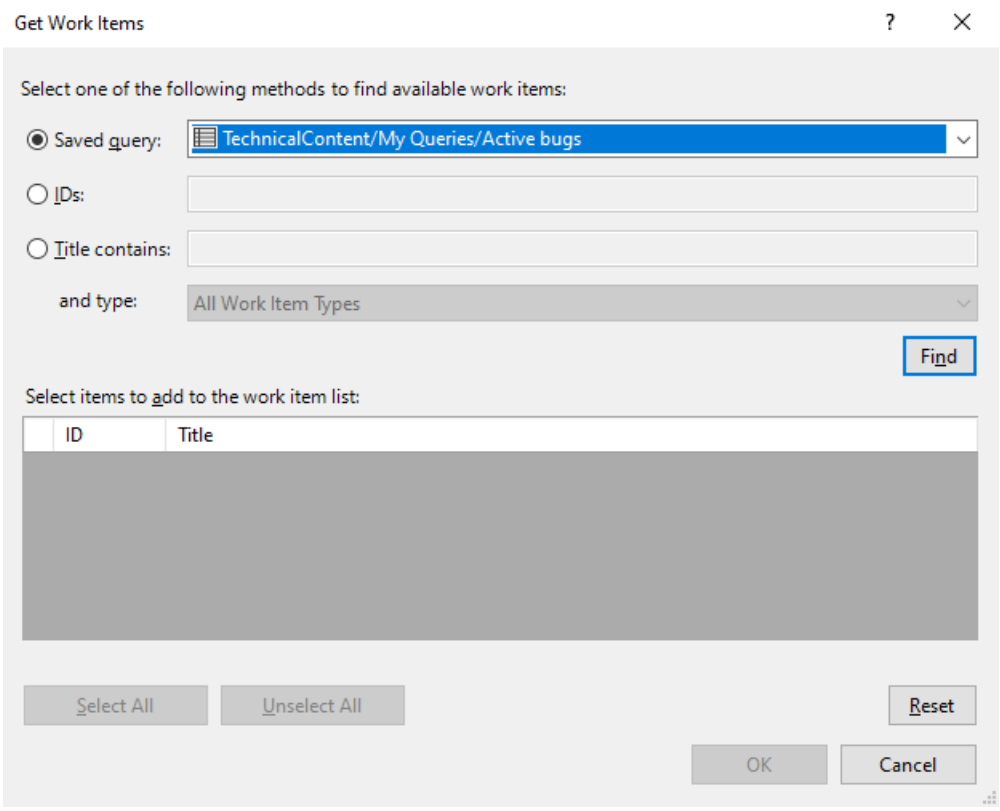
If you're working from a query, modify your query to contain the work items you want. Then refresh your list. The other work items will appear in your list.

If you're working with an input list, complete these steps.

1. From the **Team** ribbon, choose **Get Work Items**.



2. Choose the method you want from the three options available.



If the work items are defined in another project, then first select the Project. Then, make your selections:

- **Query**. Use this method when you've defined a query that contains the set or superset of work items you want.
- **IDs**. Use this method when you know the IDs of the work items that you want to link to. In the **IDs** box, type the IDs of the work items that you want to find, separated by commas or spaces.
- **Title contains**. Use this method to find work items that have a common word or phrase in the title field. In the **and type** list, select the type of work item that you want to retrieve.

> **NOTE**
> To minimize the time required to run the query, narrow the filter criteria of the search.

3. Choose **Find**.

   Only those work items defined for the selected project and specified work item type are listed. To sort on a column field, choose the column **Title**.

4. In the list of returned work items, select the check-box of one or more work items.

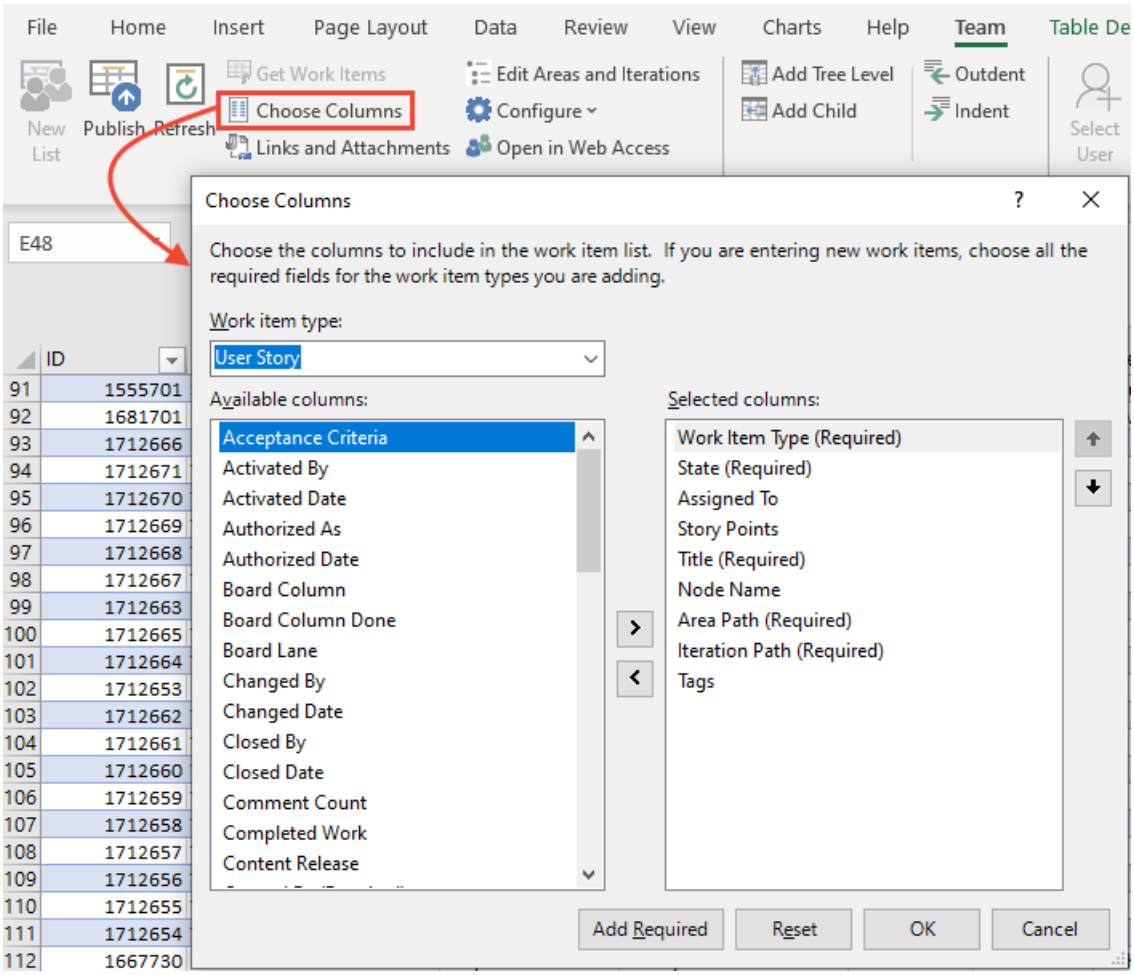
   - Select each work item that should link to the current work item. You can also press the SHIFT key while clicking to select a range of work items, or press the CTRL key while clicking to select multiple work items.
   - Choose **Select All** to select all work items in the list.

Add or remove column fields

If you start your worksheet with a **New List**, you'll see only a set of default field columns. You can add columns using the **Choose Columns** on the Team ribbon.

If you start your worksheet from an existing query, you'll see all the column fields defined for the query. From there, you can add columns using the **Choose Columns**. However, your additions don't modify the underlying query.

1. To assign values to other fields, choose **Column Options** to add the fields of interest.



   - To filter the fields based on work item type, select the **Work item type**.
   - To move or remove a field, choose the field and then select the > or < icons.
   - To change the field sequence, move the field up or down in the list using the up and down arrows.
   - You can add a rich-text field, such as the **Description** field, however you may lose some of the formatting upon publish.
2. Once the fields appear in the worksheet, assign values and publish your updates. When working with identity fields, ones that accept user accounts, see the next section, Select user accounts.
3. Save your worksheet.

Select user accounts

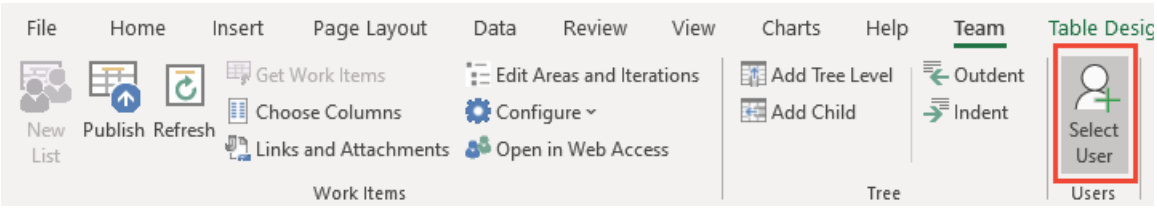
You can use the Select User feature to find user accounts and assign values to person named fields. Also, this feature provides access to the most recently used (MRU) values. If your team contains several hundreds or thousands of user accounts, you'll want to use this feature.

TIP

Without the **Select User** feature, you must enter user names exactly as they are in the database, or you'll receive data validation errors upon trying to publish.

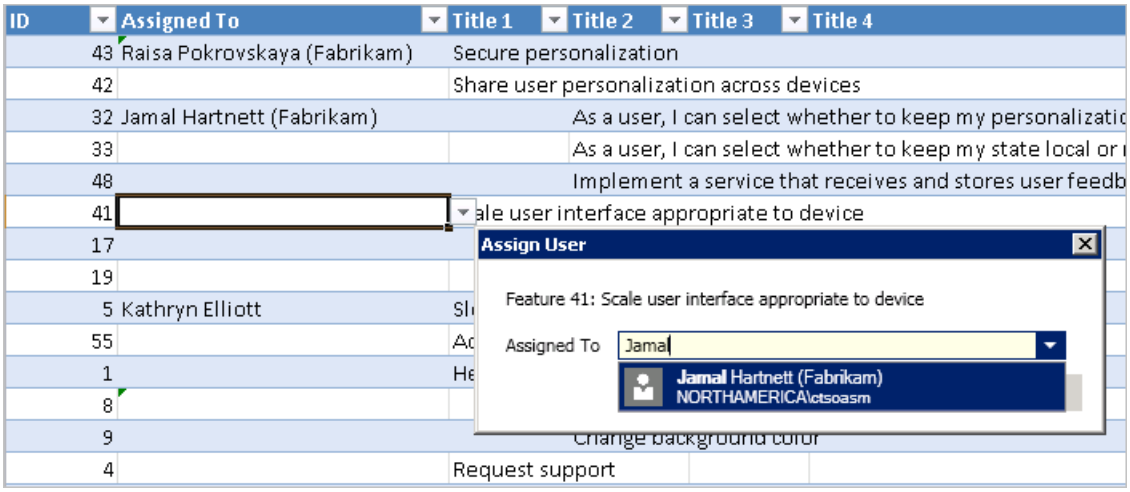
1. If you haven't installed or updated to the latest version of Visual Studio (at least VS 2015.1 or later version, do that now. You need the latest update to access the Select User feature.

2. Choose an identity or person-named field to activate the **Select User** feature in the Team ribbon.



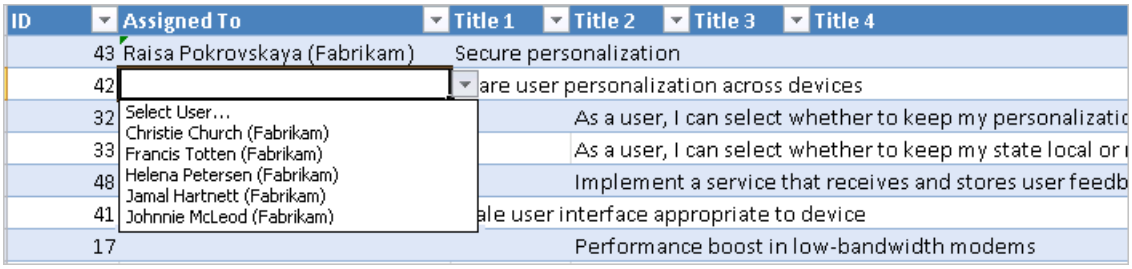
An identity or person-named field is a field that contains a user identity. These fields are typically synchronized to a database of user accounts, such as Azure Active Directory, Active Directory, or a Workgroup.

3. Begin typing the name of the user account and the Assign User dialog will automatically filter the results until you can select the account of interest.



Enter a letter to tab to the start of names beginning with that letter. Enter only user names as account aliases aren't recognized.

You'll notice that as you select user names, Excel remembers your recent selections and you can select the user accounts directly from the field.



Link work items

You can complete many actions from the **Links** tab of the **Links and Attachments** dialog. Specifically, you can:

- Review the existing links defined for the selected work item

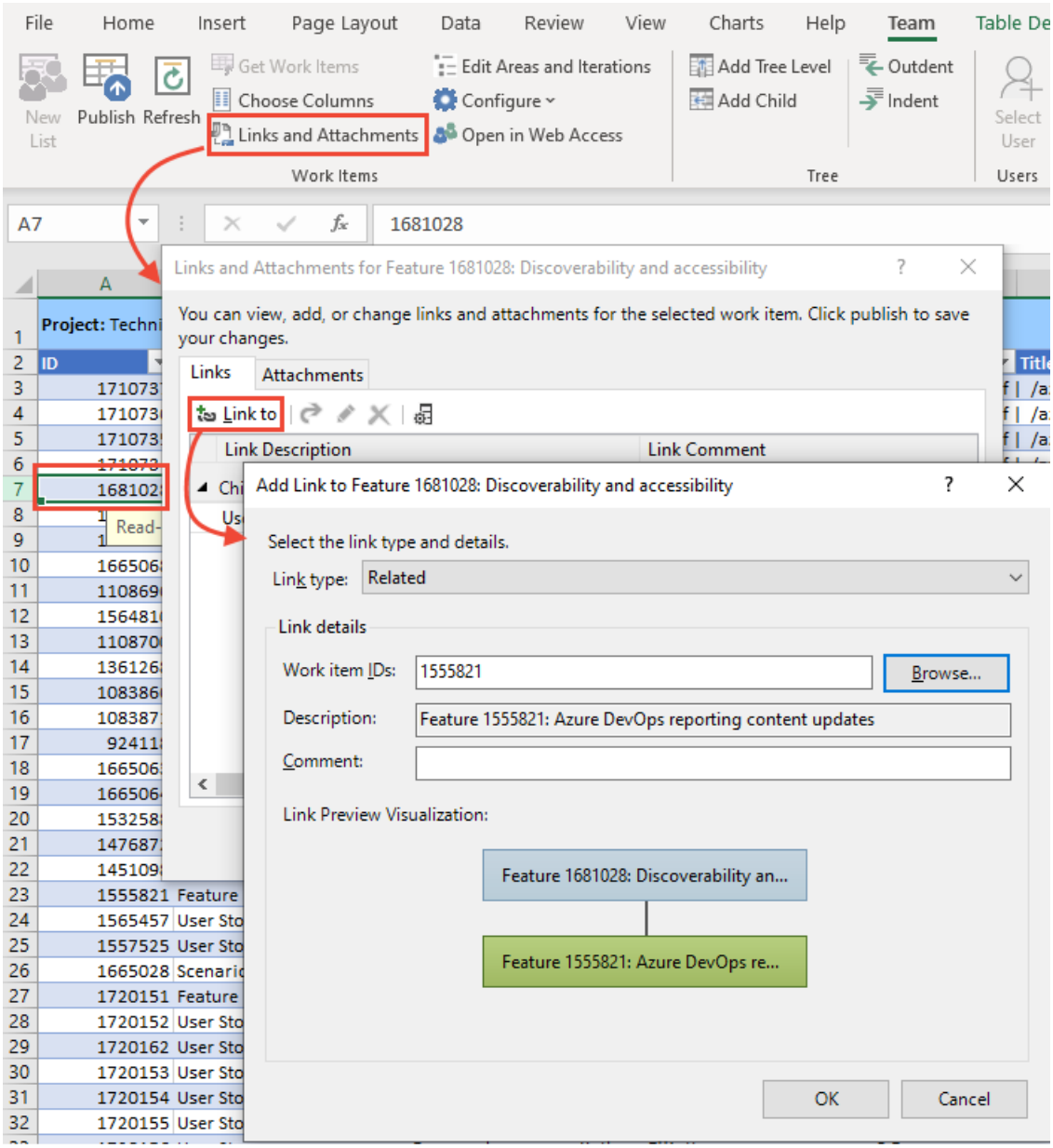
- Add links to selected work items to one or more work items or select objects
- Delete links
- Open a linked work item (opens in the web portal)
- Edit the link type of an existing link
- Add columns to the Link list and sort on that list

For more information on linking work items, see Link user stories, issues, bugs, and other work items.

View and add links

You can't use the Links and Attachments dialog to bulk update work item links. You can only bulk update tree-topology link types using a tree list.

1. To link a work item to other work items, choose the work item and then choose **Links and Attachments**. From the Links tab, choose **Link to** and then choose the **Link Type** and work item(s) you want to link to. Choose **OK** and then **Publish**.



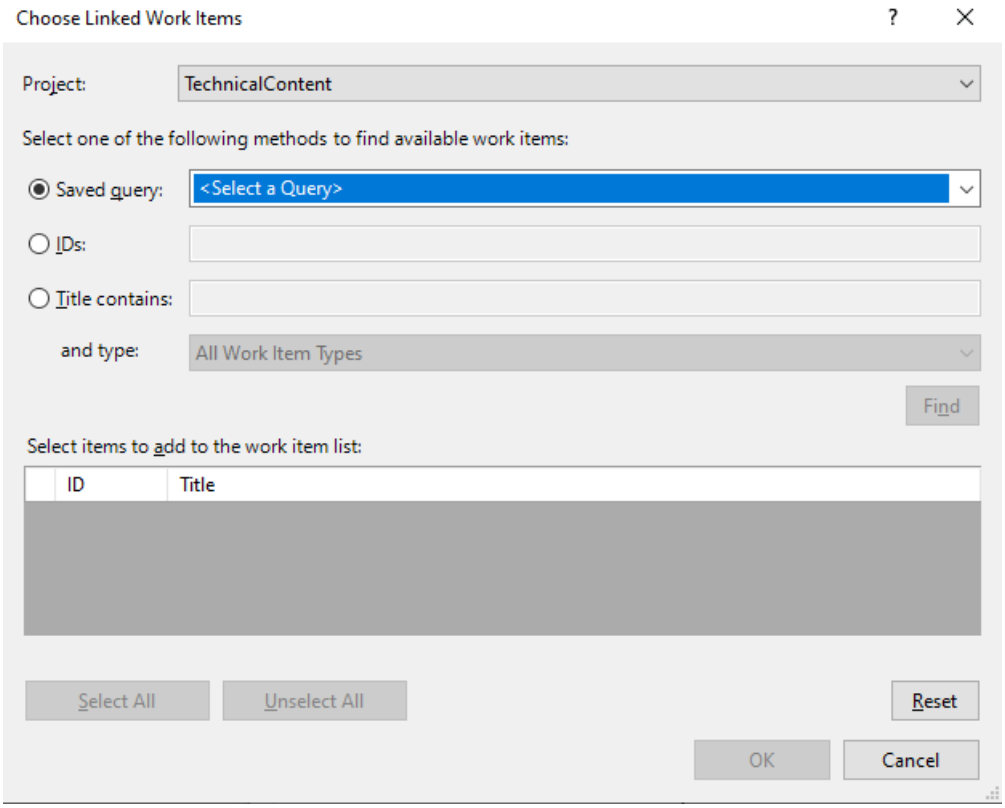
When finished, choose **Close** to dismiss the dialog.

2. To link several work items to the same work item(s), multi-select them by using **Ctrl-click** for consecutive rows, or **Shift-click** for non-consecutive rows.

Find work items to link to

From the Add link dialog, you can open a secondary dialog to choose one or more work items to link to. If you're going to find and list work items to link to by using a saved query, first define the query to use.

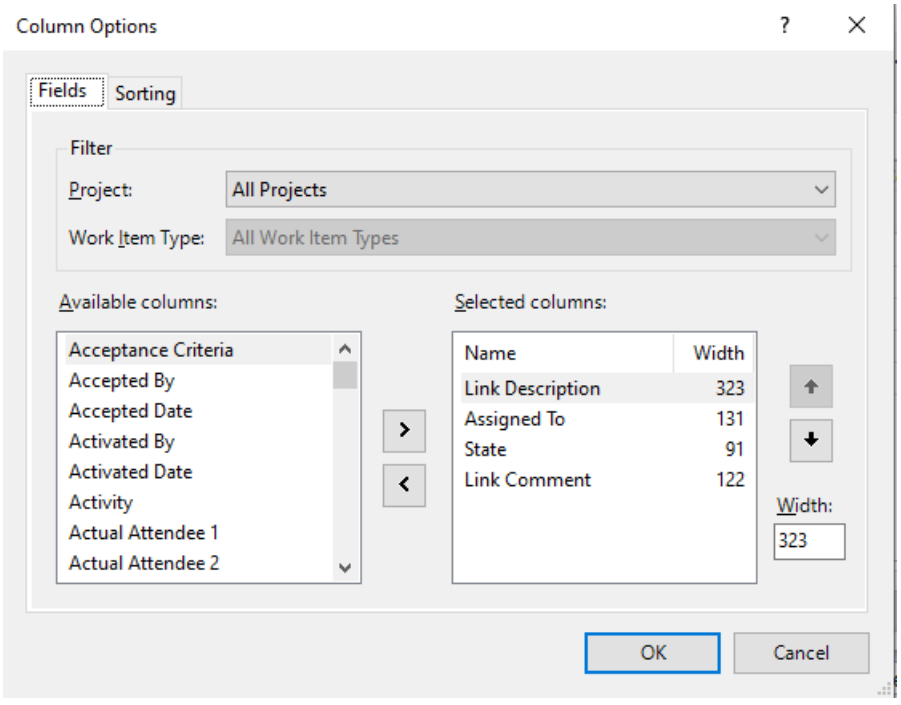
From the Add link dialog, choose the **Browse** button (Visual Studio) to open the following dialog.



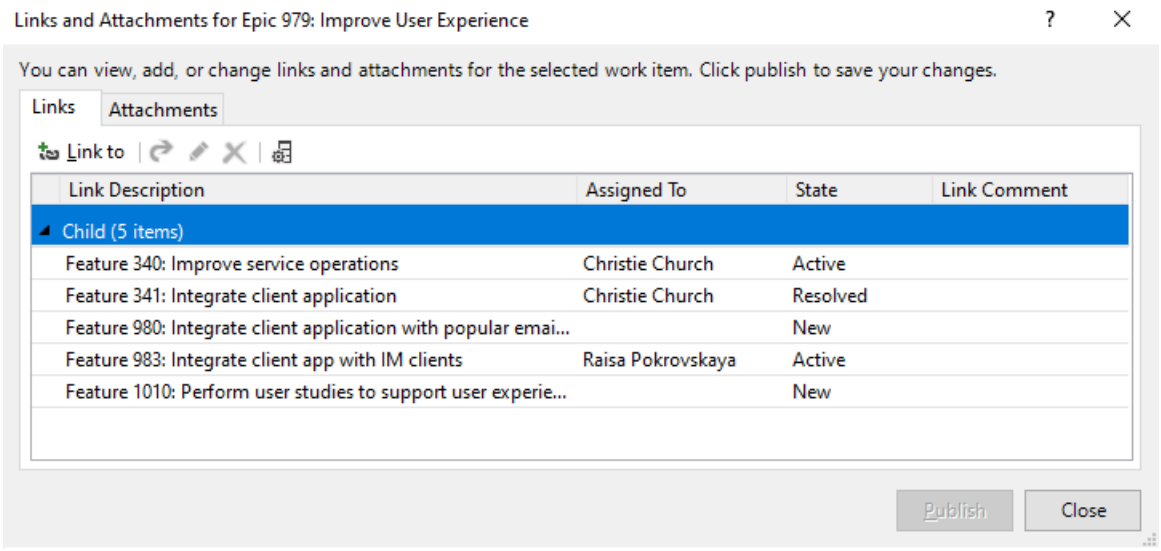
The **Choose Linked Work Items** dialog works in the same way as the **Get Work Items** dialog. To learn more, see Add existing work items to your worksheet described earlier in this article.

Add columns to the links list

1. From the **Links** tab, choose the **Columns** icon, and add the fields you want displayed. Here we add the Assigned to and State fields.



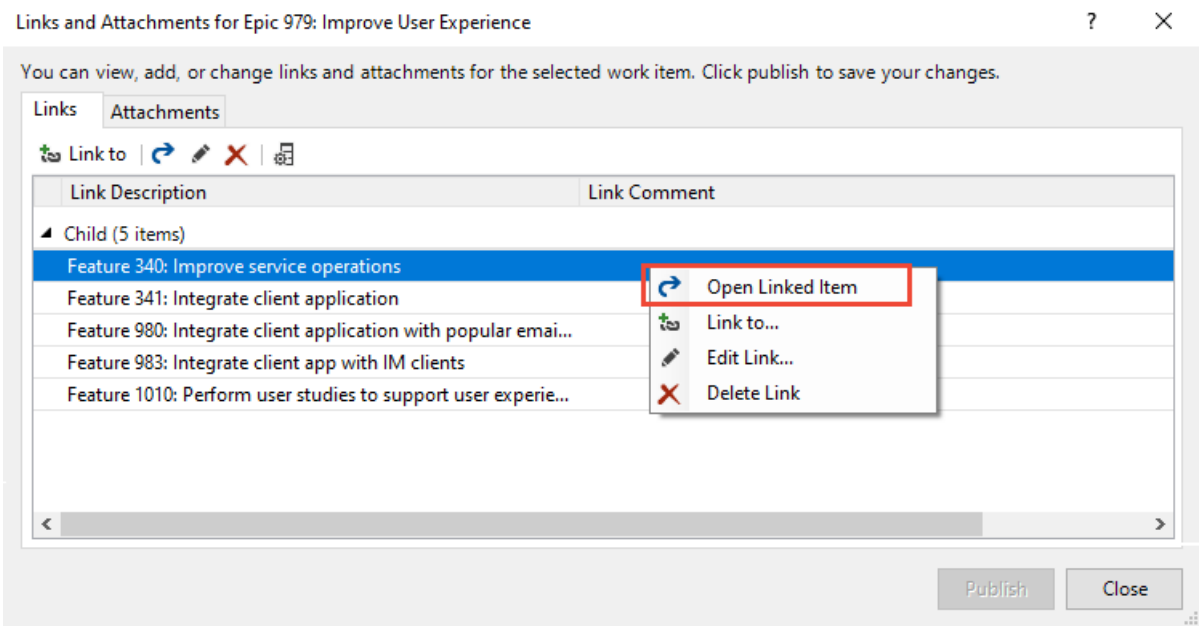
2. To reorder the links, choose the field to sort the list on that field.



This dialog works in the same way as the **Get Work Items** dialog. See Add existing work items to your worksheet described earlier in this article.

Open a linked work item

From the **Links** tab, choose the linked work item, right-click to open the context menu, and choose **Open Linked Item**.

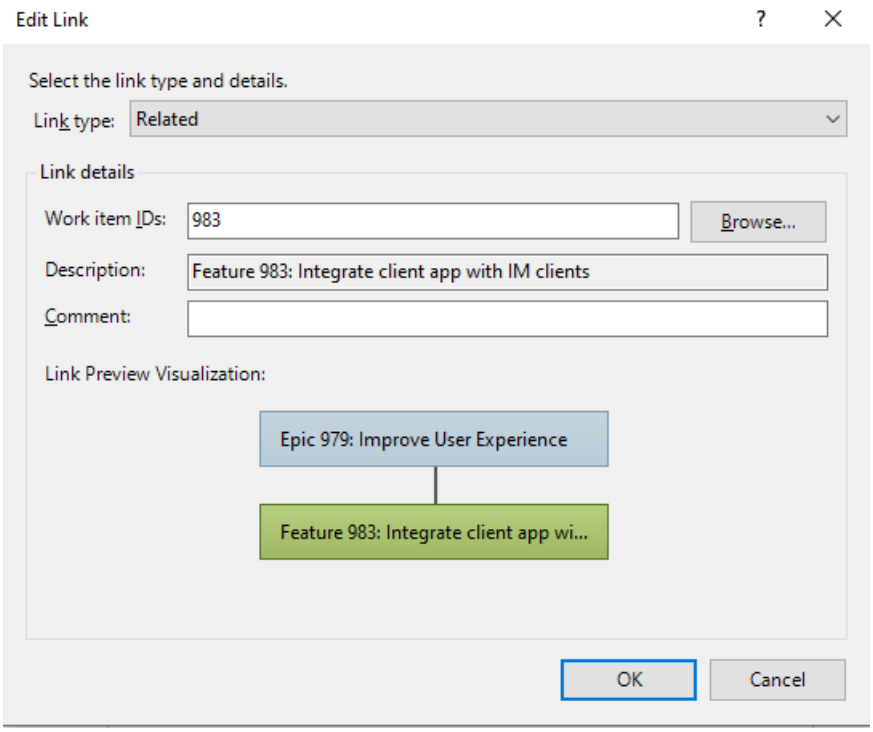


The work item opens in your web portal.

Edit the link and change the link type

You can edit any link listed. You can change the link type and the work items linked to.

1. Choose the link and choose the **Edit** icon.
2. Change the link type as needed.



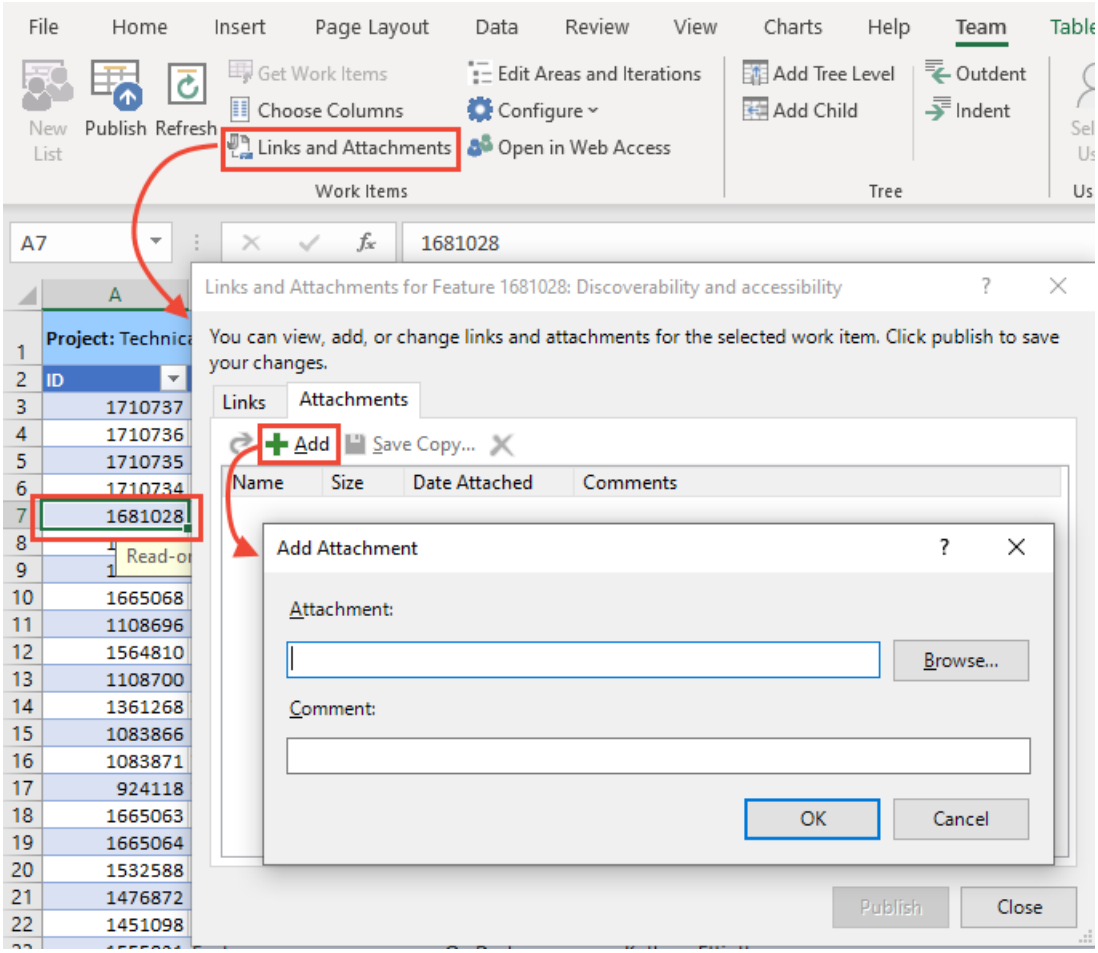
3. To change the work item linked to, enter the ID of the work item, or choose **Browse** to find the work item(s) to link to.

   The **Choose Linked Work Items** dialog works in the same way as the **Get Work Items** dialog. To learn more, see Add existing work items to your worksheet described earlier in this article.

Add attachments

- To add attachments, choose the work item, then **Links and Attachments**, and then the **Attachments** tab.

  Choose the file you want to attach, then choose **OK** and then **Publish**.



When finished, choose **Close** to dismiss the dialog.

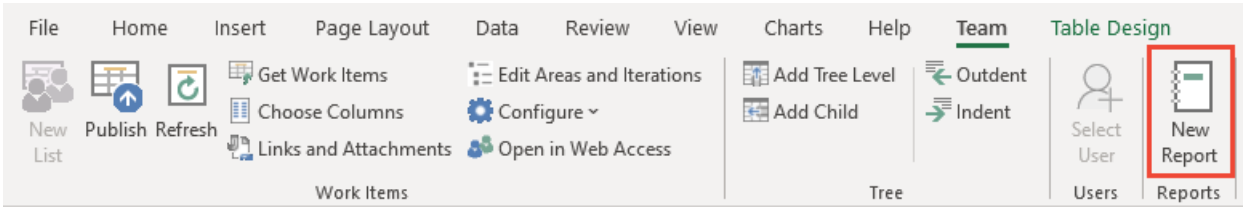
- To add the same attachment(s) to several work items, multi-select them by using **Ctrl-click** for consecutive rows, or **Shift-click** for non-consecutive rows.

Create a report

You can create a report or chart from the web portal for flat-list queries. See Track progress by creating status and trend query-based charts.

> **IMPORTANT**
> You can only create an Excel report using **New Report** from an on-premises Azure DevOps Server. These reports require that your project's project collection is configured to support SQL Server Analytics Server.

You can create a report using the **New Report** feature based on a flat list of work items.



To learn more, see Create Excel reports from a work item query.

Resolve publishing errors

To resolve publishing errors that arise when working in Excel, see one of the following articles:

- Resolve data conflicts: A data conflict occurs when a field value is changed in Azure Boards since the last time you published from Excel.
- Resolve data validation errors: A data validation error occurs if a field value violates the rules for that field and work item type.
- Resolve invalid links in a tree hierarchy: An invalid link occurs if a team member view work items in Excel as a hierarchy or tree list, and then moves a work item or sorts the list such that it breaks the dependencies between work items. You can resolve this error by reviewing the error message and repositioning work items to reflect the work item structure.
- Address Error TF208104: Hierarchical Link Relationship Is Locked:
  If you receive error TF208104, the changes you made to the fields are published. But your changes to the link hierarchy aren't published. At least one of the link relationships defined for the work item is locked by another process, such as Project Server integration.

Related articles

- Bulk modify work items (web portal)
- Azure DevOps Office integration issues
- FAQs: Work in Excel connected to Azure Boards
- Bulk import or update work items using CSV files
- View and add work items
- Basic Excel tasks

- Bulk modify work items (web portal)
- Azure DevOps Office integration issues
- FAQs: Work in Excel connected to Azure Boards

- Create Excel reports from a work item query

- Basic Excel tasks

Import or update work items in bulk by using CSV files in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

You can import and export work items in bulk using a CSV formatted file. While you can continue to use Excel for bulk import and updates, you can use the native import/export feature that doesn't require Excel. To learn more about using Excel, see Bulk add or modify work items with Excel.

You can export of work items in bulk using a CSV formatted file. While you continue to use Excel for bulk import and updates, you can use the native export feature from Queries that doesn't require Excel. To learn more about using Excel, see Bulk add or modify work items with Excel.

> **NOTE**
> The export feature is available with Azure DevOps Server 2019 Update 1 and later versions. The import feature is available with Azure DevOps Server 2020 and Azure DevOps Services.

Import new work items

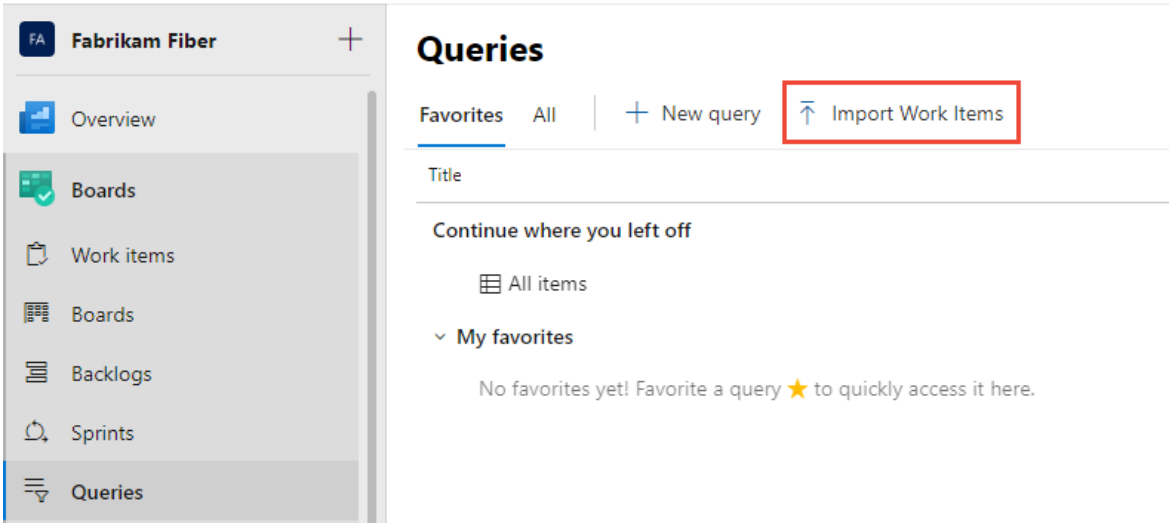
All work items you import are created in a new state. This rule means that you can't specify field values that don't meet the field rules for the new state. For example, when you create a new user story with the Agile process, the State=New and the Reason=New. No other values can be specified.

1. Create a local ***import.csv*** file and open it in Visual Studio Code or Excel.
2. The file must contain the **Work Item Type** and the **Title** fields. You can include other fields as needed. For a list of default fields, see Work item field index.

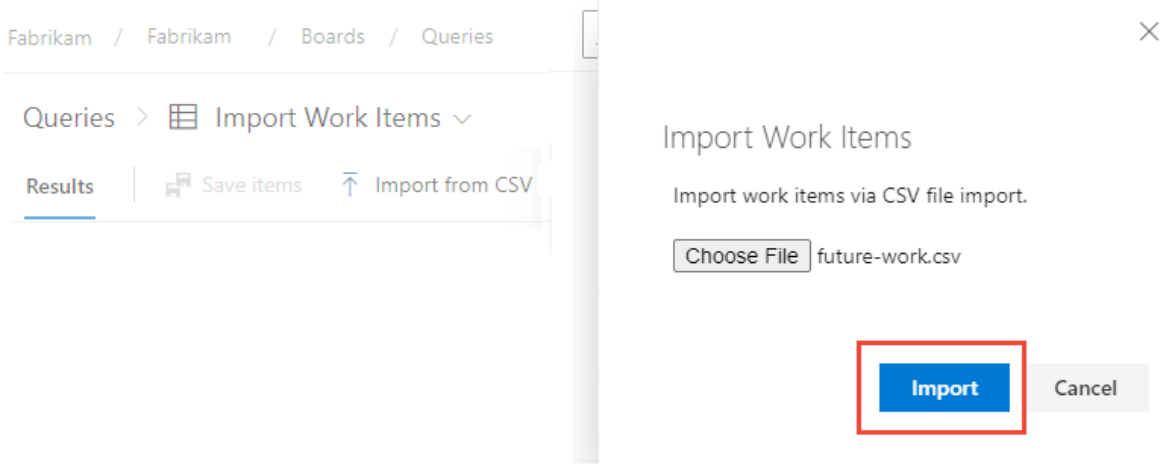
   In the following example, we include the **Priority** field.

   ```
   Work Item Type,Title,Priority
   Issue,Fix issues with code,1
   Issue,Merge testing modules,3
   Issue,Open private preview for select customers,2
   Issue,Enable feature for customer champs,2
   Issue,Remove old test code,2
   ```

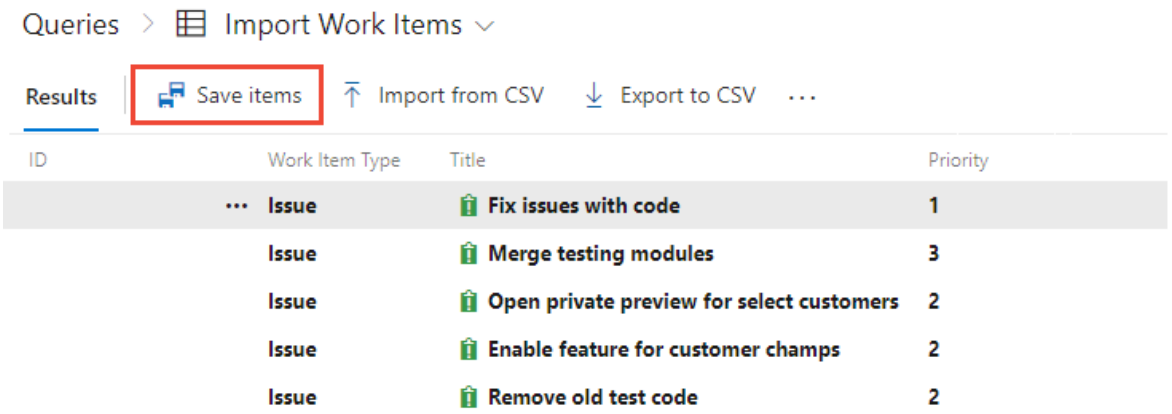
3. From the web portal for your project, open **Boards>Queries** and choose the **Import Work Items** option.



4. Select your CSV file and then choose **Import**.

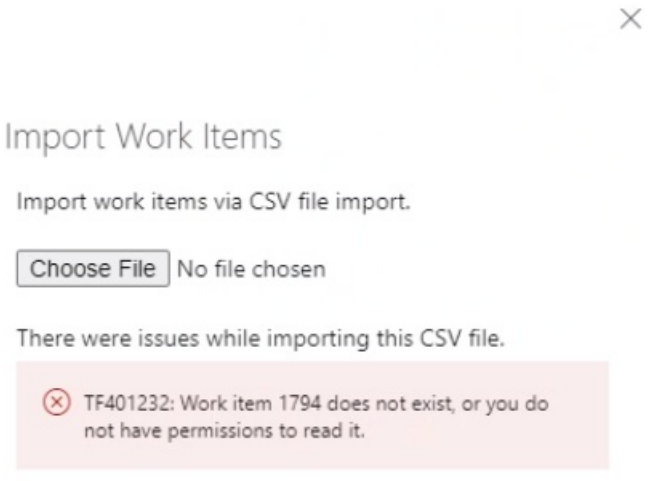


5. The import process loads the imported work items into the queries view in an **unsaved** state. No IDs are assigned. Verify the results are what you want. Then, choose **Save Items** to save the work items.

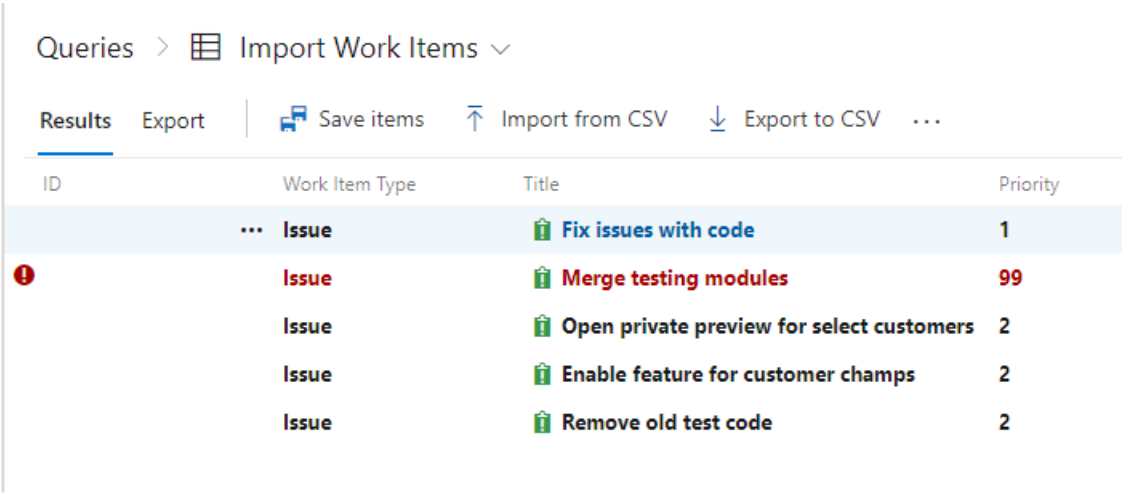


NOTE

Make sure you don't assign IDs to new work items that you're adding. You'll receive an error message similar to the following if you do so.



6. The system highlights those work items with data issues. Resolve the data issues before you save the work items. In this example, an invalid value has been entered into the Priority field. Fix the data by opening the work item directly. Instead, use bulk edit to fix several work items with the same issue.

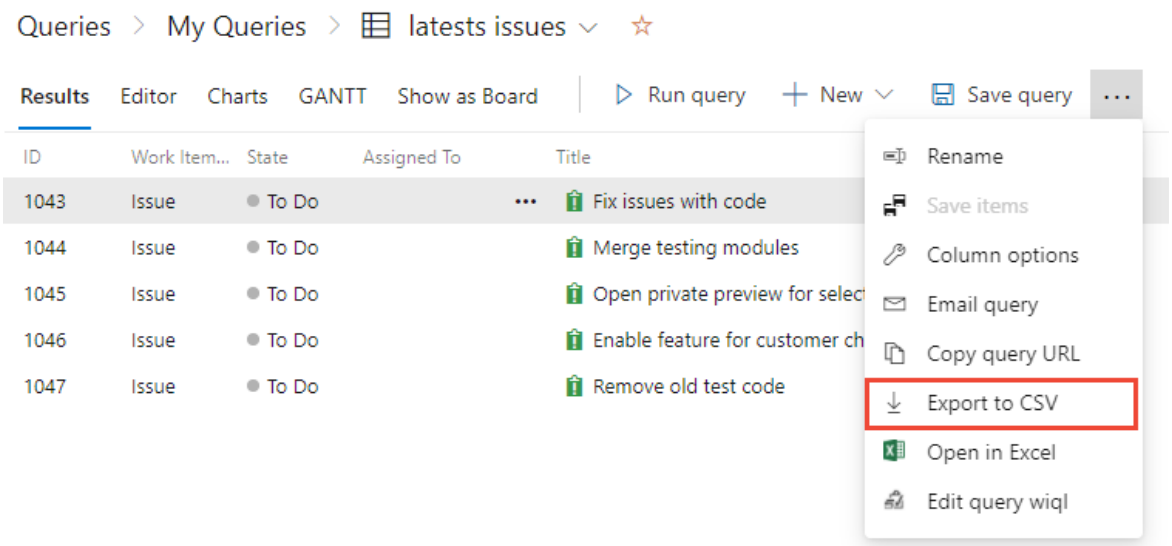


TIP

You can add parent-child links between work items you import by indenting the title columns as shown in the example later in this article, Can I import a CSV file that have parent-child links?. However, you can't specify any other link types when importing or updating work items.

Update existing work items

1. To update work items, create a query that contains all the columns you want to export and possibly edit. Save your query and select **Export to CSV** to save the data.csv file to your local machine.



The exported file should look similar to the following syntax:

```
ID,Work Item Type,State,Assigned To,Title,Tags
"1043","Issue","To Do",,"Fix issues with code",
"1044","Issue","To Do",,"Merge testing modules",
"1045","Issue","To Do",,"Open private preview for select customers",
"1046","Issue","To Do",,"Enable feature for customer champs",
"1047","Issue","To Do",,"Remove old test code",
```

2. Make the edits to your work items. Your CSV file must contain the **ID**, **Work Item Type**, **Title**, and **State** fields. Any other fields you want to include are optional.

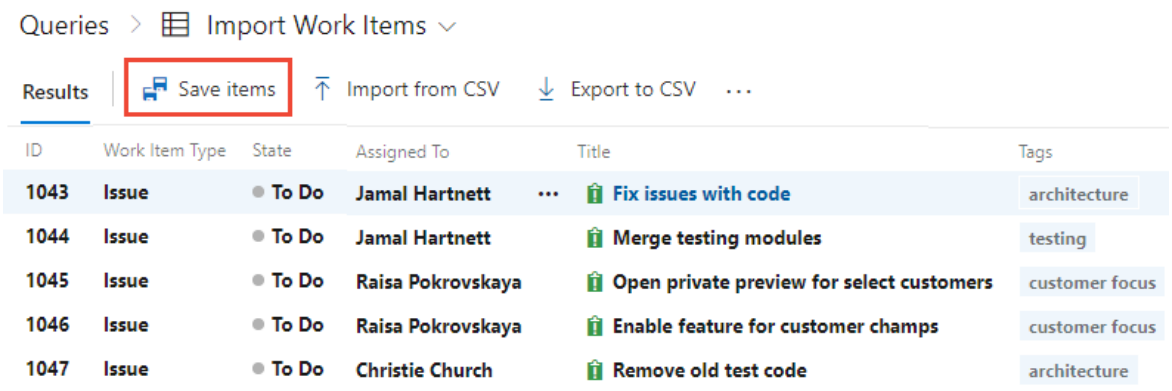
> **NOTE**
>
> When importing identity fields, the name and email must be entered in the following format `"Display Name <email>"`. For example, to assign work to Jamal Hartnett, specify `"Jamal Hartnett <fabrikamfiber4@hotmail.com>"`. If you specify a value that isn't recognized as a valid user to the system, you may encounter problems with the import.

In the following example, we change several values on existing working items.

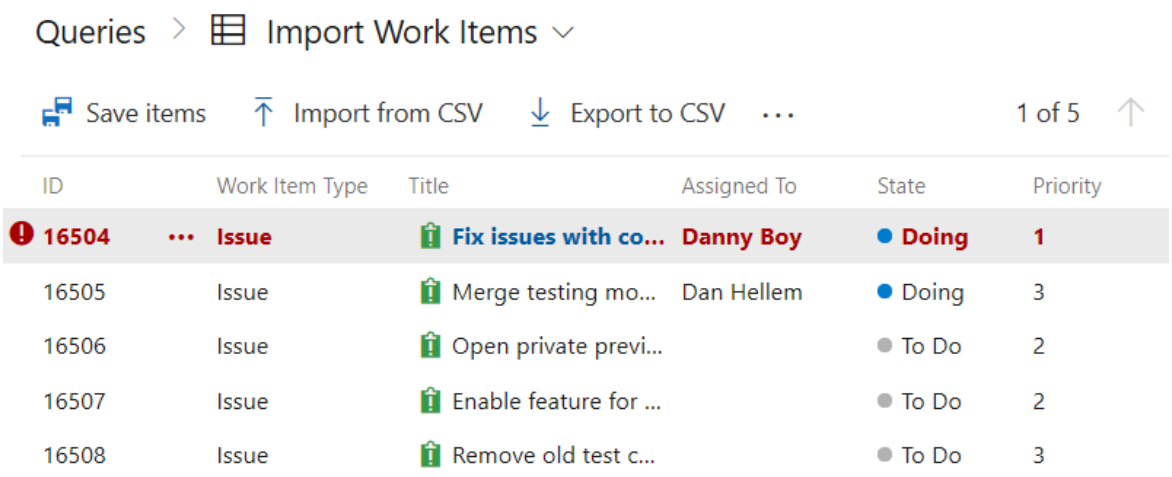
```
ID,Work Item Type,State,Assigned To,Title,Tags
"1043","Issue","To Do","Jamal Hartnett <fabrikamfiber4@hotmail.com>","Fix issues with
code",architecture
"1044","Issue","To Do","Jamal Hartnett <fabrikamfiber4@hotmail.com>","Merge testing modules",testing
"1045","Issue","To Do","Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>","Open private preview for
select customers","customer focus"
"1046","Issue","To Do","Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>","Enable feature for customer
champs","customer focus"
"1047","Issue","To Do","Christie Church <fabrikamfiber1@hotmail.com>","Remove old test
code",architecture
```

3. Save the file and import (see steps 4-6 from the previous import section.)

4. The results list with work items that contain value changes appear highlighted in bold. Choose **Save Items** to apply the changes.



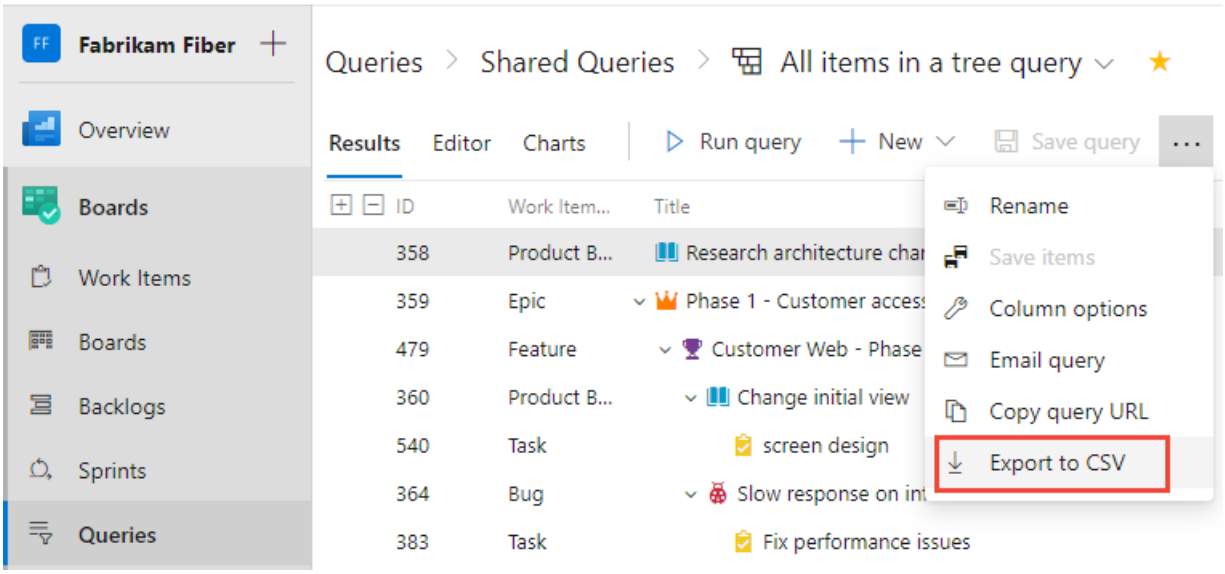
5. Work items with data issues are highlighted in red and need to be resolved before you can save them. In this example, an invalid value appears in the Assigned To field. Fix the data by opening the work item directly. Instead, you can use bulk edit if you have many work items with the same issue.



Export list as a CSV file

From any query, you can export a list of work items as a comma-delimited list. Open the query, choose the ··· actions icon, and choose **Export to CSV**.

> **NOTE**
> Requires Azure DevOps Server 2019 Update 1 or later version.



Export and import work items to a different project

You can use this feature to export work items from one project and import them to another project. However, before importing them to another project, you must remove the work item ID. You come across an error if you attempt to import new work items to a project with an ID specified.

Import or update rich-text fields

You can update or import rich-text fields such as the **Description** or **Acceptance Criteria** fields. Rich-text fields are HTML formatted fields. Replace lines ending in CRLF by surrounding sentences with `<p>... </p>`.

For example, you can import the following work item, which includes three lines of text in the Description field.

```
Work Item Type,Title,Description
"Product Backlog Item","Hello World Web Site - 8","<p><strong> You can include bold text</strong></p>
<p><em> And italic text</em></p><p><u> Underline text</u></p>"
```

Q & A

Can I import new items and update existing items in the same CSV file?

Absolutely! Leave the ID field empty for any new work items. In the following example, the last entry for an Epic doesn't specify an ID.

```
ID,Work Item Type,Title,Assigned To,State,Priority,Tags
"16504","Issue","Fix issues with code",,"To Do","1",
"16505","Issue","Merge testing modules",,"To Do","3",
"16506","Issue","Open private preview for select customers",,"To Do","2",
"16507","Issue","Enable feature for customer champs",,"To Do","2",
"16508","Issue","Remove old test code",,"To Do","2",
,"Epic","Track Telementry for data imports",,"To Do","2",
```

Can I import a CSV file that has parent-child links?

Yes, you can add child work items by indenting title columns. The following example adds three child Issues under the already defined Epic.

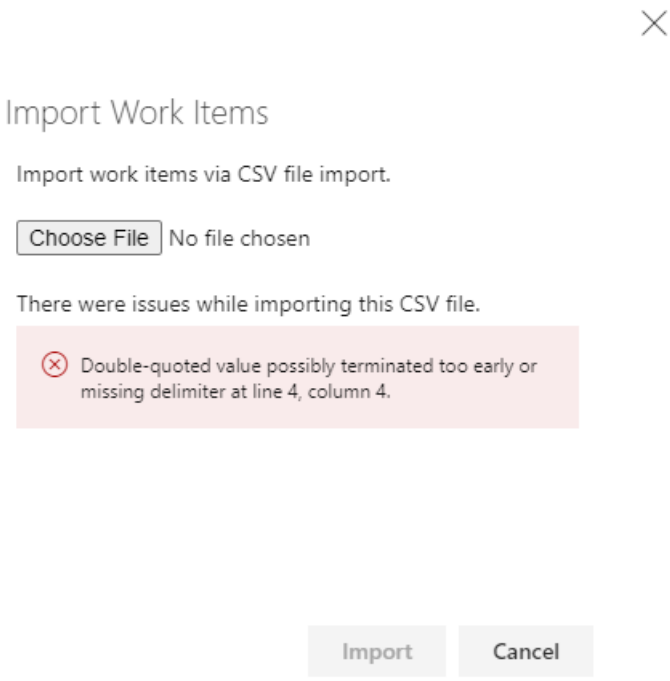
```
ID,Work Item Type,Title 1,Title 2,Assigned To,State,Priority,Tags
"165","Epic","Track Telementry for data imports",,,"To Do","2",
,"Issue",,"Fix issues with code",,"To Do","1",
,"Issue",,"Open private preview for select customers",,"To Do","2",
,"Issue",,"Enable feature for customer champs",,"To Do","2",
```

Here's a better visual in Excel.

	A	B	C	D	E	F	G	H
1	**ID**	**Work Item Type**	**Title 1**	**Title 2**	**Assigned To**	**State**	**Priority**	**Tags**
2	165	Epic	Track Telementry for data imports			To Do	2	
3		Issue		Fix issues with code		To Do	1	
4		Issue		Open private preview for select customers		To Do	2	
5		Issue		Enable feature for customer champs		To Do	2	
6								

How do I know if my imported file has errors?

Any problems with the formatting of your CSV file appear in the Results page of the import view. You can't import the work items until the formatting and syntax is correct.



The work items' results always lists the data errors found for individual work items. Fix each error either from the web portal, or in the CSV file and import again.

Related articles

- Work item field index
- Bulk add or modify work items with Excel
- FAQs: Work in Excel connected to Azure Boards

Bulk move work items and change the work item type in Azure Boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Often you find that someone created a work item of the wrong work item type (WIT) or within an incorrect project. You can correct these issues for individual work items or bulk modify several work items. You can also remove work items added to your backlog or taskboard that aren't relevant anymore.

> **TIP**
> For TFS 2018 and earlier versions, you can't change the work item type for an existing work item, but you can copy the work item and specify a new type. Also, if you have several work items with type changes you want to make, you can export them using Excel, and then re-add them as a new type.

To remove, delete, or restore deleted work items, see Remove, delete, or restore work items.

In this article you'll learn:

- How to change the work item type of one or more work items
- How to move one or more work items to another project

> **TIP**
> From the web portal, you can multi-select several work items from a backlog or query results page and perform a bulk update using the associated feature. To change, move, delete, or restore several work items at the same time, see Bulk modify work items.

Prerequisites

- You must be a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To change the work item type, you must be granted **Stakeholder** access or higher.
- To move work items to another project, you must be a member of the **Project Administrators** group or have the **Move work items out of this project** permission set to **Allow**. By default, the **Contributors** group doesn't have this permission set. Users granted **Stakeholder** access don't have access to this feature.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to all work tracking features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.

- To modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To change the work item type, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To move work items to another project, the project must use an Inherited process model.
- To move work items to another project, you must be a member of the **Project Administrators** group or have the **Move work items out of this project** permission set to **Allow**. By default, the **Contributors** group doesn't have this permission set. Users granted **Stakeholder** access don't have access to this feature.

  > **IMPORTANT**
  > You can change the work item type or move work items to another project within a project collection. These features require that the data warehouse is disabled. With the data warehouse disabled, you'll use the Analytics Service to support your reporting needs. To learn more about disabling the data warehouse, see Disable the data warehouse and cube.

To learn more, see Set permissions and access for work tracking or Change project-level permissions.

> **IMPORTANT**
> You can't change type or move work items whose work item types support test management or that belong to the Hidden Types Category. This includes all work items that track tests—such as test cases, shared steps, and shared parameters—code review requests and responses, and feedback requests and responses.

> **IMPORTANT**
> You can't change type, move work items, or delete/restore work items whose work item types support test management or that belong to the Hidden Types Category. This includes all work items that track tests—such as test cases, shared steps, and shared parameters—code review requests and responses, and feedback requests and responses.
>
> Also, you can't change the work item type if the project is defined on a collection that uses the On-premises XML process model.

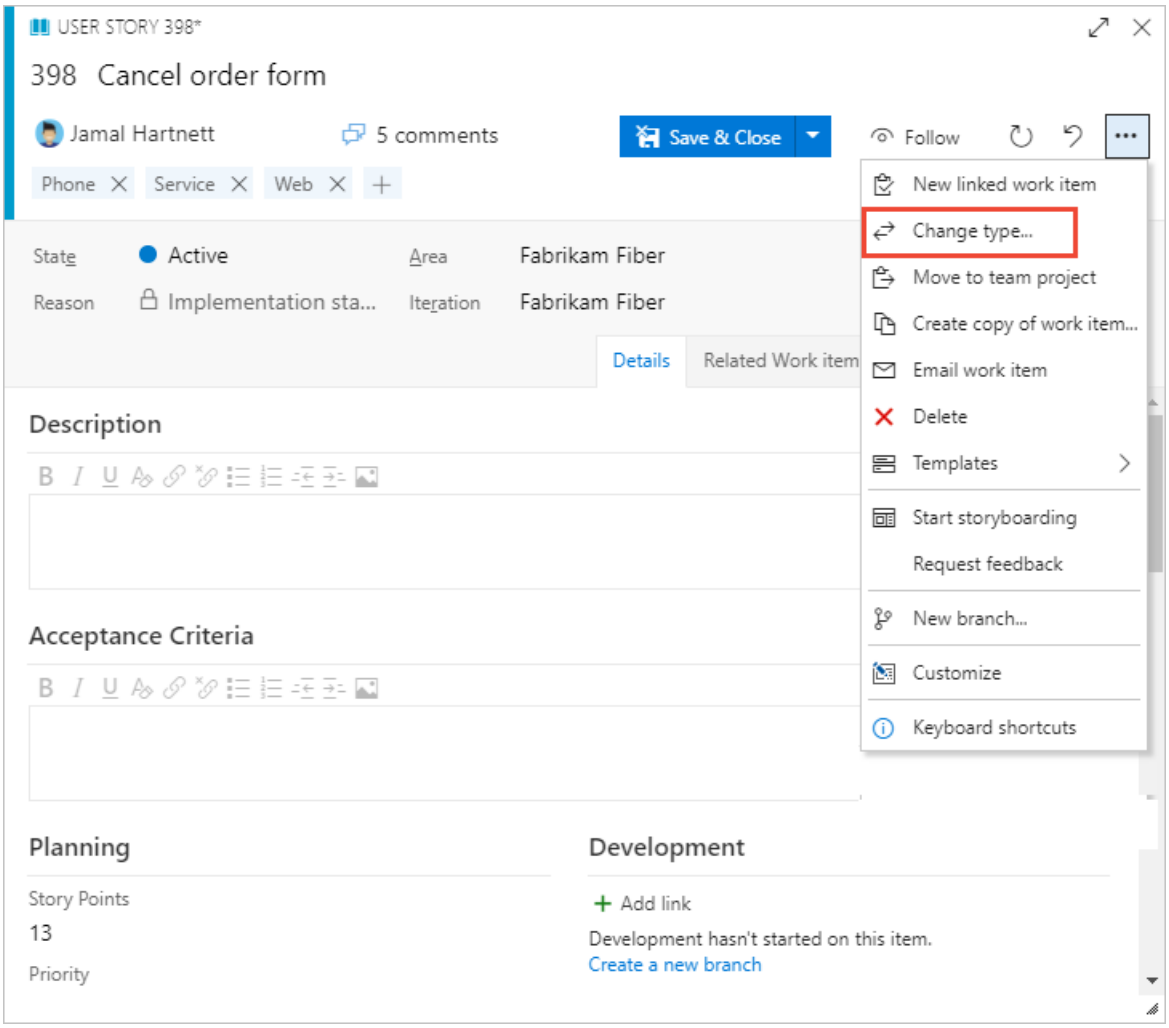
Change the work item type

Changing the work item type refreshes the work item form with the fields defined for the type selected. For example, you can change a bug to a task and the form will refresh with the fields defined for a task.

> **NOTE**
> You can't change the work item type if the project is defined on a collection that uses the On-premises XML process model. Also, you can't change the work item type of work items associated with test management.

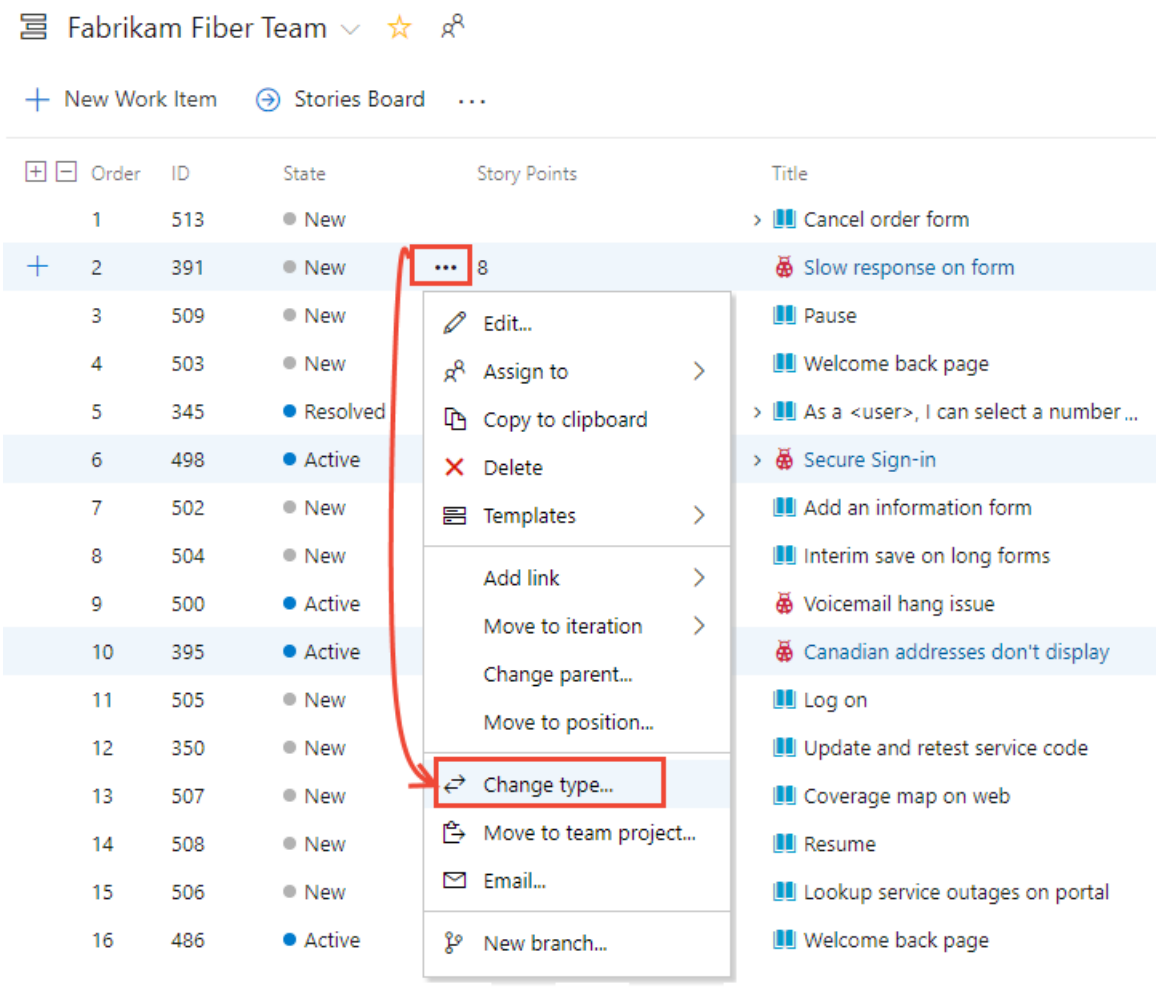
You can change a single work item or several multi-selected work items to a new type.

1. Open a work item, choose the ··· actions icon, and select the ⇄ **Change type...** option.



Or, from the backlog or query results page, multi-select several work items whose type you want to change. You can select several work items of the same type or different type so long as you want to change them all to the same work item type.

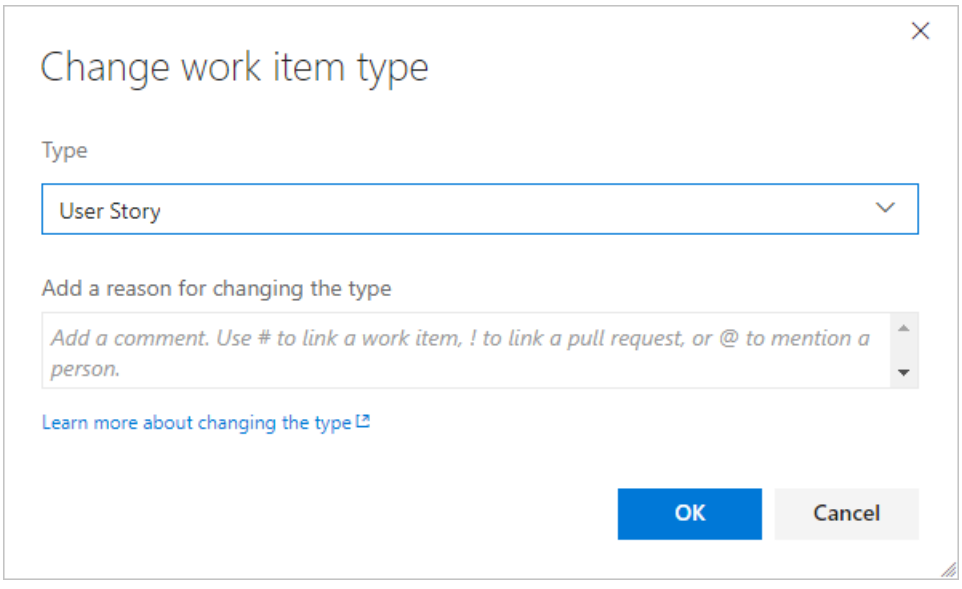
Choose the ··· actions icon, and select the **Change type...** option.



IMPORTANT

From the **Query Results page**, the **Change type...** option becomes unavailable if you have checked the Query Editor's **Query across projects** checkbox.

2. Select the type and optionally enter a comment.



Comments are automatically added to the Discussion control and an entry is made to the **History** control. Also, the system automatically resets the State and Reason fields to the default initial values for the work item type that you move.

3. Save the work item(s) to complete the change.

> **NOTE**
>
> The system automatically resets the **State** and **Reason** fields to the default initial values of the specified type. However, in some cases you may need to open the work item to change the **State** or **Reason** field to a value supported by the changed-to work item type.

From the **Query Results** page, save all work items that you bulk-modified. When you bulk modify items from the backlog, they're automatically saved. Work items shown in bold text indicate that local changes haven't yet been saved to the data store. The system automatically saves each work item. Refresh the page to reflect your changes.

Move a work item to another project

When you discover that a work item belongs to a different project within your organization or collection, you can move it where it belongs. You can move a single work item or several multi-selected work items.

Note the following:

- You can only move work items from one project to another project within the organization or collection.
- You can only move work items whose type exists in the project you're moving it to. For example, you can't move User Stories to a project based on the Scrum process that doesn't contain User Story as a work item type.
- You can't move work items associated with test management.
- To move work items to another project, you must be a member of the Project Administrators group or be granted explicit permissions to move work items.
- Users granted **Stakeholder** access don't have access to this feature even if granted permission.

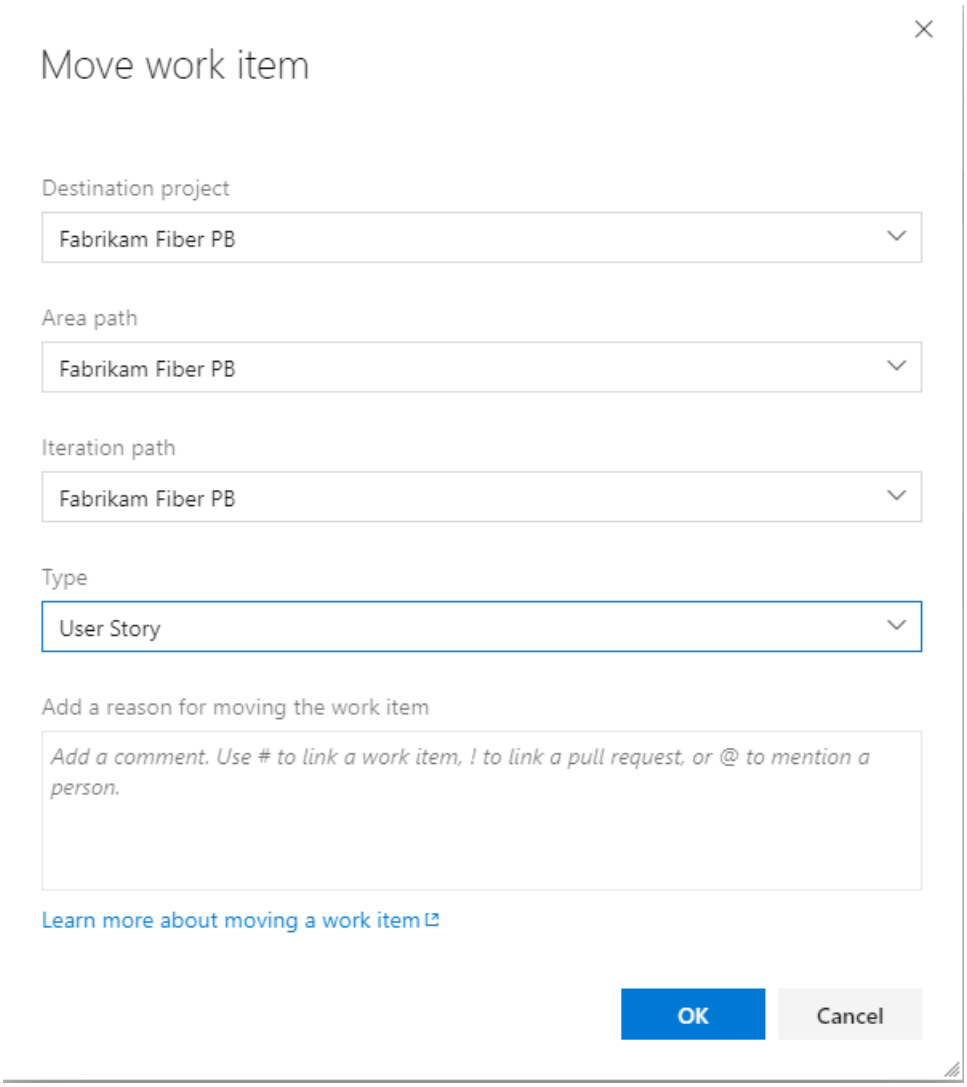
1. Open the work item and choose the **Move...** option from the work item form's ··· Actions menu.

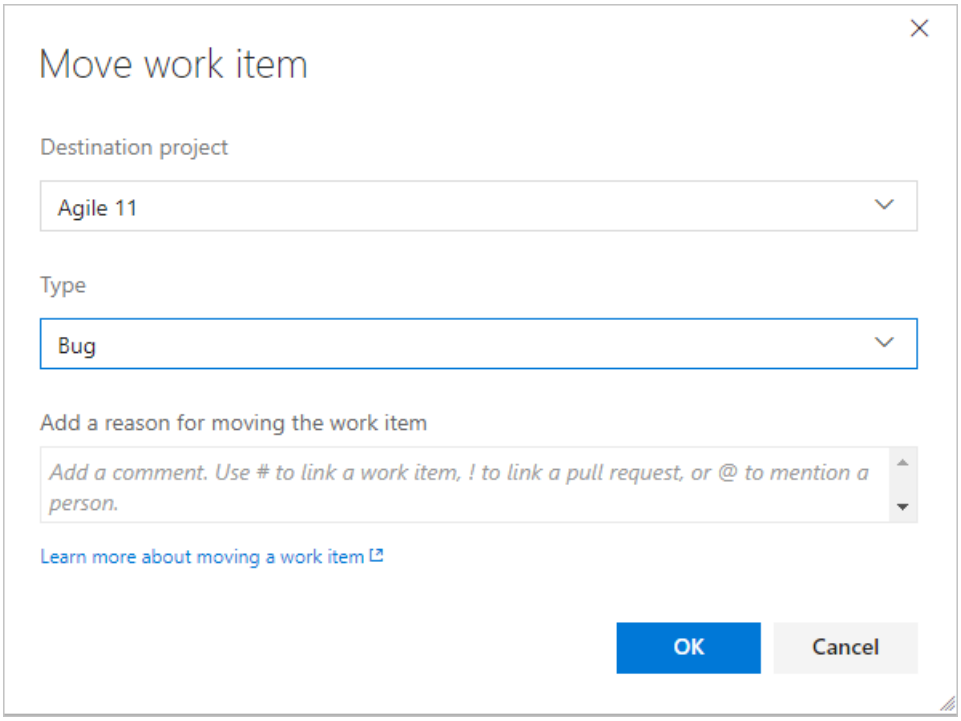
   If you don't see the option, then you haven't been granted permissions to move work items out of the project.

   Or, from the backlog or query results page, multi-select several work items that you want to move to another project. You can select several work items so long as you want to move them all to the same project.

   Choose the ··· actions icon to open the context menu of one of the selected work items, and choose the **Move...** option.

2. Select the destination project and choose the other options available, including changing the work item type. Optionally enter a comment.





NOTE

Child work items are not moved. They remain in the origin project, but the Parent-Child links remain in place.

Comments are automatically added to the **Discussion** control and an entry is made to the **History**

control. Also, the system automatically resets the State and Reason fields to the default initial values for the work item type that you move.

Related articles

- Remove or delete work items
- View and add work items using the Work Items page
- Best tool to add, update, and link work items

Remove, delete, or restore work items in Azure Boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Work items can live forever in your work tracking data store. You never have to delete them. However, you might want to set up a work item management process for one of the following actions:

- **Change state**: Remove work items from appearing on backlogs and boards by changing the work item **State** to *Remove* or *Cut*. The state available to you is based on the workflow assigned to the work item type.
- **Delete**: Remove work items from backlogs, boards, and queries. Deleted work items are moved to a **Recycle Bin**.
- **Restore**: Recover deleted work items by restoring them from the **Recycle Bin**.
- **Destroy**: Permanently delete work items, which deletes all data from the work tracking data store.

> **NOTE**
> The ability to archive work items or projects isn't a supported feature at this time.

To move a work item from one project to another, or to change the work item type, see Move work items, change work item type.

> **NOTE**
> For information about the Azure Artifacts Recycle Bin, see Delete and recover packages.

Prerequisites

In general, members of the **Contributors** group can remove, delete, and restore work items. To permanently delete work items, you must be a member of the **Project Administrators** group, or be granted the required permission. Users with **Stakeholder** access can view the contents of the **Recycle Bin**, but can't restore or permanently delete items in the bin regardless of the permissions they're granted.

Task

Required permission(s)

Change state to **Remove** or **Cut**

- Have the **Area Path** permission set to **Allow**: **Edit work items in this node**
- By default, members of the **Contributors** group have this permission.

Delete work items and
Restore work items

- Have the project-level permission set to **Allow**: **Delete and restore work items**
- Have **Basic** access or higher.
- By default, members of the **Contributors** group have this permission.

Permanently delete or destroy work items from the command line

Be a member of the **Project Administrators** group.

Permanently delete or destroy work items

- Have the project-level permission set to **Allow**: **Permanently delete work items**
- By default, members of the **Project Administrators** group have this permission.

Delete or destroy work items from the command line

- Have the project-level permission set to **Allow**: **Permanently delete work items**
- By default, members of the **Project Administrators** group have this permission.

For a simplified view of permissions assigned to built-in groups, see Permissions and access.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to all work tracking features just like users with **Basic** access. For details, see Stakeholder access quick reference.

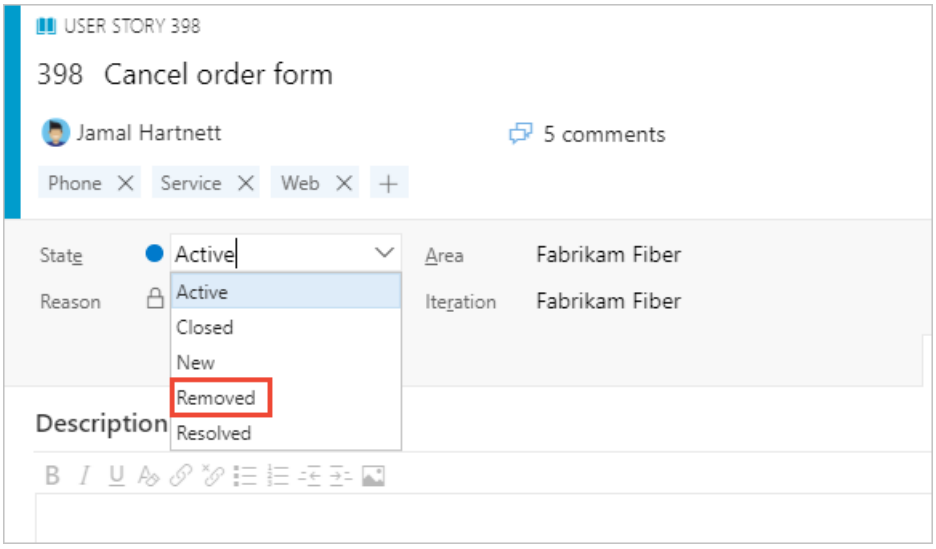
Remove or delete multiple work items

You can act on individual work items or bulk modify several work items.

From the web portal, you can multi-select several work items from a backlog or query results page. You can also do a bulk update using the associated feature. To delete or restore several work items at the same time, see Bulk modify work items.

Remove work items

By changing the **State** of a work item to *Removed*, you effectively remove it from a backlog or board view (product, portfolio, and sprint backlogs, Kanban board, and Taskboards). The *Removed* state corresponds to the **Removed** workflow category state. If you define custom workflow states, any state you map to the **Removed** workflow category state will act in a similar way.



To cause removed items to not show up in queries, you must add a clause that filters on the **State** field.

> **NOTE**
>
> The *Removed* state isn't supported with the Basic process. It is only supported with the Agile, Scrum, and CMMI process work item types. The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1.

Delete work items

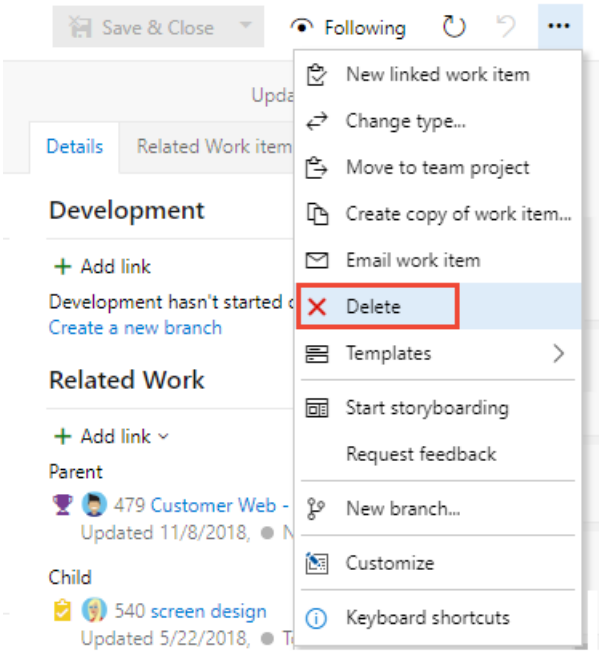
Deleted work items won't appear in your backlogs, boards, or queries. Deleted items are moved to a **Recycle Bin** from which you can recover them if needed. To delete a test case, test plan, or other test-related work item types, see Delete test artifacts.

You can delete work items in one of the following ways:

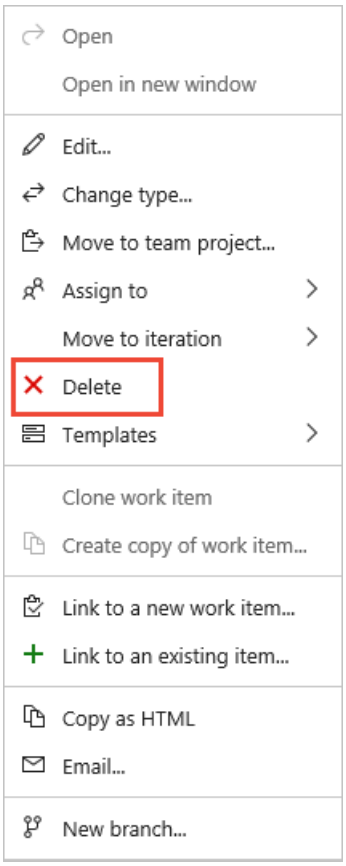
- From within the work item form
- From the **Work Items** page ··· **More Actions** menu
- From the Kanban board card ··· context menu
- From a backlog or query results page.

1. To initiate a delete operation:

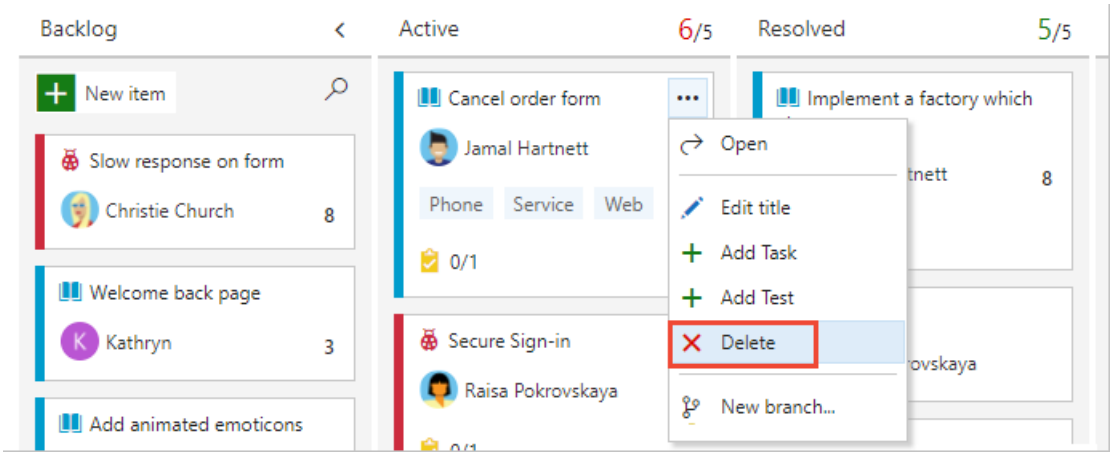
   From the work item form, open the work item, choose ··· **Actions**, and select **Delete**.



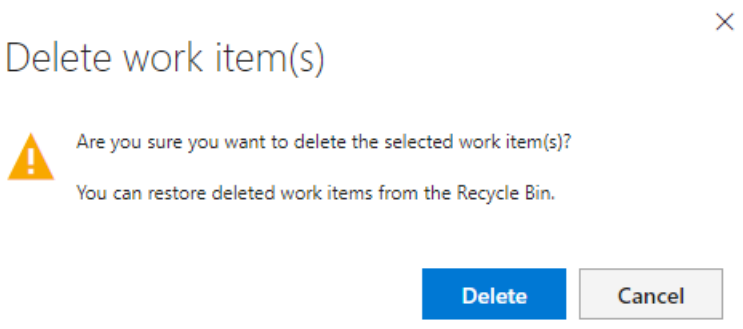
   To delete several work items, multi-select them from a backlog or a query results list, choose the ··· context menu, and then select **Delete**.



To delete a work item from your Kanban or taskboard, choose the ··· context menu for the card and select **Delete**.

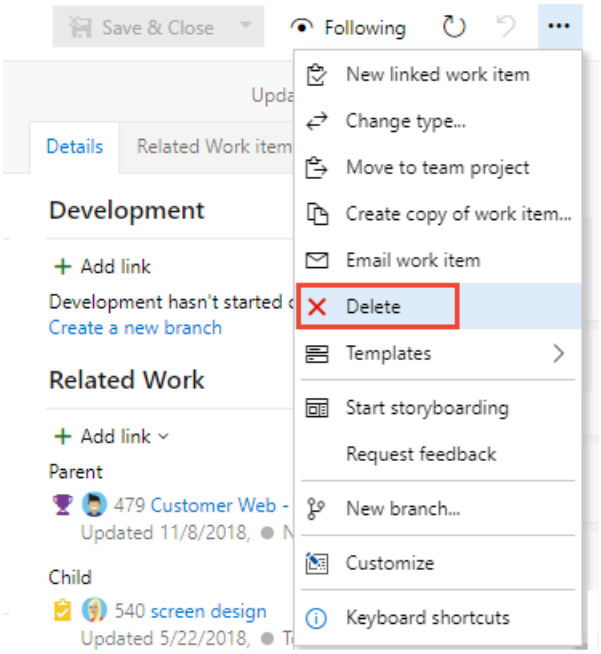


2. Confirm you want to actually delete the item(s).

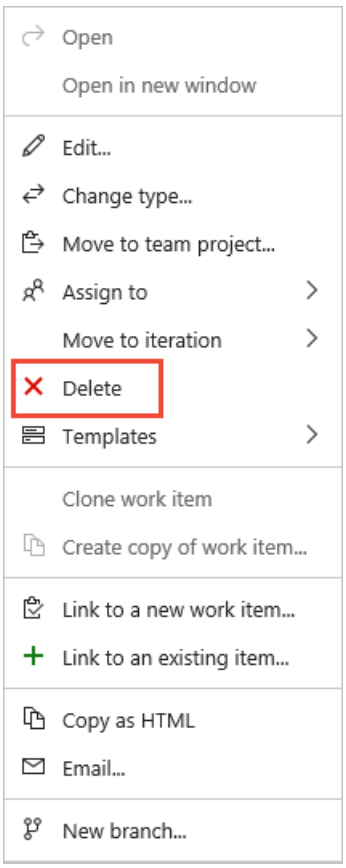


1. You can delete a work item from within the work item form, or by multi-selecting work items from a backlog or query results page.

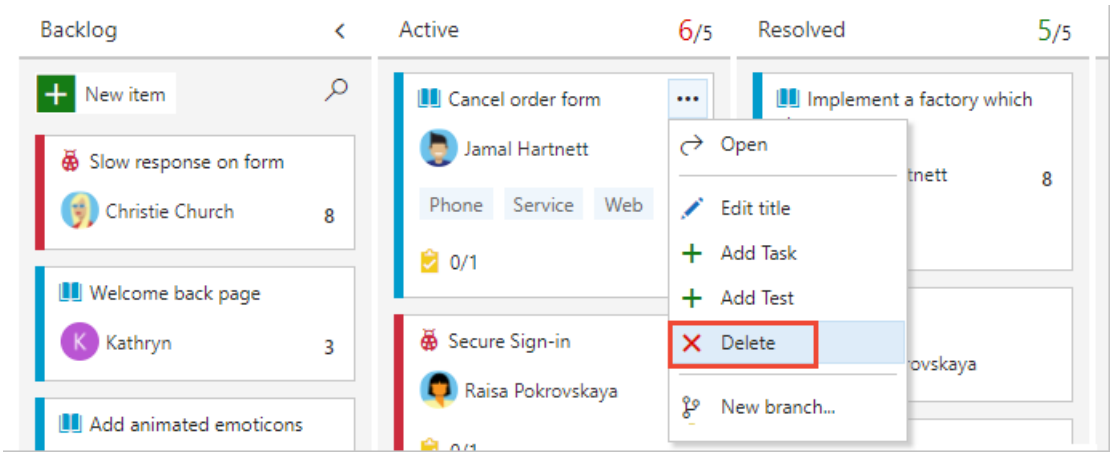
   To delete a single work item, open the work item, choose the ··· **Actions**, and select **Delete**.



To delete several work items, multi-select them from a backlog or a query results list. Then, choose the ··· actions icon and select **Delete**.

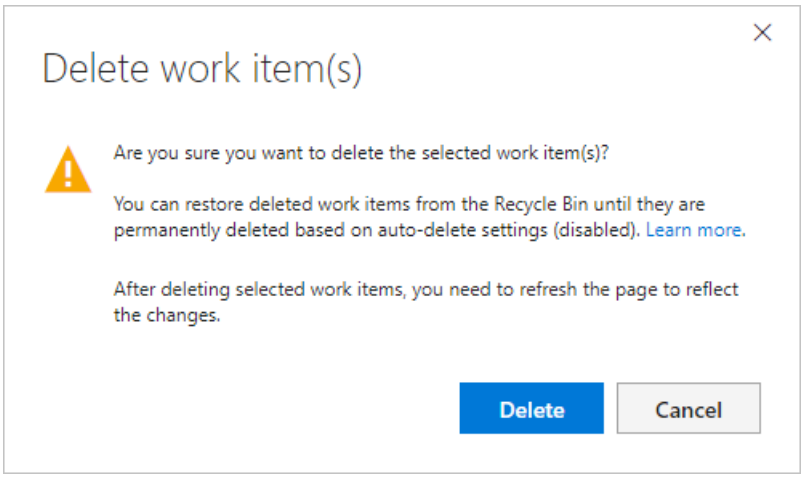


You can also delete work items from your Kanban or taskboard.



Or, you can drag them to the Recycle Bin **Recycle bin**. You can only access the **Recycle bin** from the **Work** hub.

2. Confirm you want to actually delete the item(s).



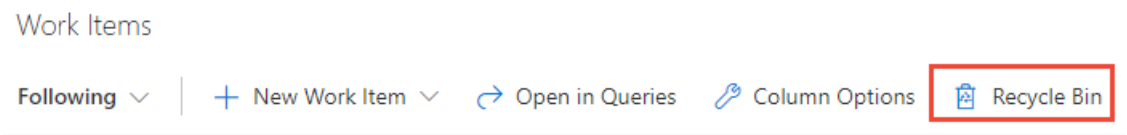
> **NOTE**
> The Delete work items confirmation dialog for on-premises Azure DevOps may indicate there are auto-delete settings (disabled). There are no settings you can enable or disable. There is only a background process which permanently deletes work items that have been set to delete.

Restore or destroy work items

You can't open work items that have been moved to the **Recycle Bin**. Also, you'll only see the **Permanently delete option** if your Permanently delete work items project-level permission is set to **Allow**.

You restore deleted work items or permanently delete them from the web portal **Recycle Bin**.

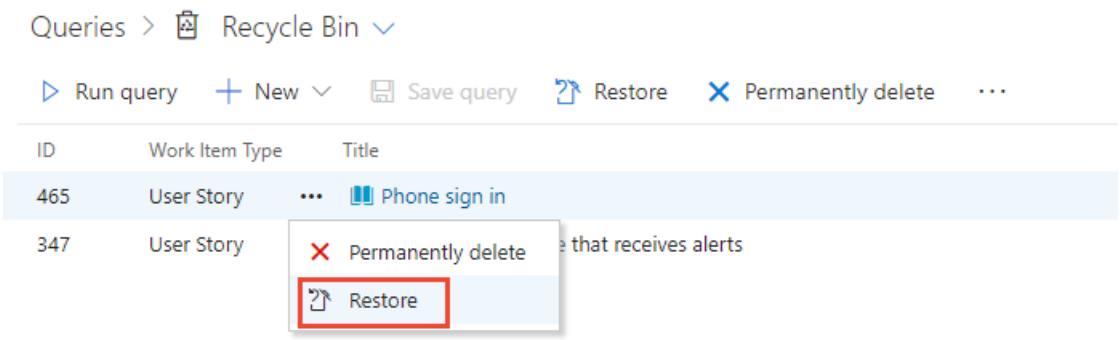
1. Choose **Boards>Work Items** and then choose the **Recycle Bin**.



If you don't see the **Recycle Bin** option, choose **More commands** ... and choose it from the menu of options.

2. A new browser tab opens with the query that lists work items added to the **Recycle Bin**.

3. Select the items you want to restore and then choose **Restore**.

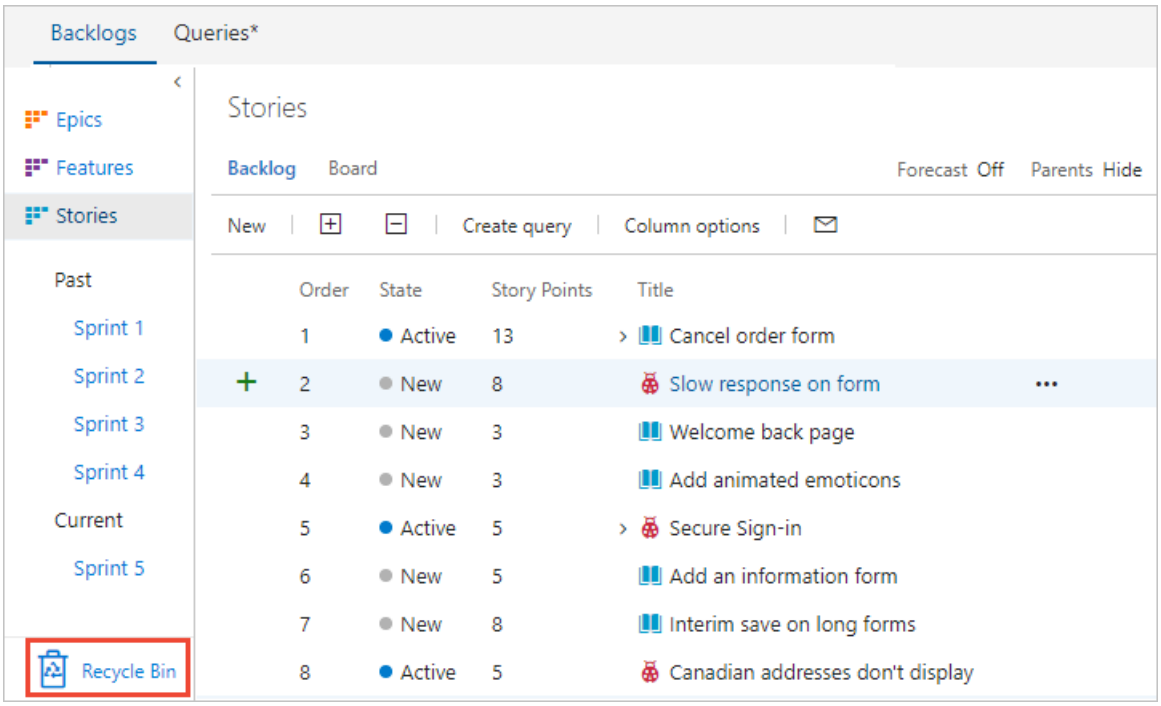


Optionally, you can choose to permanently delete the items.

4. Confirm your selection.

You restore deleted work items from the web portal **Recycle Bin**.

1. Choose **Work > Backlogs** or **Work > Queries** and then choose the **Recycle Bin**.



A new browser tab opens with the query that lists work items added to the **Recycle Bin**.

2. Select the items you want to restore and then choose **Restore**.

   ![Screenshot of Restore selected items, TFS 2018 version.

   Optionally, you can choose to permanently delete the items.

3. Confirm your selection.

> **NOTE**
>
> Deleted test artifacts won't appear in the **Recycle Bin** and can't be restored. Deletion of test artifacts deletes the selected test artifact and all of its associated child items, such as child test suites, test points across all configurations, testers (the underlying test case work item doesn't get deleted), test results history, and other associated history.

Delete or destroy work items from the command line

You can delete or destroy a work item with the az boards work-item delete command. To get started, see Get started with Azure DevOps CLI.

> **NOTE**
> You can restore work items you **delete**, but you can't restore work items you choose to **destroy**.

```
az boards work-item delete --id
                           [--destroy]
                           [--org]
                           [--project]
                           [--yes]
```

Parameters

- **id**: Required. The ID of the work item.
- **destroy**: Optional. Permanently delete this work item.
- **org**: Azure DevOps organization URL. You can configure the default organization using `az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using `git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.
- **project**: Name or ID of the project. You can configure the default project using `az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up using `git config`.
- **yes**: Optional. Don't prompt for confirmation.

Example

The following command permanently deletes the bug with the ID 864 and doesn't prompt you for confirmation.

```
az boards work-item delete --id 864 --destroy --yes
```

Destroy work items from the command line

Use the `witadmin destroywi` command to permanently remove work items from the data store. A permanent delete means all information in the work tracking data store is deleted and can't be restored nor reactivated.

> **NOTE**
> Deleting work items from the `witadmin` command line is deprecated for TFS 2018.2 and later versions, and not supported for Azure Boards cloud service.

Open a Command Prompt window where the latest version of Visual Studio is installed and change the directory to where the `witadmin.exe` tool has been installed.

For example, you would change to the following directory for TFS 2018. (For other versions, see Remove work items permanently (witadmin destroywi)).

```
%programfiles(x86)%\Microsoft Visual
Studio\2018\Professional\Common7\IDE\CommonExtensions\Microsoft\TeamFoundation\Team Explorer
```

On a 32-bit edition of Windows, replace %programfiles(x86)% with %programfiles%.

The `witadmin` command-line tool installs with any version of Visual Studio or Team Explorer. You can access this

tool by installing the free version of Visual Studio Community.

- To delete several work items, enter the server name and directory path to the collection. For example:

```
witadmin destroywi /collection:http://TFSServerName:8080/tfs/DefaultCollection /id:12,15,23
```

- To delete a single work item, enter the ID as shown:

```
witadmin destroywi /collection:http://TFSServerName:8080/tfs/DefaultCollection /id:2003
```

Delete and restore processes

When you delete a work item, the following actions occur:

- Generates a new revision of the work item
- Updates the **Changed By/Changed Date** fields to support traceability
- Preserves the work item completely, including all field assignments, attachments, tags, and links
- Causes the work item to become non-queryable and, as such, won't appear in any work tracking experience, query result, or report
- Updates charts correctly. The CFD, velocity, burndown, and lightweight charts are updated to remove deleted work items
- Removes work tracking extensions
- Preserves trend data except for the latest value
- Removes the work item from the data warehouse/cube similar to as if it was permanently removed.

When you restore a work item, the following actions occur:

- Causes a new revision of the work item to be made
- Updates the **Changed By/Changed Date** fields to support traceability
- Becomes queryable
- All fields remain unchanged
- History contains two new revisions, one for deletion, and one for restore
- Reattaches work tracking extensions
- Updates charts correctly. The CFD, velocity, burndown, and lightweight charts are updated to include the restored work items
- Restores trend data
- Adds the work item back to the data warehouse/cube
- Sets the area or iteration path fields to the root node if the previous area path or iteration paths were deleted.

Use a REST API to delete, restore, and destroy work items

To programmatically delete, restore, and destroy work items, see one of the following REST API resources:

- Recycle bin REST API Reference
- Work Items - Delete REST API Reference

Related articles

- Best tool to add, update, and link work items
- View and add work items using the Work Items page
- Delete test artifacts

- Set permissions and access for work tracking
- Change project-level permissions
- Stakeholder access quick reference

Delete test artifacts in Azure Boards

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

While test artifacts such as test plans, test suites, test cases, and so on, are all types of work items, the method for deleting them differs from deleting non-test work items.

> **IMPORTANT**
> We only support permanent deletion of test artifacts such as test plans, test suites, test cases, shared steps and shared parameters. Deleted test artifacts won't appear in the recycle bin and cannot be restored. Deletion of test artifacts not only deletes the selected test artifact but also all its associated child items such as child test suites, test points across all configurations, testers (the underlying test case work item doesn't get deleted), test results history, and other associated history.

When you delete test artifacts, the following actions occur:

1. Removes the deleted test artifact from the test case management (TCM) data store and deletes the underlying work item
2. Runs a job to delete all the child items both from the TCM side and the underlying work items. This action may take time (up to a few minutes) depending on the number of artifacts to be deleted.
3. Causes all information in the work item tracking data store and TCM data store to be deleted and cannot be reactivated nor restored.

Prerequisites

- To delete test runs, you must be a member of the Project Administrators group or have the project-level **Delete test runs** permission set to **Allow**.
- To delete test plans and test suites, you must be a member of the Project Administrators group or have the Area Path node-level **Manage test plans** or **Manage test suites** permission set to **Allow**.
- To manage or delete test artifacts, you must also have your access level set to **Basic + Test Plans** or **Visual Studio Enterprise**. This level provides access to the full Test Plans feature set. Users with Basic access and with permissions to permanently delete work items and manage test artifacts can only delete orphaned test cases. That is, they can delete test cases created from **Work** that aren't linked to any test plans or test suites.

- To delete test runs, you must be a member of the Project Administrators group or have the project-level **Delete test runs** permission set to **Allow**.
- To delete test plans and test suites, you must be a member of the Project Administrators group or have the Area Path node-level **Manage test plans** or **Manage test suites** permission set to **Allow**.
- You must also have your access level set to Basic+Test Plans or Advanced, which provides access to the full Test feature set. Users with Basic access and with permissions to permanently delete work items and manage test artifacts can only delete orphaned test cases. That is, they can delete test cases created from **Work** that aren't linked to any test plans or test suites.

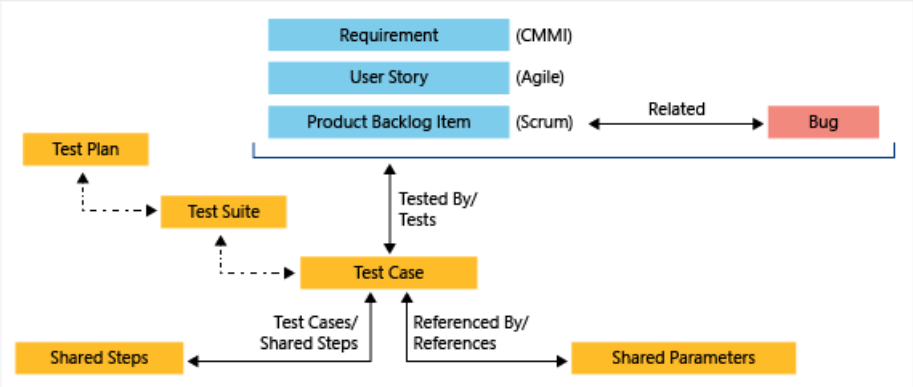
To delete test artifacts, the following restrictions and operations apply:

- Users with Basic access and with permissions to permanently delete work items and manage test artifacts can only delete orphaned test cases. That is, they can delete test cases created from **Work** that aren't linked to any test plans or test suites.

- When you delete a test plan, test suite, test case, shared steps, or shared parameters, you not only permanently delete them, you also delete all associated test artifacts such as test results.
- You can't bulk delete test artifacts. If test artifacts are part of a bulk selection to be deleted, all other work items except the test artifact(s) will get deleted.

Work item types that support the test experience

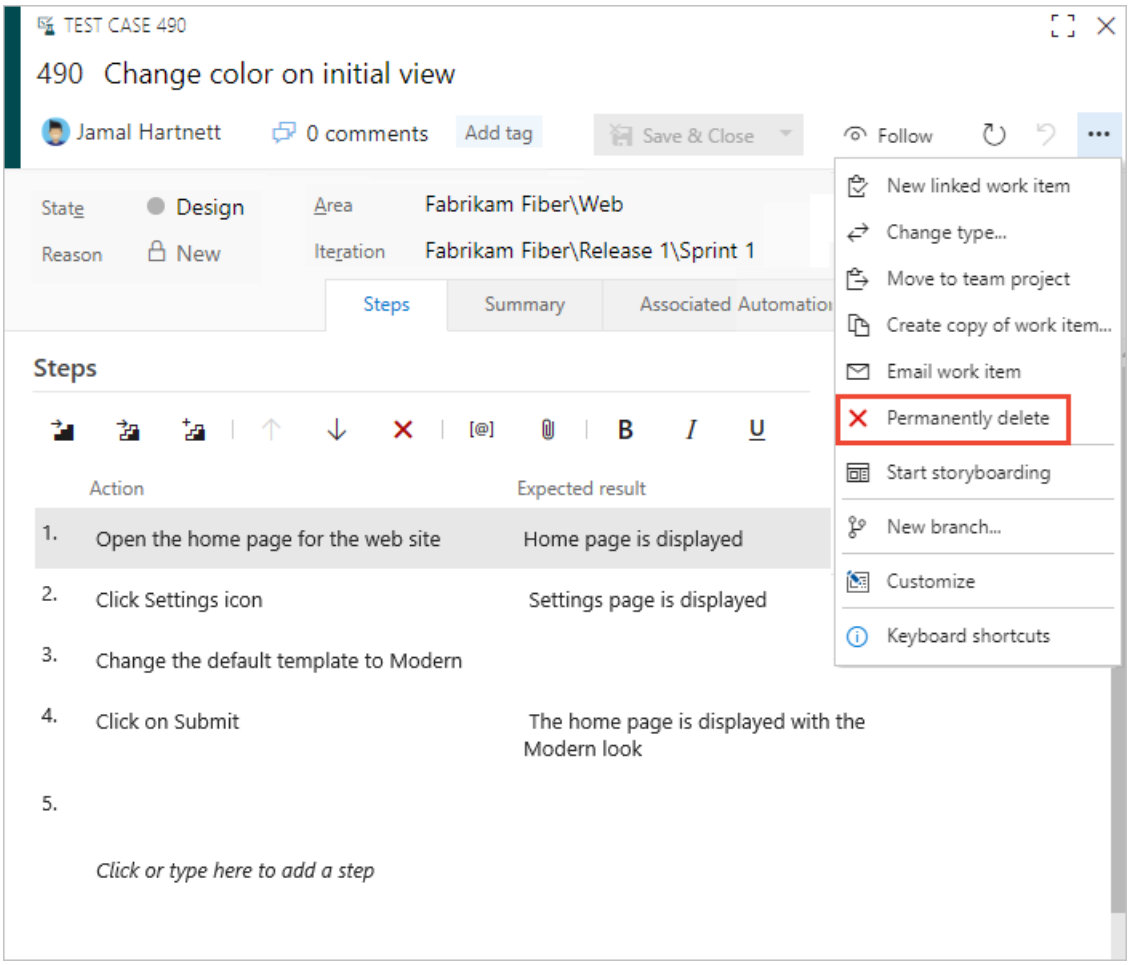
The following image illustrates the set of work item types that support the test experience and work with Microsoft Test Manager. These work item types are linked together using the link types shown.



From the web portal or Microsoft Test Manager, you can view which test cases are defined for a test suite, and which test suites are defined for a test plan. However, these objects aren't connected to each other through link types. For definitions of each field used in these work item types, see Query based on build and test integration fields.

Delete a test case, test suite, or test plan

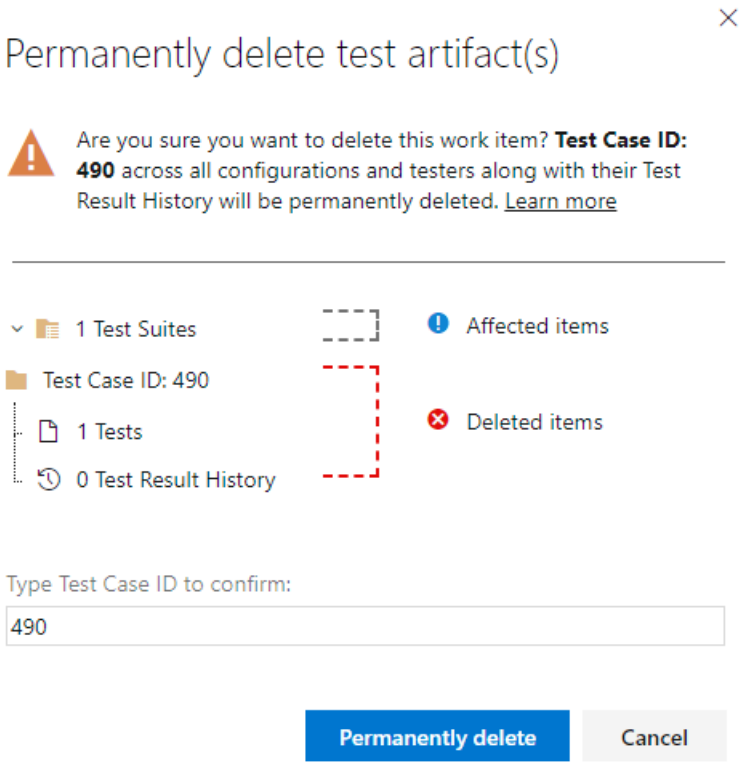
1. To delete a test case, test suite, or test plan, open it from the web portal and choose the **Permanently delete** option from the actions menu. (Bulk deletion isn't supported from a query results page.)



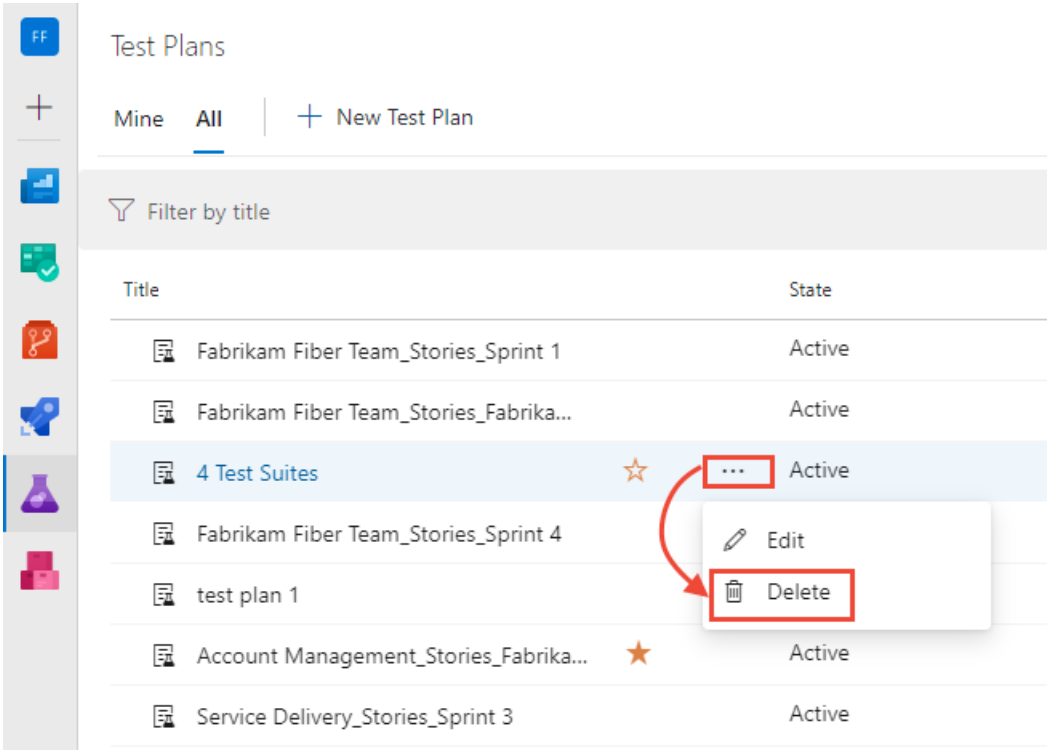
NOTE

You'll only see the **Permanently delete** option if you have the necessary permissions and access.

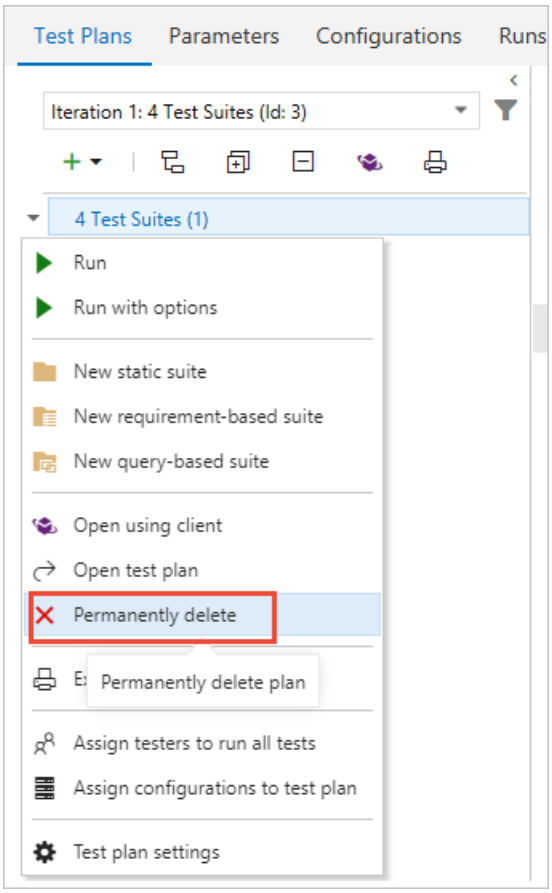
2. Confirm you want to actually delete the item.



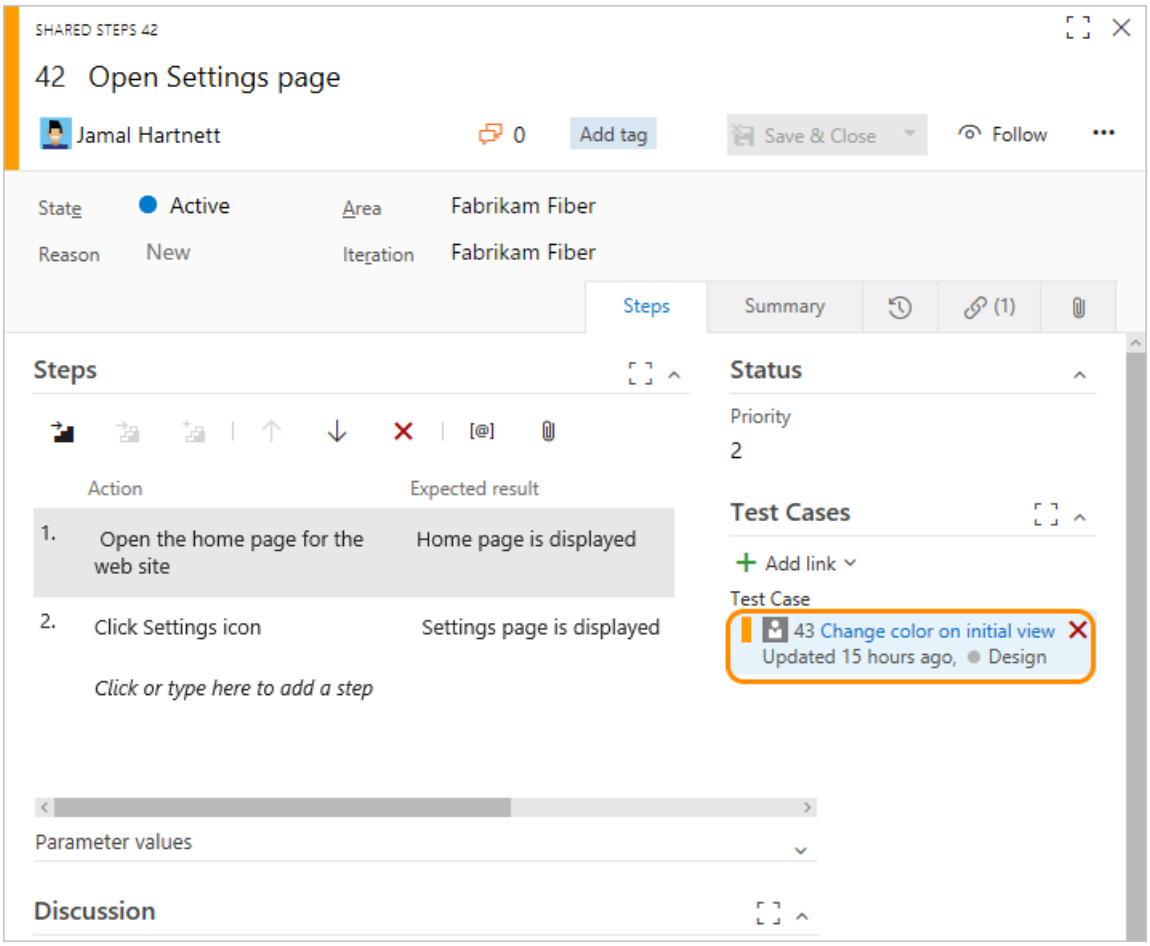
3. You can also delete a test plan directly from **Test Plans**. To delete a test plan, open **Test Plans** and choose ··· **More Actions** for the plan you want to delete, and choose **Delete**.



3. You can also delete a test plan directly from **Test** or **Test Plans**.



4. To delete shared steps and shared parameters, you need to first manually remove all references to them before you can delete them.



Related articles

- Create a test plan
- Control how long to keep test results
- Set permissions and access for work tracking, Manage test artifacts

Use templates to add and update work items in Azure Boards and Visual Studio

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015

With work item templates, you can quickly create work items that have pre-populated values for your team's commonly used fields. For example, create a template that defines the area path, iteration path, and activity to use when creating a task.

You can use work item templates to create work items or bulk update several work items. For examples that show usage of work item templates, see Sample work item templates.

> **NOTE**
> Work item templates are distinct from process templates. For information on process templates, see Choose a process template or these specific topics for the default process templates: Basic, Agile, Scrum, or CMMI.

Supported template tasks

The availability of template task options depends on the client and platform version available to you. Add and manage work item templates from the web portal or from Visual Studio 2015 or earlier versions. Visual Studio 2015 and earlier versions support working with templates by installing the Microsoft Visual Studio Team Foundation Server 2015 Power Tools. These templates only appear in your view of Team Explorer.

As shown in the following table, a ✔ indicates the task is available from the web portal or from Visual Studio 2015 or earlier versions. *(Make sure that you've selected the content version based on your platform version)*.

Task	Web portal	Visual Studio 2015
Capture a work item as a template	✔	✔
Manage work item templates (Define, edit, delete, copy link, create copy, and rename)	✔	✔
Copy link (URL) of a template	✔	

✔

Add a work item using a template

✔

✔

Define a work item template

✔

✔

Apply a template to one or more work items

✔

✔

Add or remove tags from templates

✔

Define a template using a hyperlink

✔

TIP

The templates you define through the web portal are distinct from those you define through Visual Studio. Web portal templates can only be managed and applied to work items through the web portal. Similarly, Visual Studio templates can only be managed, viewed, and applied to work items in Visual Studio. However, you can use the URLs of both template types to add work items through the web portal.

Prerequisites

- To add, capture, edit, or delete work item templates through the web portal, you must be a member of the team under which you add them.
- To apply a work item template, you must be a Contributor of the project and a member of the team under which the work item template is defined.
- To add, capture, edit, or delete work item templates through the web portal, you must be a team administrator.
- To apply a team template, you must be a Contributor of the project and a member of the team under which the work item template is defined.
- To add, capture, or edit work item templates through Visual Studio Team Explorer, install the Microsoft Visual Studio Team Foundation Server 2015 Power Tools. These templates only appear in your view of Team Explorer.

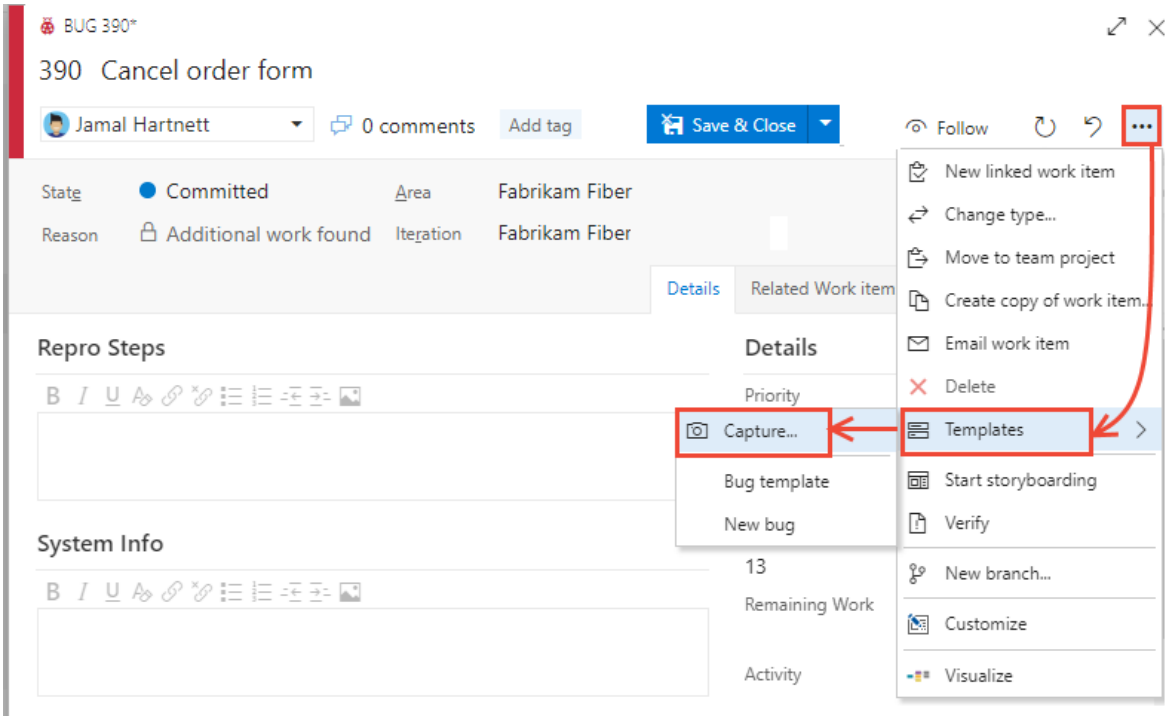
Capture a work item as a template

The steps to capture a work item differ based on the platform, version, and client you use.

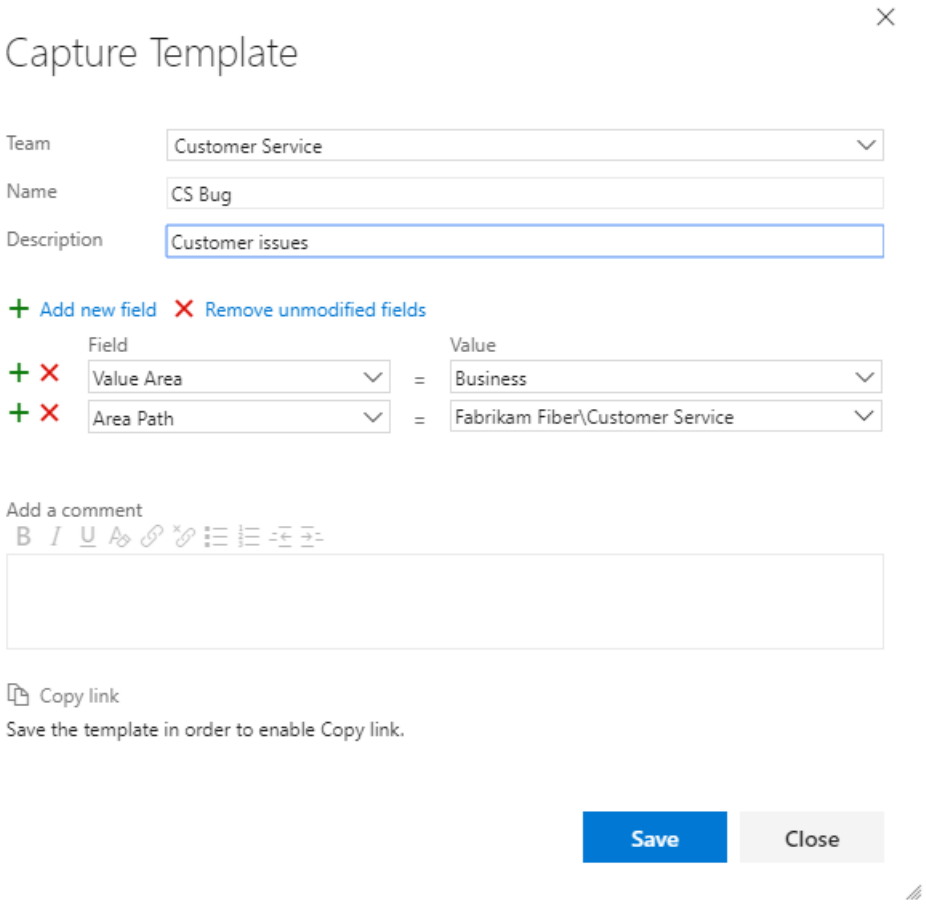
- Web portal
- Visual Studio 2015

Templates captured through the web portal are assigned a GUID.

1. From the web portal, open a work item that you'll use as the basis for a template.
2. Choose the ··· actions icon to open the menu. Choose **Templates** and then **Capture**.



Name the template, select the team for which you want to save it under, and optionally define or clear fields. Save the template when finished.



3. Once you've saved the template, choose **Copy link** to capture the URL for the template.

4. You can paste the URL link into a browser to create a work item, or provide it to others for their use to add work items. For example, you can add it as a hyperlink to a project wiki, a dashboard via a Markdown widget, or other shared network resource.

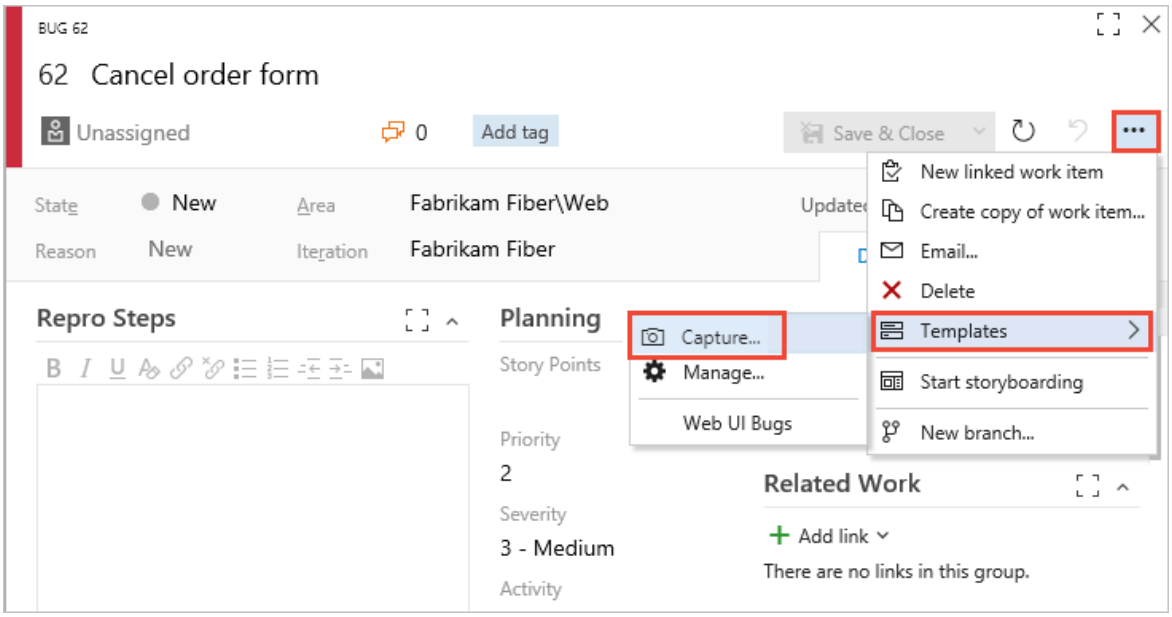
   Use the URL whenever you want to add a work item of the type you've defined with its predefined values.

Templates captured through the web portal are assigned a GUID.

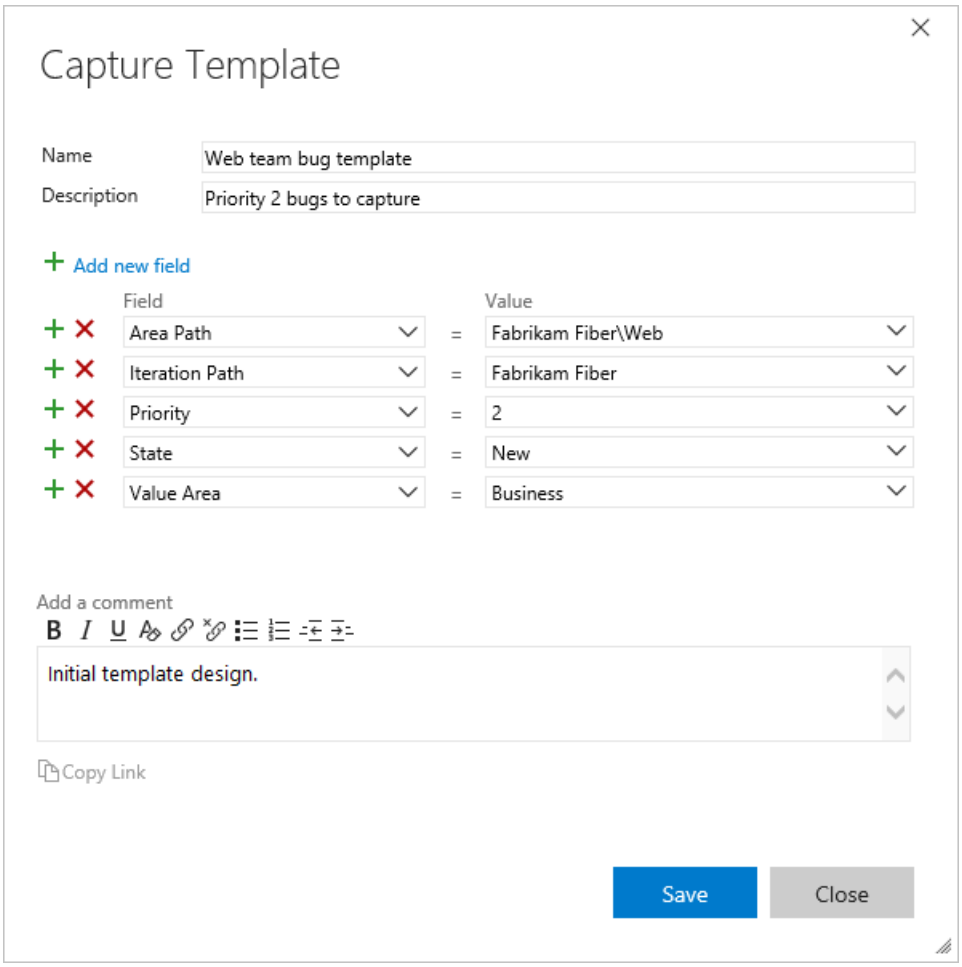
1. From the web portal, open a work item that you'll use as the basis for a template.

   Within the web portal, work item templates are associated with a team. Only those templates that are defined for a team are accessible when you go to apply a template to work items using the web portal.

2. Choose the ··· actions icon to open the menu. Choose **Templates** and then **Capture**.



   Name the template and optionally define or clear fields. Save the template when finished. The template is saved under the team you selected in the first step.



3. Once you've saved the template, choose **Copy link** to capture the URL for the template that you can use to add work items using the template.
4. Use the URL whenever you want to add a work item of the type you've defined with its predefined values.

   You can save the URL as a text file or add the URL to a dashboard or web page as a hyperlink.

Manage work item templates

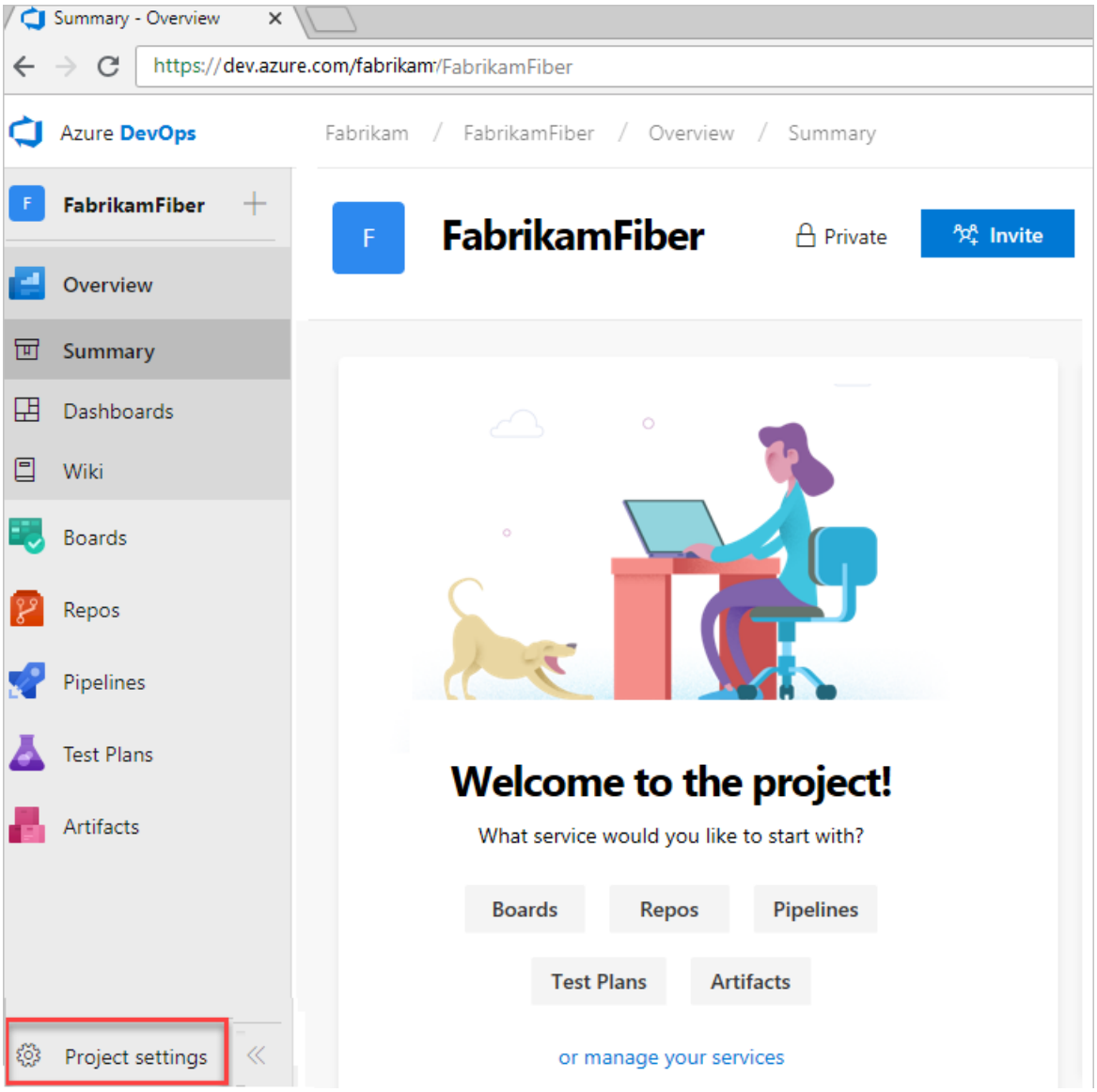
You can view the list of templates defined for each work item type, and also add, edit, copy, delete, rename, and copy the link of a template. All templates are defined and managed for a team.

- Web portal
- Visual Studio 2015

You must be a member of the team to add or edit team templates.

1. From the web portal, open settings for a team.

   Choose **Project Settings**.



Expand **Boards** and choose **Team configuration**. If you need to switch to a different team, use the team selector.

2. Choose **Templates**.

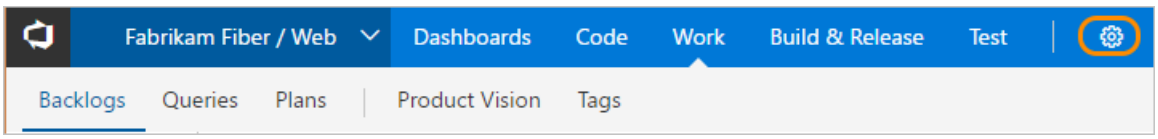
   From here, you can choose any work item type to view or add templates for that type.

You manage templates from team settings. All templates are defined for a team. If you're not a team administrator, get added as one. Only team or project administrators can change work item templates.

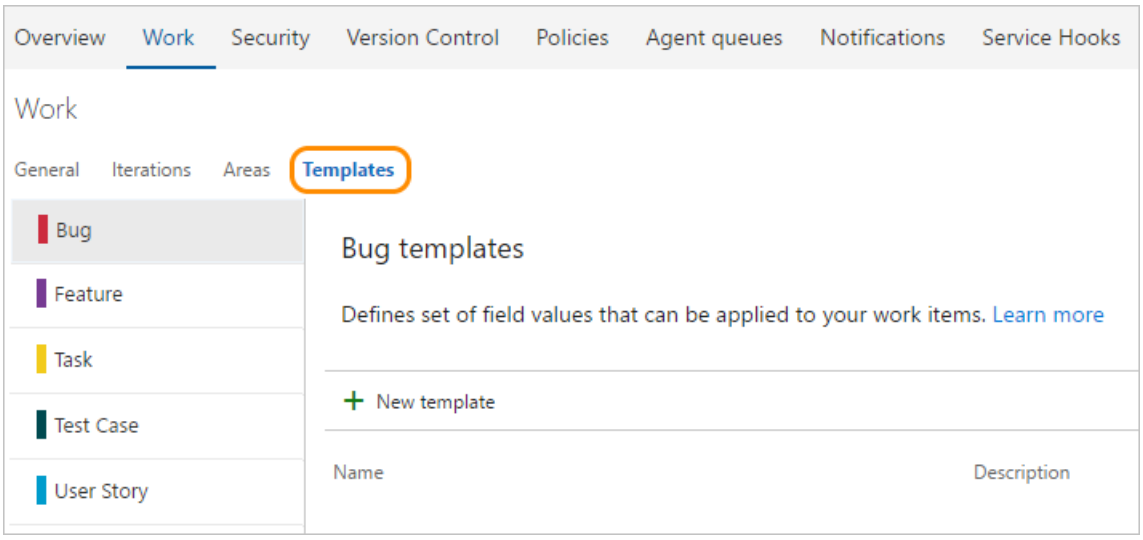
1. From the web portal, open settings for a team.

   Choose the ⚙ gear icon to open the settings for a team.

   Here we open the admin page for the Web team.



2. Choose **Work > Templates**.

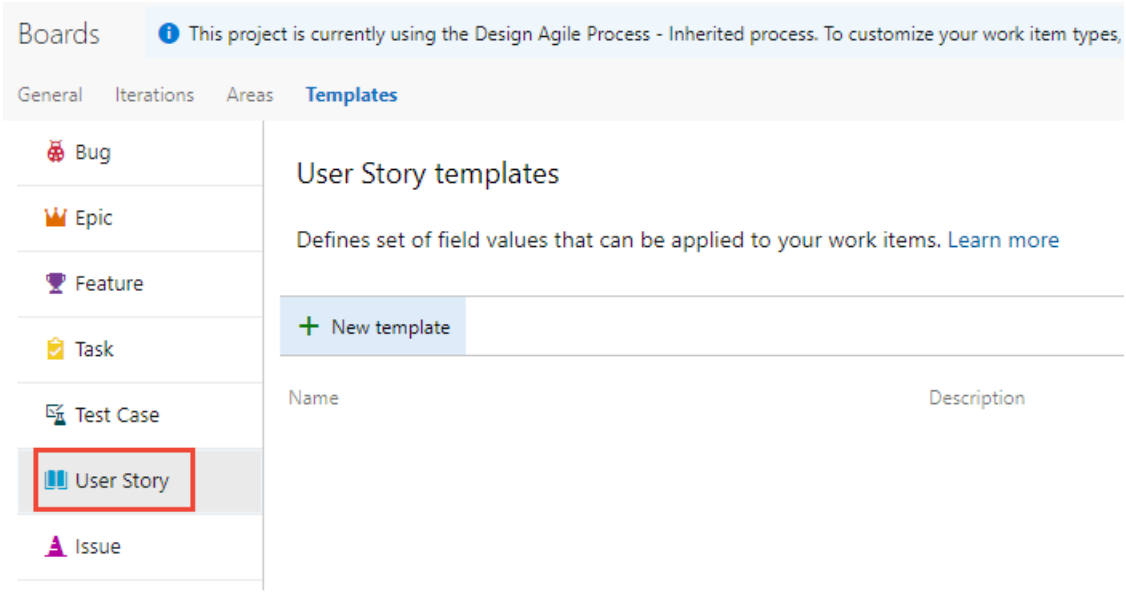


From here, you can select any work item type to view or add templates for that type.

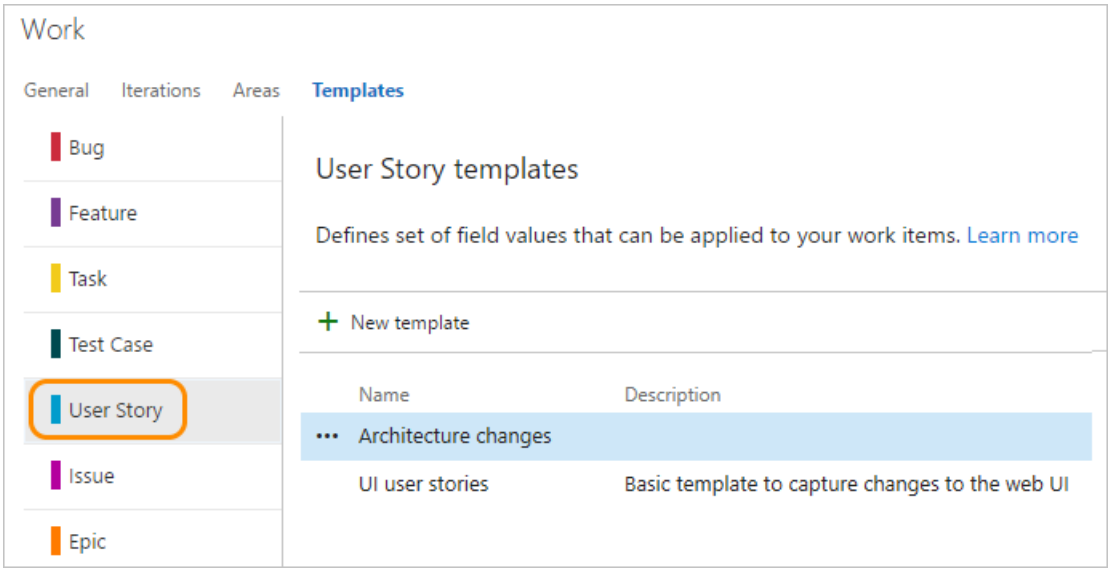
Manage templates for a work item type

- Choose the work item type to view the templates defined for each type.

  For example, choose User Story to view templates defined to capture user stories.

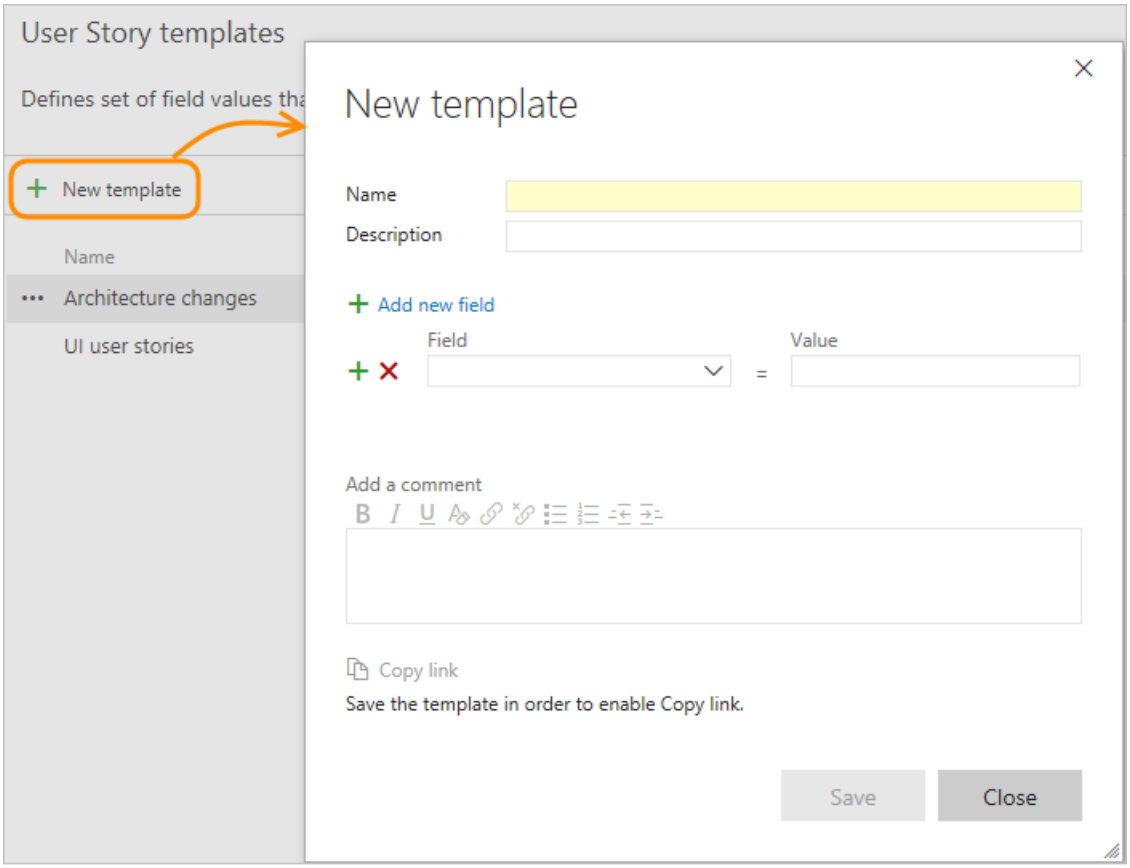


- Choose the work item type to view the templates defined for each type. For example, choose User Story to view templates defined to capture user stories.



Create a work item template

1. From the work item type page, choose the **New template** to create a template from scratch.

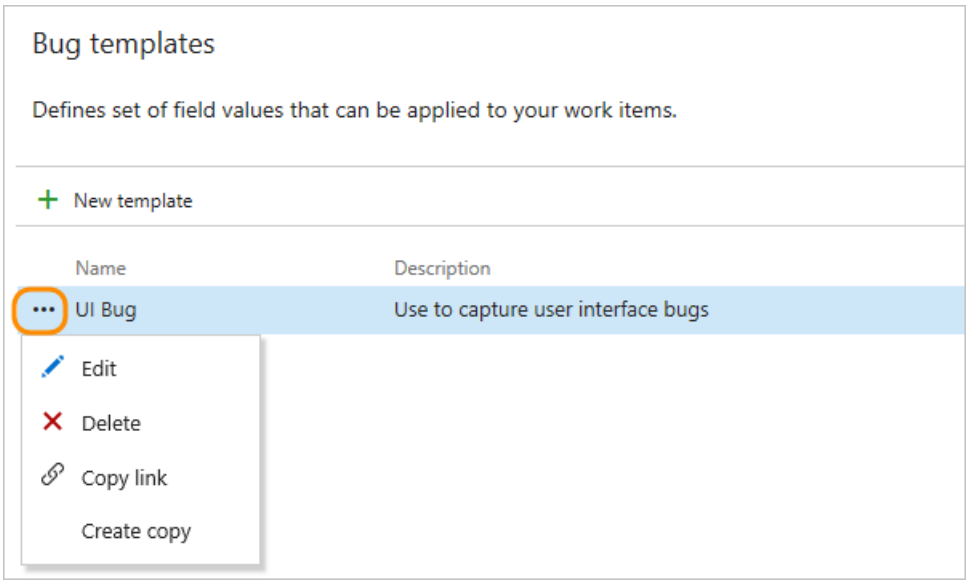


2. Name the template and optionally add and remove fields. Save the template when finished.

   Once you've saved the template, select Copy link to capture the URL for the template that you can use to add work items using the template.

Edit, rename, or copy a link to a template

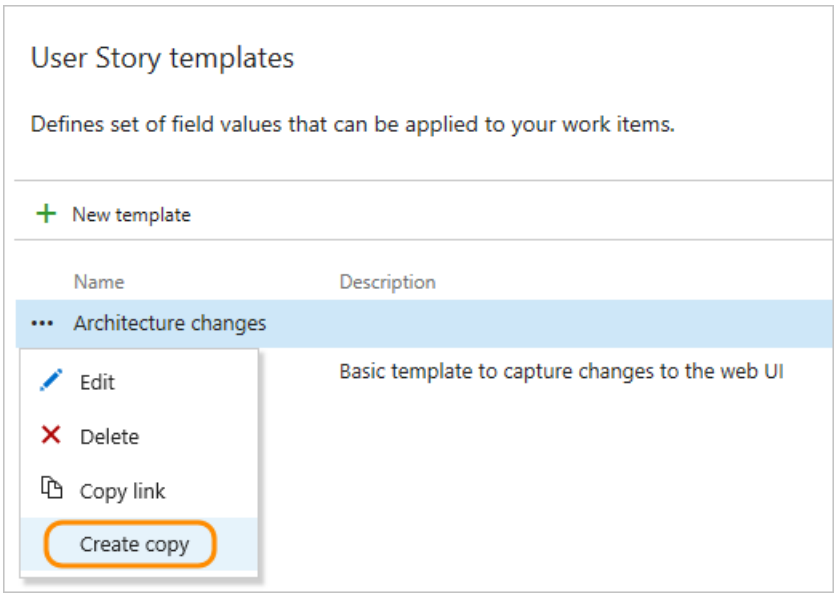
1. From the work item type page, choose ··· the actions icon for an existing template to access the menu options to **Edit**, **Delete**, or **Copy link**.



2. To rename a template, choose to edit the template, change the name, and then save your changes.

Create a copy of a template

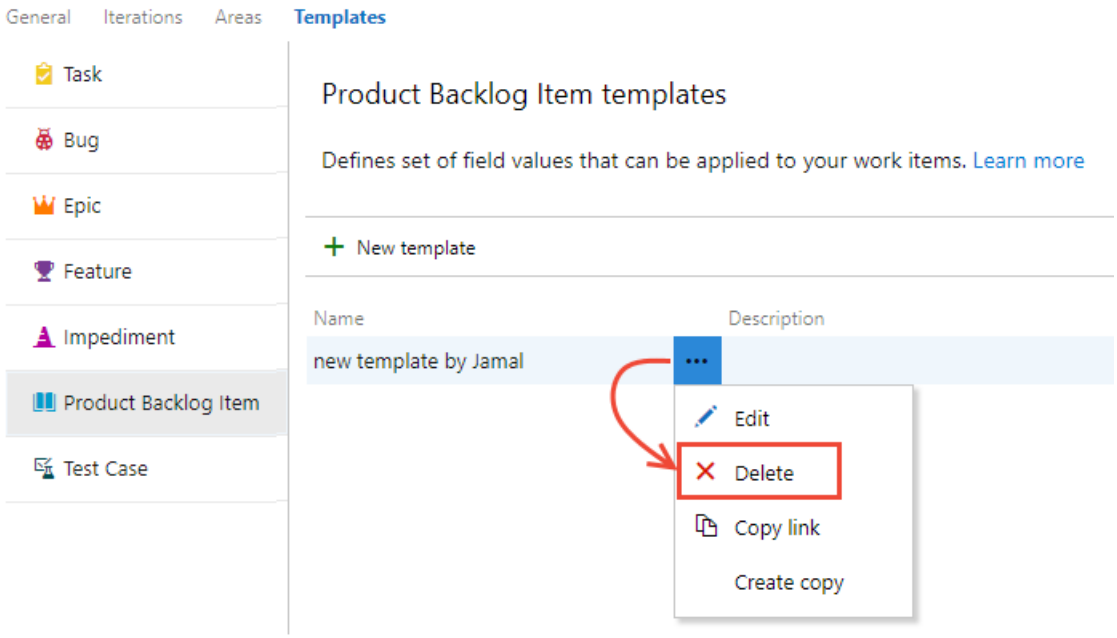
1. To duplicate an existing template, choose ··· the actions icon for an existing template and select the **Create copy** option.



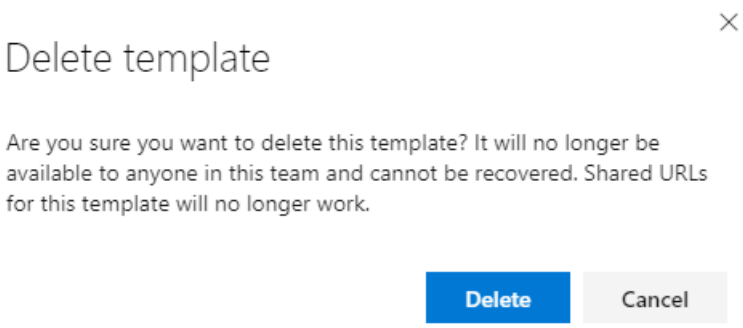
2. Name the template and optionally add and remove fields. Save the template when finished.

Delete a template

1. From the web portal, open **Project Settings > Team configuration > Templates**.
2. Choose the ··· actions icon to open the menu, and choose **Delete**.



3. Choose **Delete** from the Delete template confirmation dialog that displays.



As indicated in the confirmation dialog, you can't recover a template once it's deleted.

Add a work item using a template

The main method for adding a work item using a template is to open the template link within a browser window. To get the template link, see the next section Copy the link to a template.

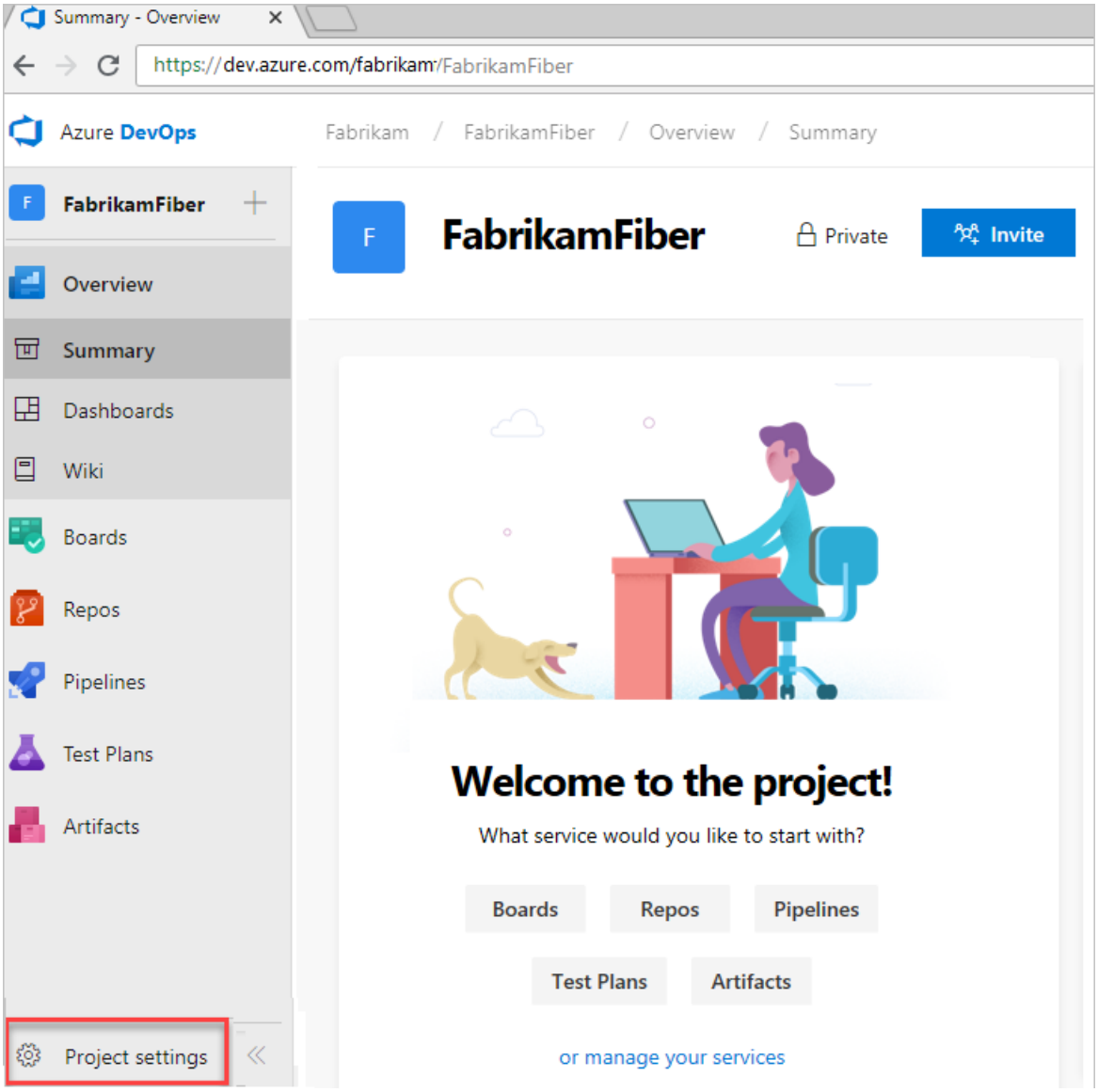
You can share these links through email, a network share, or a team dashboard.

Copy the link to a template

Use the URL whenever you want to add a work item of the type you've defined with its predefined values. Copy the link to a shared network or send to your team via email. Also, consider adding a link to the team dashboard.

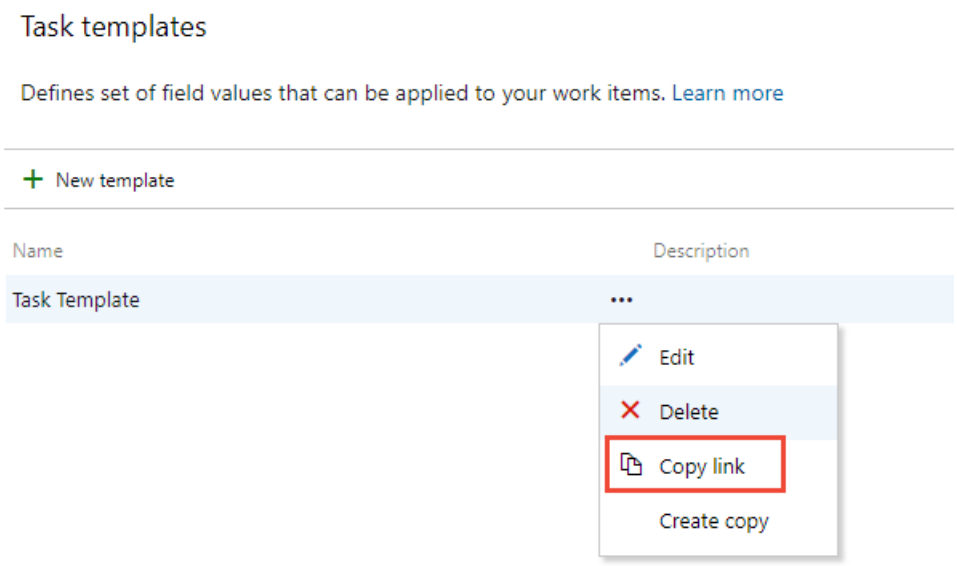
- Web portal
- Visual Studio 2015

1. Go to **Project Settings**.



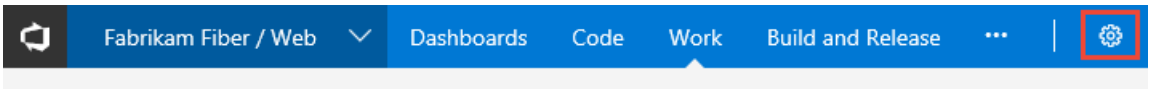
   Expand **Boards** and choose **Team configuration**. Then, choose **Templates**.

2. Choose the work item type whose template you want to copy.
3. Choose the ··· actions icon to open the menu, and choose **Copy link**.

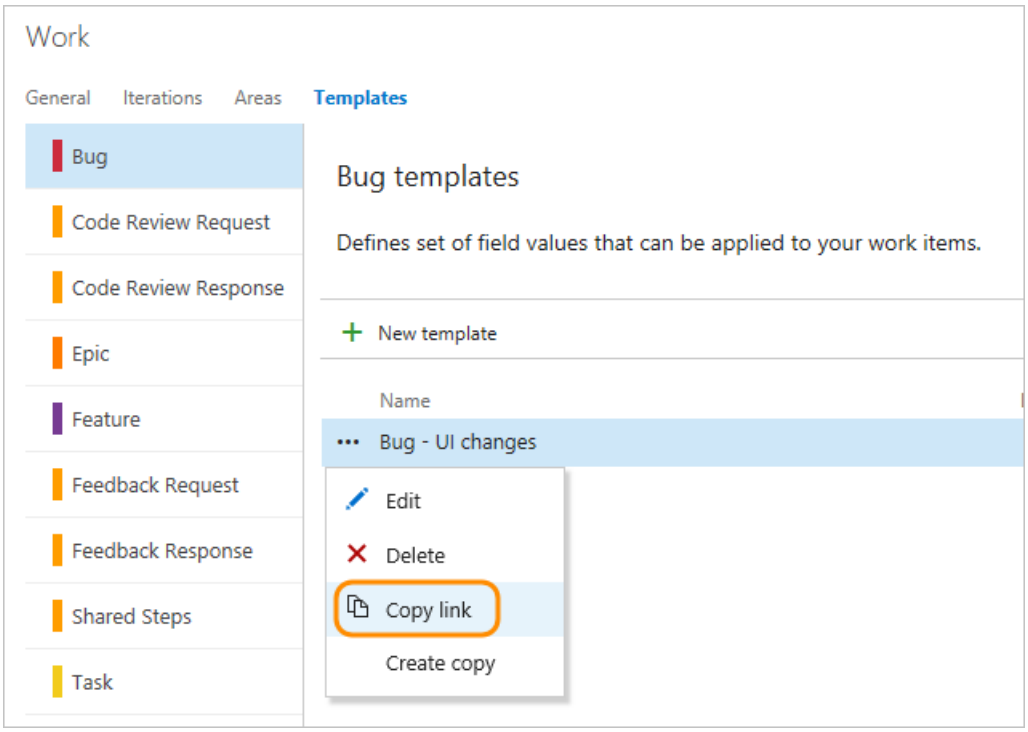


You can save the URL as a text file or add the URL to a web page as a hyperlink.

1. Choose the gear icon to open the settings for a team.



2. Choose the **Work > Templates** tab. Then, choose the ··· actions icon for the template you want to copy and select **Copy link**.



You can save the URL as a text file or add the URL to a web page as a hyperlink.

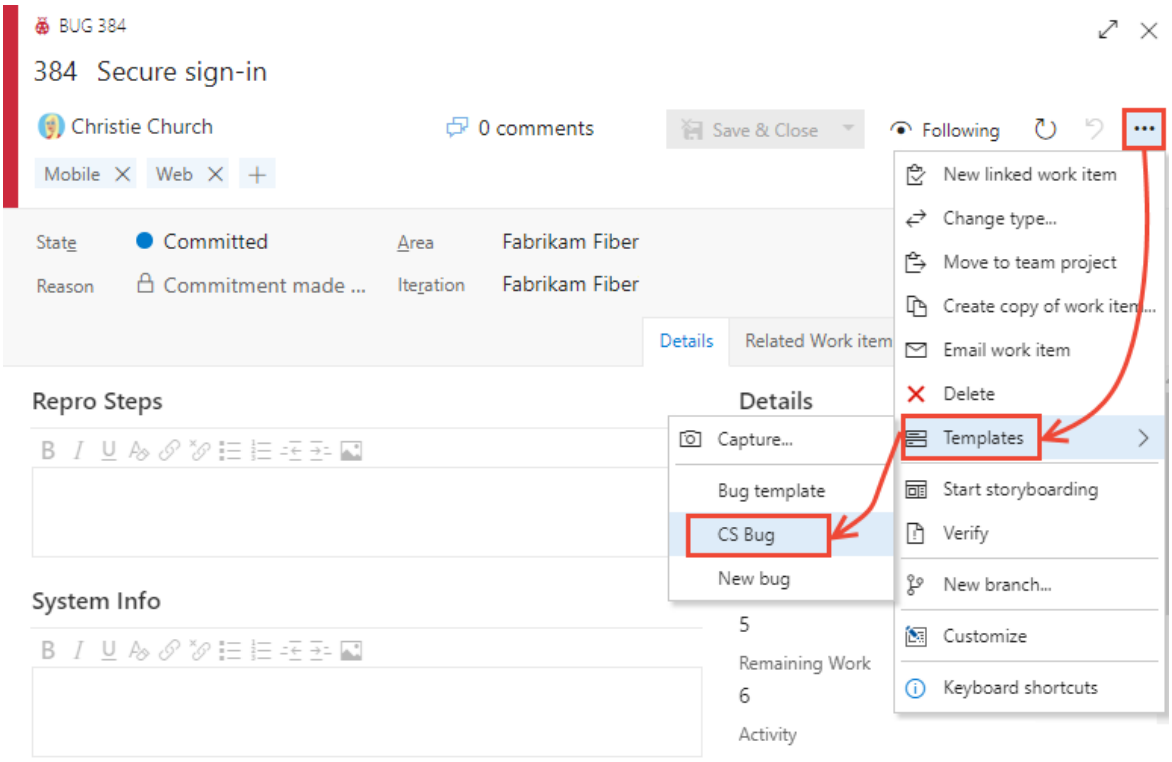
Apply a template to new or existing work item(s)

You can apply a template to a single work item or do a bulk update of several work items.

- Web portal
- Visual Studio 2015

Apply a template within a work item

1. Open a new work item or an existing work item that you want to update using the fields defined within a template, choose the ··· actions icon to open the menu, select **Templates**, and then select the name of a predefined template.



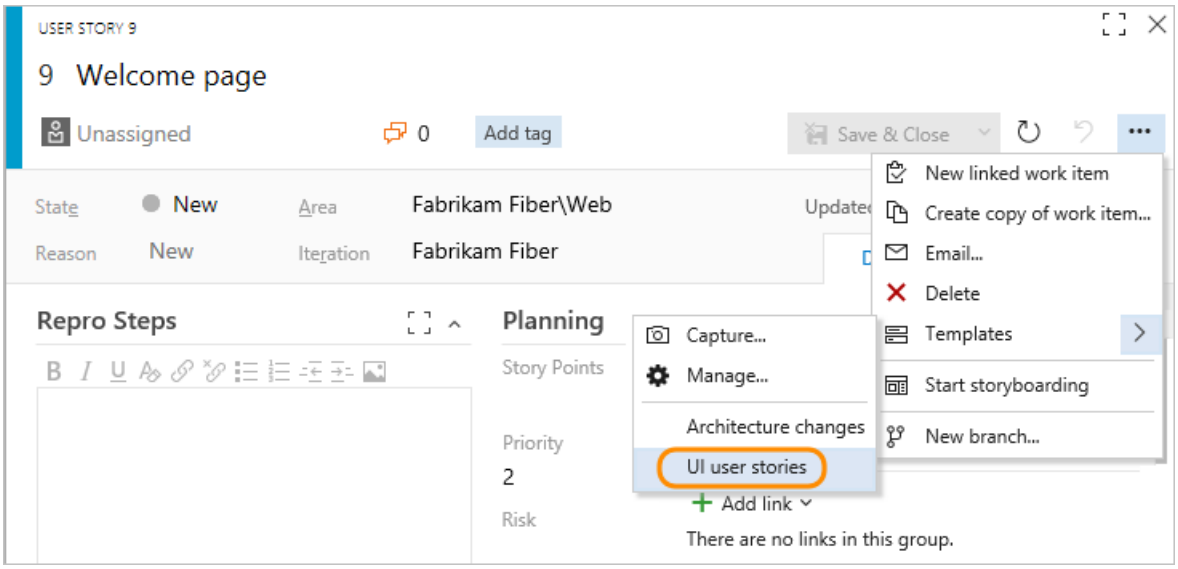
Only those templates defined for teams that you belong to appear.

> **TIP**
> Refresh your browser to discover the latest templates that have been added. If you don't see any templates, it may be that there are none defined for the work item type.

2. Save the work item for the changes to be applied. The fields changed are noted in the History field.

Apply a template within a work item

1. Open the work item that you want to update using the fields defined within a template, choose the ··· actions icon to open the menu, select **Templates**, and then select the name of a predefined template.



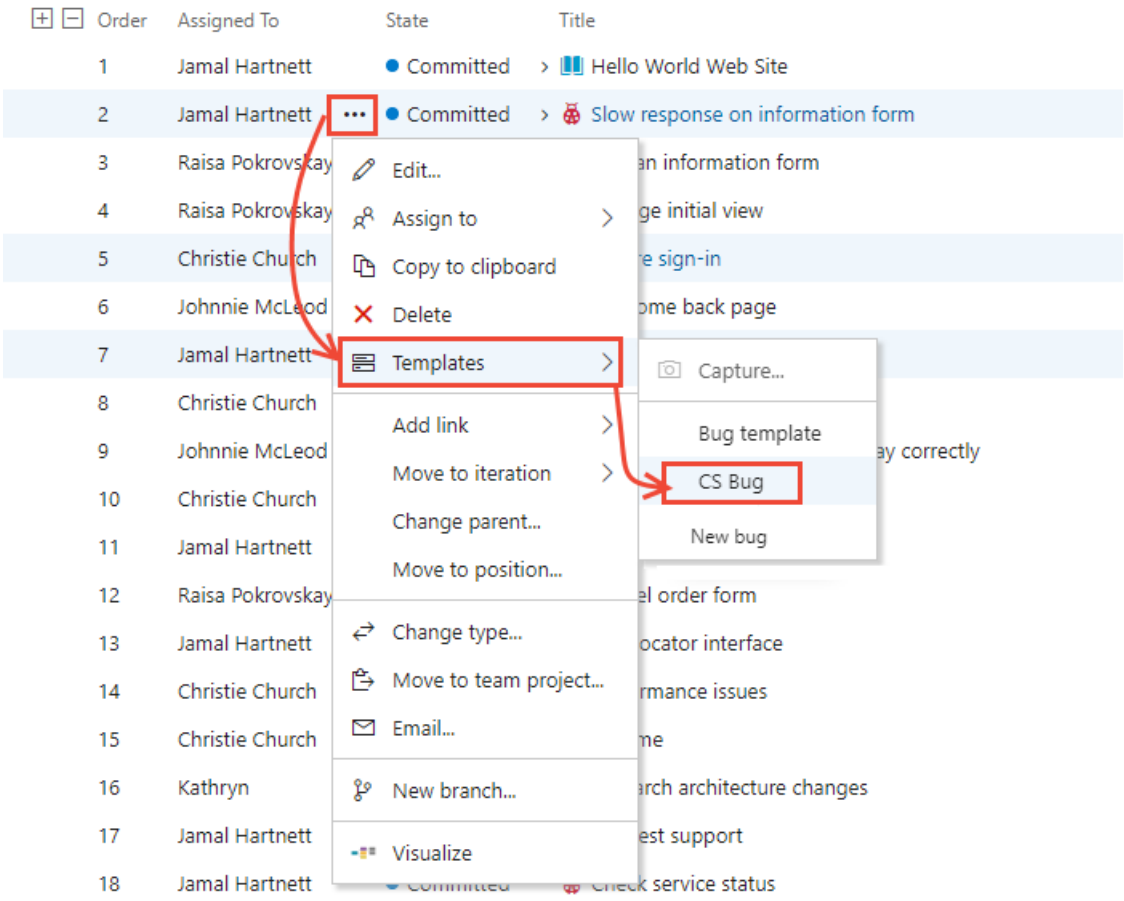
> **TIP**
> Refresh your browser to discover the latest templates that have been added. If you don't see any templates, it may be that there are none defined for the work item type.

2. Save the work item for the changes to be applied. The fields changed are noted in the History field.

Apply a template to several work items

1. To bulk update several work items, first select them from the backlog or a query results list, and then open the actions menu for one of them. All work items you select must be of the same work item type. For example, all user stories or all bugs.

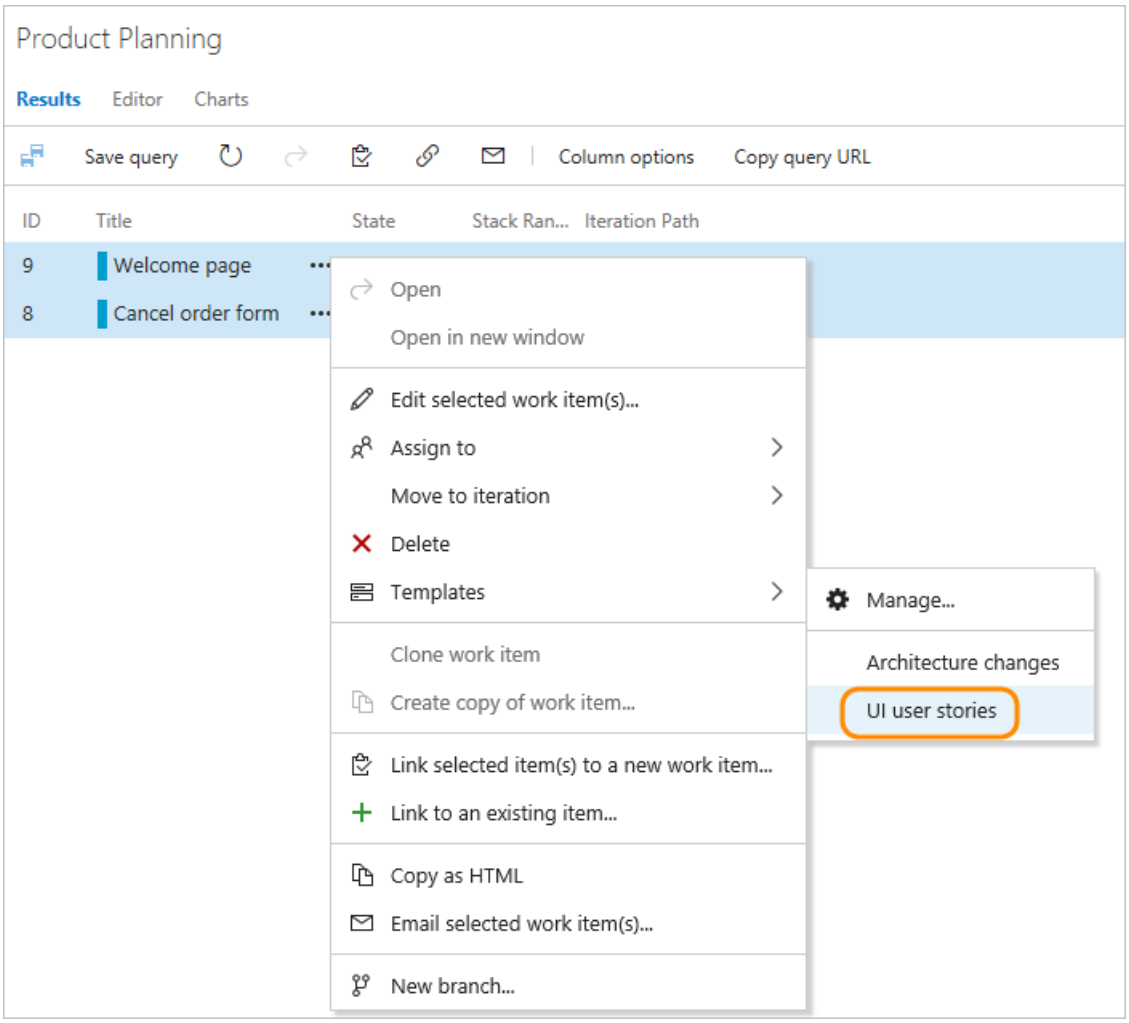
2. Choose the template to apply.



3. Field changes are automatically applied and work items saved. To learn more about bulk updates, see Bulk modify work items.

1. To bulk update several work items, first select them from the backlog or a query results list, and then open the actions menu for one of them. All work items you select must be of the same work item type. For example, all user stories or all bugs.

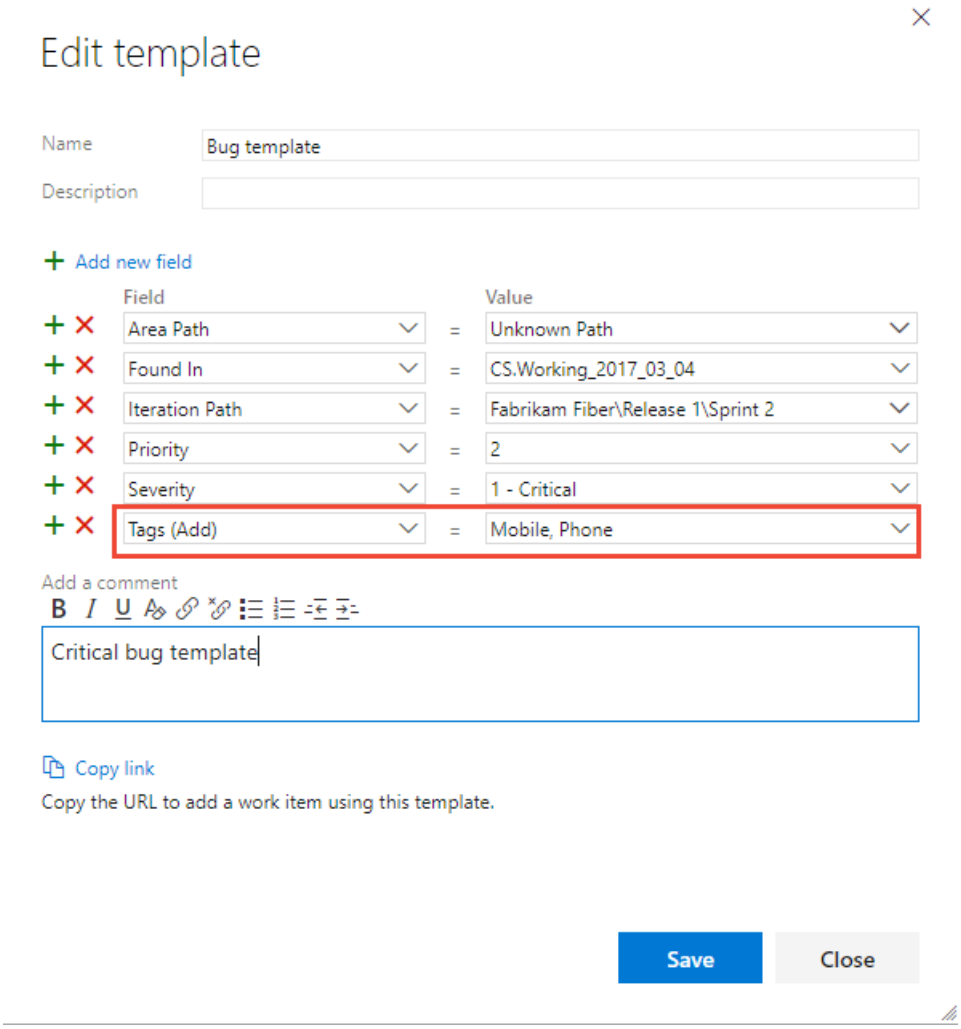
2. Choose the template to apply.



3. Field changes are automatically applied and work items saved. To learn more about bulk updates, see Bulk modify work items.

Add or remove tags from templates

You can add tags to a template and they'll be applied to the work item when you use the template. To add two or more tags, delimit them with a comma (,).



If you don't specify tags to remove, then all tags present in a work item will remain defined. They'll remain defined even when you apply a work item template to an existing work item.

Define an unplanned work item template using a hyperlink

You can specify a work item template that specifies several field values using the following syntax. Use the URL whenever you want to add a work item of the type you've defined with its predefined values.

```
https://dev.azure.com/{OrganizationName}/{ProjectName}/_workItems/create/{WorkItemType}?
[FieldReferenceName 1]={FieldValue 1}&
[FieldReferenceName 2]={FieldValue 2}&
[FieldReferenceName 3]={FieldValue 3}&
. . .
```

```
http://{ServerName}:8080/tfs/DefaultCollection/{ProjectName}/_workItems/create/{WorkItemType}?
[FieldReferenceName 1]={FieldValue 1}&
[FieldReferenceName 2]={FieldValue 2}&
[FieldReferenceName 3]={FieldValue 3}&
. . .
```

For example, the following syntax specifies a work item task with title *TaskTitle*. It specifies values for the Assigned To, Description, Tags, Activity, and Iteration Path fields.

```
https://dev.azure.com/{OrganizationName}/{ProjectName}/_workItems/create/Task?
[System.Title]=TaskTitle&
[System.AssignedTo]=Jamal+Hartnett&
[System.Description]=<p>Always+include+Remaining+Work+and+links+to+any+related+bugs+or+user+stories.</p>&
[System.Tags]=Web;+Phone;+Service&
[Microsoft.VSTS.Common.Activity]=Development&
[System.IterationPath]=Fabrikam+Fiber%5CIteration+1
```

```
http://{ServerName}:8080/tfs/DefaultCollection/{ProjectName}/_workItems/create/Task?
[System.AssignedTo]=Jamal+Hartnett&
[System.Description]=<p>Always+include+Remaining+Work+and+links+to+any+related+bugs+or+user+stories.</p>&
[System.Tags]=Web;+Phone;+Service&
[Microsoft.VSTS.Common.Activity]=Development&
[System.IterationPath]=Fabrikam+Fiber%5CIteration+1
```

TIP

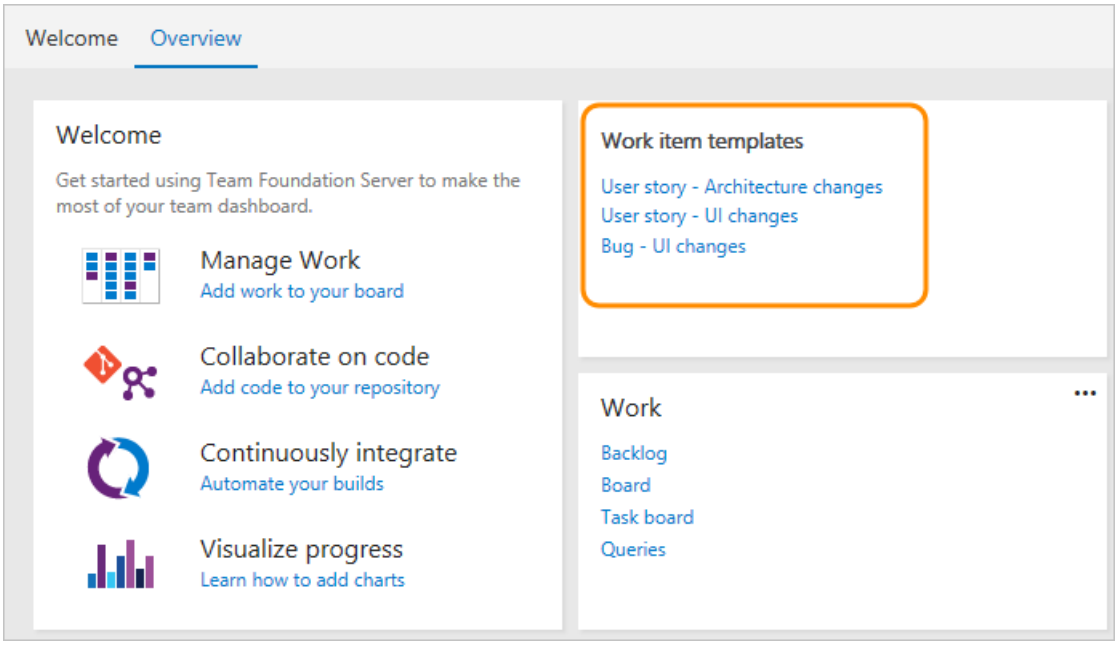
There is a 2000 character limit imposed by some browser clients.

You can save the URL as a text file or add the URL to a dashboard or web page as a hyperlink.

Add a template link to a team dashboard

You can add links to a Markdown widget that appear on your team dashboard in the web portal. These links open a work item with the template defined fields predefined.

For example, the following widget contains links to three templates.



To learn more about the Markdown widget see Add Markdown to a dashboard, Markdown widgets.

Related articles

- Azure Boards FAQs
- Sample work item templates

View sample work item templates in Azure Boards

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Work item templates can help save time and provide guidance to your team when defining user stories, features, bugs, or tasks. Teams use templates to support the following goals:

- Create bugs for specific product areas
- Provide guidance to fill out the work item
- Create work items with specific tags
- Define a bug template for use with another application or extension, such as Bug Bash Pro.

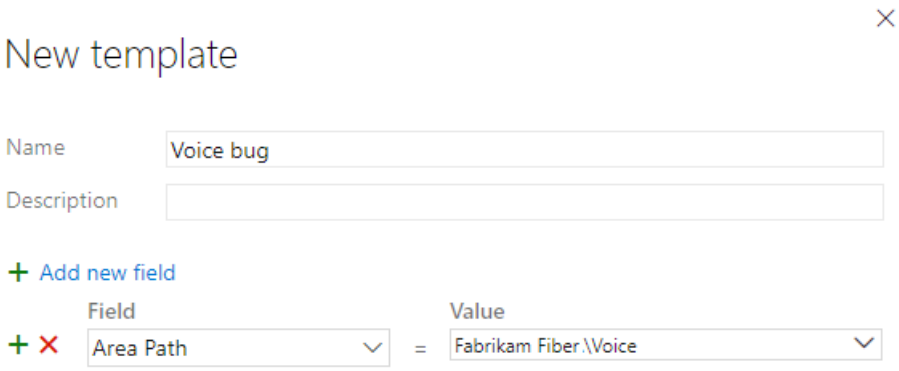
Review this article for examples of defining specific values of work item templates. For guidance on adding, managing, and applying work item templates, see Use templates to add and update work items.

> **NOTE**
> Work item templates are distinct from process templates. For information on process templates, see Choose a process template or these specific topics for the default process templates: Basic, Agile, Scrum, or CMMI.

Specify the Area Path

As an organization grows, the number of product or feature areas and number of teams can grow in number. To support work items to appear on a team's backlog or board, templates should specify the Area Path that the team owns.

In this example, the Voice team sets the **Area Path** = `Fabrikam Fiber/Voice`.



To learn more about area paths, see About area and iteration (sprint) paths.

Add guidance in a rich-text field

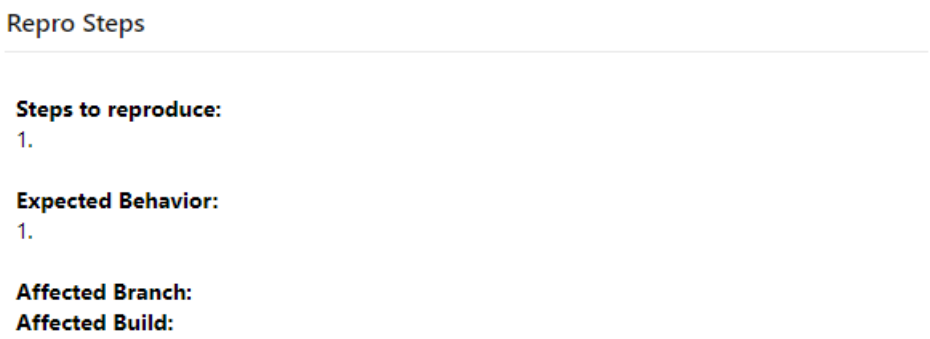
To provide guidance, enter HTML syntax into a rich-text field's value.

For example, here we add the following syntax into the **Repos Steps** field

```
<p><b>Steps to reproduce:</b><br>1.  <br><br><b>Expected Behavior:</b><br>1. <br><br><b>Affected Branch:</b> <br> <b>Affected Build:</b><br></p>
```



The work item form renders this example as shown.



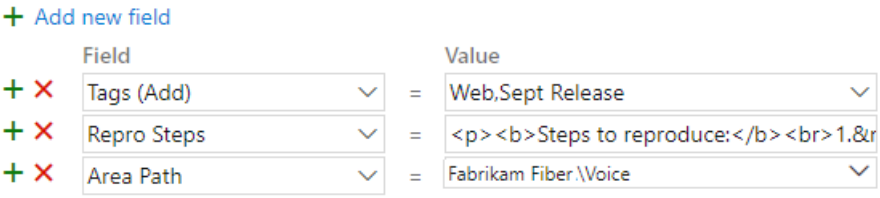
For more information about rich-text fields, see Share information within work items and social tools.

Add work item tags

Work item tags are useful to quickly filter backlogs, boards, or queries. You can add tags to track anything of interest, for example: Customer issue, Release, Milestone.

To add two or more tags, add them all within a single **Tags (Add)** field, entering a comma between tags.

For example, here we add Web and Sept Release.



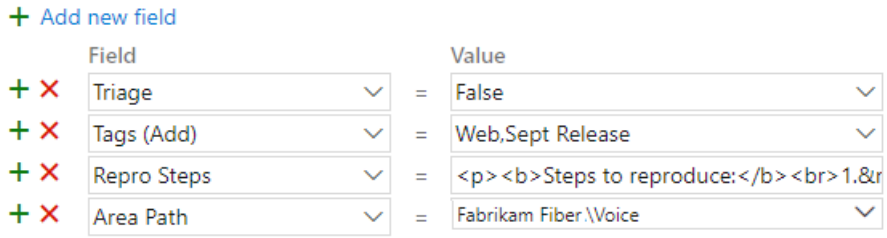
To learn more about tags, see Add work item tags to categorize and filter lists and boards.

You can also use the **Tags (Remove)** template field to remove tags from work items. For example, if many work items were tagged with Milestone 1, and that no longer applies, you could query for all those work items and remove the tag by doing a bulk apply of a template that removed the Milestone 1 tag.

Define and pre-populate custom fields

You can pre-populate a custom field that has been added to the work item type. Before adding it to a template, you must first add it to the work item type. For inherited process models, see Add and manage fields for an inherited process. For On-premises XML process models, see Add or modify a field to track work.

For example, the **Triage** custom field can be set to `False`, indicating the bug needs to be triaged.



Access other features through extensions

An often requested feature is to allow the creation of a work item that automatically links to one or more work items. For example, a user story that links to five tasks. Work item templates don't support this capability. However, you may find a Marketplace extension will support this feature. For example, see the following extensions:

- Work item form one click actions
- 1-Click Tasks
- 1-Click Child-Links

Customize the work item web form

If your project uses the On-premises XML process model to customize work tracking, you can add help text, hyperlinks, or web content to a form to provide guidance.

To learn more, see the following articles:

- WebLayout and Control elements
- Provide help text, hyperlinks, or web content on a work item form
- Edit a WIT definition to add web content to a work item form

Extensibility

You can programmatically interact with work item templates to create, get, list, and delete using the Templates REST APIs.

Related articles

- Use templates to add and update work items

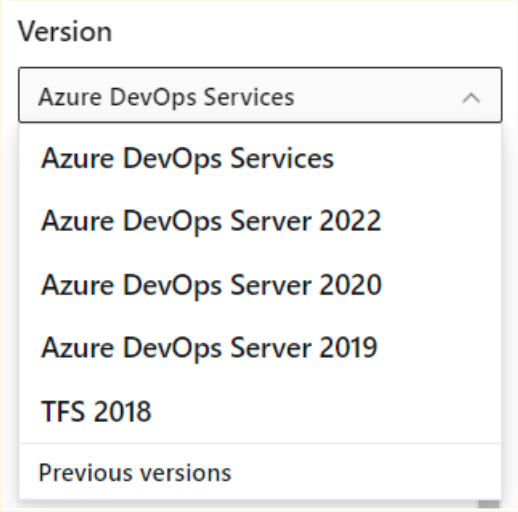
Define the URL for a work item

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can define the URL for a work item using the syntax provided based on the version or platform you work from.

IMPORTANT



To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

Examples in this article use the following conventions:

- *OrganizationName* specifies the name of the Azure Boards organization
- *ServerName* specifies the name of the TFS application tier server
- *Port* specifies the port, default=8080
- *CollectionName* specifies the name of the project collection.
- *ProjectName* specifies the project name
- *WorkItemNumber* specifies the ID of the bug, task, or other work item.

Azure DevOps Services

```
https://dev.azure.com/OrganizationName/ProjectName/_workitems/edit/WorkItemNumber
```

or

```
https://OrganizationName.visualstudio.com/DefaultCollection/ProjectName/_workitems/edit/WorkItemNumber
```

Examples:

```
https://dev.azure.com/fabrikam/Phone%20Saver/_workitems/edit/390
```

or

```
https://fabrikam/DefaultCollection/Phone%20Saver/_workitems/edit/390
```

Azure DevOps Server 2020

http://*ServerName:Port/CollectionName/ProjectName/*_**workitems?id=** *WorkItemNumber***&_a=edit**

Example:

```
http://fabrikamprime:8080/DefaultCollection/Phone%20Saver/_workitems/133&_a=edit
```

Azure DevOps Server 2019, TFS 2018 through TFS 2015

http://*ServerName:Port*/**tfs**/*CollectionName/ProjectName/*_**workitems?id**=*WorkItemNumber***&_a=edit**

Example:

```
http://fabrikamprime:8080/tfs/DefaultCollection/Phone%20Saver/_workitems/133&_a=edit
```

TFS 2013.2

http://*ServerName:Port***/tfs/***CollectionName/ProjectName/***_workitems/edit/***WorkItemNumber*

Example:

```
http://fabrikamprime:8080/tfs/DefaultCollection/Phone%20Saver/_workitems/edit/133
```

TFS 2013.1 and earlier versions

```
http://ServerName:Port/tfs/CollectionName/TeamProjectName_workitems#_a=edit&id=WorkItemNumber
```

Example:

```
http://fabrikamprime:8080/tfs/DefaultCollection/Phone%20Saver/_workitems#_a=edit&id=133
```

Share links in context with your board or backlog

Share links to work items within the context of your board or backlog with the work item URL parameter. When you open a work item form from your board, backlog, or sprint backlog, the parameter `?workitem=[ID]` is appended to the URL.

For example:

```
https://dev.azure.com/fabrikam/FabrikamFiber/_boards/board/t/Voice/Stories/?showParents=true&workitem=191
```

```
https://ServerName/DefaultCollection/ProjectName/_boards/board/t/Voice/Stories/?workitem=56
```

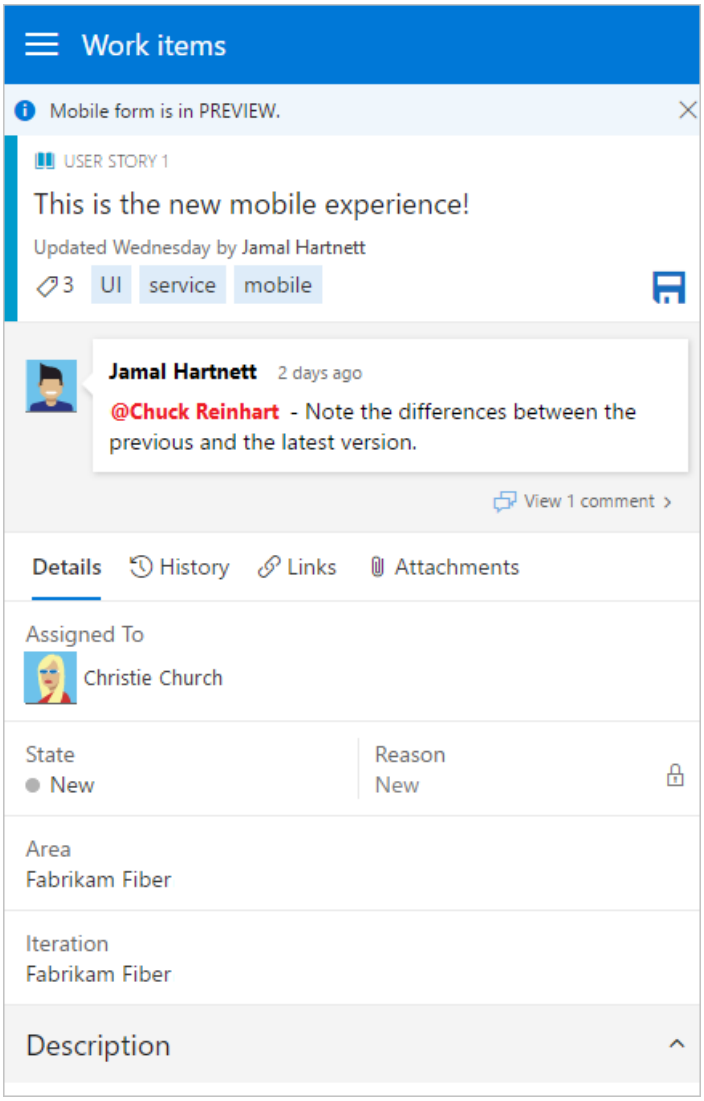
Anyone you share the link with, opens the work item within the same context you had when you shared the link.

View and update work items via the mobile browser

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With the mobile browser and work item form, you gain on-the-go features to stay on top of the latest updates made to work tracking. When you click any work item link on your mobile device, it will open a mobile-friendly version of the work item. From there, you can update the work item or access all work items assigned to you or that you're following.



NOTE

The mobile browser supports Azure DevOps work tracking. To sign up for free, go to Azure DevOps Services. The mobile browser is not an app, but a mobile view into select features. There is nothing to download. You access the mobile browser by clicking a link from a work item you receive in your mobile email application.

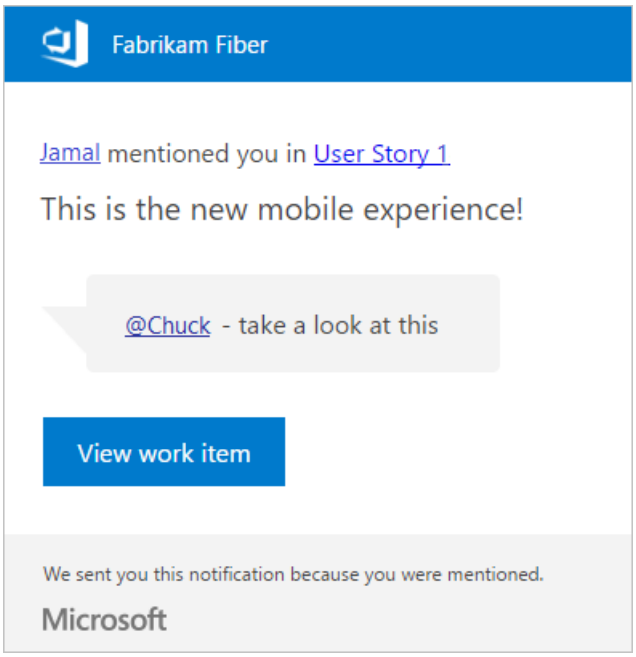
NOTE

The mobile browser is available for TFS 2018 and Azure DevOps Server 2019 and later versions. For downloads, see Downloads. The mobile browser is not an app, but a mobile view into select features. There is nothing to download. You access the mobile browser by clicking a link from a work item you receive in your mobile email application.

Open the mobile work item form

The mobile work item form will open when you click **View work item** from an email you receive from your mobile device. You'll receive this type of email under these circumstances:

- Changes were made to a work item you're following
- You were **@mentioned** in a discussion
- A notification is sent based on the work item alerts you've set using Manage personal notifications.



Update a work item

Within the mobile form, you can do almost everything you can do from the web portal form. Here are the actions you can take in the order they appear in the mobile form:

- Add and remove tags
- View and add to the discussion, click on the comment to add to the discussion
- View and update any field within the form (Assign to, State, Area, Iteration, Description, and more)
- View and open a link within the Development section
- View History
- View and open a link from the Links tab
- Open and add an attachment from the Attachments tab

Actions not available to you within the mobile work item form:

- You can't create or add new work items
- You can't initiate a development operation
- You can't add a link

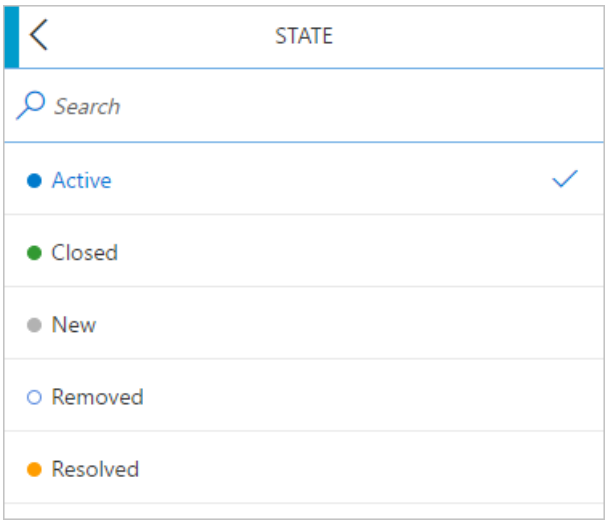
Interact with mobile form controls

Mobile form controls operate as follows:

- Click any field to edit it and the form changes to a full-screen experience. For example, some of the most common actions such as changing the state of an item, moving to a different area path, adding an attachment, and creating/removing tags are all supported.
- When done, click the return option.
- Remember to click the save icon to save your changes!

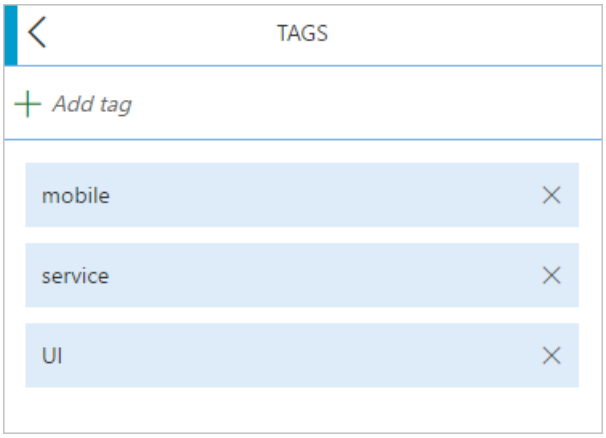
Update status (change State)

To update the state, click the state you want.



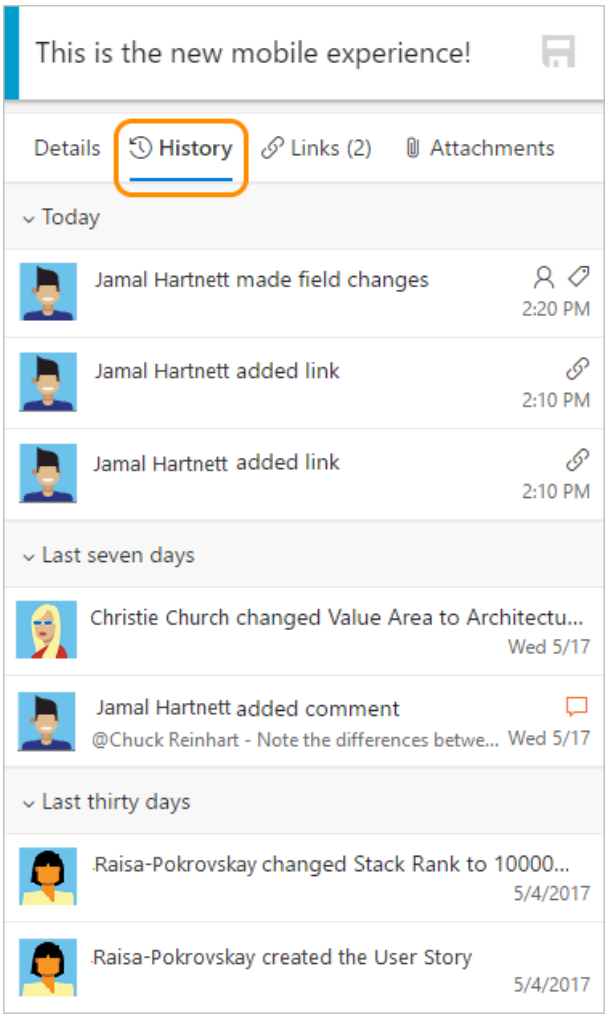
Add or remove tags

To add a tag, type the text you want.



View history

Click the History tab to view history.



View and open work items in your activity lists

From within the mobile work item form, you can access your work items. The mobile browser allows you to view and open work items which fall into these categories:

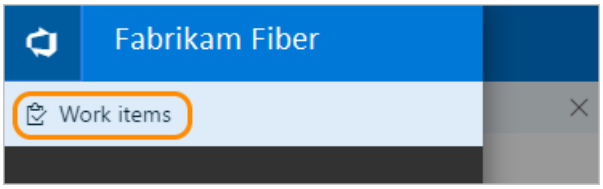
- **Assigned to me**: lists all work items assigned to you
- **Following**: lists all work items that you have elected to follow
- **My activity**: lists all work items that you have recently viewed or updated.

The lists available from each page span all team projects that you work in.

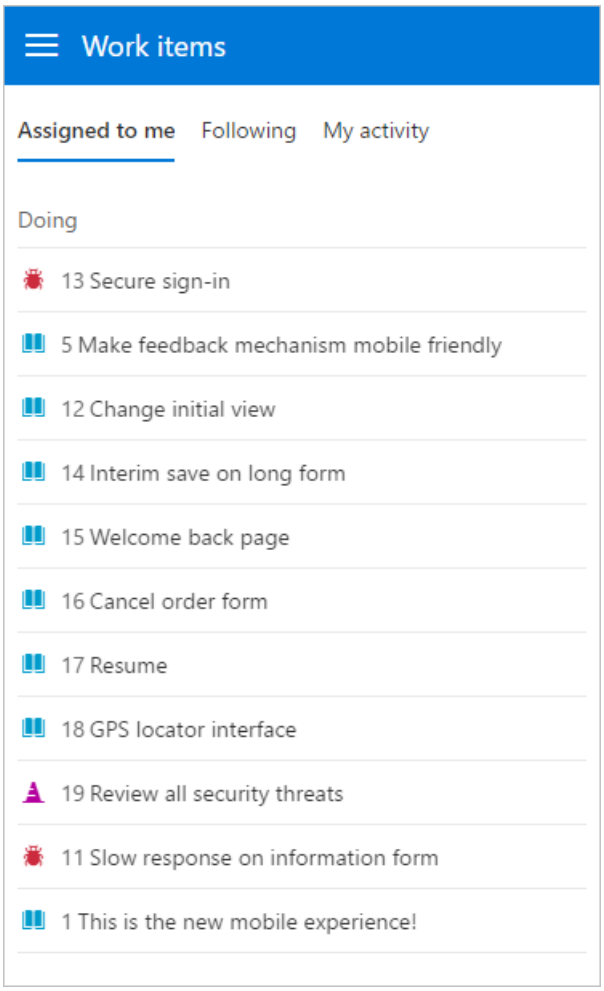
To access a list, first click the list control from the work item form you've opened.



Then, click **Work items**.



The browser opens to the **Assigned to me** page. From there, you can choose **Following** or **My activity** to access the other pages. To learn more about the **Work Items** view, see View and add work items.



Related articles

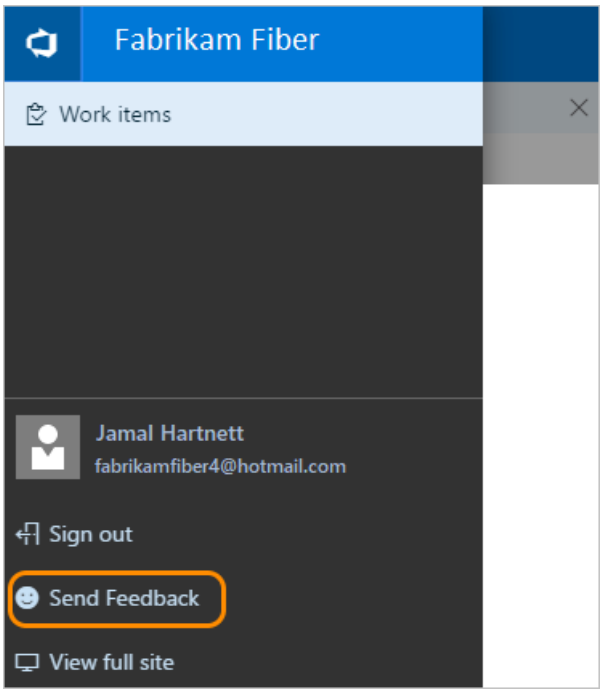
Additional experiences are in the works to improve and expand on the mobile experience. For more information, see the blog post: The mobile work item form (preview).

- Set personal notifications
- Set team notifications
- Follow a work item

Provide feedback for the mobile experience

Help us improve the mobile experience.

To provide feedback, choose the list control from the work item form and then click **Send Feedback**. To complete the feedback, select either the smile or frown and optionally enter a comment.


Fabrikam Fiber
Work items
Jamal Hartnett
fabrikamfiber4@hotmail.com
Sign out
Send Feedback
View full site

Add, review, and update work items in Azure Boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Collaborate with others by adding, updating, and reviewing your work items as cards on a Kanban board.

If you're a project administrator just getting started, review the Configure settings and manage your Azure Boards project to learn more about defining area and iteration paths and customizing your work item types. To add another Kanban board, you do that by adding a team. For details, see About teams and Agile tools.

Use Kanban boards to plan and show progress

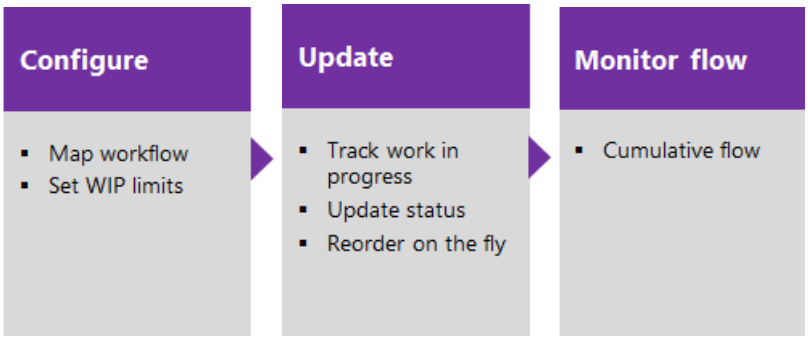
Your Kanban board provides you with a visual interactive space for you and your team to plan and show progress. With it, your team can track the critical information they need by seeing which work items are in progress, where the bottlenecks are, who work is assigned to, and more.

Boards present work items as cards and support quick status updates through drag-and-drop, similar to sticky notes on a physical whiteboard.

In a nutshell, you use boards to:

- Implement Kanban methods
- Define work your team is tasked with by defining user stories, product backlog items, or requirements
- Monitor progress and update the status of work items by drag-and-drop to a column
- Add details and estimates to your backlog items
- Quickly define tasks or child items for backlog items
- Add, run, and update inline tests

To get started, you'll want to follow the steps outlined below.

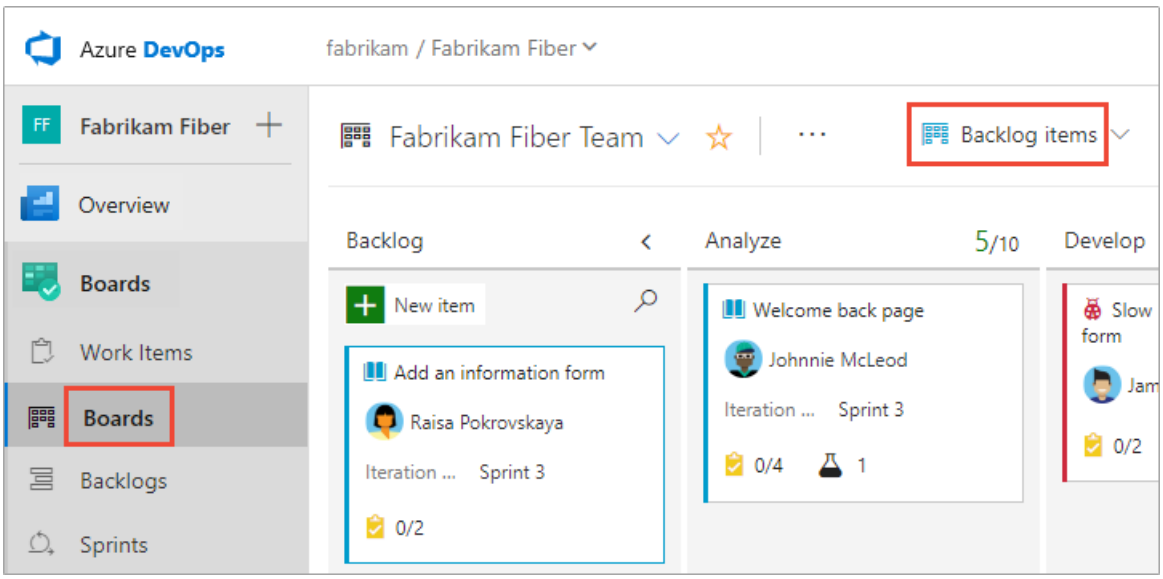


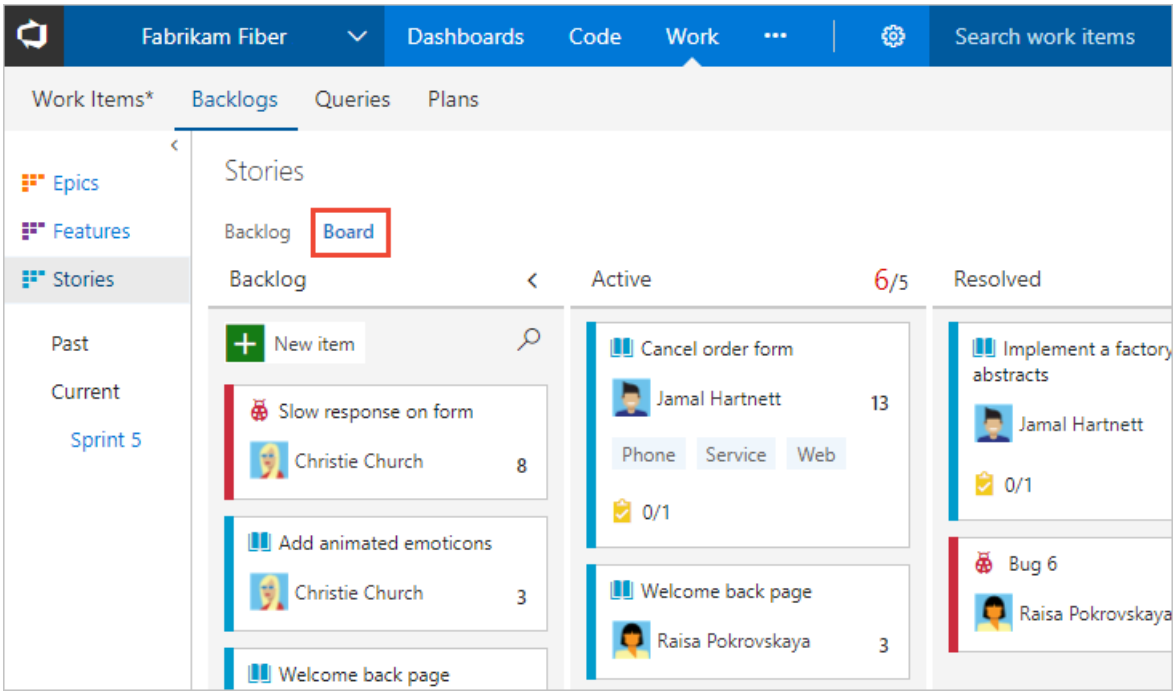
> **NOTE**
> A Kanban board is provisioned with the addition of each project and each team. You can only create or add Kanban boards to a project by adding another team. To learn more, see About teams and Agile tools.

Product and portfolio Kanban boards

Each product and portfolio backlog has a corresponding Kanban board. Both backlogs and boards are associated with a team, and display work items based on the area and iteration paths selected by the team as described in Define iteration (sprint) paths and configure team iterations.

Each board supports many Kanban practices such as defining columns and swimlanes, setting Work-in-Progress (WIP) limits, defining the Definition of Done, and more. To get started, see Kanban quickstart.





Configure and customize your Kanban board

Your Kanban board is highly configurable to support your team's workflow. Each team can configure each board in the following areas:

- Configure boards:
  - Add columns
  - Set WIP limits
  - Set Definition of Done
  - Add swimlanes
  - Define card reordering
  - Enable backlog and board levels
  - Working with bugs
- Configure card displays:
  - Add or remove fields from cards

- Define card styles
- Apply tag colors
- Enable/disable annotations
- Define inline test behavior on cards

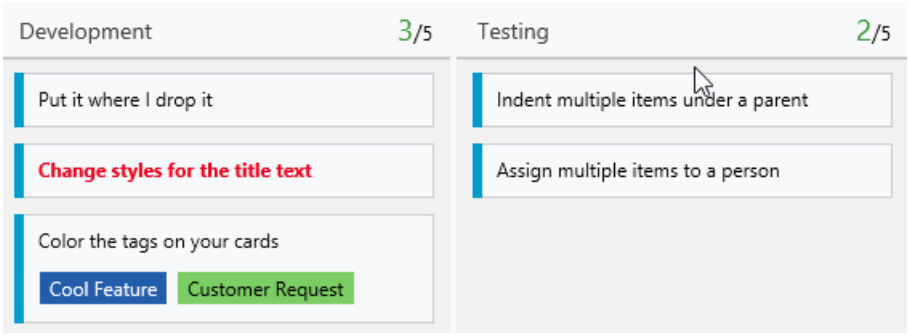
Along with these team configurations, you can customize a project by adding or modifying work item types, the workflow, and add customized portfolio backlogs and boards.

Common board configurations for multiple teams

Question: *Can you define a board configuration that multiple teams can subscribe to?* **Answer**: *No. Each team controls their own team settings and board configurations.*

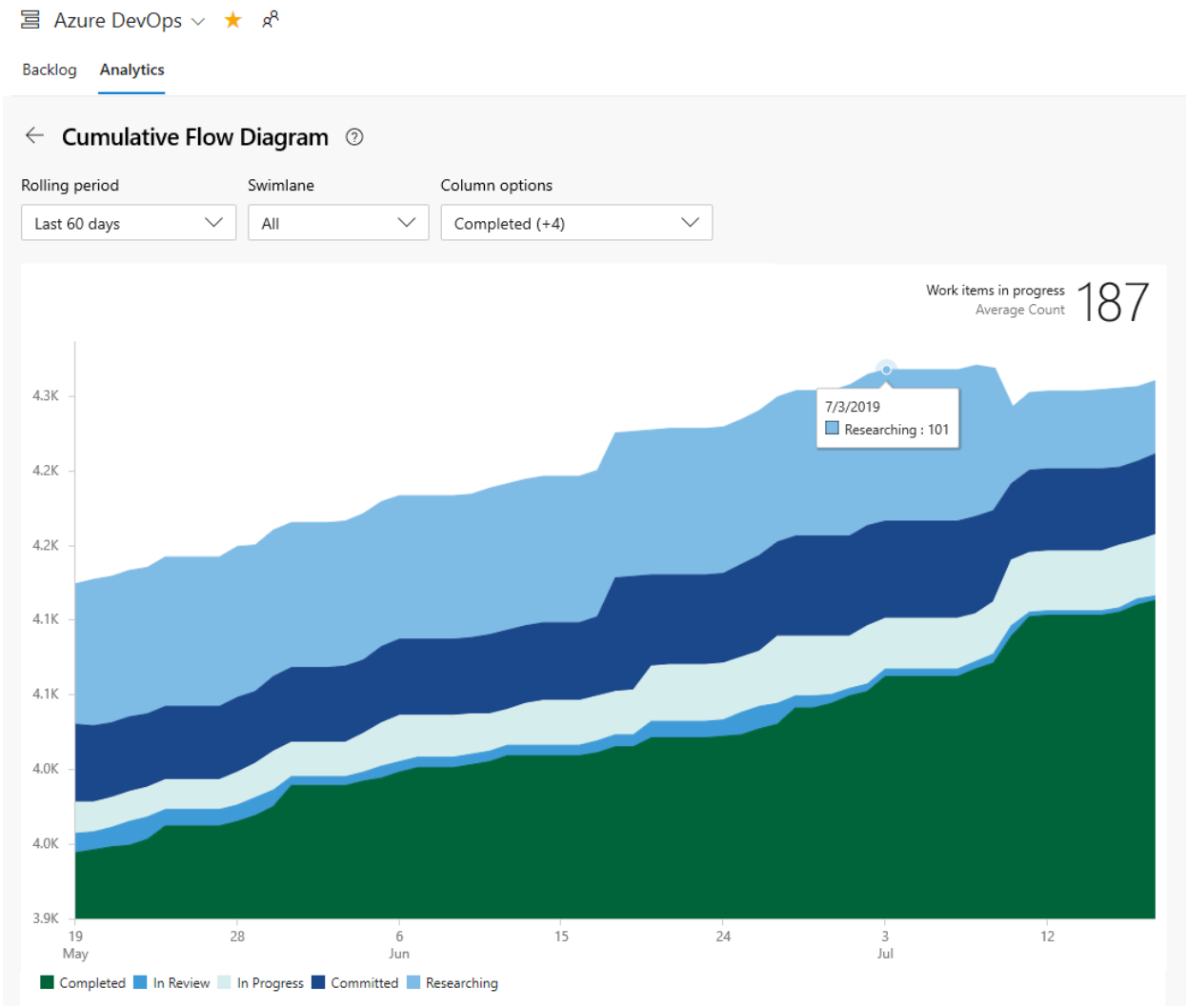
Update work item status

Once you've configured your Kanban board, you can add work items directly to the board. You can then update the status of work by dragging a card to another column on the Kanban board. You can even change the order of items as you move a card to a new column. For more information, see Workflow states and state categories.

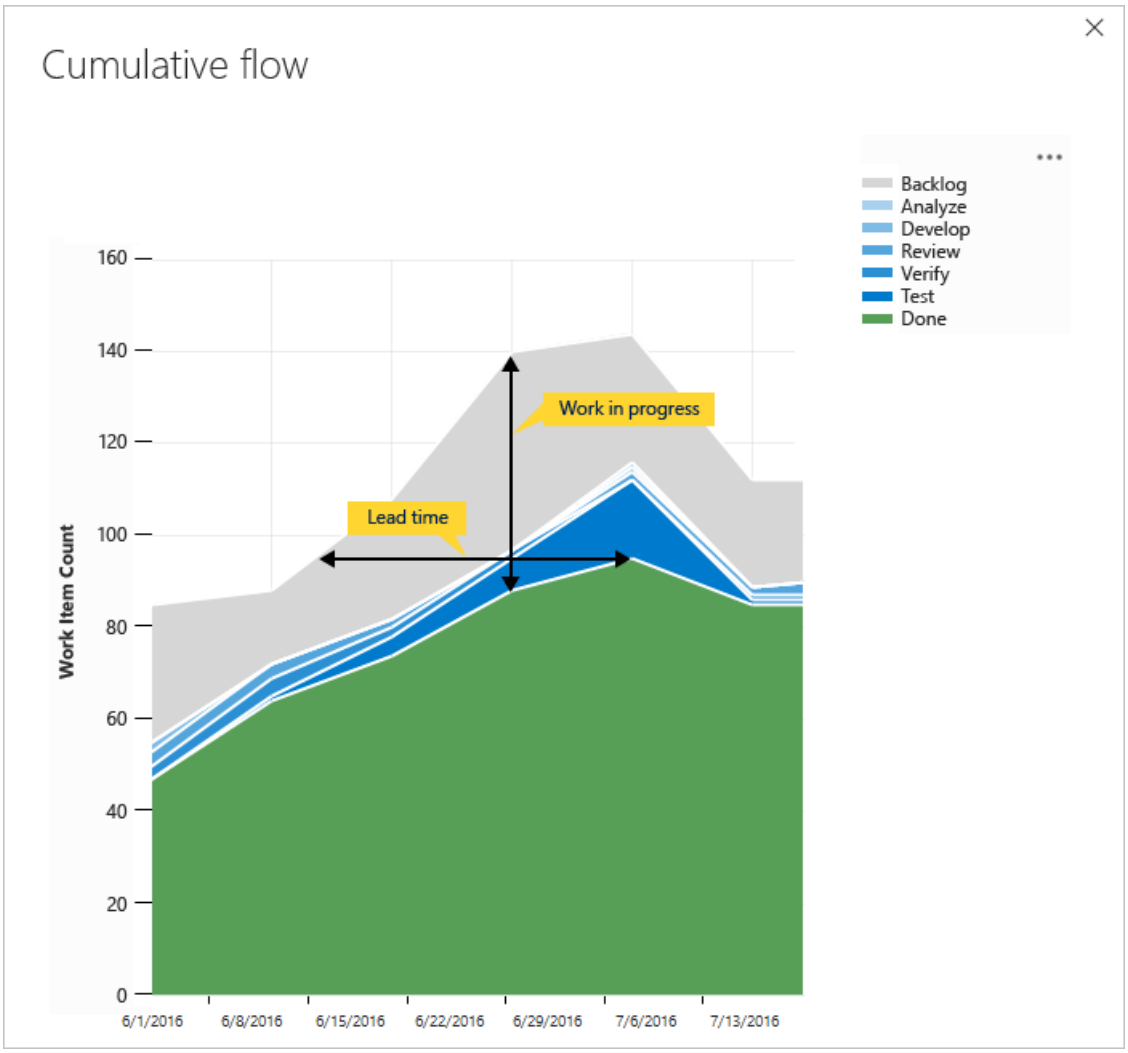


Use the Cumulative Flow Diagram to monitor metrics

The Cumulative Flow Diagram (CFD) is an in-context report available for both product and portfolio backlog levels. Use this chart to monitor key metrics to fine-tune your processes.



The Average work in progress value excludes completed work items. Use the interactive controls to choose the time frame, swimlanes, and workflow states or Kanban board columns.



The CFD shows the count of items in each Kanban column for the past 30 weeks or less. From this chart, you can gain an idea of the amount of work in progress and lead time. Work in progress counts unfinished requirements. Lead time indicates the amount of time it takes to complete a requirement from the time it was first proposed.

By monitoring these metrics, you can gain insight into how to optimize your processes and minimize lead time.

To learn more, see the following articles:

- Configure a cumulative flow chart
- Cumulative flow, lead time, and cycle time guidance
- Lead time and cycle time widgets

Display of leaf node work items

While you can create a hierarchy of backlog items, tasks, and bugs—we don't recommend that you create same-category hierarchies. That is, don't create parent-child links among work items of the same type, such as story-story, bug-bug, task-task. The reason is that the display of the leaf node, the last node in a same-category hierarchy, may only appear on Kanban boards, sprint backlogs, and taskboards. For example, if you link items within a same-category hierarchy that is four levels deep, only the items at the fourth level appear on the Kanban board, sprint backlog, and taskboard.

Instead of nesting requirements, bugs, and tasks, we recommend that you maintain a flat list. In other words, only create parent-child links one level deep between items that belong to a different category. To learn more, see Fix re-ordering and nesting issues, How backlogs and boards display hierarchical (nested) items.

Exercising select features with multi-team ownership

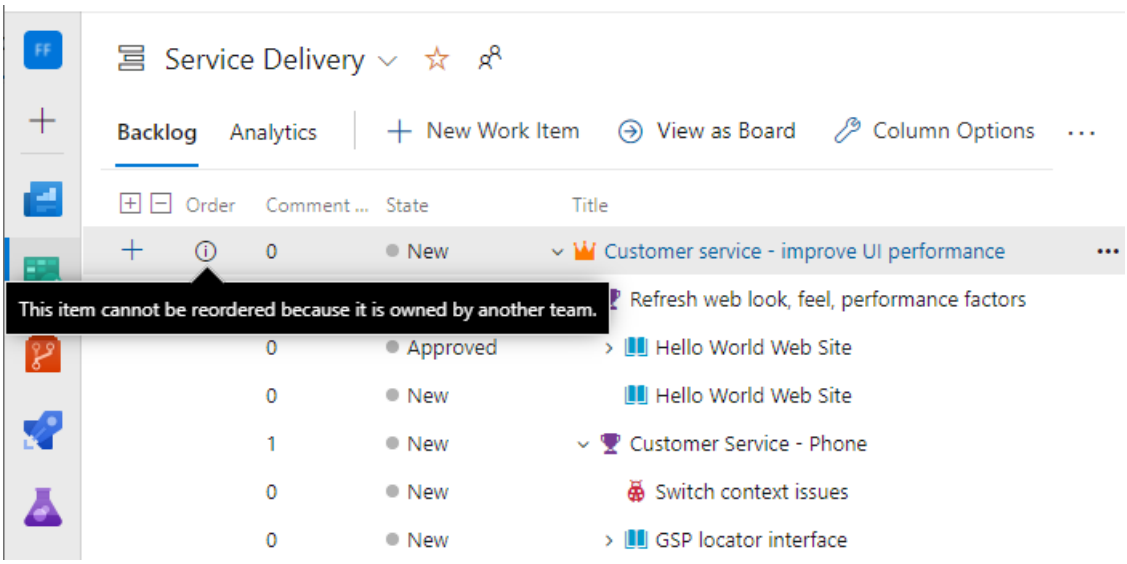
When you share area paths across two or more teams, you'll want to understand how Azure Boards manages conflicts that can arise when exercising these features:

- Reordering or reparenting work items on a backlog or board
- Updates made to Kanban Board Column, Done, and Lane fields when dragging items to a different column

Reordering and reparenting work items

All backlogs and boards support drag-and-drop to reorder and reparent work items. Updates made to one team's backlogs and boards are reflected in other team backlogs and boards that share the same area path. You may need to refresh the page to view the changes.

You can only use drag-and-drop to reorder or reparent work items assigned to area paths selected for your team. When the **Parents** view option is enabled, work items may appear on your backlog that your team doesn't own. Anything that appears with the ⓘ information icon can't be reordered nor reparented as it's owned by another team.



Kanban board column updates

Because each team can customize the Kanban board columns and swimlanes, the values assigned to Kanban board fields may differ from what you expect when another team updates the work item from a different board. Even if the management team and the feature teams configure their Feature Kanban board columns with identical workflow mapping, updating work items on one team's Kanban board won't be reflected on another team's Kanban board. Only when the work item moves to a column that maps to a workflow state does the card column reflect the same on all boards.

By design, the team with the longest area path wins the conflict and determines the values for the Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields. If the shared area paths are of equal depth, the results are non-deterministic.

The primary work-around for this issue is to maintain single ownership of work items by Defining area paths and assign to a team. Another option is to add custom workflow states that all teams can use. For details, see Customize the workflow (Inheritance process).

The primary work-around for this issue is to maintain single ownership of work items by Defining area paths and assign to a team. Another option is to add custom workflow states that all teams can use. For details, see Change the workflow for a work item type.

Provide permissions and access

As a member added to the Contributors group of a project, you can use most features provided under **Boards** or **Work**. Users with Basic access have full access to all features. Users with Stakeholder access are limited to certain features. For details, see Stakeholder access quick reference.

To learn more about permissions and access, see Permissions and access for work tracking and Stakeholder access quick reference.

To add users to a project, see Add users to a project or team.

Customize your project and boards

If you need more than three board levels, you can add more. To learn how, see Customize your backlogs or boards for a process.

You can also add or modify the fields defined for a work item type (WIT), add a custom WIT, or modify the workflow. To learn more, see Customize an inheritance process.

If you need more than three board levels, you can add more. To learn how, see Add portfolio backlogs.

You can also add or modify the fields defined for a work item type (WIT), add a custom WIT, or modify the workflow. To learn more, see Customize the On-premises XML process model.

Can I view a Kanban board of work items defined by a query?

The Query Based Boards Marketplace extension supports viewing a flat-list query of work items as a Kanban board. The query can contain different work item types and work items defined in different projects.

Try this next

Take these tools for a test run by signing up for free. From there, you're ready to add items to your Kanban board and customize it.

Kanban quickstart

Related articles

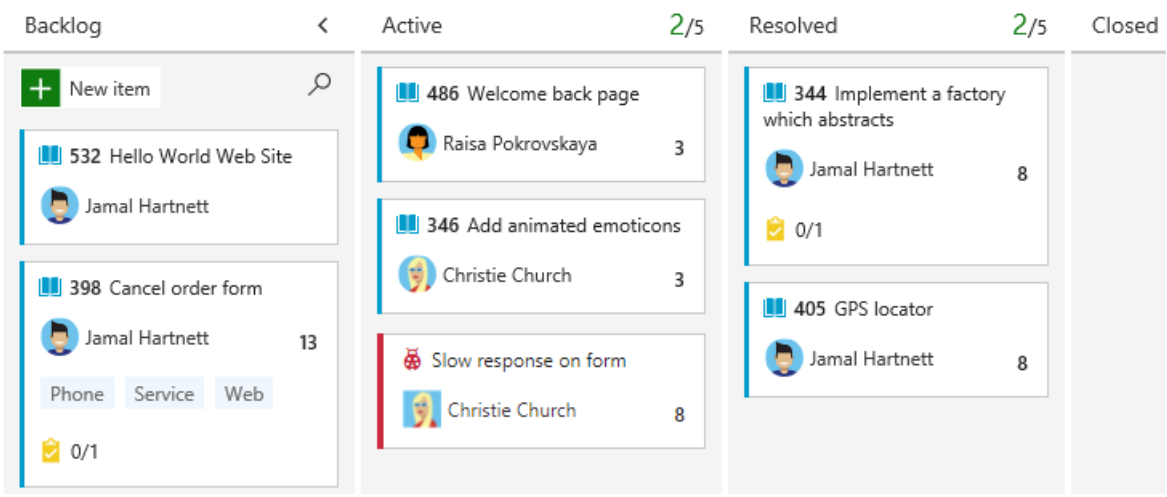
- Kanban key concepts
- Web portal navigation
- Backlogs, portfolios, and Agile project management
- About work items
- Work across projects FAQs
- What is Agile?
- What is Agile development?

Start using your Kanban board

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Your Kanban board turns your backlog into an interactive signboard, which provides a visual flow of work. As work progresses from idea to completion, you update the items on the board. Each column represents a work stage. Each card represents a backlog item, user story, or bug at that stage of work.



User stories and bugs correspond to types of work items. You use work items to share information, assign work to team members, update status, track dependencies, and more.

Prerequisites

Boards are automatically created when you create a project or add a team. Each team has access to their own product and portfolio boards as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.
- Users with **Stakeholder** access for a public project have full access to board features just like users with **Basic** access.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access can't exercise these board features: add work items, drag-and-drop work items to update status, or update fields displayed on cards. They can add tasks and change task status.

Add a Kanban board

If you have a project, you have a Kanban board. Each project defines a default team and set of boards for that team. You only need to add a Kanban board when you want to support a new team. When you add a team, you add many team assets. A team admin can configure the assets to support the way the team works. To add a set of Kanban boards to support a new team, see Add a team.

Each team's Kanban boards are associated with one or more work item types. The association depends on the process selected when the project was created, team configurations, and process customizations. The boards defined for each process are:

- **Agile**: **Stories**, **Features**, and **Epics**
- **Basic**: **Issues** and **Epics**
- **Scrum**: **Backlog items**, **Features**, and **Epics**
- **CMMI**: **Requirements**, **Features**, and **Epics**

- **Agile**: **Stories**, **Features**, and **Epics**
- **Scrum**: **Backlog items**, **Features**, and **Epics**
- **CMMI**: **Requirements**, **Features**, and **Epics**

You choose the board from the backlog selector as shown in the following image.



To customize your Kanban boards to add custom work item types, add portfolio backlogs, or other supported options, see the following articles, depending on the process your project uses:

- Inherited process model
- On-premises XML process model

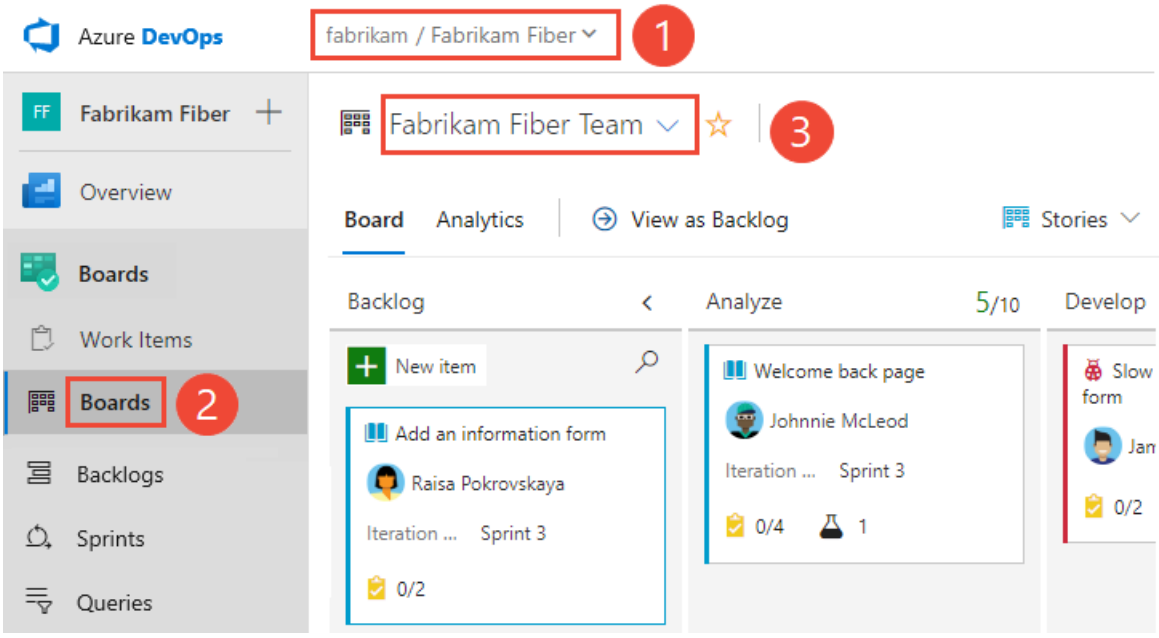
To customize your Kanban boards to add custom work item types, add portfolio backlogs, or other supported options, see On-premises XML process model.

> **NOTE**
> Both Kanban boards and Taskboards support visualizing the flow of work and monitoring metrics to optimize that flow. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard.

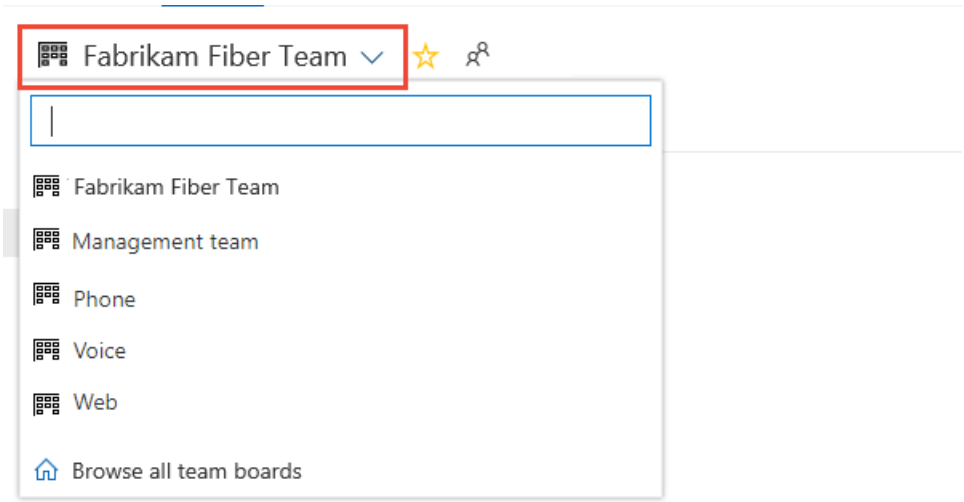
Open your Kanban board from the web portal

Your Kanban board is one of two types of boards available to you. The other is the sprint Taskboard. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard. For an overview of the features supported on each backlog and board, see Backlogs, boards, and plans.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



TIP

Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

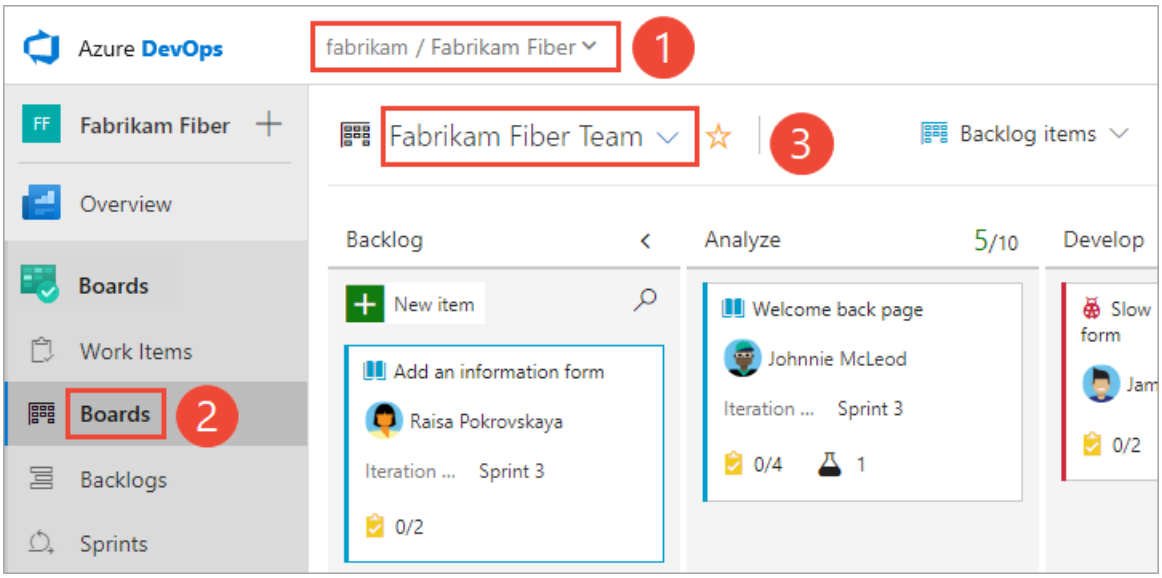
2. Check that you selected **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI as the

backlog level.

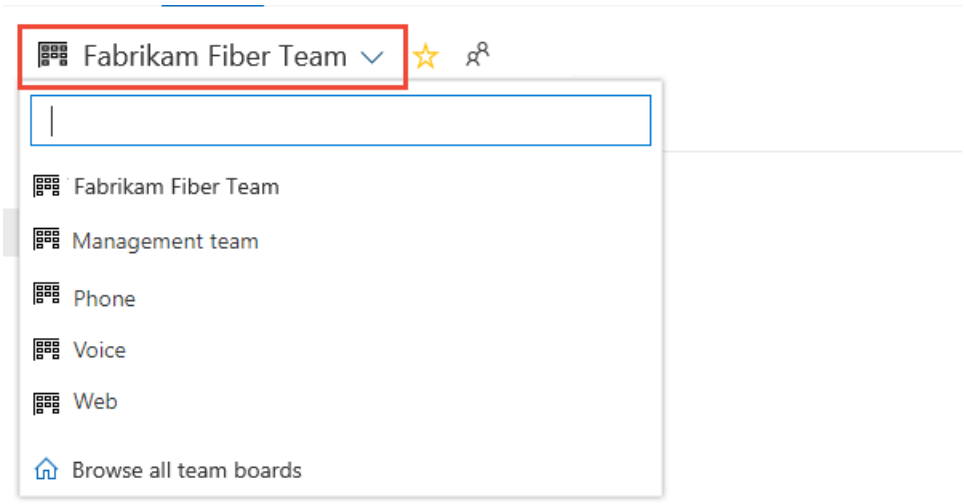


To switch to the product backlog, select **Stories backlog**. To switch to a Taskboard, see Update and monitor your Taskboard.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



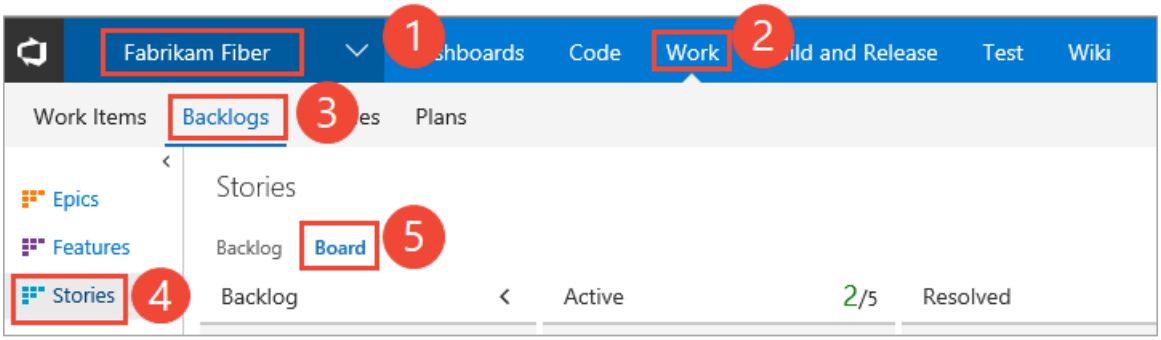
> **TIP**
> Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI as the backlog level.

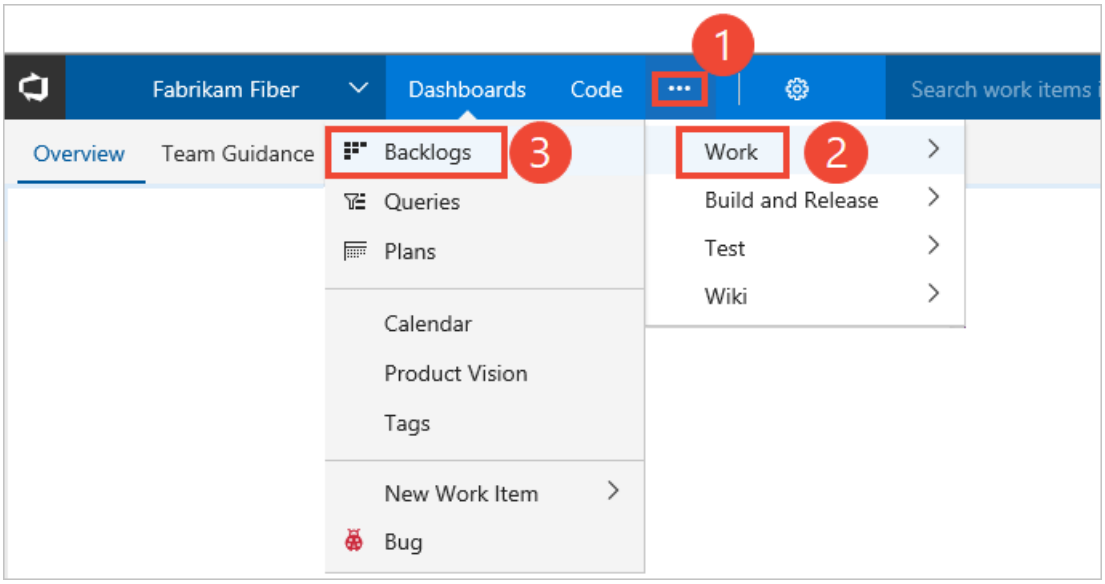


To switch to the product backlog, select **Stories backlog**. To switch to a Taskboard, see Update and monitor your Taskboard.

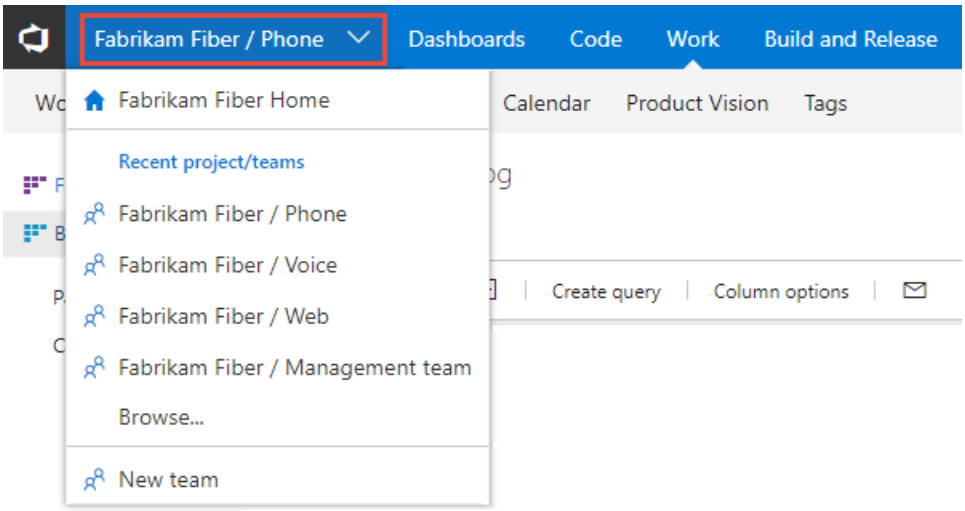
1. To view your Kanban board, open your project from a web browser. Select **Work** > **Backlogs** > **Stories**, and then select **Board**.



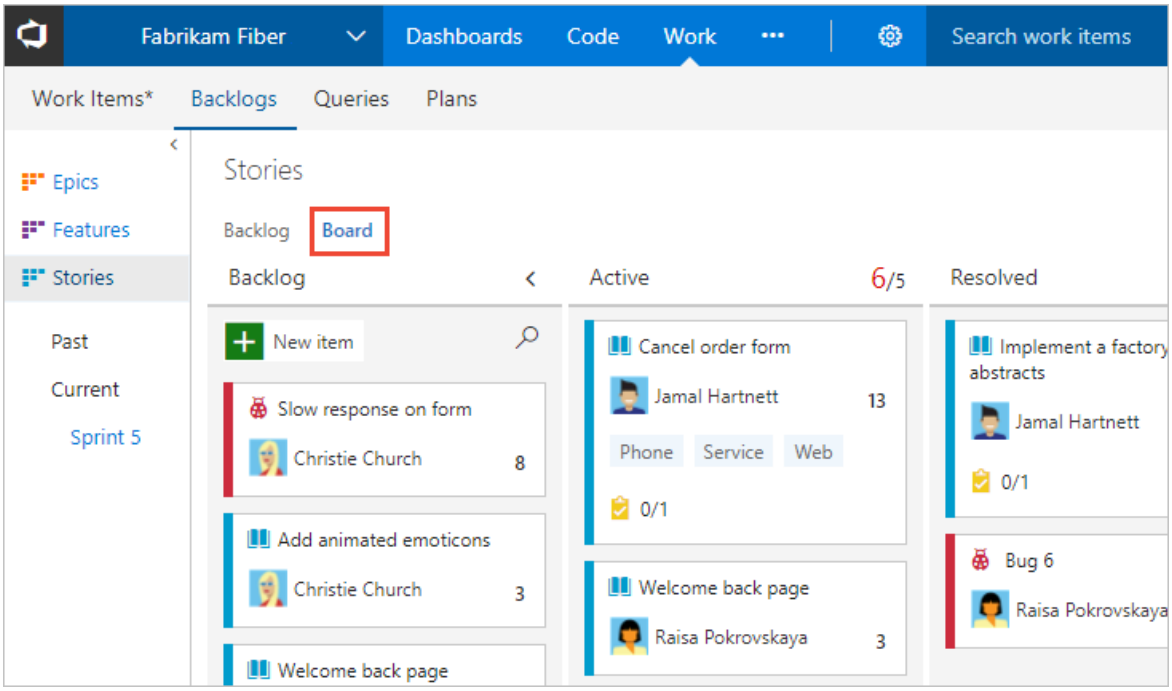
If you don't see **Work**, your screen size might be reduced. Select the three dots ( ) icon. Then select **Work** > **Backlogs** > **Board**.



2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.

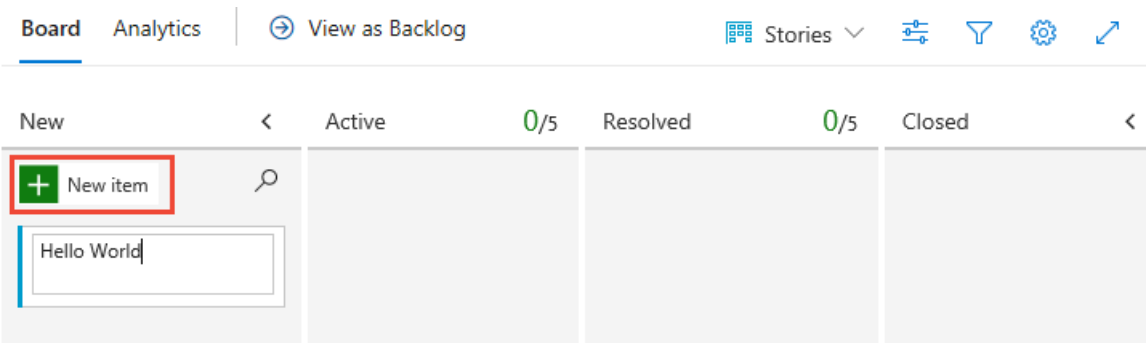


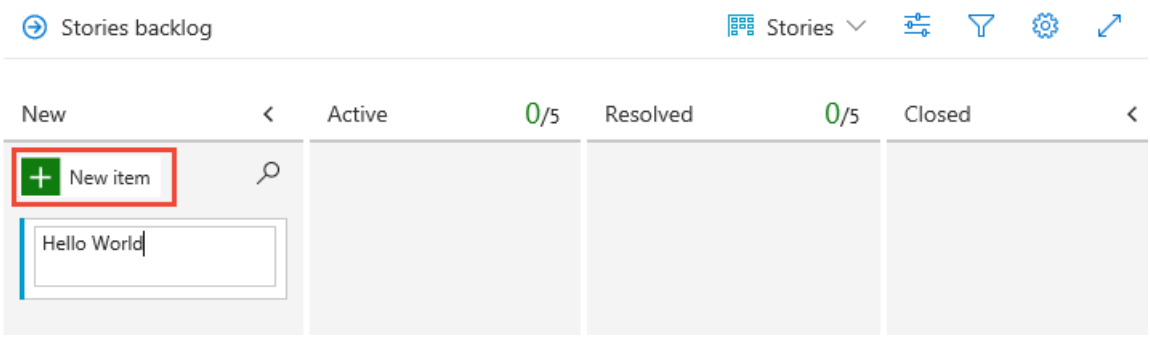
Your Kanban board appears.



Add work items

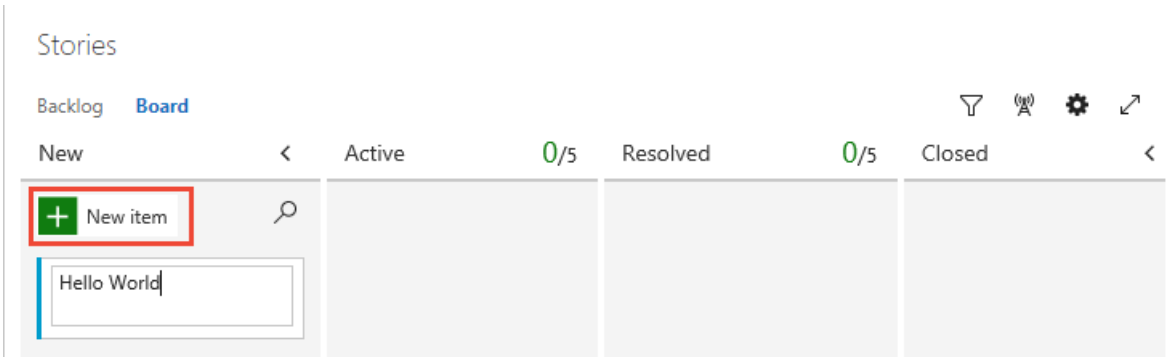
To add a work item, select the plus sign, enter a title, and then press Enter.





The system automatically saves the work item with the title you entered. You can add as many work items you want by using this method.

To add a work item, select the plus sign, enter a title, and then press Enter.



The system automatically saves the work item with the title you entered. You can add as many work items you want by using this method.

To add details to any work item, select the title. Or, you can directly modify any field that displays. For example, you can reassign a work item by selecting **Assigned To**. For a description of each field, see Create your backlog, Add details and estimates.

> **NOTE**
> You can only assign work to a single user. If you need to assign work to more than one user, add a work item for each user and distinguish the work to be done by title and description. The Assigned To field only accepts user accounts that have been added to a project or team.

To customize the set of fields displayed on the card, see Customize cards.

Update the status of a work item

As work completes in one stage, update the status of an item by dragging it to a downstream stage.

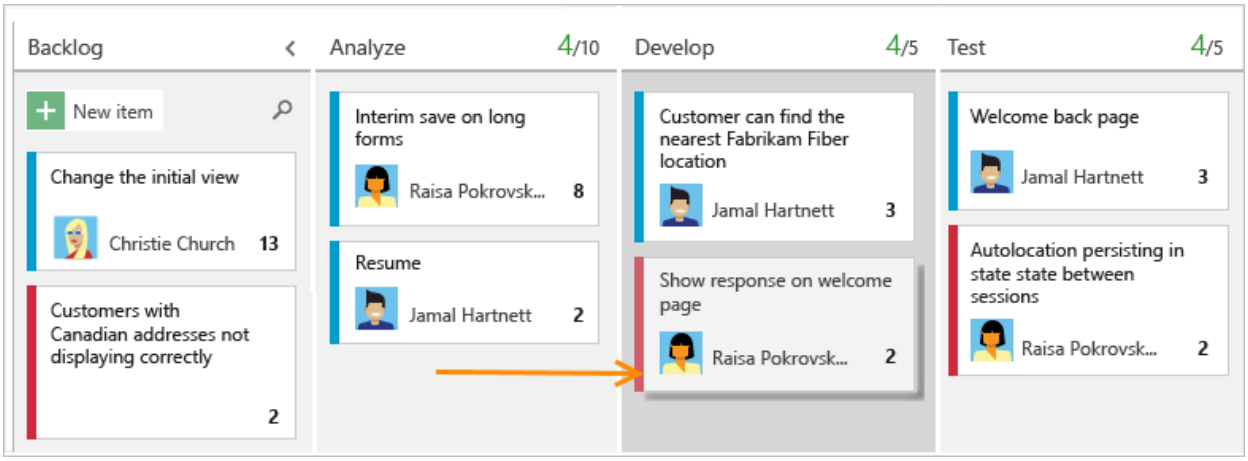
> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than 183 days (about a half a year). You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

NOTE

Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than a year old. You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

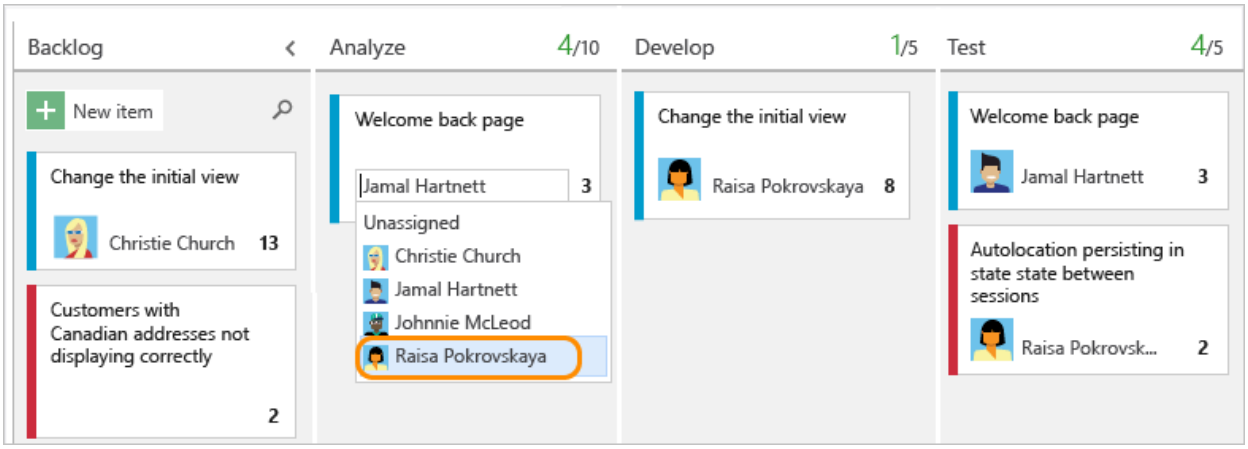
NOTE

Users assigned Stakeholder access aren't able to use the drag-and-drop feature to update status.



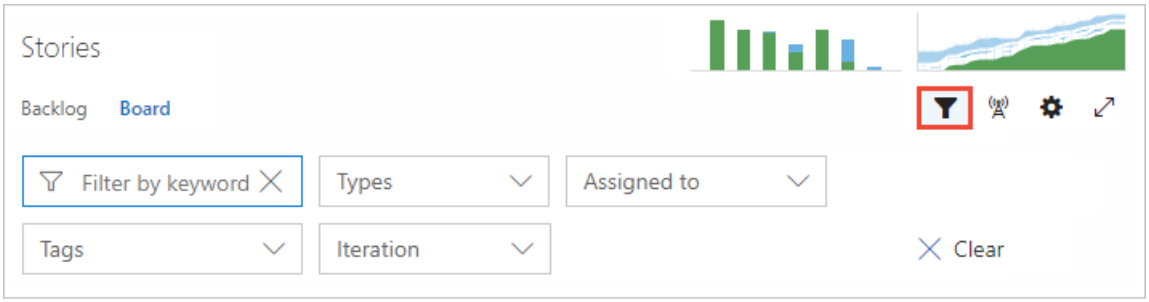
Update fields from the card

You can quickly update a field or reassign ownership directly from the board. If the field you want to update isn't showing, then customize the card to show it.

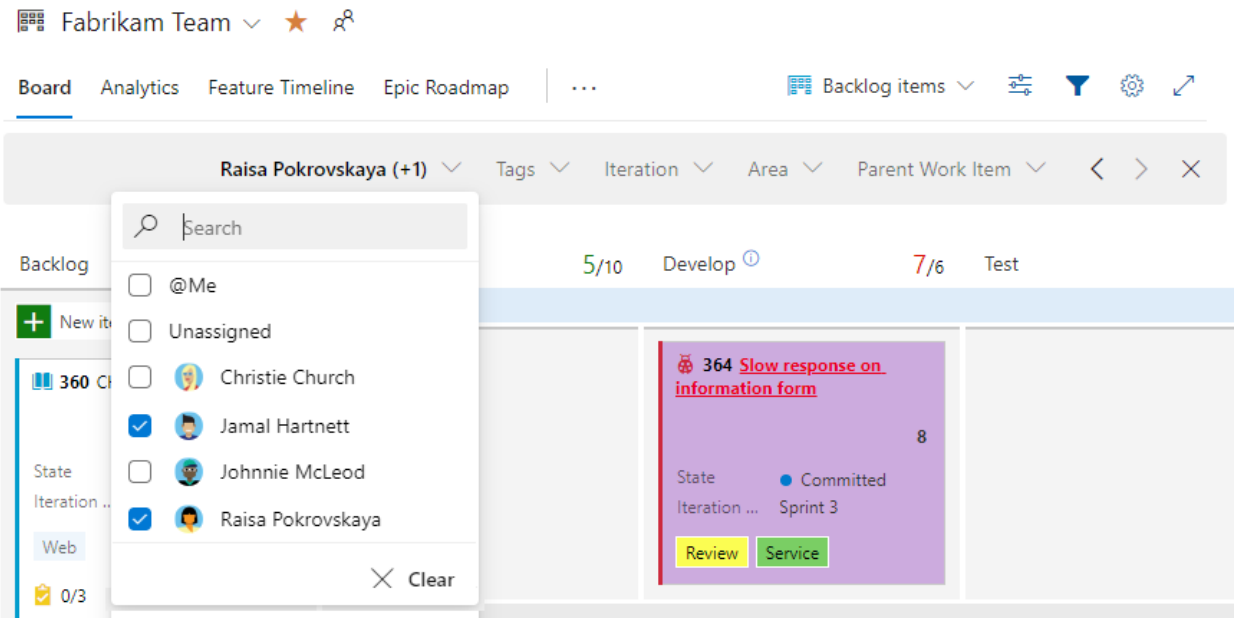


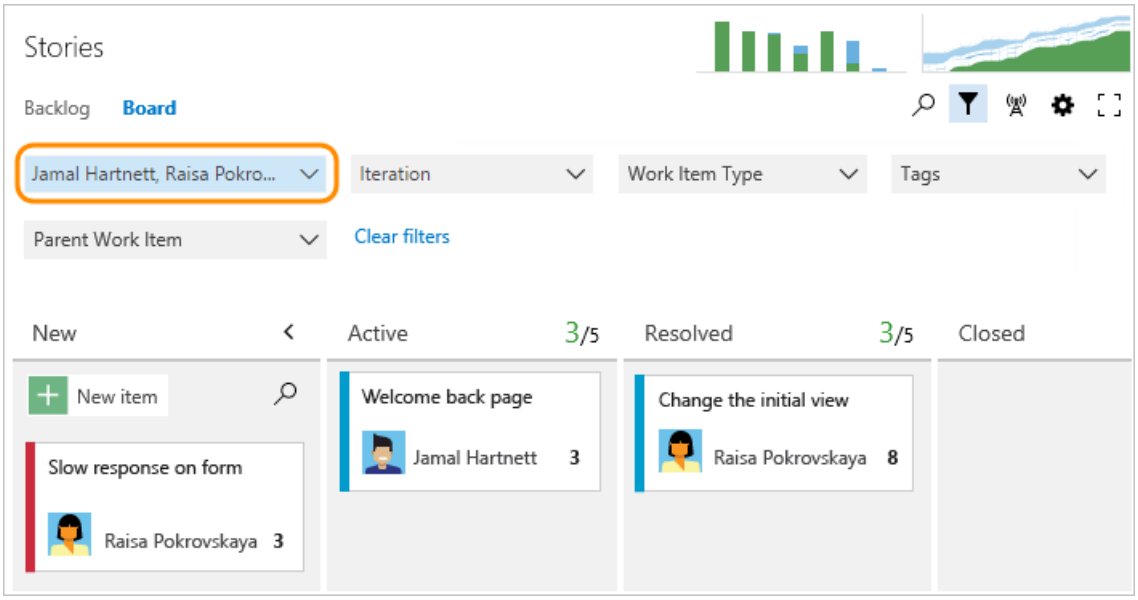
Filter your board using keywords, field values, or tags

You can apply filters interactively to focus on a subset of work. For example, you can filter the board to focus on work assigned to at team member for a specific sprint. To start filtering, choose **Filter** . To learn more about applying filters, see Interactively filter your backlogs, boards, and plans.



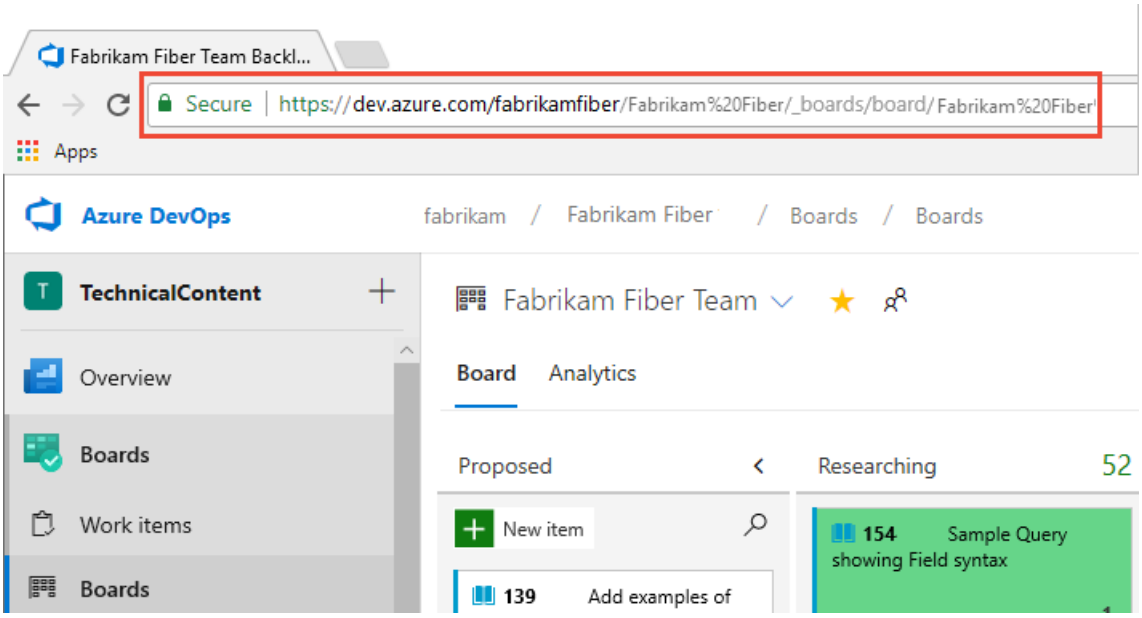
For example, here we filter for all items assigned to Jamal and Raisa.

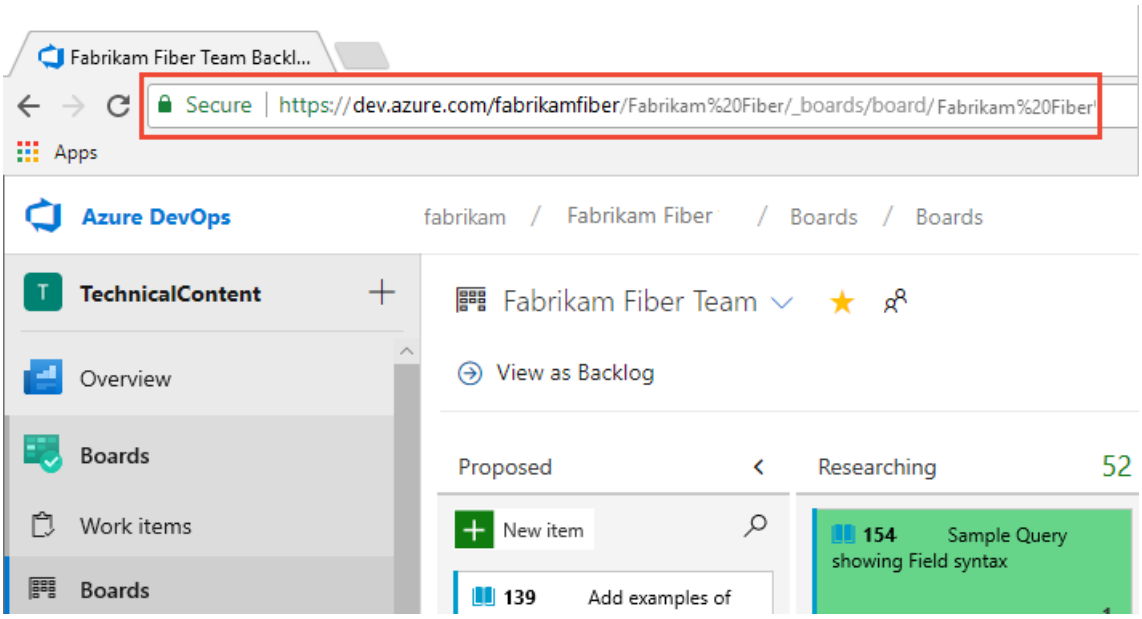


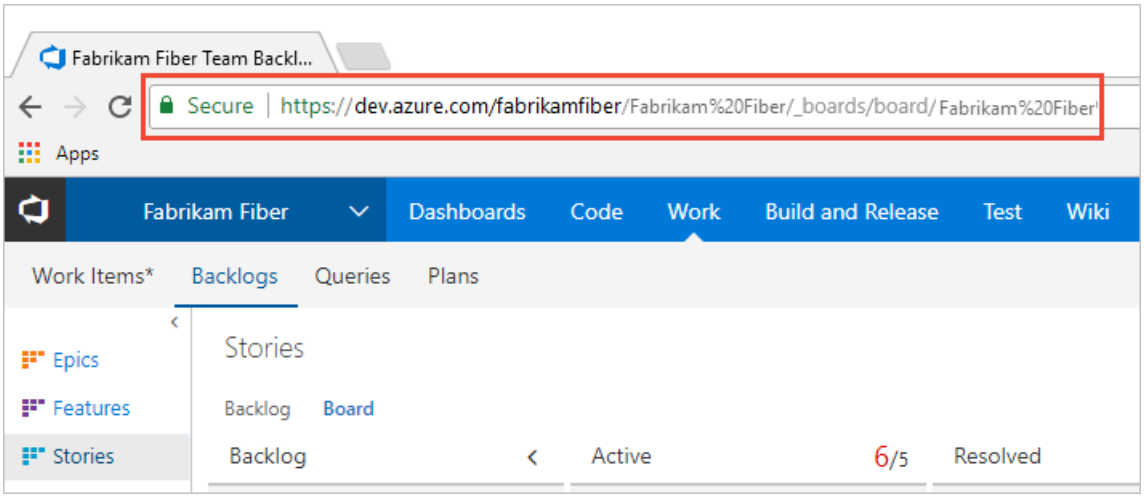


Invite others to work on your Kanban board

All members of a project can view and contribute to your Kanban board. To invite users to contribute, copy the URL of your Kanban board and email it to people you want to invite to your project.







To add users to your project, see Add users to a project.

Try this next

To get the full power of the Kanban board, configure it to map the flow of work and set WIP limits for your team. To configure the Kanban board, you must be added as a team administrator or be a member of the Project Administrators group. If you're the organization owner or creator of the project, then you have these

permissions.

Kanban basics

Related articles

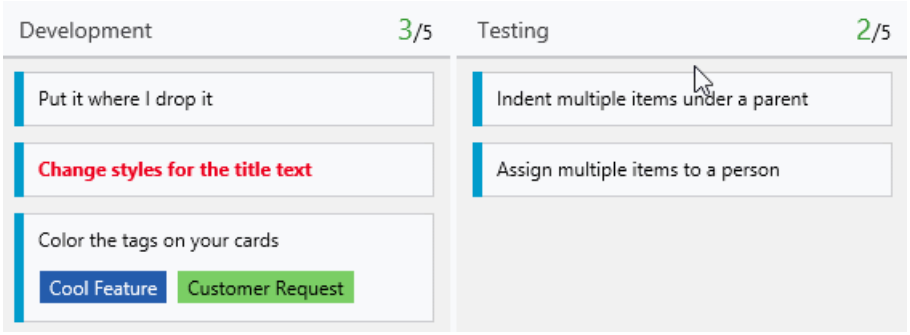
- Interactively filter backlogs, boards, queries, and plans
- Kanban key concepts
- Kanban best practices
- Cumulative flow diagram
- Cumulative flow, lead time, and cycle time guidance

Reorder cards on your Kanban board

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can drag any work item to any column or swimlane on the Kanban board. You can even change the order of items as you move a card to a new column.



In addition to the dynamic card reordering, you can also move a card to a specific column position.

> **NOTE**
> The last column, typically the **Closed** or **Done** column, is always ordered by *Closed Date* with the most recently closed items appearing towards the top of the column. In all other columns, cards are ordered by the backlog order or they're reordered based on the Card reordering setting selected.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

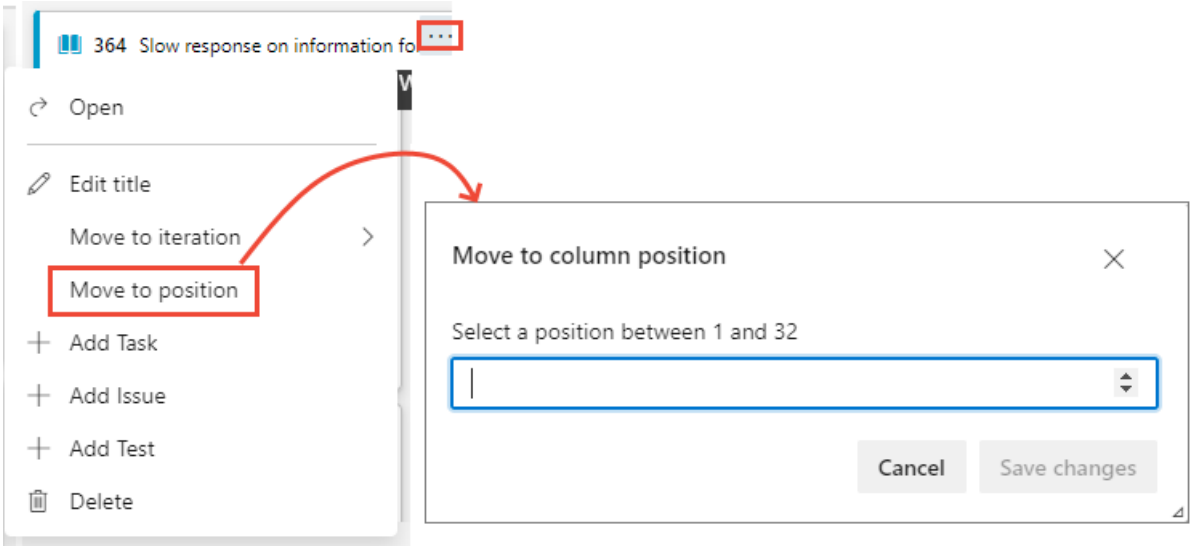
- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Move a card to a specific column position

You can re-order the work items within a Kanban board column by choosing ...**Work items action menu**, selecting **Move to position**, and then specifying a value in the dialog.

> **NOTE**
> The **Move to column position** feature requires you to enable the **New Boards Hub** preview feature. To enable this feature, see Manage or enable features.

Specify a value within the range listed, which corresponds to the number of items currently in the column.



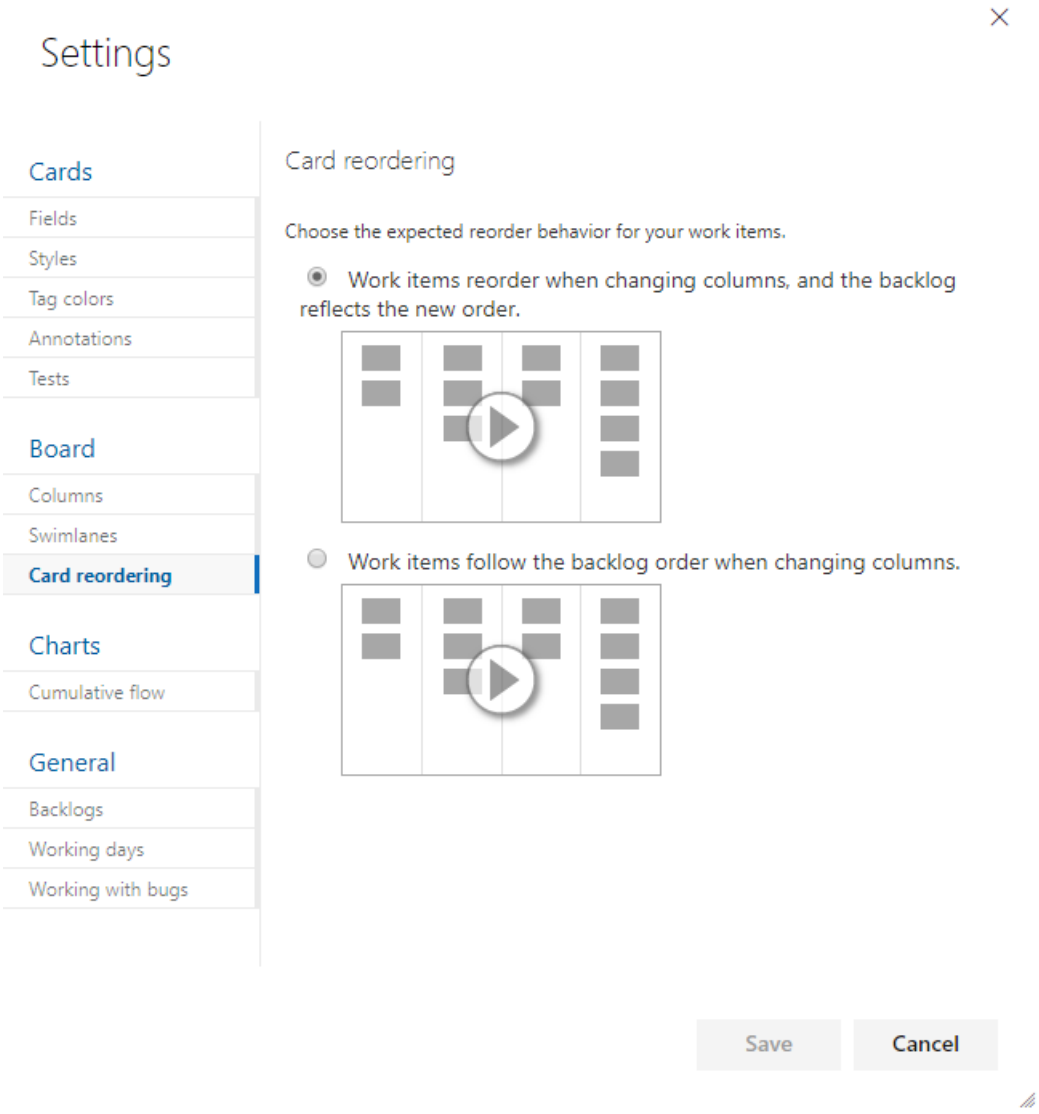
Set the team preference for reordering cards

If you want to preserve the backlog priority when you move a card to a new column, you can change the Kanban board card reordering setting for your team.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose the gear icon to configure the board and set general team settings.



3. Choose **Card reordering** and select from the two reordering behaviors listed.



The setting you choose applies to all active Kanban boards for your team.

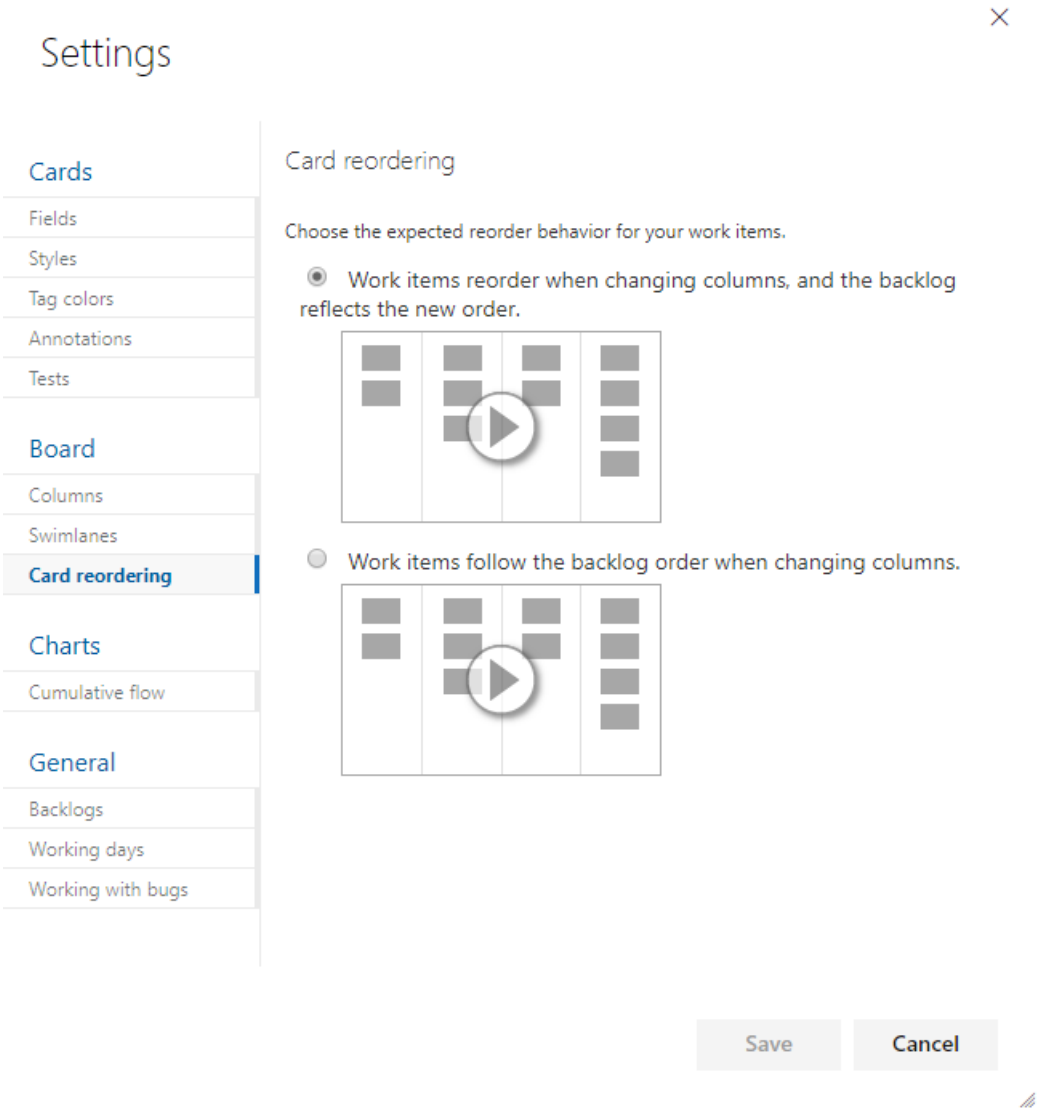
4. When done with your changes, choose **Save**.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.

2. Choose ⚙ to open the common configuration settings dialog for the Kanban board.



3. Choose **Card reordering** and select from the two reordering behaviors listed.



The setting you choose applies to all active Kanban boards for your team.

4. When done with your changes, choose **Save**.

> **TIP**
> You can drag-and-drop work items onto a sprint from any backlog or board. To add sprints to a team backlog, see Define iteration (sprint) paths and configure team iterations.

Related articles

- Backlog priority or stack rank order
- Customize cards

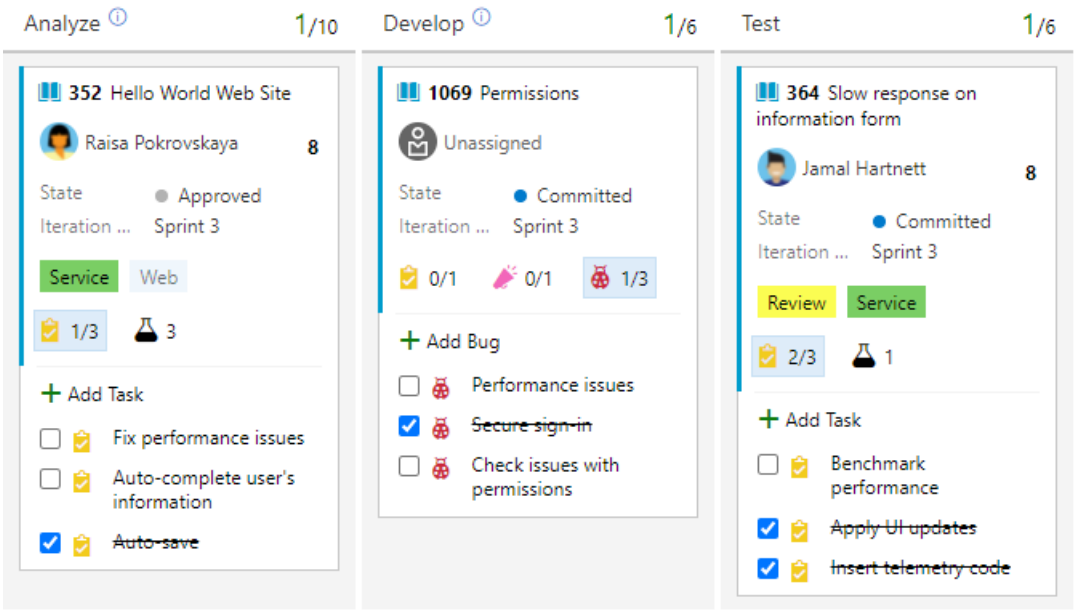
Add tasks or child items as checklist items

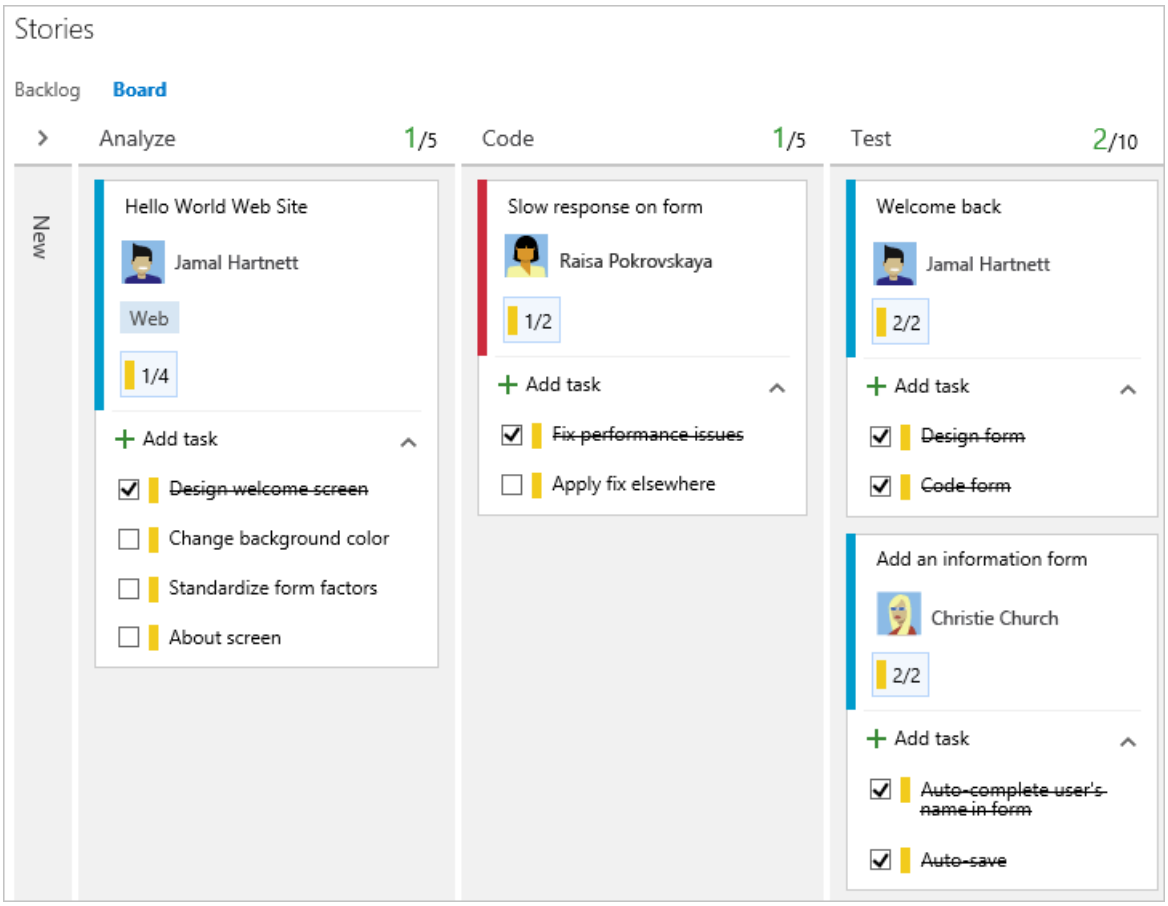
10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Many teams find Kanban boards ideal for tracking work. Kanban boards are ideal because they support visualization the flow of work that is in progress. It also allows team members to quickly add new items and update work item status in a Kanban board. If you're new to working with the Kanban board, see Kanban basics.

With checklists or to do lists, you continue to enjoy lightweight tracking. You gain visibility into which tasks, bugs, or other child items are in progress or completed. For example, here we show several tasks and bugs for work in progress, both yet to do and those items marked as completed. By adding the Issue work item type to the Iteration backlog, issues can be added as checklists.





In this article, you'll learn:

- Summary of checklist features
- How to add checklist items from your Kanban board
- How to mark a checklist item as done
- How to expand or collapse a checklist
- How to reorder and reparent checklist items or reassign them to a sprint

Overview of checklist features

Checklists are a feature of all Kanban boards, both product and portfolio backlog levels.

- All Kanban board levels support checklists. For a view of default backlog hierarchies, see Plan and track work.
- Adding a checklist item automatically adds a parent-child link between the parent work item and the checklist item.
- For the product-level board:
  - When **Bugs are managed with tasks**, they're available to add and track within a checklist.
  - When **Bugs are managed with requirements**, you can add tasks to bugs that are tracked on the board. Teams make this choice via Board settings, **Working with bugs**.
- Marking any checklist item as "done" moves the work item **State** to done, closed, or completed.
- Teams can remove select checklist features by disabling them on the **Annotations** tab of the Board settings.
- Tasks or other child items that you create from the Kanban board are automatically assigned to the sprint/iteration path of the parent work item under which you define them.
- Any work item type added to the backlog hierarchy is available to add using the checklist feature.
  To add work item types to track as checklists, add them to the next-lower hierarchical backlog. To learn how, see Customize your backlogs or boards (Inheritance process) or Process configuration XML element reference (On-premises XML process).

TIP

If you find that you don't use this feature, you can disable it from the common configurations dialog.

Prerequisites

Boards are automatically created when you create a project or add a team. Each team has access to their own product and portfolio boards as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.
- Users with **Stakeholder** access for a public project have full access to board features just like users with **Basic** access.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access can't exercise these board features: add work items, drag-and-drop work items to update status, or update fields displayed on cards. They can add tasks and change task status.

NOTE

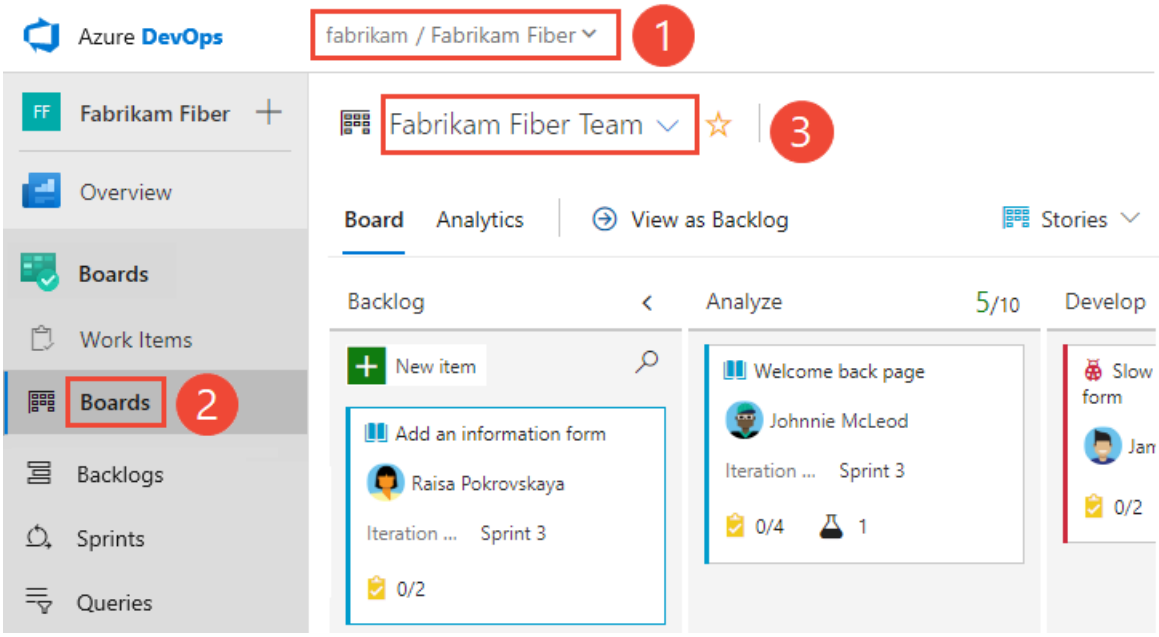
Both Kanban boards and Taskboards support visualizing the flow of work and monitoring metrics to optimize that flow. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard.

Open your Kanban board from the web portal

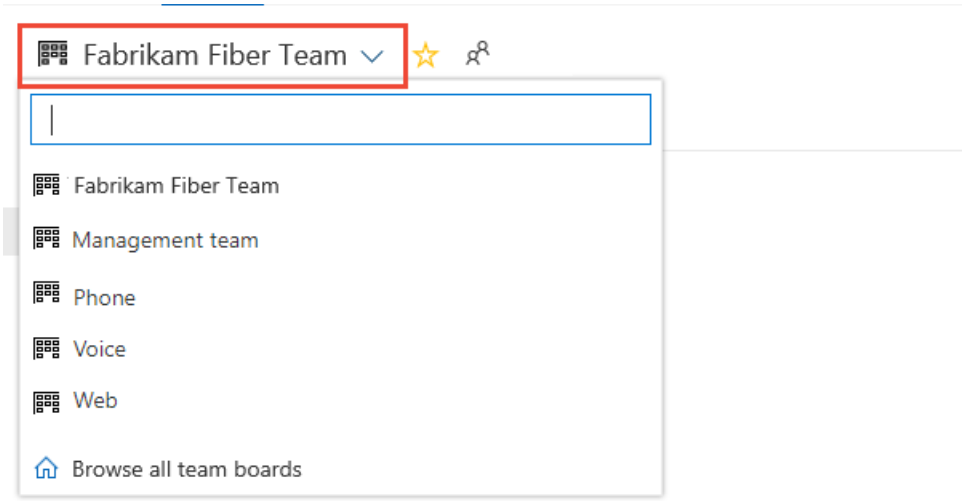
Your Kanban board is one of two types of boards available to you. The other is the sprint Taskboard. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can

monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard. For an overview of the features supported on each backlog and board, see Backlogs, boards, and plans.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



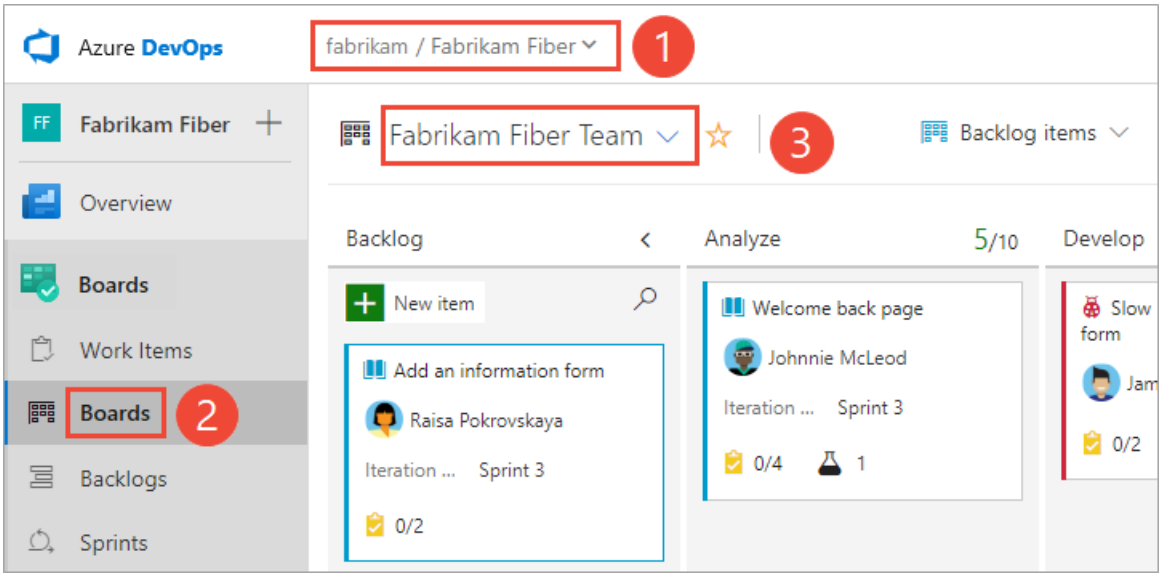
> **TIP**
> Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI as the backlog level.

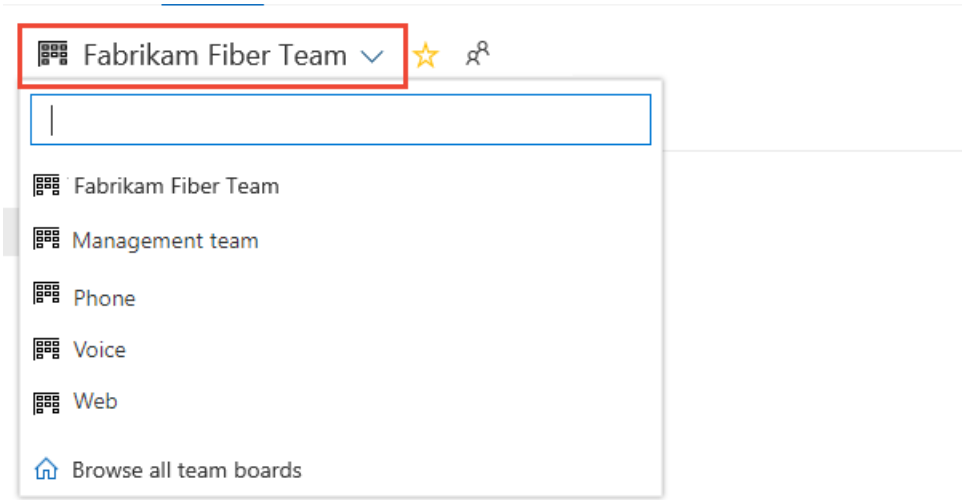


To switch to the product backlog, select **Stories backlog**. To switch to a Taskboard, see Update and monitor your Taskboard.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



TIP

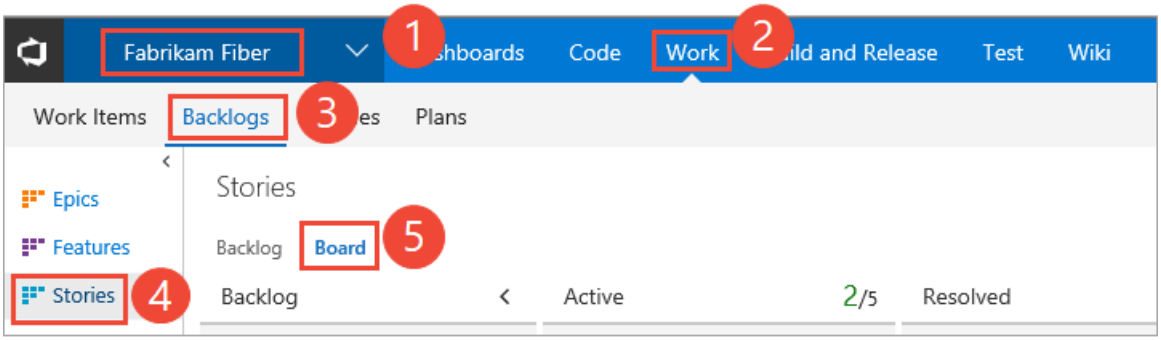
Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI as the backlog level.

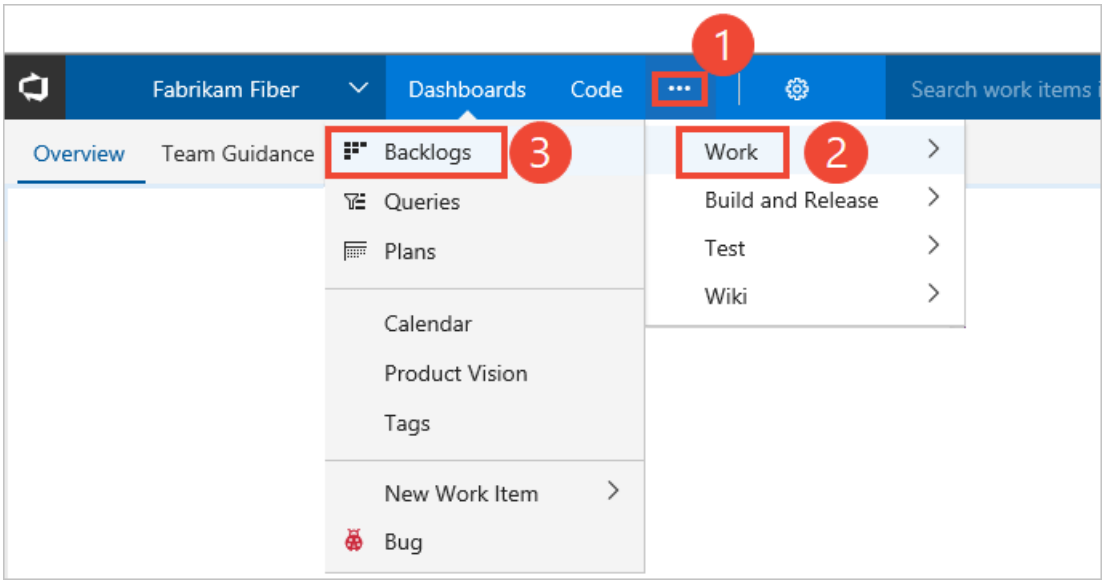


To switch to the product backlog, select **Stories backlog**. To switch to a Taskboard, see Update and monitor your Taskboard.

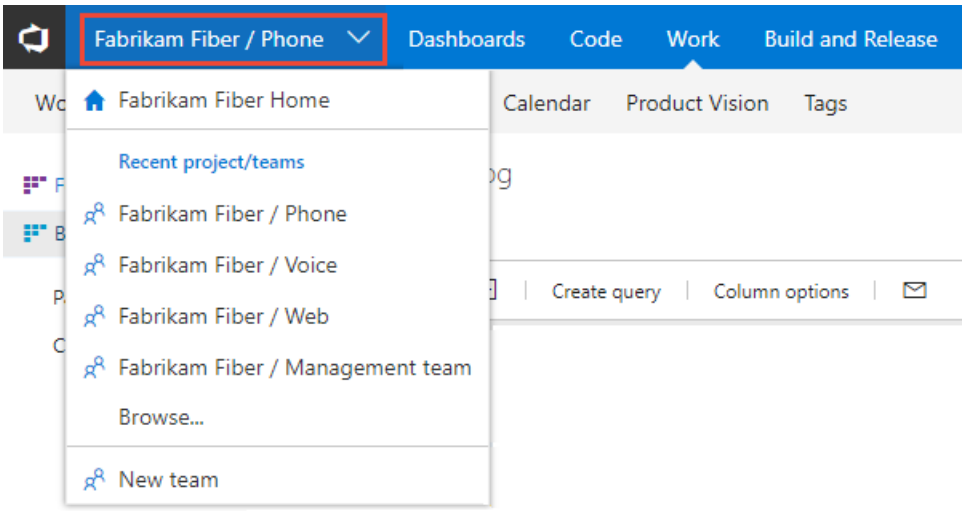
1. To view your Kanban board, open your project from a web browser. Select **Work** > **Backlogs** > **Stories**, and then select **Board**.



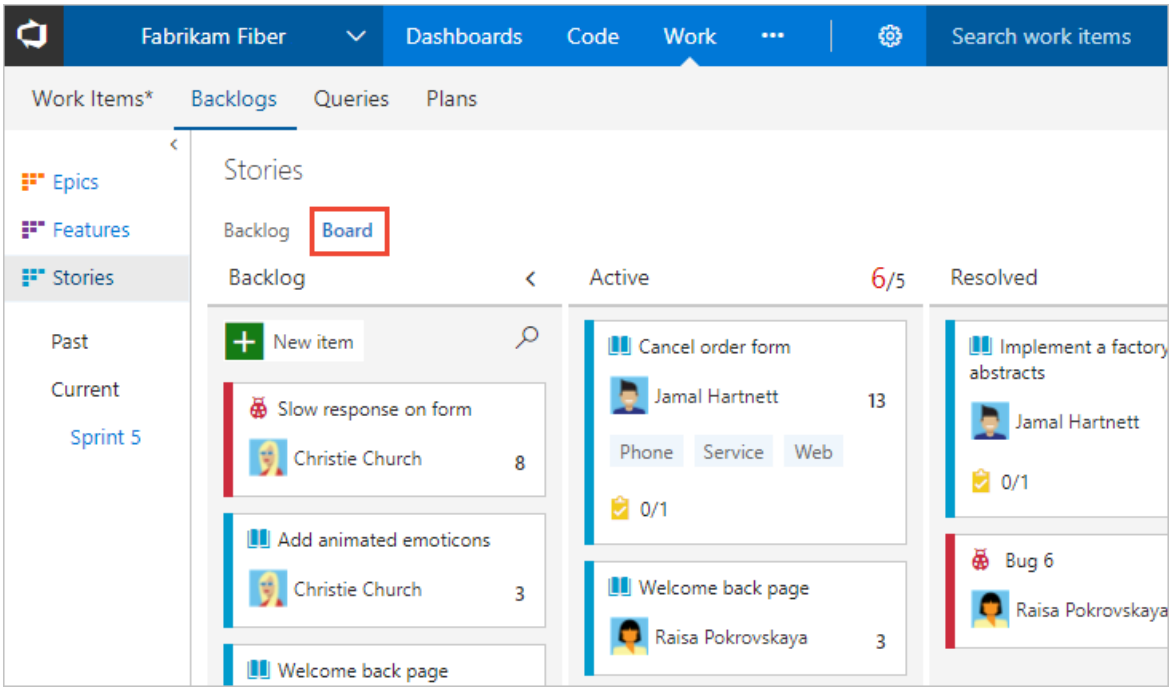
If you don't see **Work**, your screen size might be reduced. Select the three dots ( ) icon. Then select **Work** > **Backlogs** > **Board**.



2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.



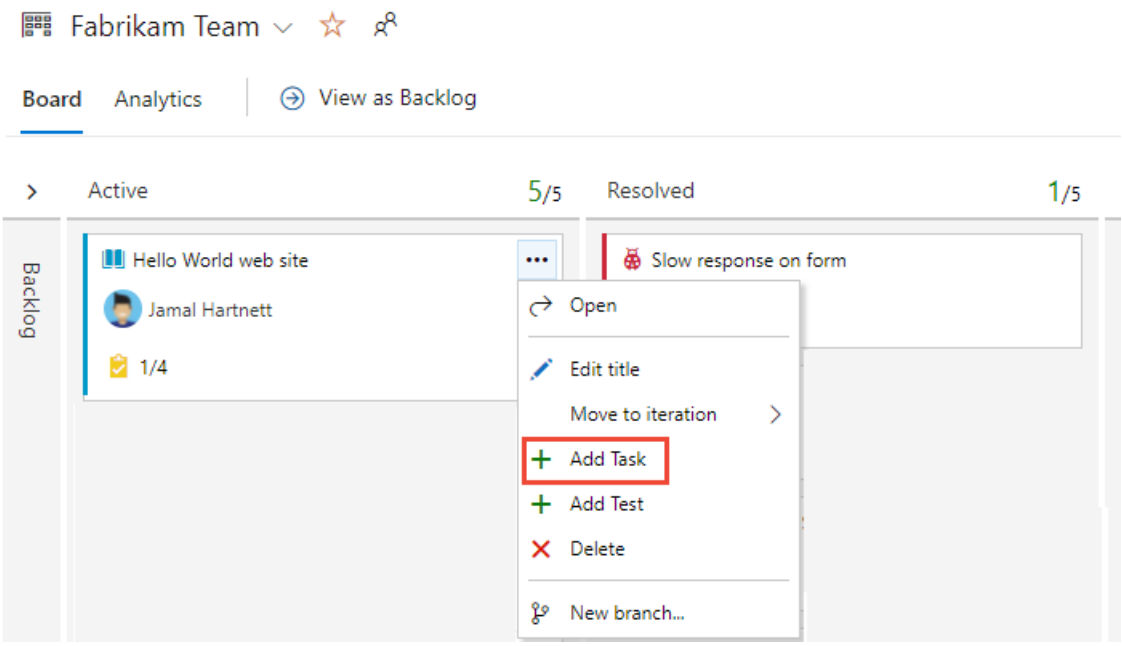
Your Kanban board appears.

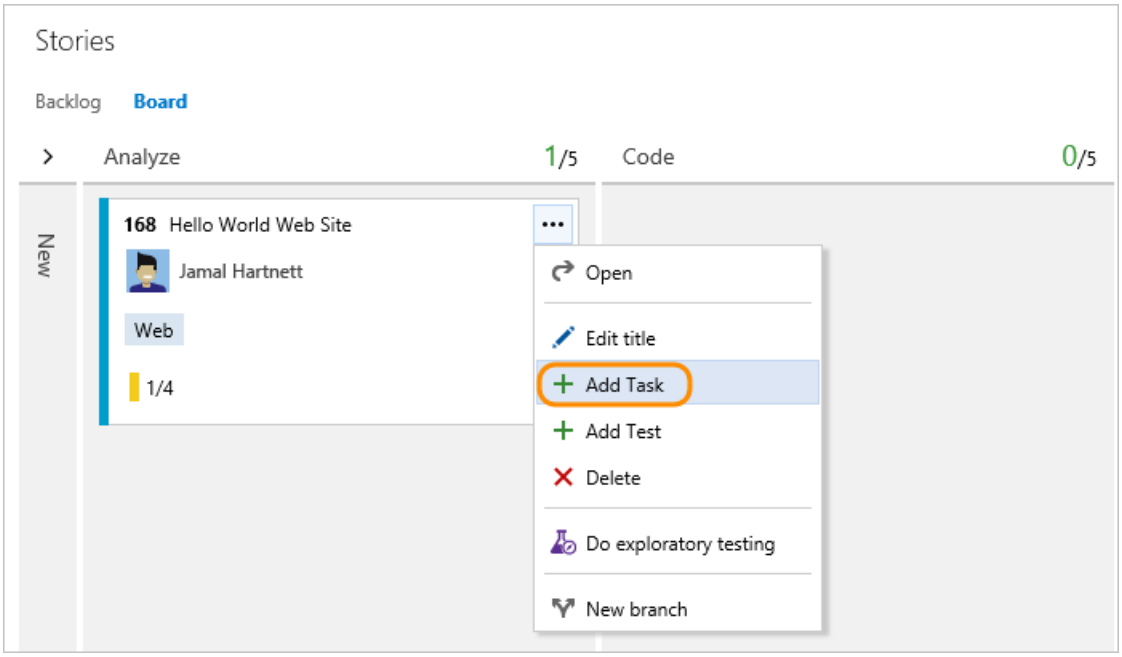


Add one or more child items to a checklist

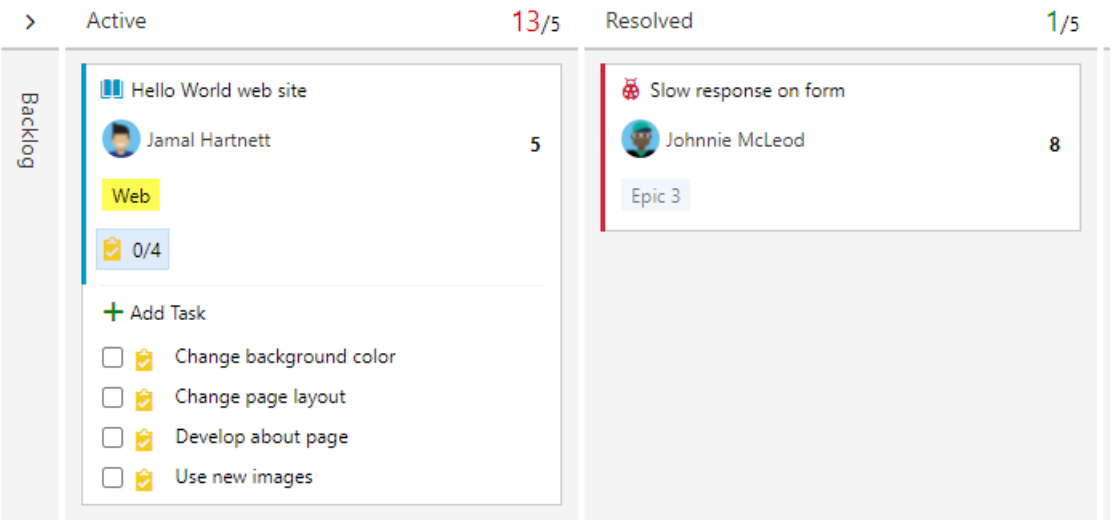
In this example, tasks are added to the product Kanban board using the checklist feature. You can use the same procedures to add any other supported checklist item from your board.

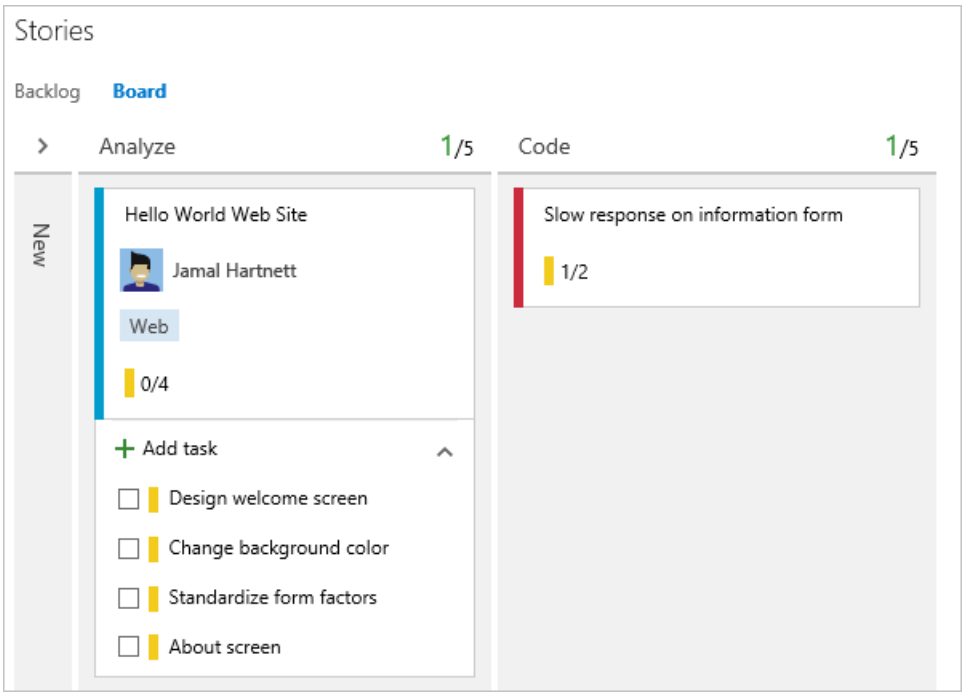
1. To start adding tasks, open the menu for the work item.



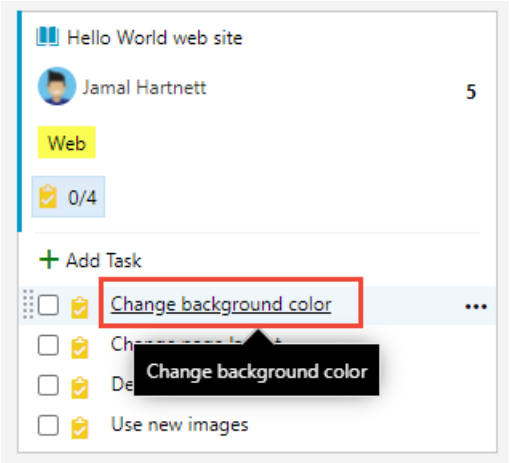


2. If you have many tasks to add, keep typing their titles and choose Enter after each title.





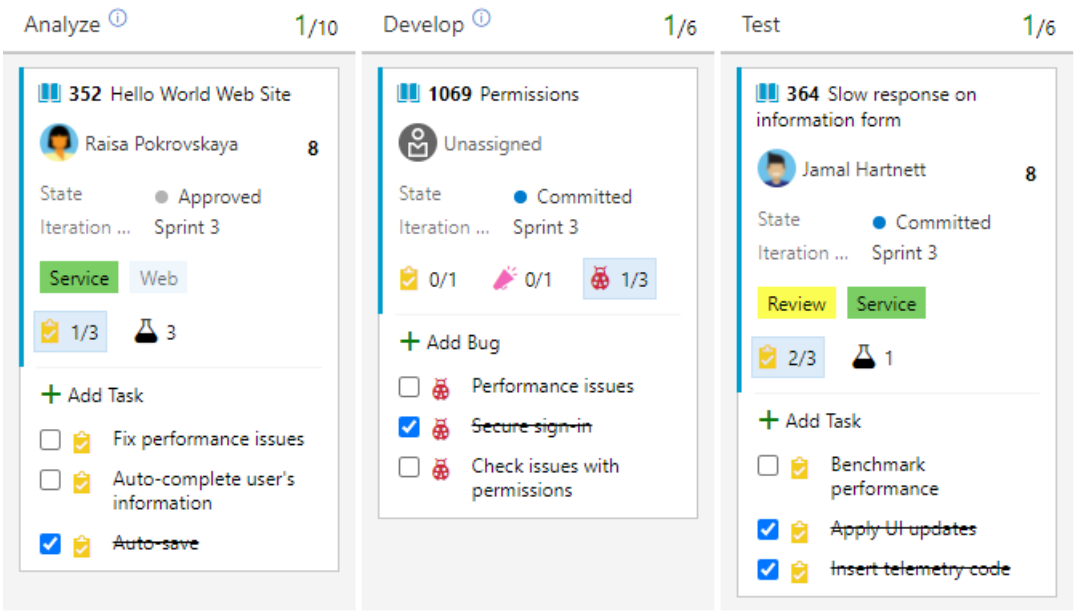
3. If you have details you want to add about a task, open the item by choosing the title.

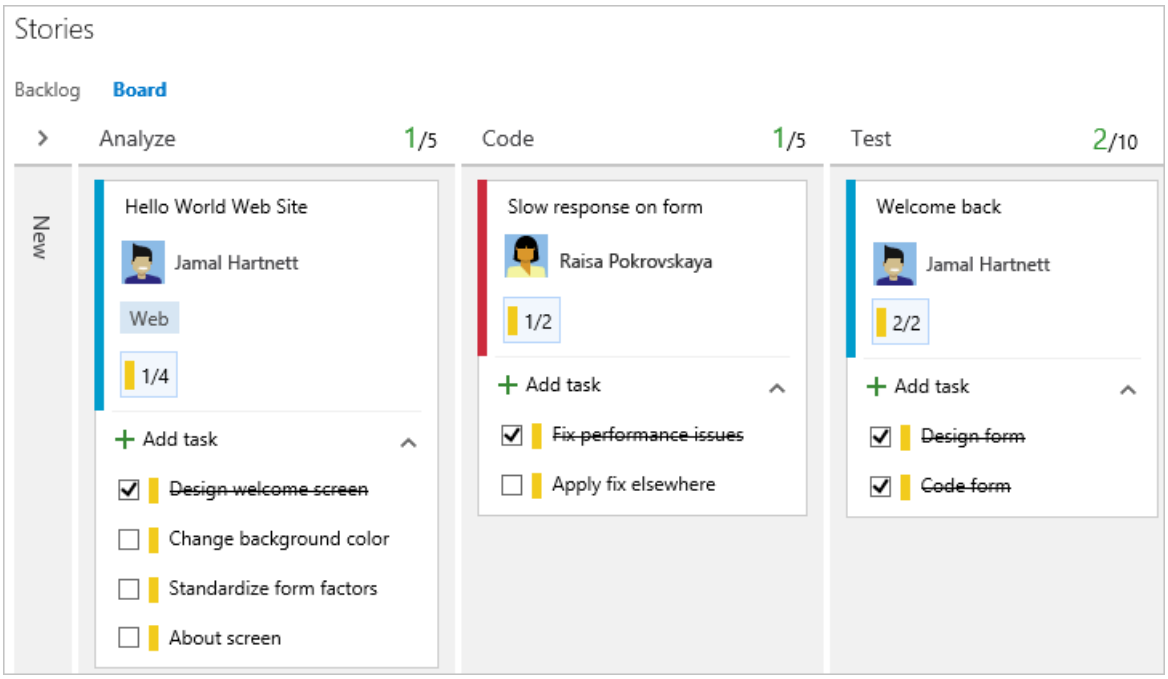


> **NOTE**
> Tasks that you create from the Kanban board will show up on your sprint Taskboard. Also, tasks that you create from the sprint backlog or taskboard will show up within tasks checklists on the Kanban board.

Mark a checklist item as done

When you complete a task or other checklist item, choose the checkbox to change its status to Done, Closed, or Completed.



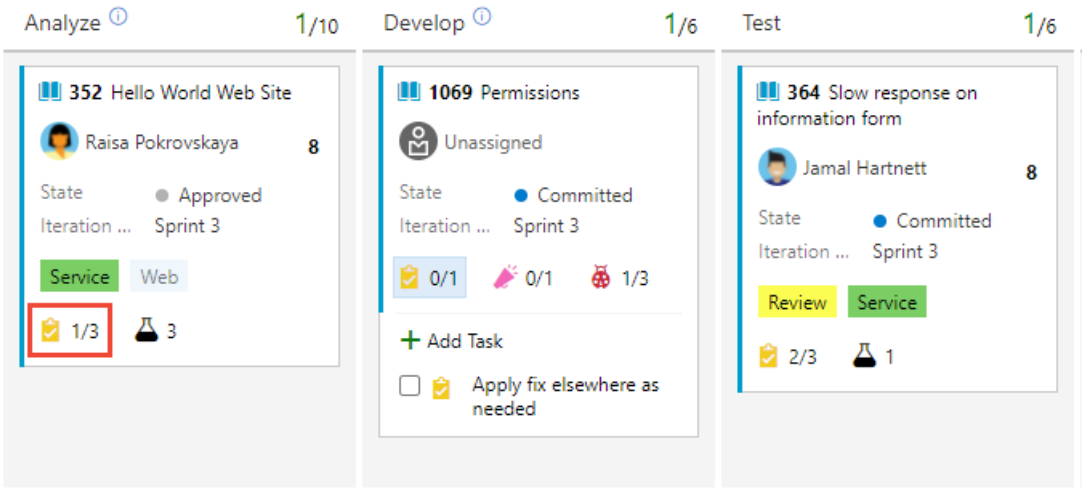


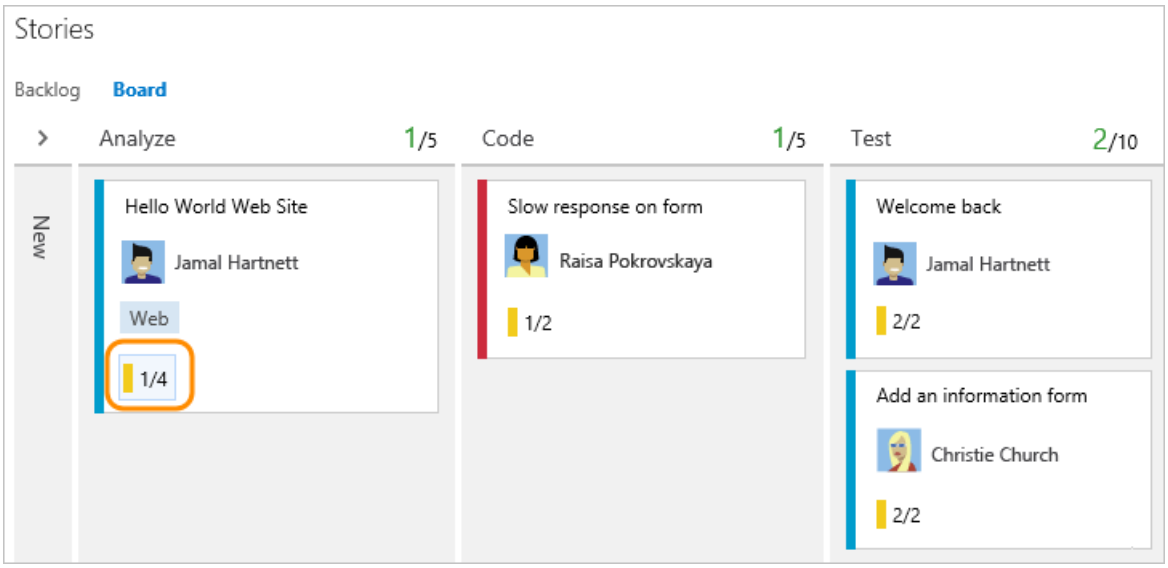
The State of the work item is updated from **Active** to **Closed** for Agile and CMMI projects and from **To Do** to **Done** for Scrum projects, and

> **TIP**
> No matter the number of workflow states a checklist item has, checking it moves it to its closed state.

Expand or collapse a checklist on a Kanban board

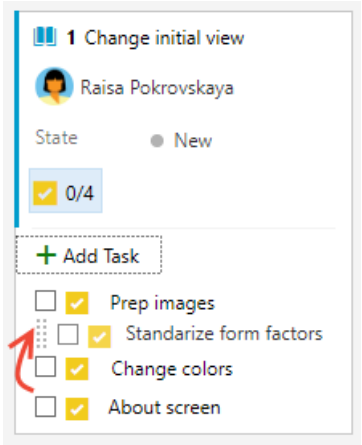
Upon first opening the Kanban board, you'll see an unexpanded view of checklists. Choose the checklist summary to expand a collapsed checklist. Choose the same summary to collapse an expanded checklist.

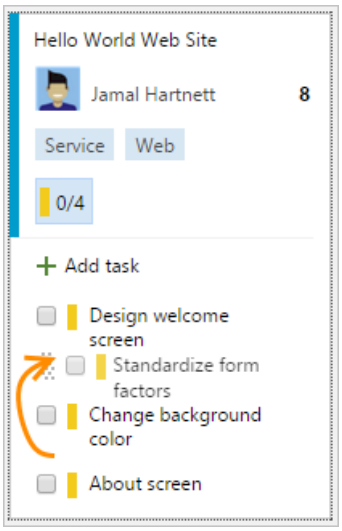




Reorder tasks, reparent tasks, or reassign tasks to a sprint

You can drag a task within a work item to reorder it. Or, you can drag the task to another work item on the Kanban board to reparent it.





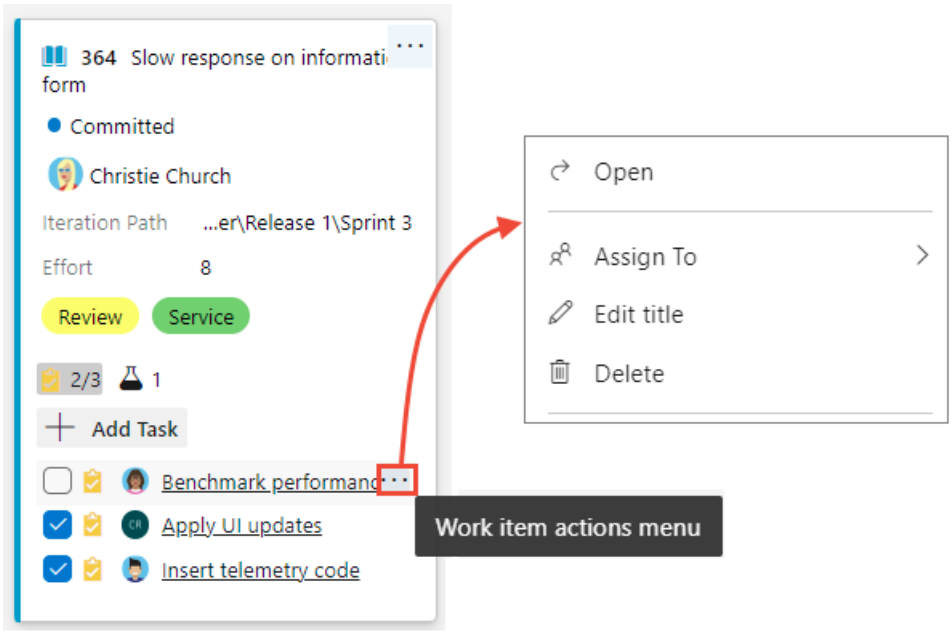
> **NOTE**
> Users with **Stakeholder** access can't drag-and-drop tasks or reorder and reparent tasks.

Tasks or other child items you add as checklists are automatically assigned to the **Iteration Path** of their parent work item. To reassign a checklist item to a different sprint, you must open the item and change its **Iteration Path**. Or, open the sprint backlogs where it's currently defined and drag it to the new sprint using the Planning pane. For details, see Assign backlog items to a sprint.

Reassign a checklist item

Checklist items show the avatars of those assigned to the item. You can view the avatar assignment of checklist items, or reassign a checklist item by choosing the item's ...**Work items action menu** and selecting **Assigned to**.

> **NOTE**
> Avatar images and the **Assign to** menu option requires you to enable the **New Boards Hub** preview feature. To enable this feature, see Manage or enable features.



Configure the Kanban board

To configure or change the layout of the Kanban board, see Customize your boards.

Related articles

- Kanban board features and epics
- Interactively filter backlogs, boards, queries, and plans
- Add, run, update manual tests
- Create a new branch, drive Git development
- Kanban board controls

REST API resources

To programmatically create work items, see the REST API, Work Items reference.

Add Kanban board features and epics

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

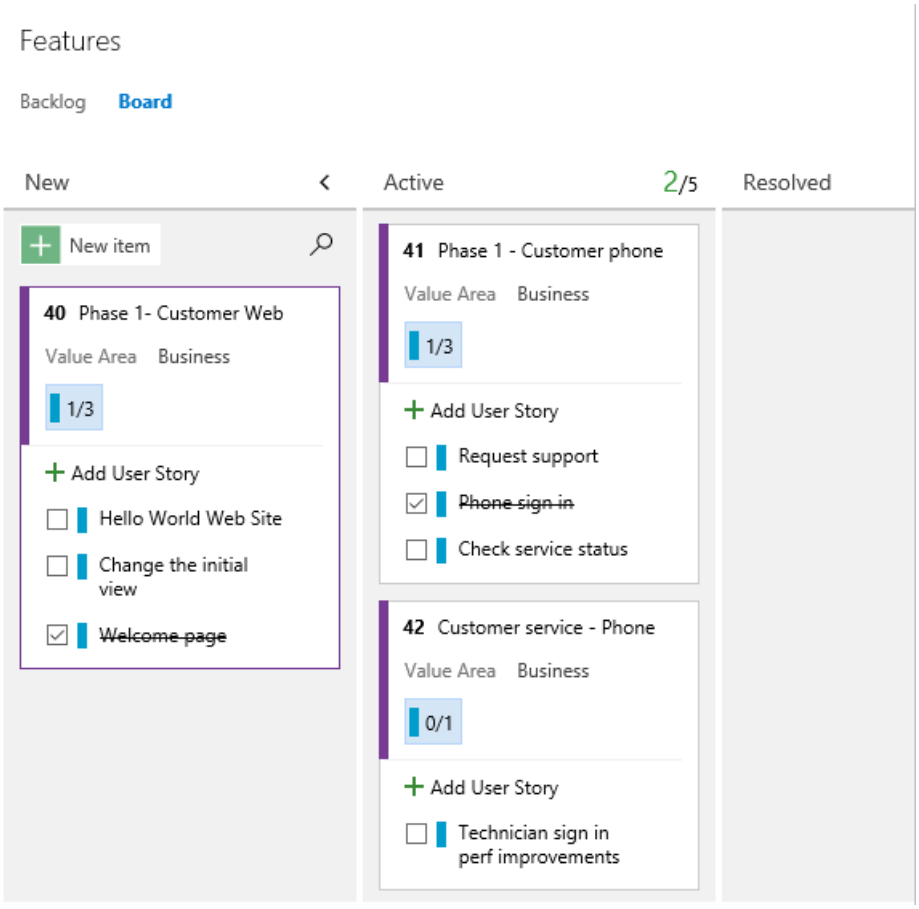
If you track progress on your backlog with Kanban, you can also use Kanban boards to track epics and features.

And, as with child task checklists for backlog items, you can quickly define and track the progress of child items for your features or epics. Here we see several stories defined for features, both in progress and completed.

In this article, you'll learn:

- How to add epics and features using your portfolio backlogs
- Keyboard shortcuts for working with the Kanban board

For information on managing features and epics as a list and examples for features and epics, see Define features and epics.



Prerequisites

Boards are automatically created when you create a project or add a team. Each team has access to their own product and portfolio boards as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node**

permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.

- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.
- Users with **Stakeholder** access for a public project have full access to board features just like users with **Basic** access.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access can't exercise these board features: add work items, drag-and-drop work items to update status, or update fields displayed on cards. They can add tasks and change task status.

Open your Kanban board from the web portal

Your Kanban board is one of two types of boards available to you. For an overview of the features supported on each backlog and board, see Backlogs, boards, and plans. To switch to the product backlog, choose **Stories backlog**. And, to switch to the taskboard, choose **Sprints** and then choose **Taskboard**.

1. (1) Check that you've selected the right project, (2) choose **Boards > Boards**, and then (3) select the correct team from the team selector menu.

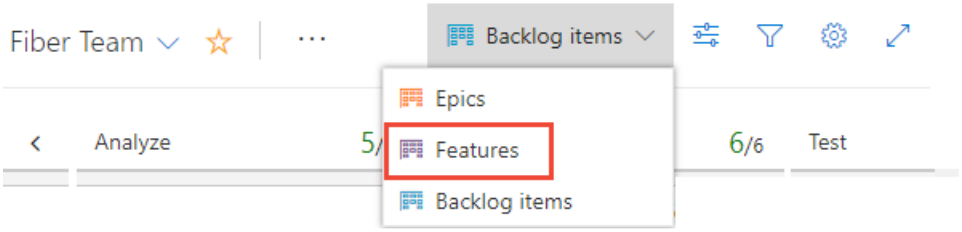
   

   To choose another team's board, open the selector and select a different team or choose the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

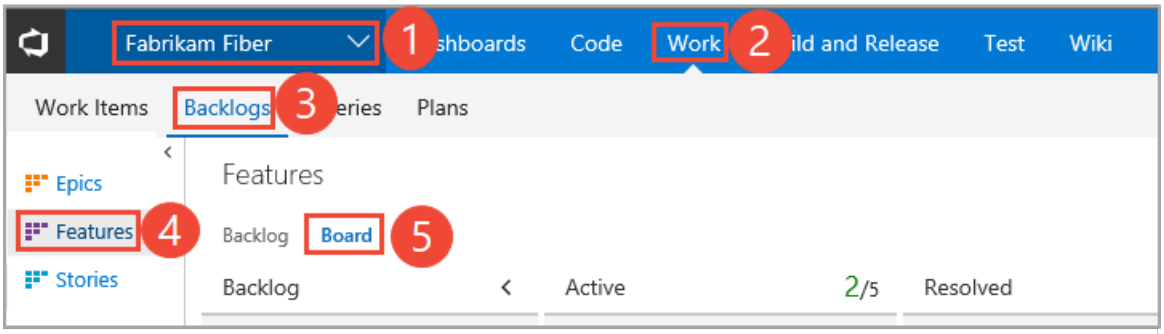
   

   > **TIP**
   > Choose the ☆ star icon to favorite a team board. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

2. Select **Features** or **Epics** from the backlog selector menu.



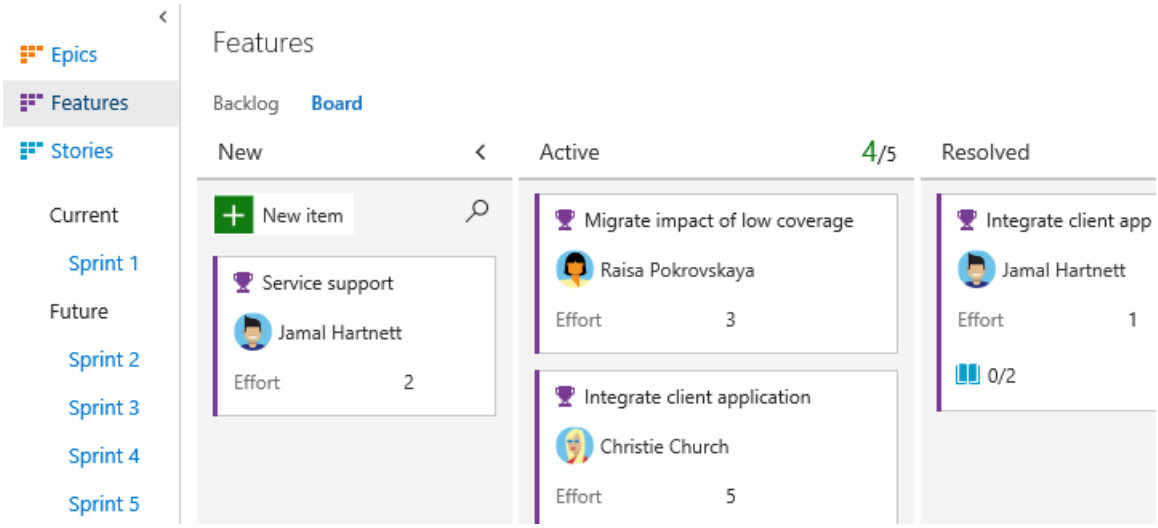
1. To view a portfolio backlog, open your (1) project from a web browser and choose (2) **Work**, (3) **Backlogs**, (4) **Features** or **Epics**, and then (5) **Board**.



If you don't see **Work**, your screen size may be reduced. Select the three dots ( ), then choose **Work**, **Backlogs**, and then **Board**.

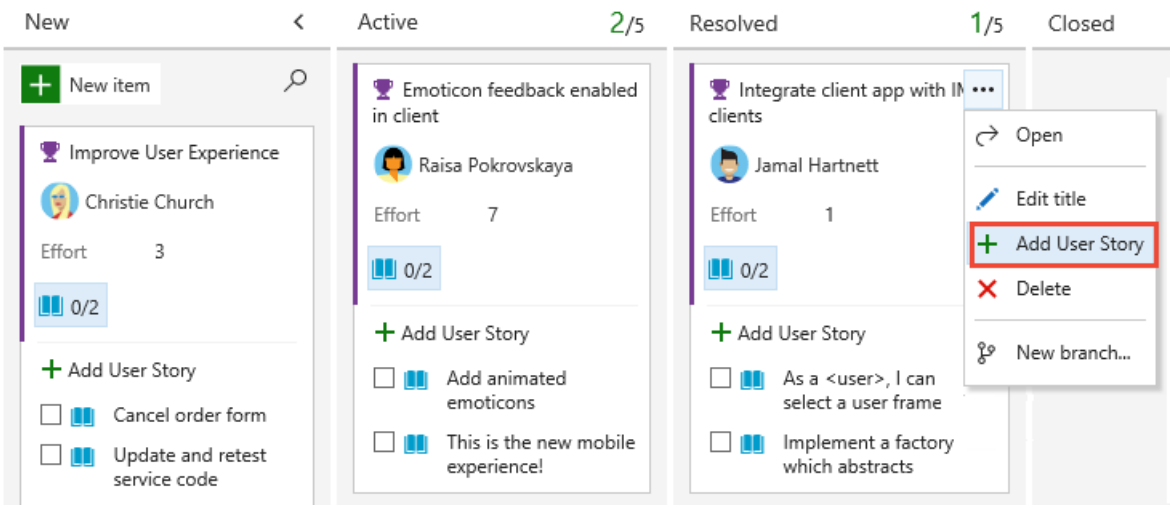
2. To choose another team, open the project/team selector and select a different team or choose the **Browse** option.

3. Your Kanban board for the selected portfolio backlog displays.



Add epics or features

Add new items to a feature or epic through the item's ··· Action menu. For descriptions of fields used to support features and epics, see Define features and epics.



When you have many items to add, keep typing your task titles and press Enter. If you have details you want to add about to a work item, hover over the item and press Enter.

Related articles

If you're new to working with the Kanban board, see Kanban basics

For more information on working with a checklist on a Kanban board, see Add tasks or child items as checklists. You can run the same operations for the features and epics Kanban boards as you do with the Kanban board for the product backlog. These operations include:

- Mark an item as done
- Reorder and reparent work items

To customize the columns, swimlanes, or cards for each Kanban board, make sure you first select the board. Then, choose the or gear icon to open the Settings dialog. See these articles for details:

- Add columns
- Customize cards

REST API resources

To programmatically interact with Kanban board and other team settings, see the REST API, Boards reference.

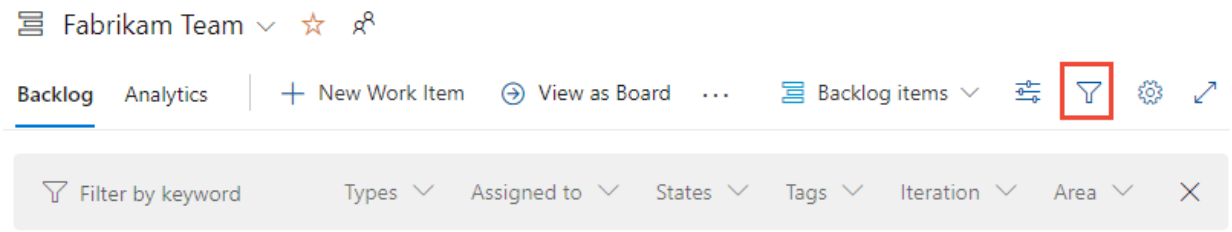
Interactively filter backlogs, boards, queries, and plans in Azure Boards

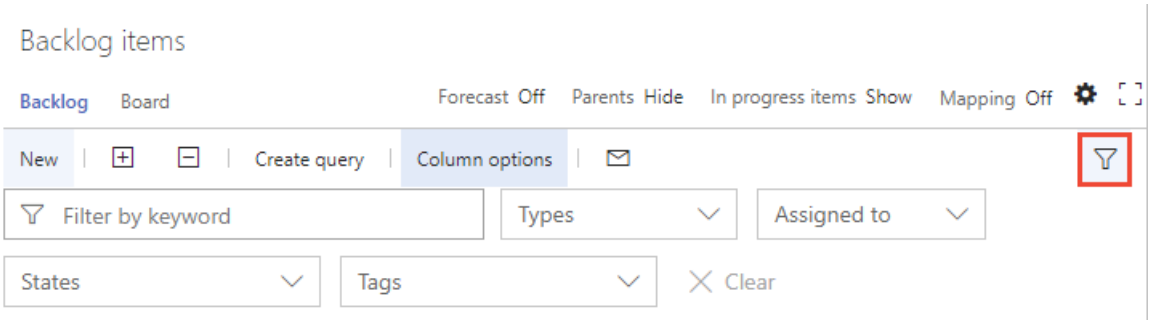
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

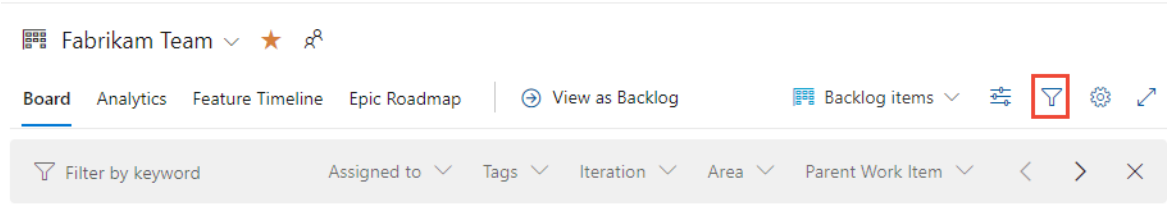
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

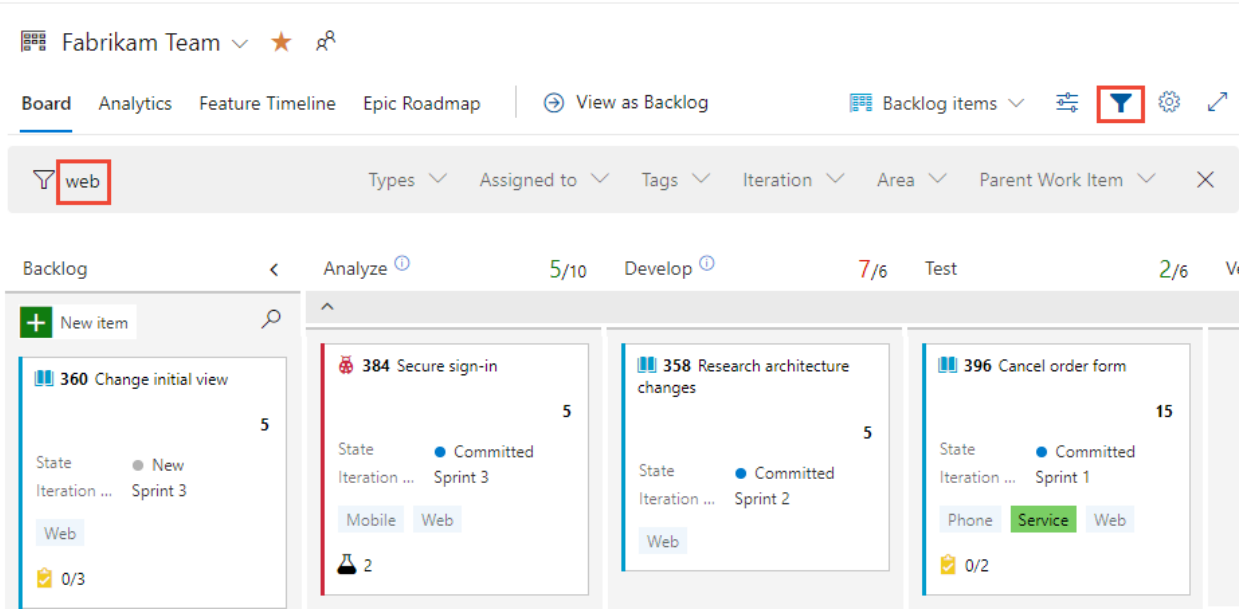
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

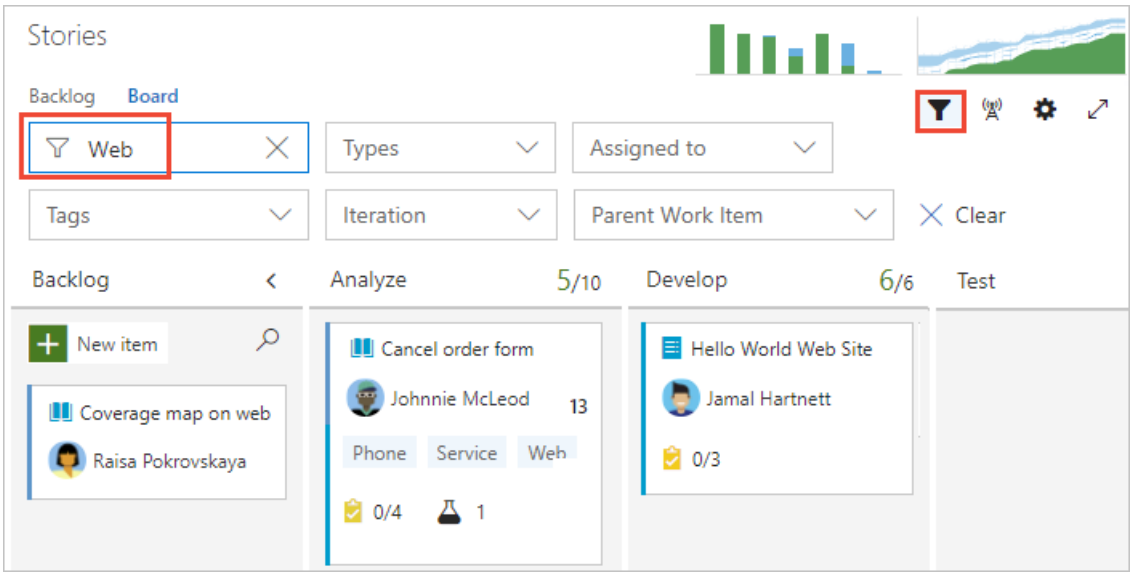
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

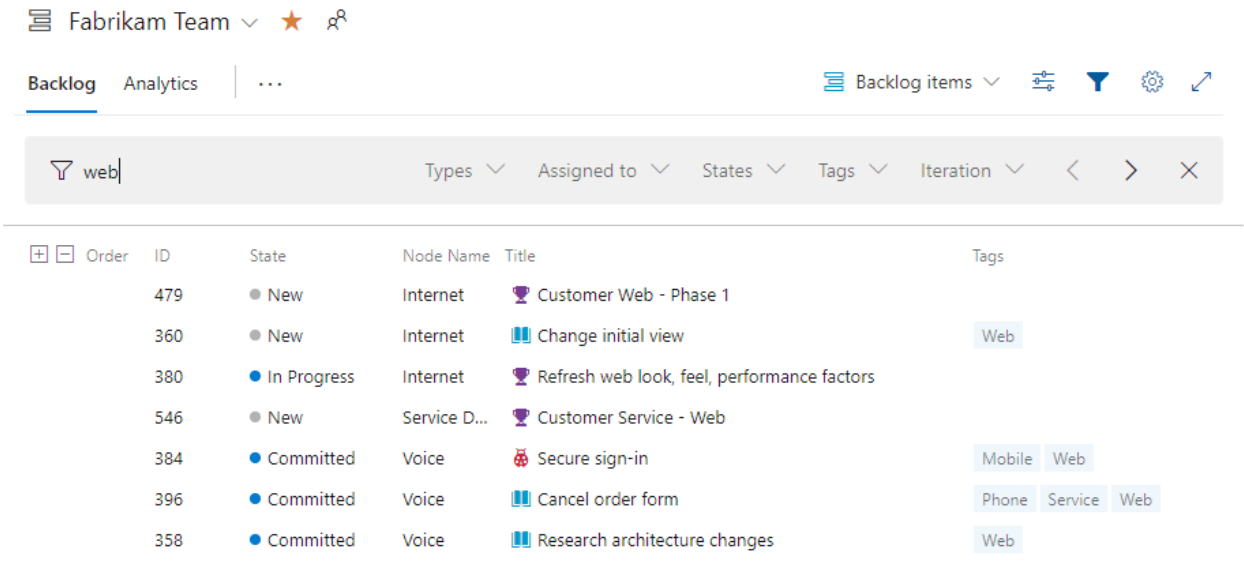


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

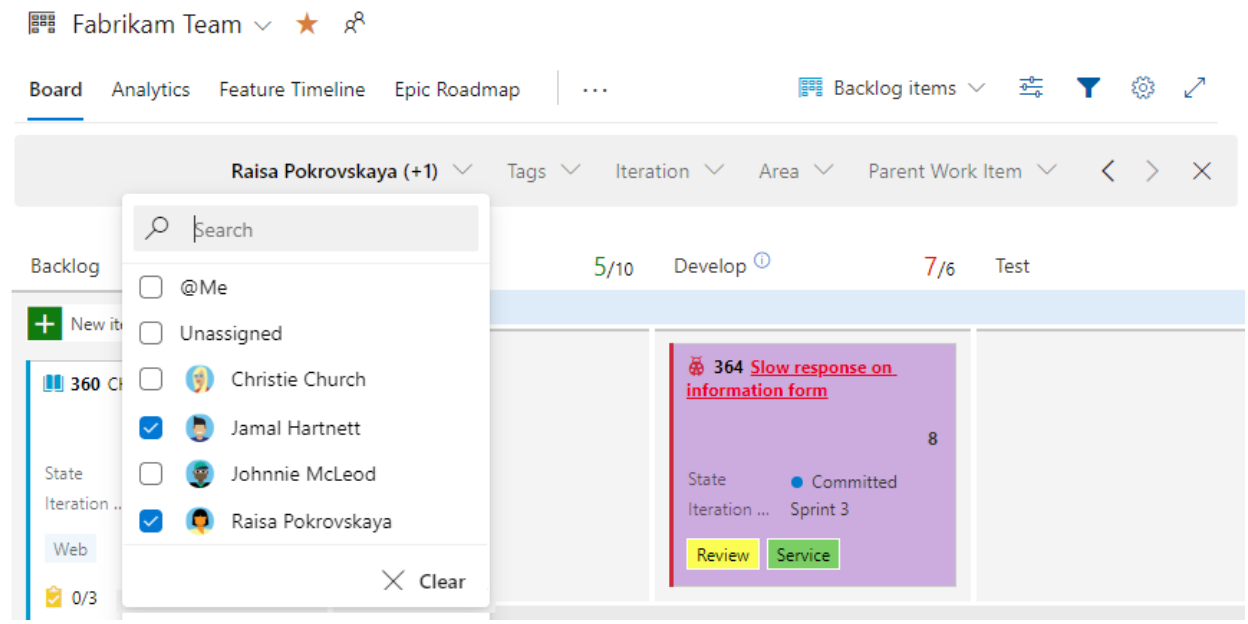
Features board

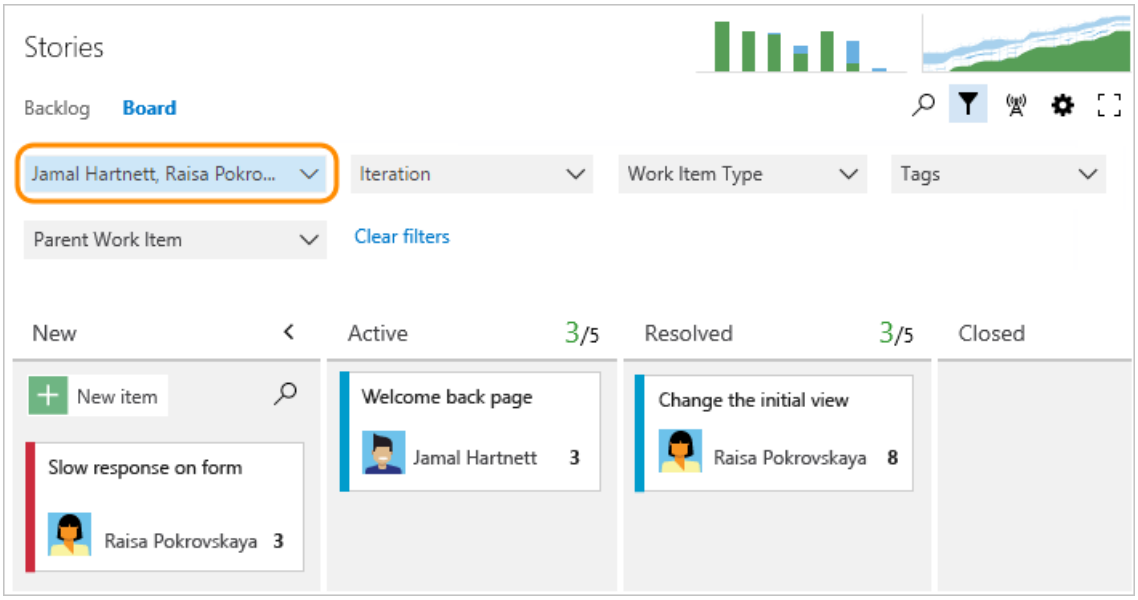
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

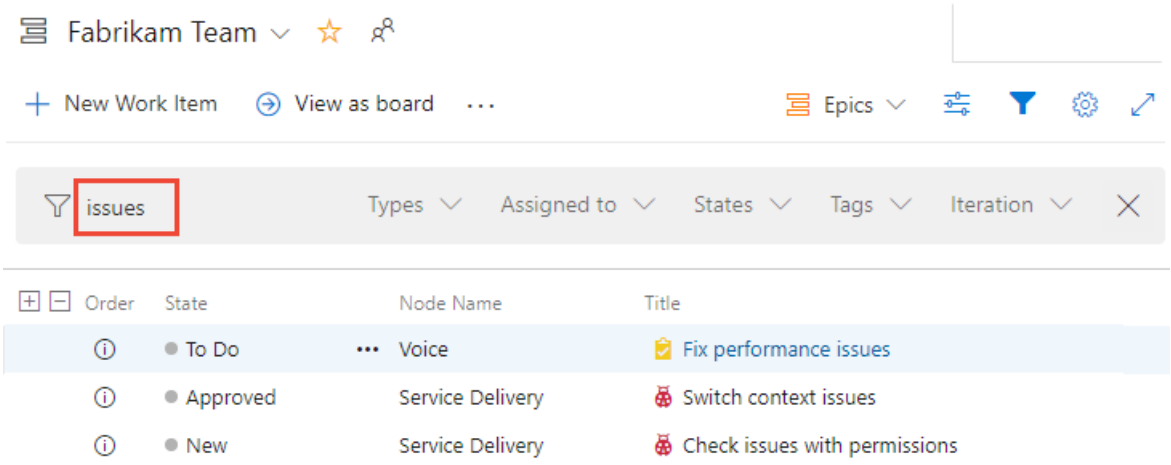
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

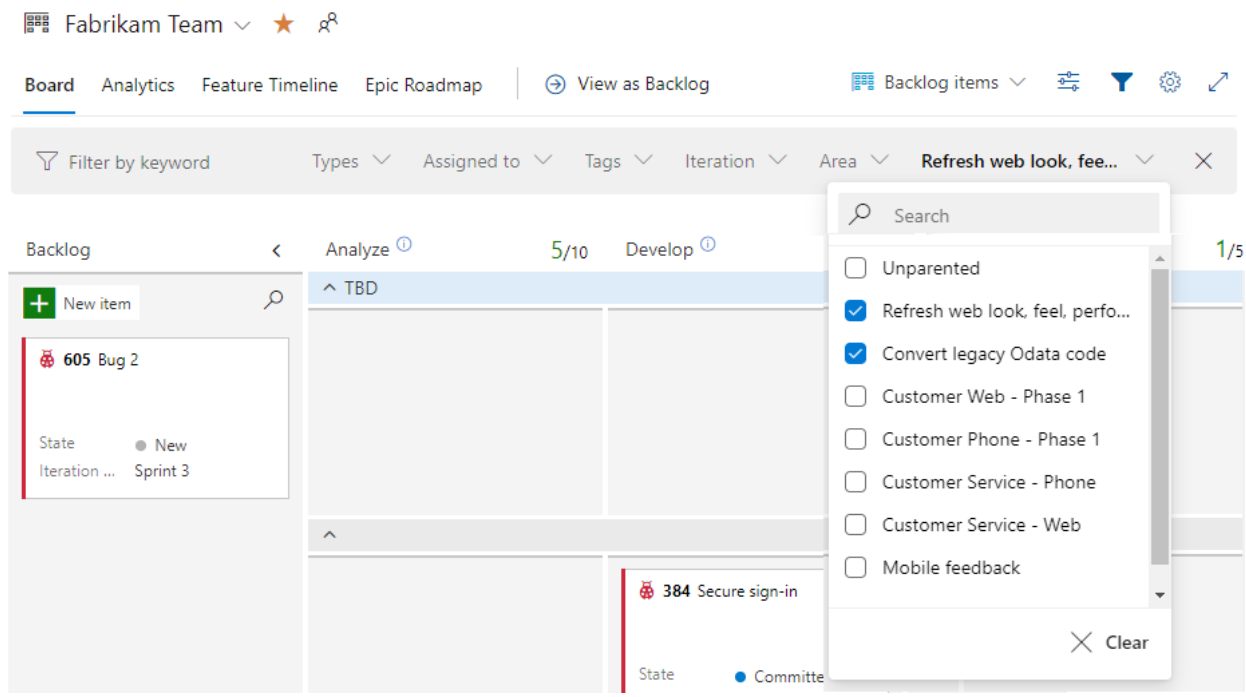
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

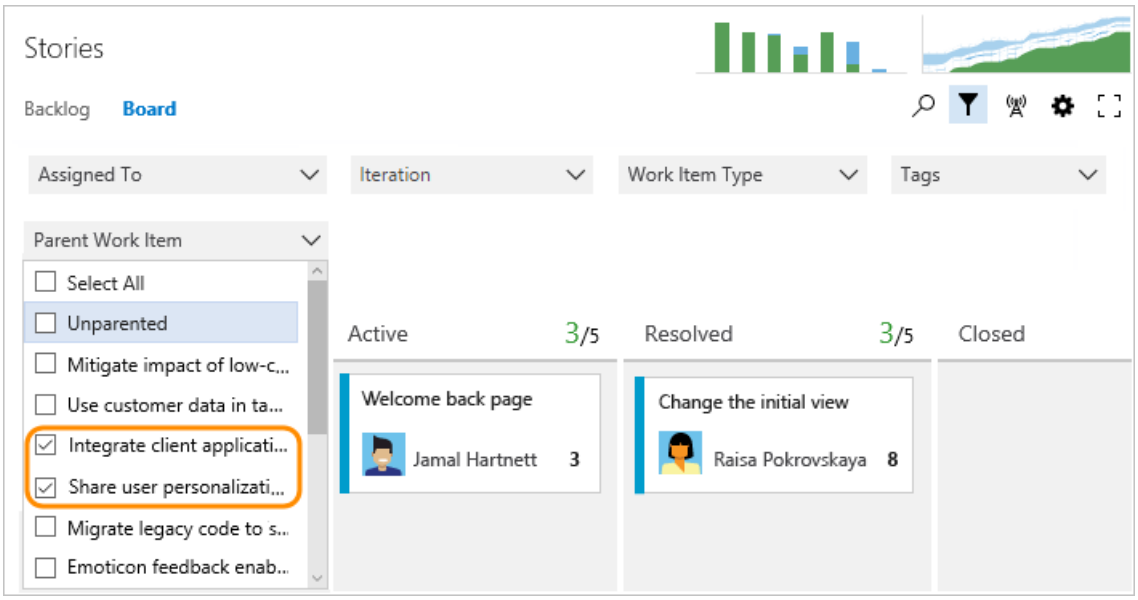
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



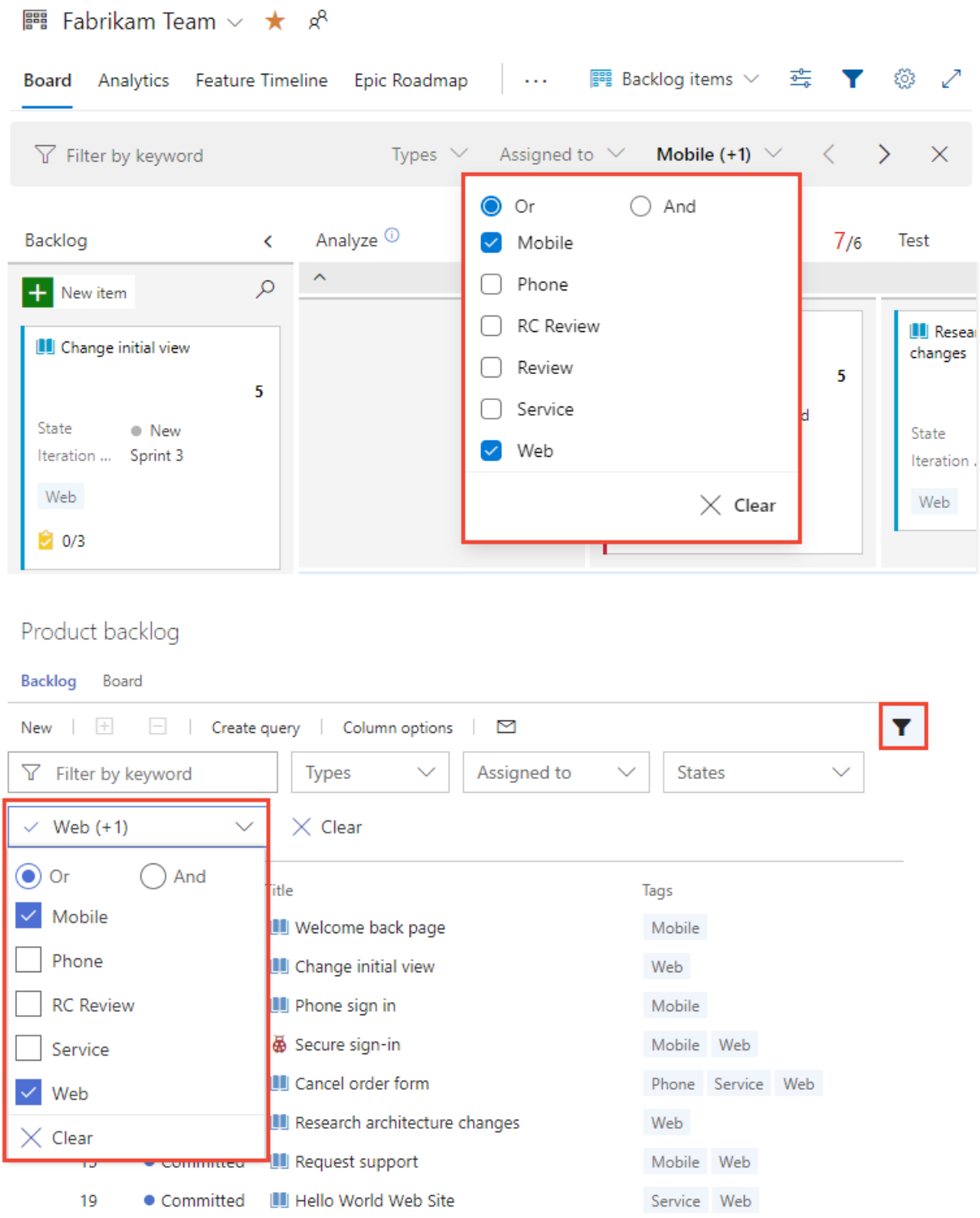


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

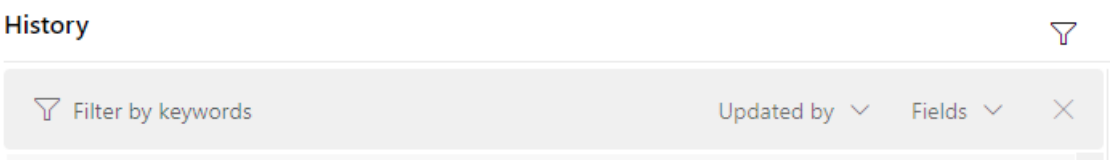


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

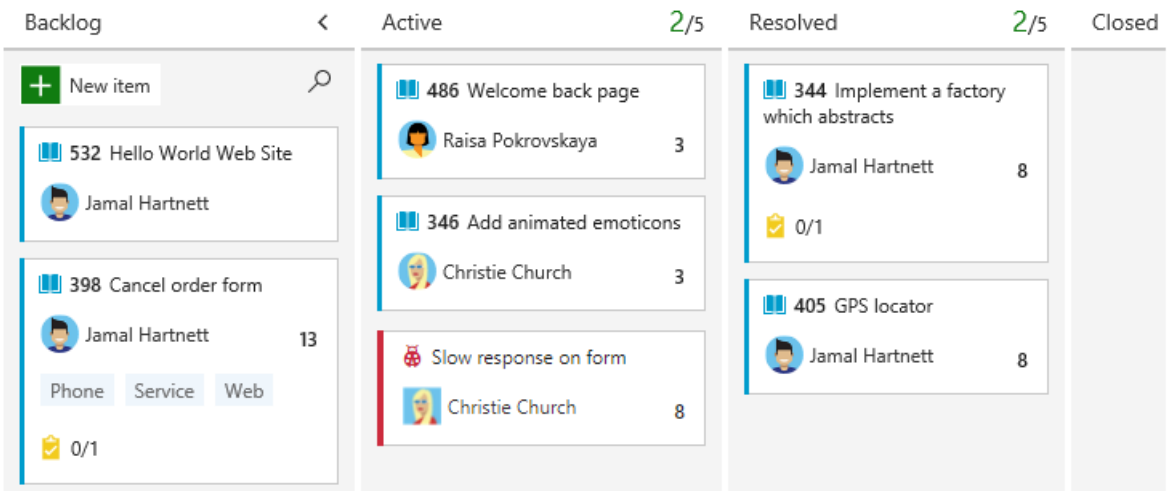
- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Implement Kanban practices using Azure Boards

10/4/2022 • 11 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To maximize a team's ability to consistently deliver high-quality software, Kanban emphasizes two main practices. The first is to visualize the flow of work. This practice requires you to map your team's workflow stages and configure your Kanban board to match. The second, constrain the amount of work in progress, requires you to set work-in-progress (WIP) limits. You're then ready to track progress on your Kanban board and monitor key metrics to reduce lead or cycle time.



Your Kanban board turns your backlog into an interactive signboard, providing a visual flow of work. As work progresses from idea to completion, you update the items on the board. Each column represents a work stage, and each card represents a user story (blue cards) or a bug (red cards) at that stage of work.

Review this article to gain an understanding of how to configure and start working with your Kanban boards:

- View your Kanban board
- Customize the columns shown on your Kanban board to support how your team works
- Set WIP limits to constrain work in progress
- Update the status of work via drag-and-drop
- View the Cumulative flow chart
- How to turn live updates on or off

> **NOTE**
> Both Kanban boards and Taskboards support visualizing the flow of work and monitoring metrics to optimize that flow. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard.

User stories and bugs correspond to types of work items. You use work items to share information, assign work to team members, update status, track dependencies, and more.

Prerequisites

Boards are automatically created when you create a project or add a team. Each team has access to their own product and portfolio boards as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.
- Users with **Stakeholder** access for a public project have full access to board features just like users with **Basic** access.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.

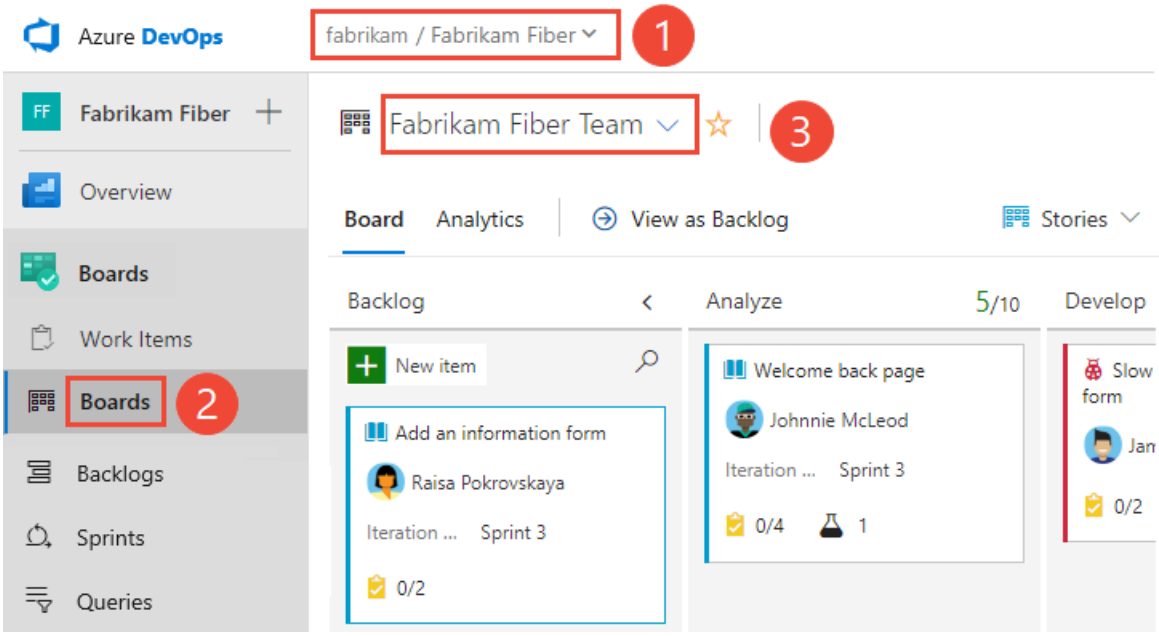
- Users with **Stakeholder** access can't exercise these board features: add work items, drag-and-drop work items to update status, or update fields displayed on cards. They can add tasks and change task status.

> **NOTE**
> Both Kanban boards and Taskboards support visualizing the flow of work and monitoring metrics to optimize that flow. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard.

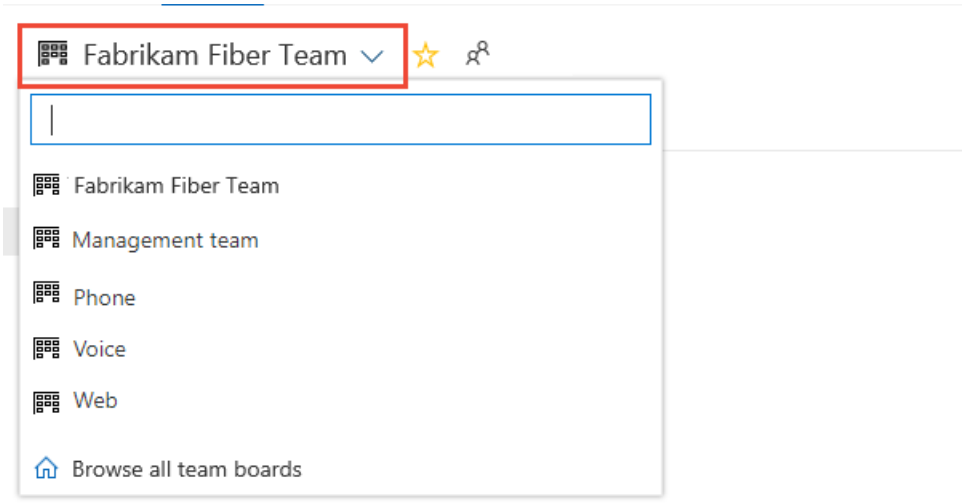
Open your Kanban board from the web portal

Your Kanban board is one of two types of boards available to you. The other is the sprint Taskboard. Kanban boards track requirements, are sprint-independent, and provide a cumulative flow chart for monitoring progress. Each sprint is associated with a Taskboard that supports tracking tasks defined for the sprint. You can monitor progress through capacity charts and the sprint burndown chart. For guidance on using the Taskboard, see Update and monitor your Taskboard. For an overview of the features supported on each backlog and board, see Backlogs, boards, and plans.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



TIP

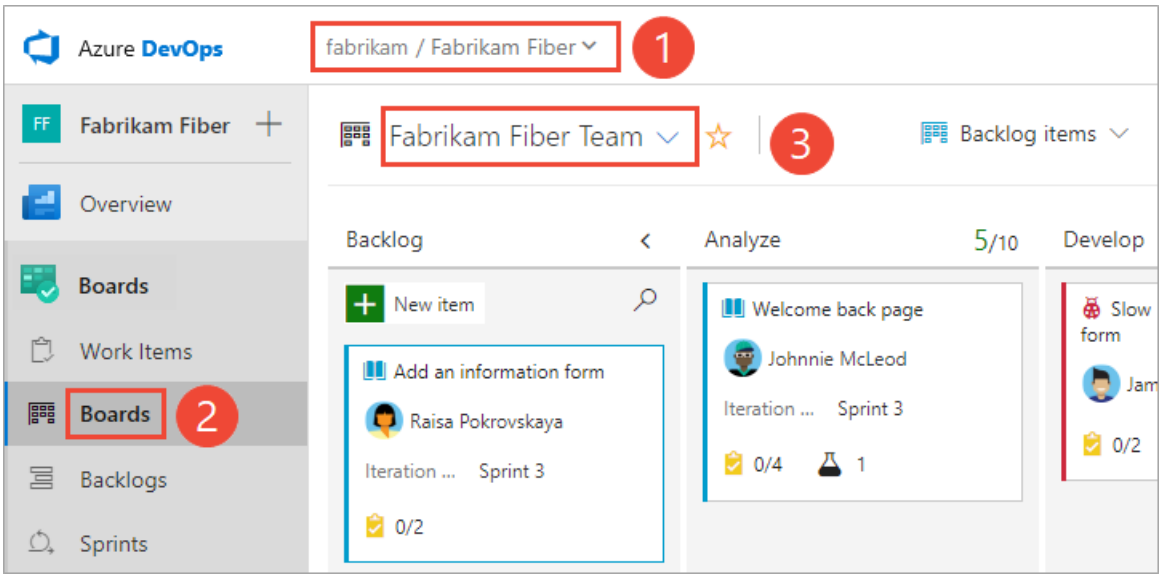
Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI as the backlog level.

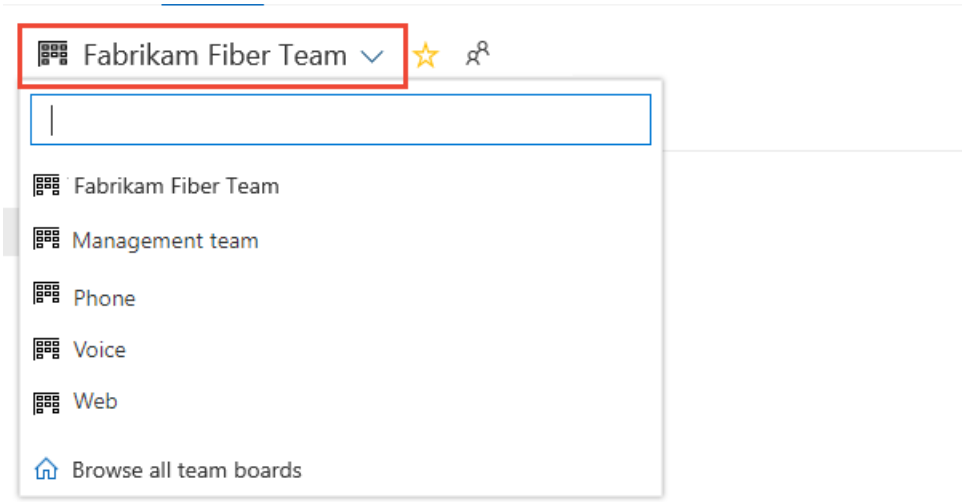


To switch to the product backlog, select **Stories backlog**. To switch to a Taskboard, see Update and monitor your Taskboard.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



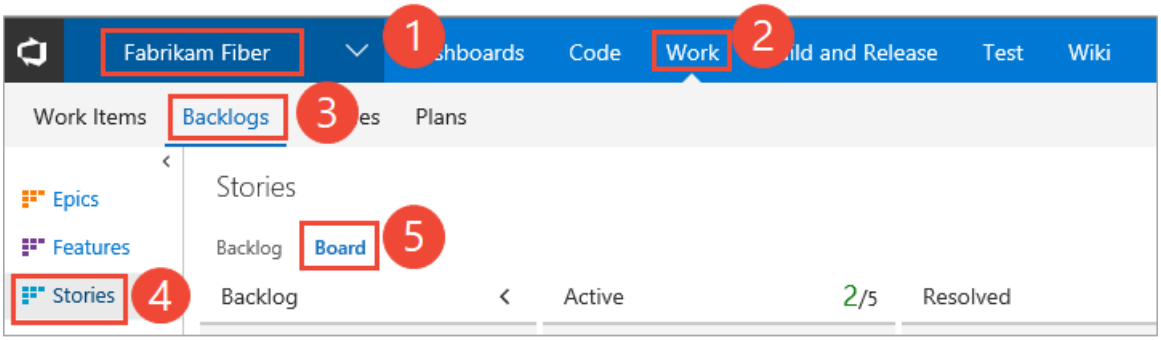
> **TIP**
> Select the ☆ star icon to make a team board a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI as the backlog level.

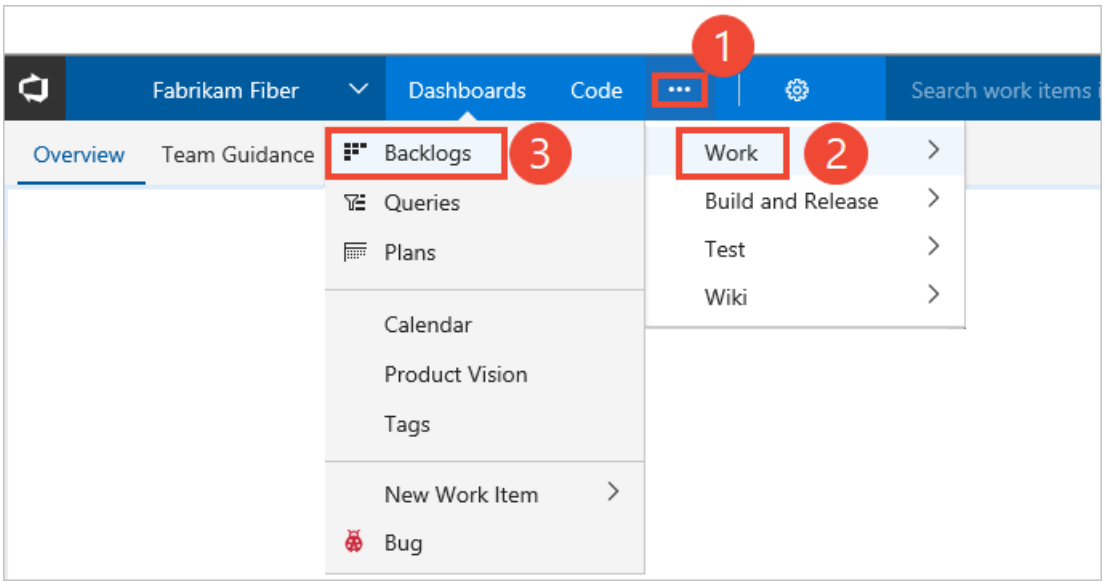


To switch to the product backlog, select **Stories backlog**. To switch to a Taskboard, see Update and monitor your Taskboard.

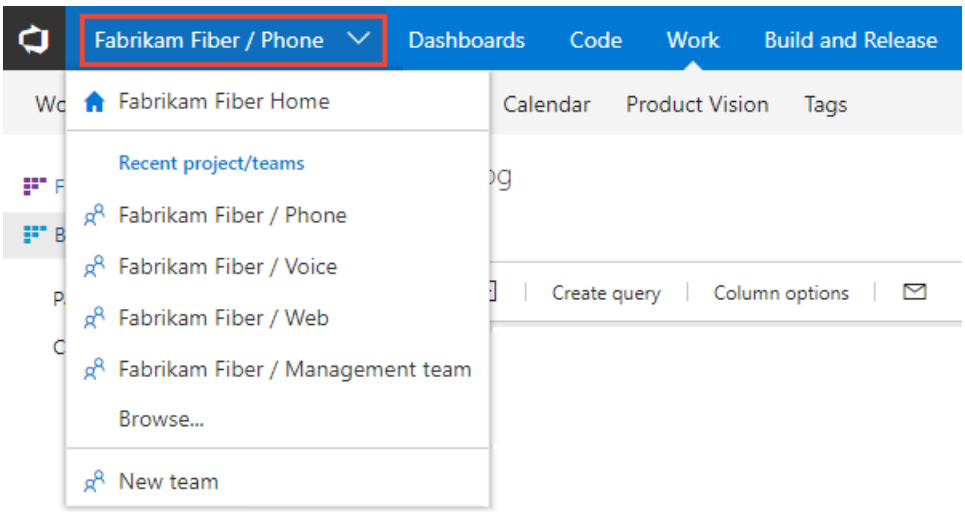
1. To view your Kanban board, open your project from a web browser. Select **Work** > **Backlogs** > **Stories**, and then select **Board**.



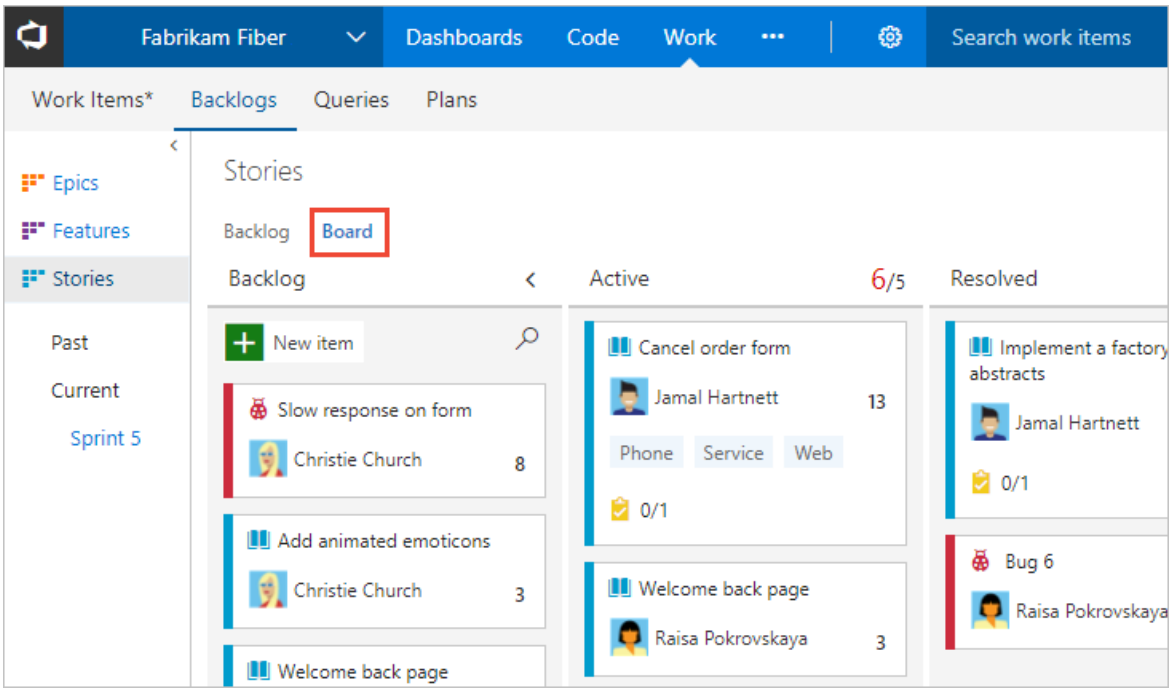
If you don't see **Work**, your screen size might be reduced. Select the three dots ( ) icon. Then select **Work** > **Backlogs** > **Board**.



2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.



Your Kanban board appears.

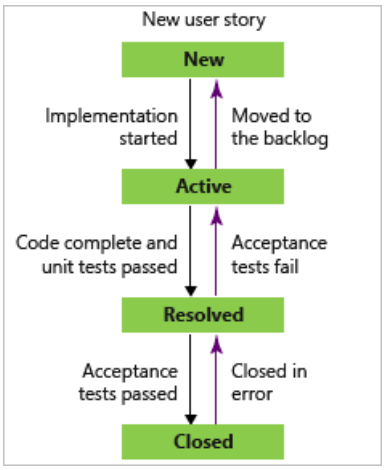


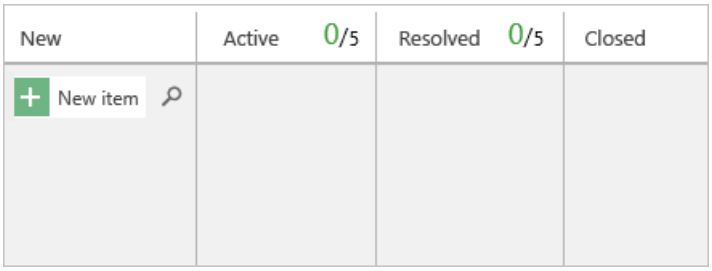
Map the flow of how your team works

Kanban literally means signboard or billboard. As a first step, you customize your board to map to how your team works.

When you first open your Kanban board, you'll see one column for each workflow state. Your actual columns vary based on the process used to create your project.

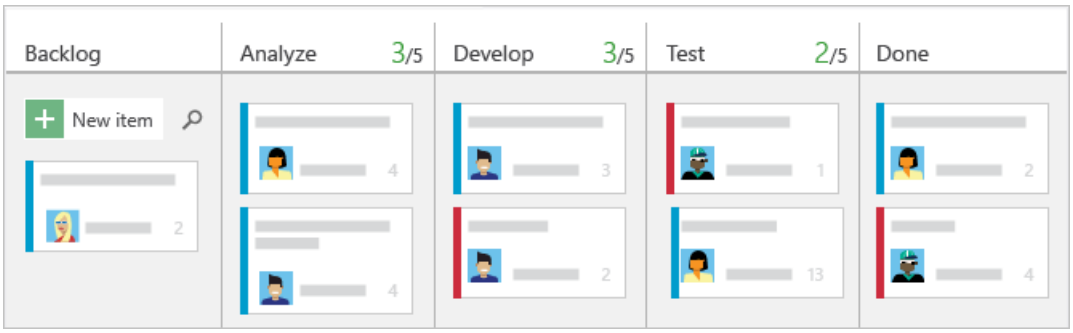
For user stories, the New, Active, Resolved, and Closed states track progress from idea to completion.





However, your team's workflow stages most likely don't map to these default states. For your team to have a functional board they must identify the stages of their workflow process and then configure the board to match.

For example, you can change your Kanban columns to map to the following five workflow stages.



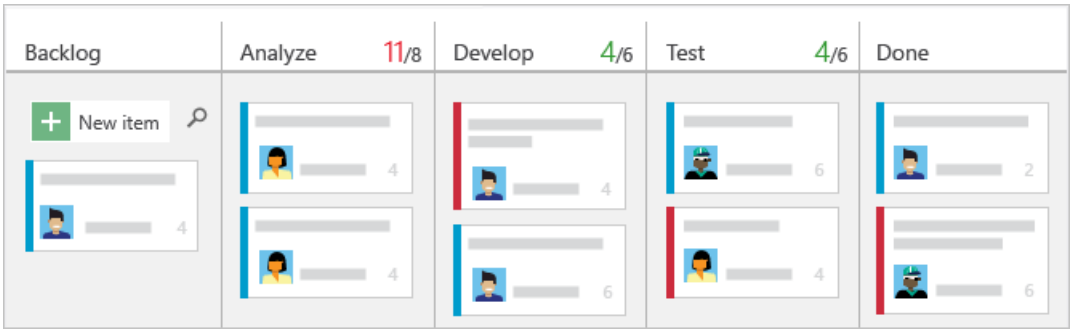
Once you've identified your stages, add and rename columns to map to them. Keep the number of columns to a minimum while still representing the key handoffs that occur for your team.

Set WIP limits to constrain work in progress

In this next step, your team sets WIP limits for each workflow stage. While setting WIP limits is easy, adhering to them takes a team commitment. Teams new to Kanban may find WIP limits counterintuitive and uncomfortable. This single practice, however, has helped teams identify bottlenecks, improve their process, and increase the quality of software they ship.

What limits should you set? Start with numbers that don't exceed two or three items per team member working within a stage. Respecting WIP limits means teams don't pull items into a column if doing so causes the number of items in the column to exceed the WIP limit.

When they do exceed the limit, the column count displays red. Teams can use this color as a signal to focus immediately on activities to bring the number of items in the column down.

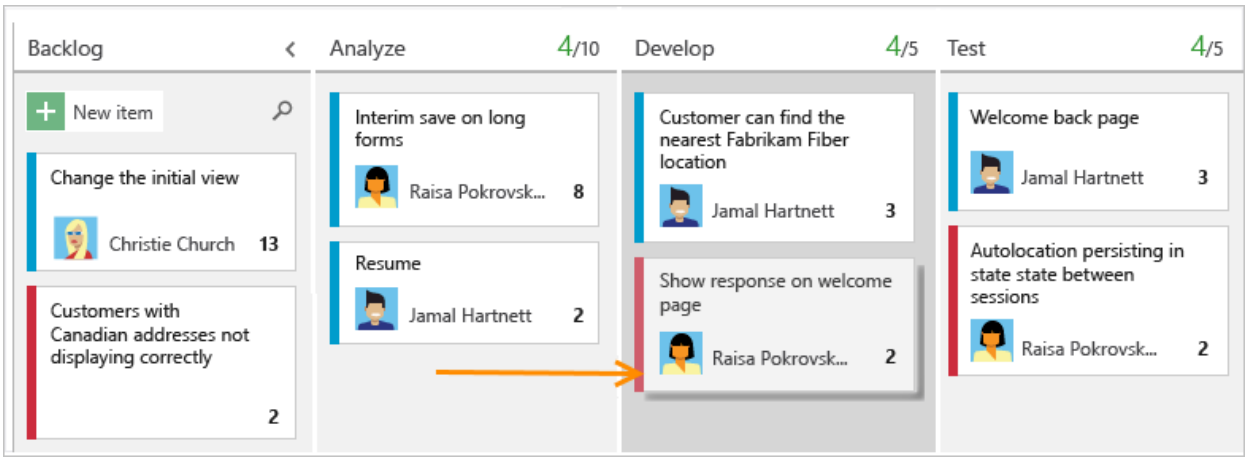


Set WIP limits based on team discussions and revisit as your team identifies ways to improve their processes. Use WIP limits to identify bottlenecks and eliminate waste from your work flow processes.

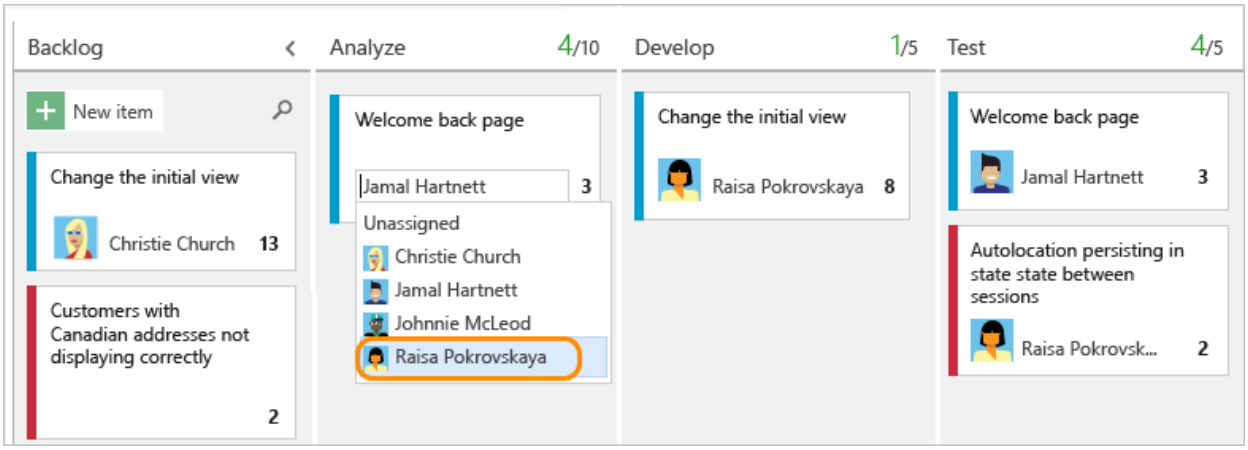
Track work in progress

Once you've configured your Kanban board to match how your team works, you're ready to use it.

Here are a few things you can do. See at a glance the estimated size of work for each item that displays at the bottom right of each card. Add items to your backlog in the first column. When priorities change, move items up and down within a column. And, as work completes in one stage, update the status of an item by moving it to a downstream stage.



Also, you can quickly update a field or reassign ownership directly from the board.



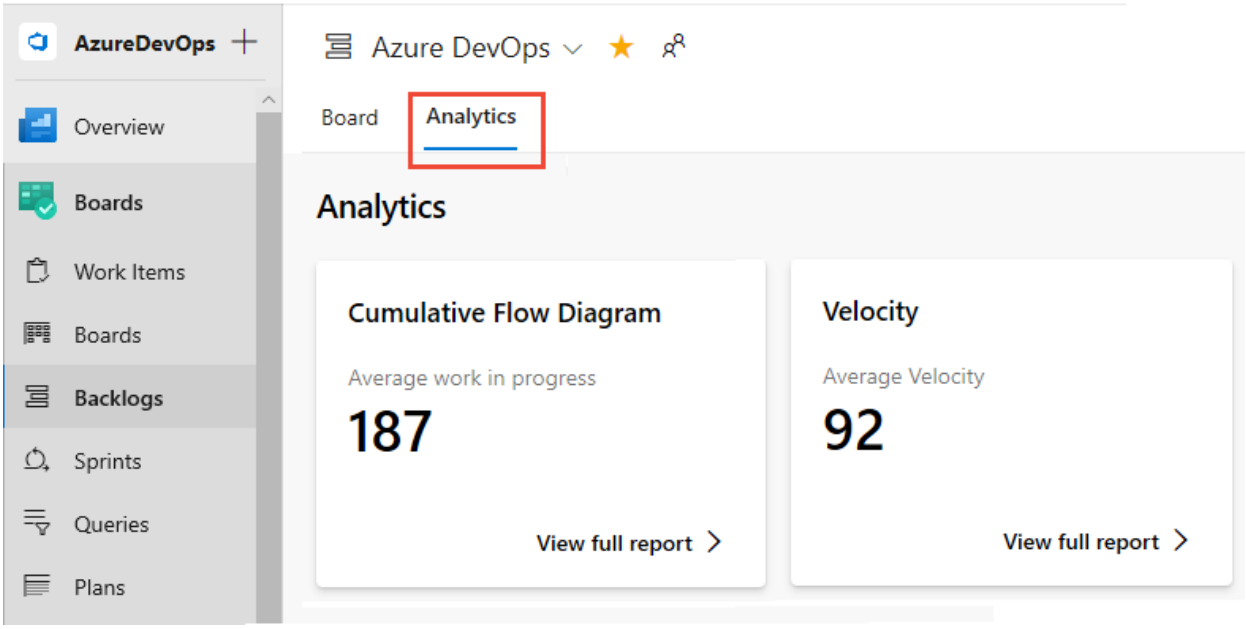
Updating your Kanban board as work progresses helps keep you and your team in sync. Also, you can see and share the value stream your team is delivering to customers.

> **IMPORTANT**
> Work items that appear on more than one team's Kanban board can yield results that don't meet your expectations because each team can customize its Kanban board columns and swimlanes. The values assigned to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields might differ from what you expect when another team updates the work item from a different board. To learn more, see Add, review, and update work items in Azure Boards.

Monitor metrics and fine tune

As with most Agile practices, Kanban encourages monitoring key metrics to fine tune your processes. After your team has used the Kanban board for several weeks, check out your Cumulative Flow Diagram (CFD).

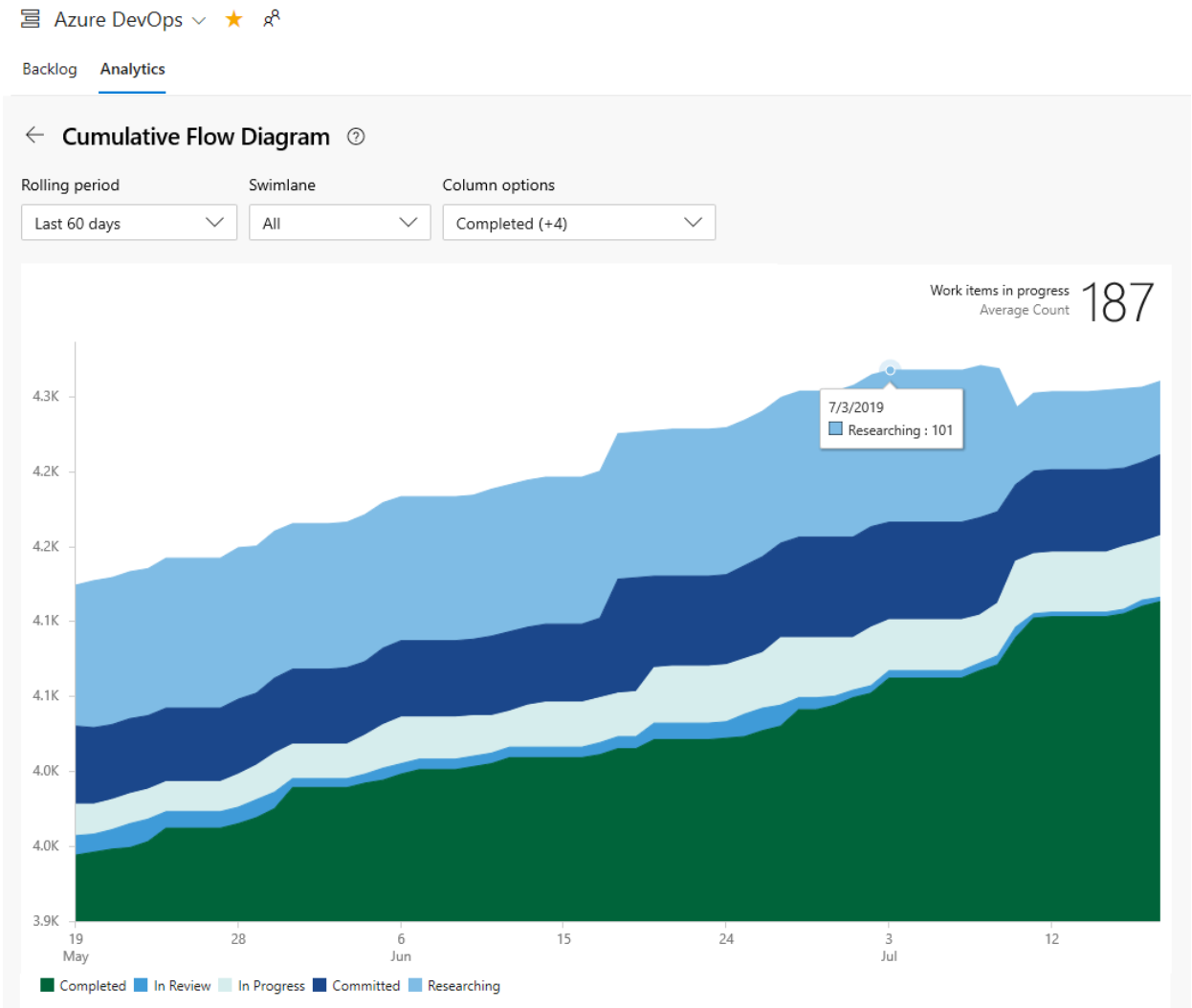
Choose the **Analytics** tab, and then choose **View full report** for the Cumulative Flow Diagram as shown in the following image.



The Average work in progress value excludes completed work items.

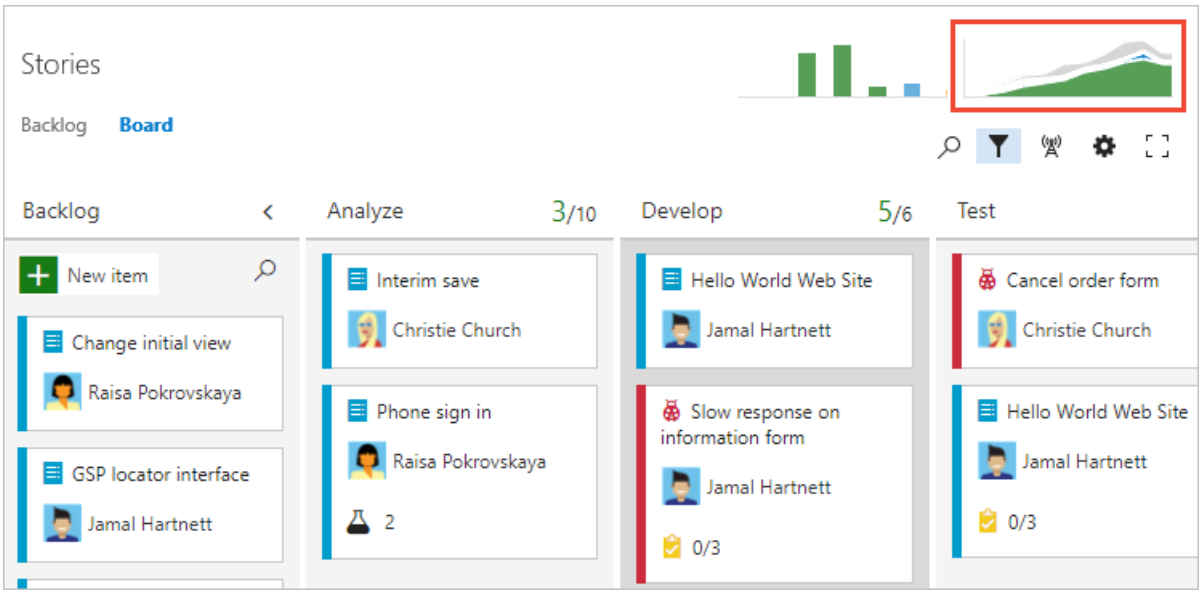
Use the interactive controls to choose the time frame, swimlanes, and workflow states or Kanban board columns. Hover over a point in time to show how many work items are in a particular state. The default settings for the Cumulative Flow Diagram-Average work in progress includes completed work items since the team started tracking work.

For example, On July 3, 101 items were in a Research state.

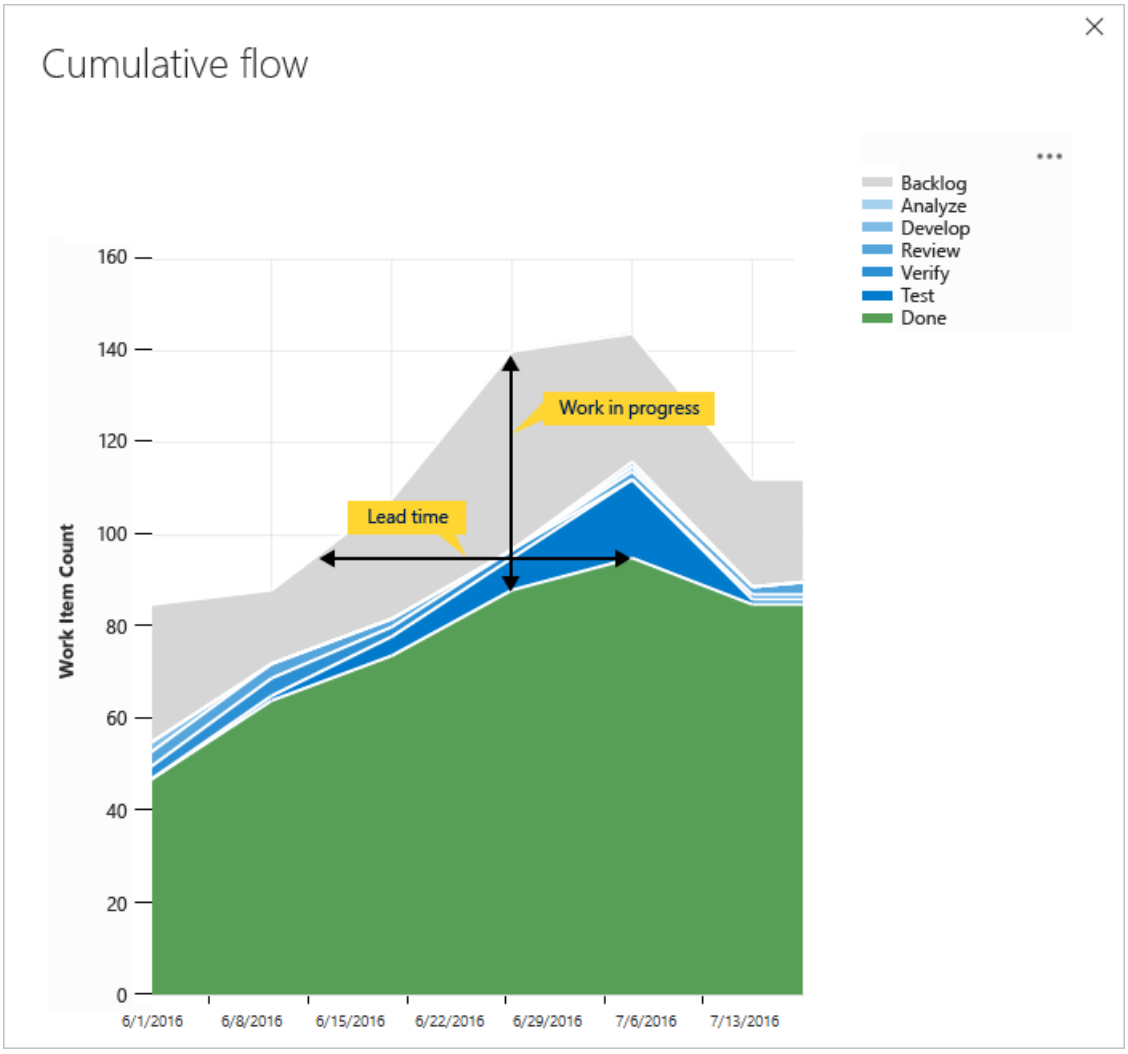


The selections you make are only set for you, and persist across sessions until you change them.

Choose the chart as shown in the following image.



The CFD shows the count of items in each Kanban column for the past 30 weeks or less. From this chart, you can gain an idea of the amount of work in progress and lead time. Work in progress counts unfinished requirements. Lead time indicates the amount of time it takes to complete a requirement from the time it was first proposed.



By monitoring these metrics, you can gain insight into how to optimize your processes and minimize lead time. For more information, see Configure a cumulative flow chart.

Along with the above chart, you can add Analytics widgets to your dashboard. The Analytics Service is in preview and provides access to several widgets. To learn more, see these articles:

- Widgets based on the Analytics Service
- Add a widget to a dashboard
- What is the Analytics Service?

Try this next

Here are some useful tips when working with the Kanban board:

- To focus on select work items, filter your Kanban board
- To quickly assign items to a team member, add the Assign To field to display on the cards, see Customize cards
- Add a swimlane to track high-priority work or track work that falls into different service level agreements, see Swimlanes
- Highlight specific work items by color coding cards based on a field value or tag, (see Customize cards)
- If you use Scrumban, drag-and-drop cards onto a sprint to quickly assign them to a sprint.

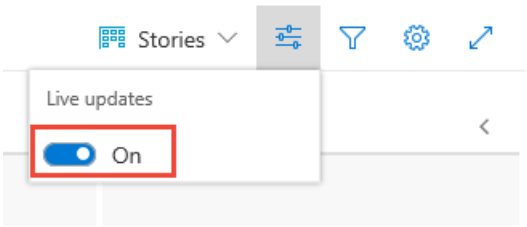
Each team can manage their backlog and customize their Kanban board. Add teams when you assign specific feature areas to different teams for development. Each team can then manage their backlog and focus on how they'll develop their deliverables.

If you're new to tracking work with Agile tools and want to understand what you can customize, see Customize your work tracking experience.

Enable live updates

Enable live updates to automatically refresh your Kanban board when changes occur. With live updates enabled, you no longer have to press **F5** to see the latest changes.

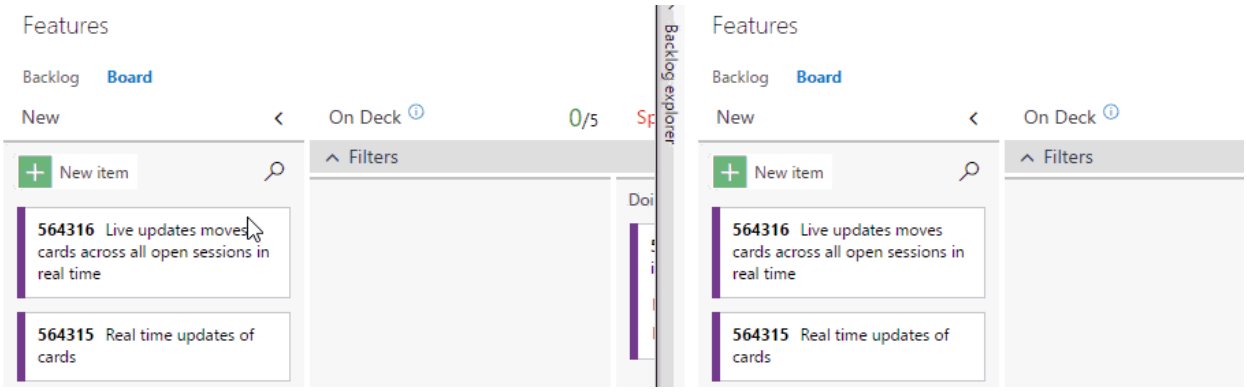
Choose the view options icon and move the slider for **Live updates** to On.



Choose the **Live updates** icon.



As one team member updates the status of a work item, other team members will see those updates in real time as they occur.



Related articles

- Kanban key concepts
- Kanban best practices
- Cumulative flow diagram
- Cumulative flow, lead time, and cycle time guidance

Add, run, and update inline tests in Azure Boards and Azure DevOps

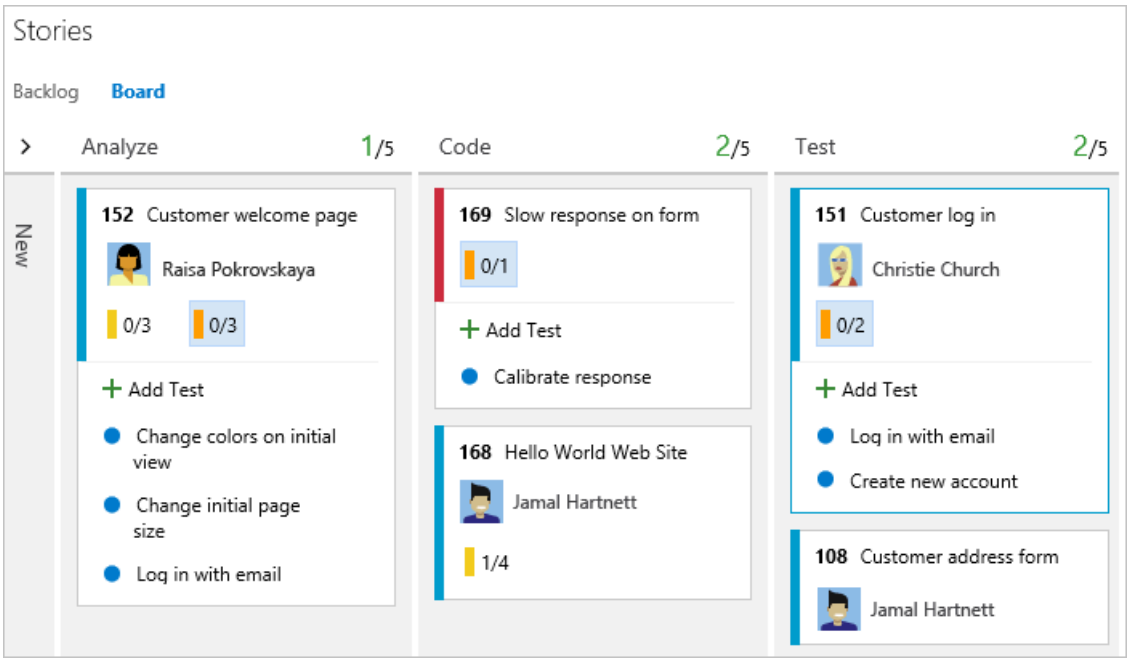
10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Similar to task checklists, you can quickly define inline tests, or a set of manual tests cases, for a backlog item from your Kanban board. Not only can you add tests, you can run them and update their status. See Kanban basics if you're new to working with the Kanban board. If you're new to testing, see Exploratory and manual testing scenarios and capabilities.

In this article, you'll learn:

- How to add inline tests to a backlog item from your Kanban board
- How to run tests and update the status of tests
- How to expand or collapse inline tests
- How to reorder or reparent inline tests



Tests you create from the Kanban board are automatically linked to the user story or backlog item.

Prerequisites

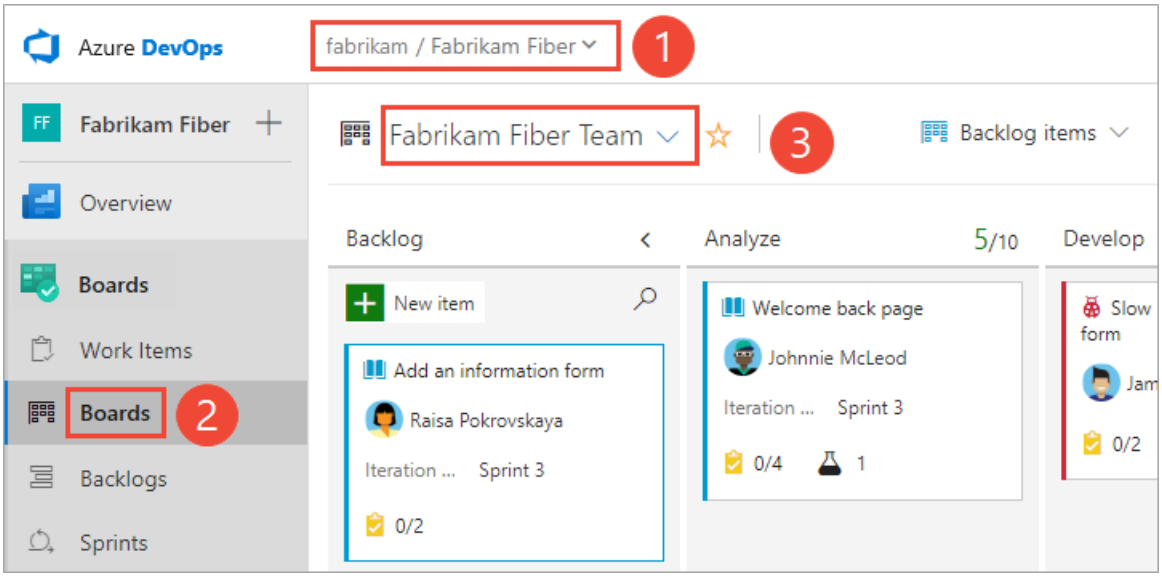
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To view or run tests, you must have **Basic** access or higher. Users with **Stakeholder** access can't view or run tests.

- You must connect to a project. If you don't have a project yet, create one.

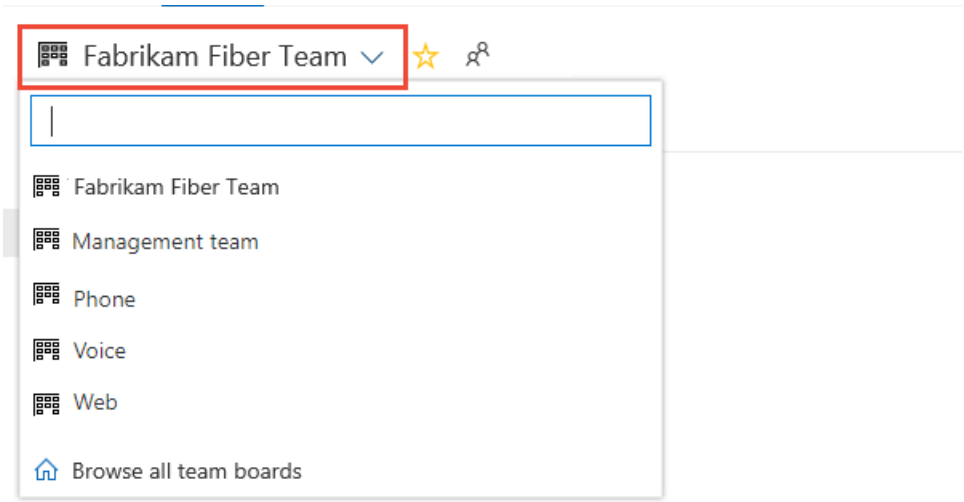
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To view or run tests, you must have **Basic** access or higher. Users with **Stakeholder** access can't view or run tests.

Open your Kanban board from the web portal

1. To view your Kanban board, open your project from a web browser and choose (1) **Work**, (2) **Boards**, and then (3) select the team's board from the selector.



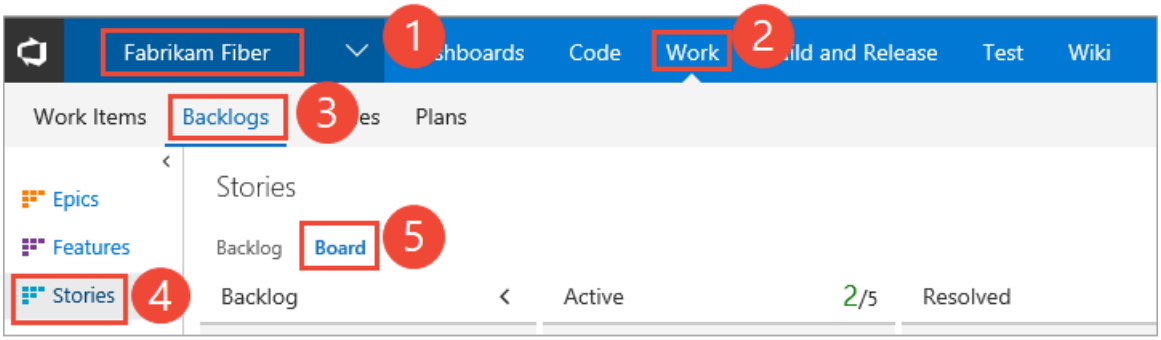
To choose another team's board, open the selector and select a different team or choose the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



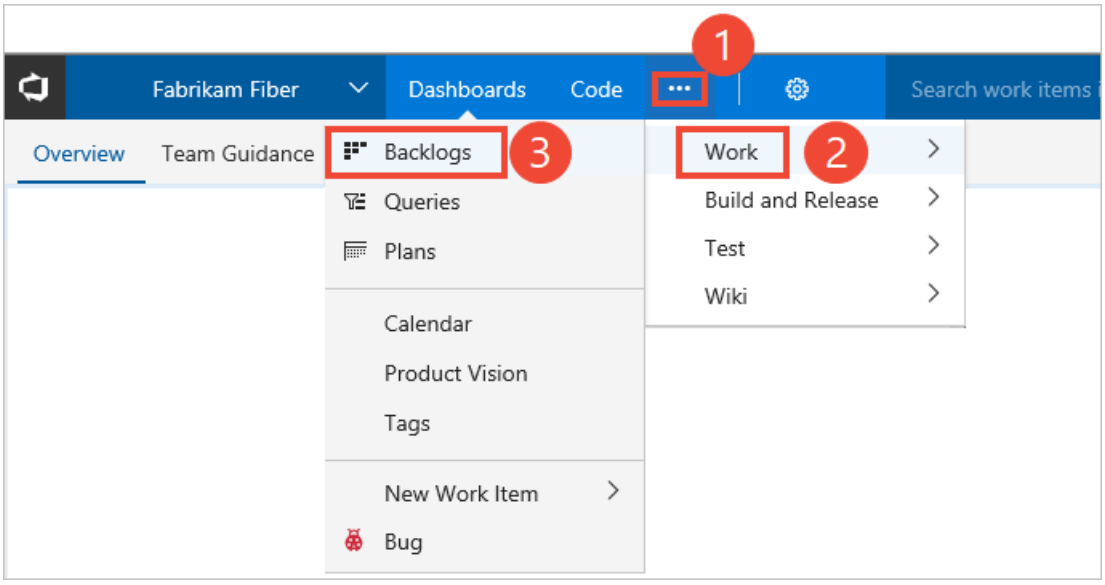
> **TIP**
> Choose the ☆ star icon to favorite a team board. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

1. To view your Kanban board, open your (1) project from a web browser and choose (2) **Work**, (3)

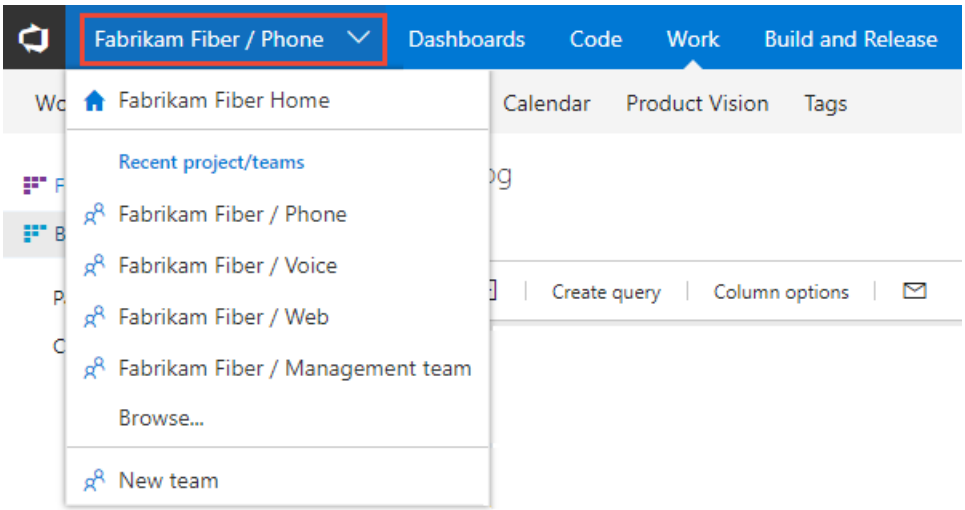
Backlogs, (4) **Stories**, and then (5) **Board**.



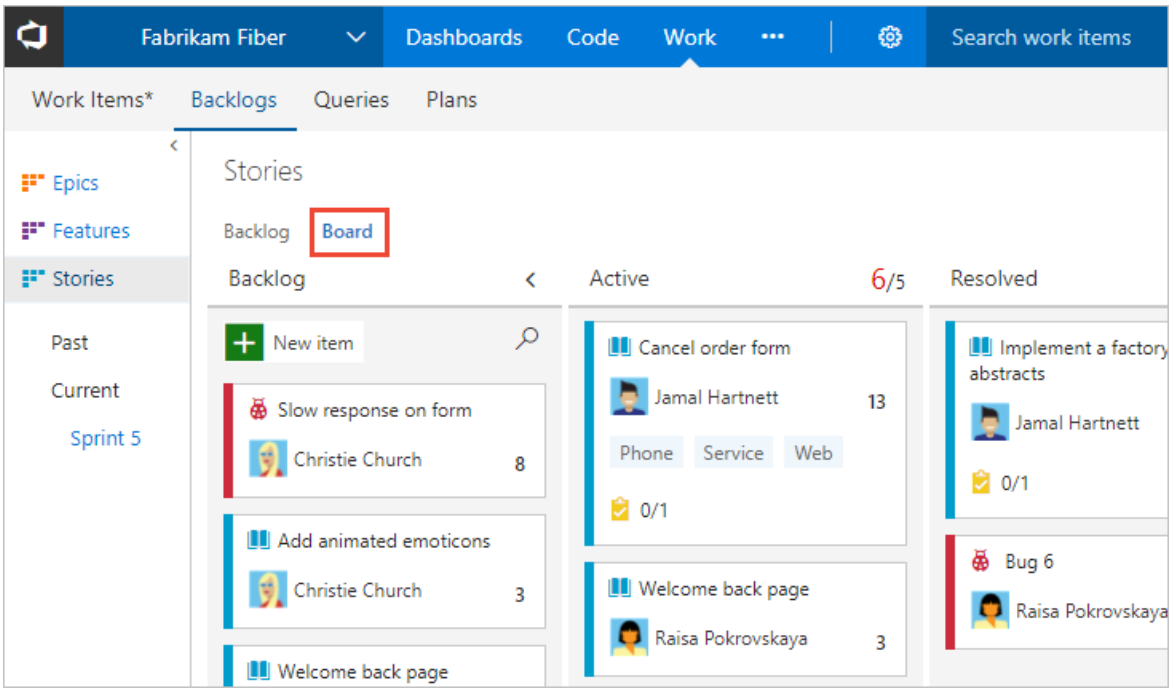
If you don't see **Work**, your screen size may be reduced. Select the three dots (, then choose **Work**, **Backlogs**, and then **Board**.



2. To choose another team, open the project/team selector and select a different team or choose the **Browse** option.

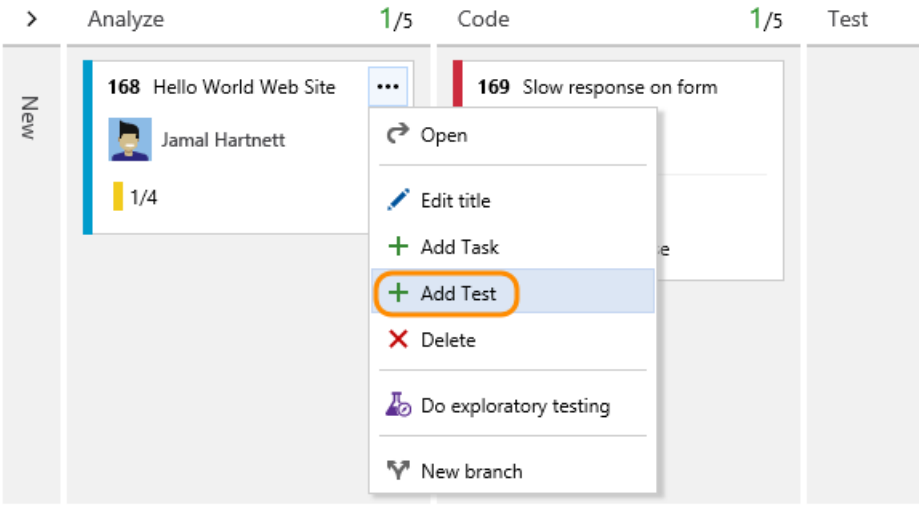


3. Your Kanban board displays.



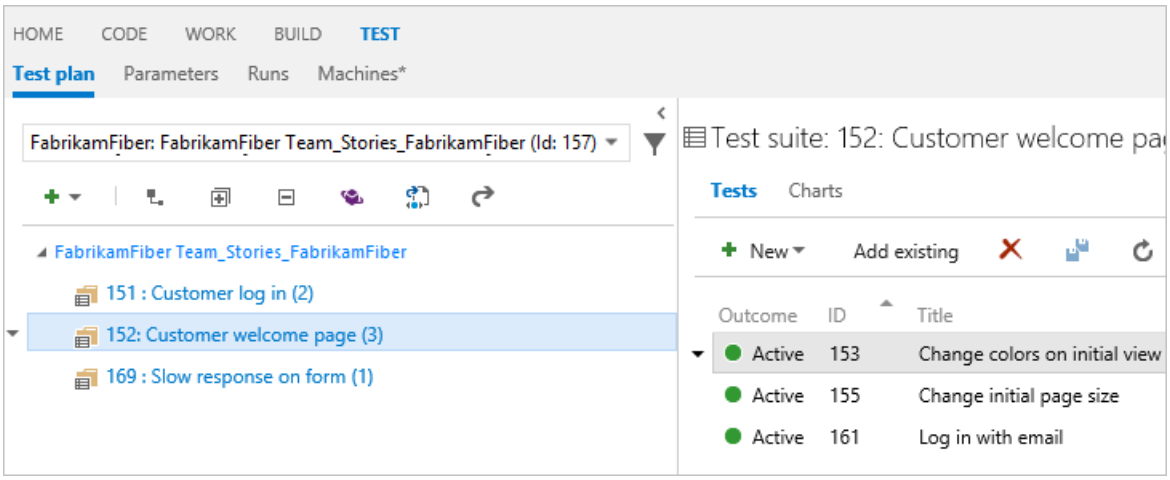
Add inline tests

1. To start adding tests, open the menu for the work item.



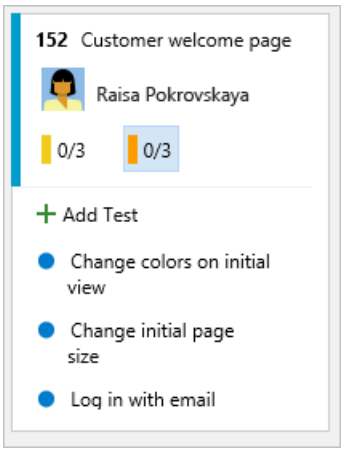
Adding inline tests is the same as adding test cases to a test suite. A default test plan and test suite are automatically created under which the manual test cases are grouped.

For example, a test suite is created for each user story, and all inline tests are added to that suite. Below, user story 152 is highlighted which has three manual tests defined with IDs of 153, 155, and 161.

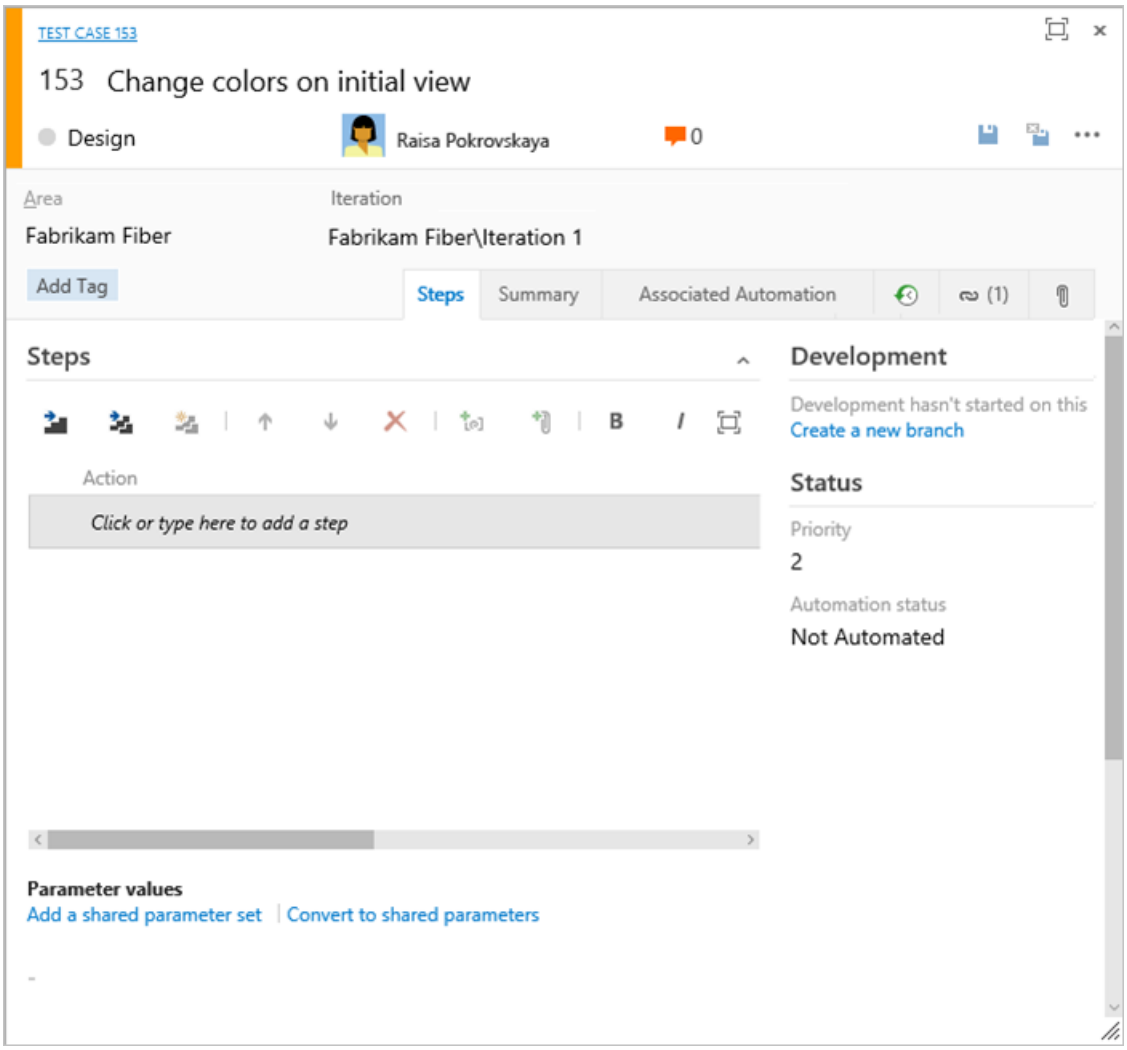


To learn more about test plans and test suites, see Plan your tests.

2. If you have many tests to add, keep typing each title and select Enter.



To add details to the test case, open it. You can select the title, double-click the inline item, or open the context menu and choose Open.

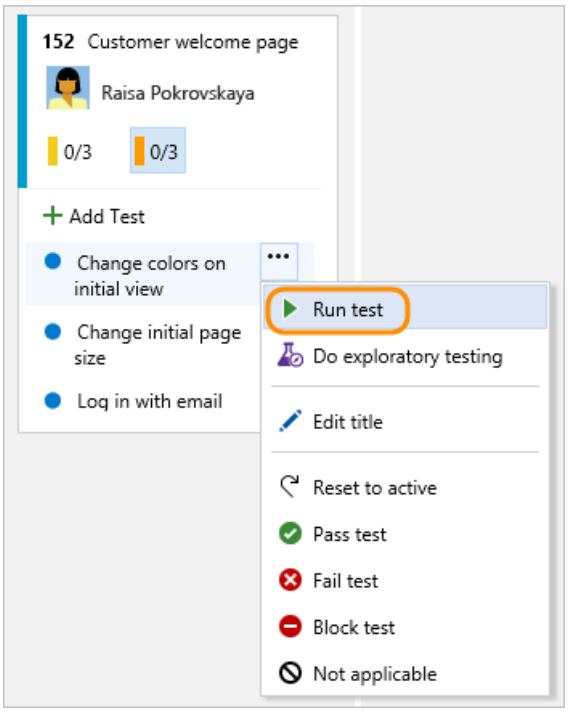


See Create manual tests to learn more about defining tests.

Before running the test, you must add details.

Run the inline test

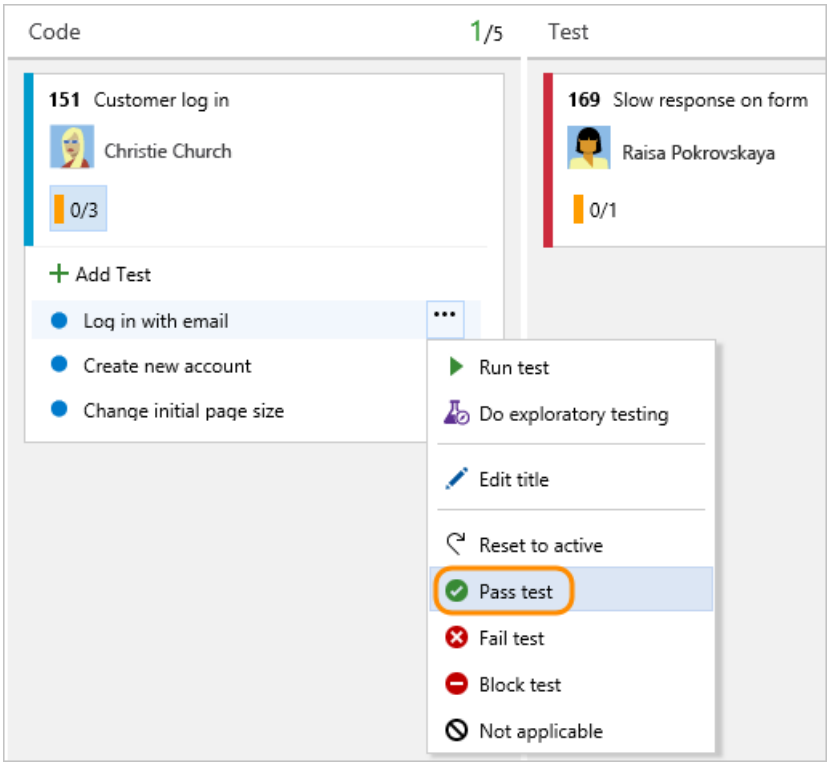
Run the test by selecting Run test from the ··· actions menu for the inline test.



Microsoft Test Runner starts in a new browser instance. For details on running a test, see Run manual tests.

Update the status of an inline test from the action menu

You can update the status of the test from the ··· actions menu.

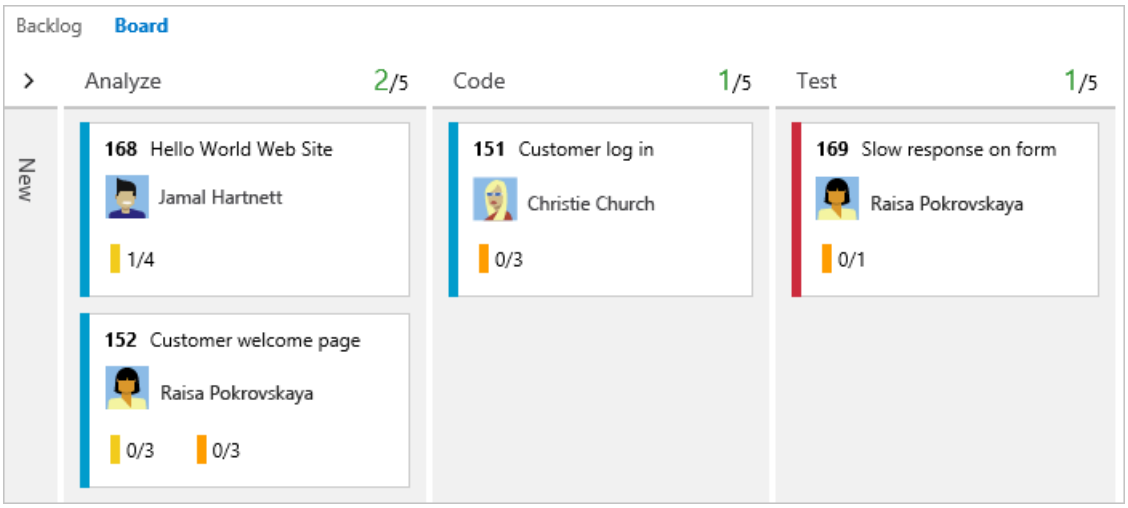


Updating the status of tests enable you to track test results.

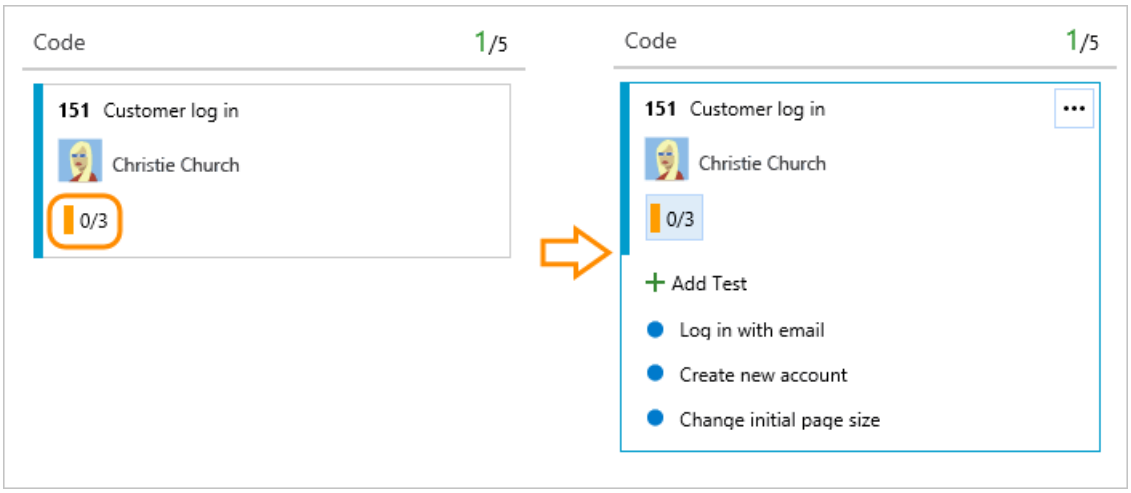
Why doesn't the Kanban board show the status for test suites and plans already created in **Test**?

Expand or collapse inline tests

Upon first opening the Kanban board, you'll see an unexpanded view of checklists.

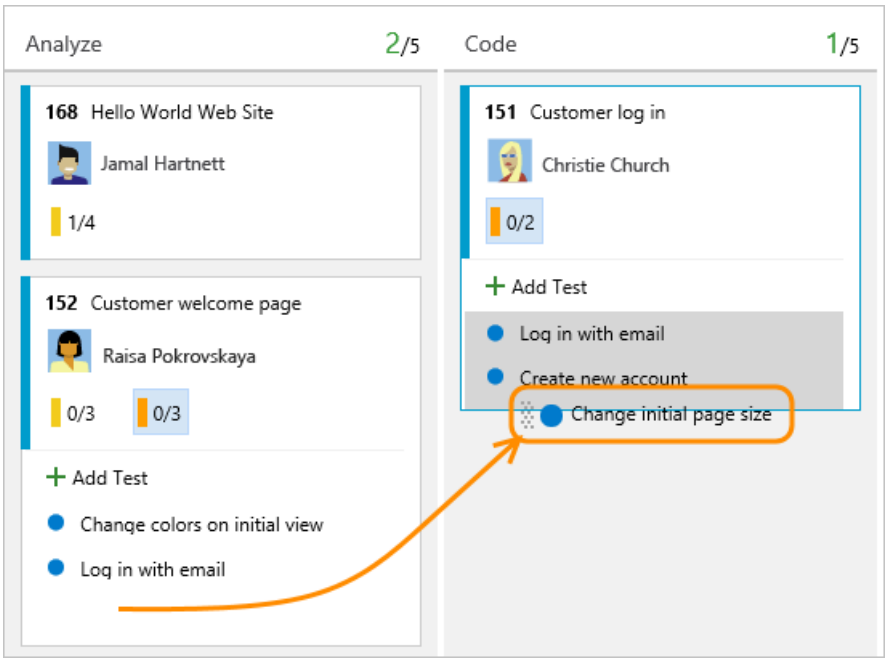


Select the inline test summary to expand a collapsed set of tests. Select the same summary to collapse an expanded list.



Copy or reparent a test

To reparent a test, drag and drop the test onto a different user story.



This action automatically changes the linked relationship of the test to point to the new user story.

To create a copy of a test to add to a different user story, select the test, press the CTRL key, and then drag and drop the test onto the card of the user story.

Related articles

Use inline tests for lightweight traceability and to manage manual tests for user stories or other backlog items that they support. To learn more about test case management, see Create manual tests.

If you find that you don't use this feature, you can disable it from the common configurations dialog.

Other ways you can quickly add linked items and objects to user stories from the Kanban board:

- Add tasks or child items as checklists
- Create a new branch, drive Git development

To start web-based exploratory testing for a user story, you need to install the Test & Feedback Marketplace extension. For details, see Install the Test & Feedback extension.

Test status in the Kanban board

Test integration with the Kanban board makes it easy for teams to get started with manual testing and then take advantage of the full testing capabilities in Test Manager later, when required. When test cases are created from the Kanban board and updated afterwards in Test Manager, or the other way around, when users create requirement-based suites with Test Manager and update them in Test Manager, the Kanban board shows the correct status. However, the Test status in Kanban board doesn't work if the requirement-based suite has more than one configuration assigned to it. In such scenario, the Kanban board only shows the test outcome for the default configuration. As such, it's recommended to use Test Manager to manage/track the testing progress across multiple configurations.

Use Kanban board controls to enable interface features

10/4/2022 • 2 minutes to read • Edit Online

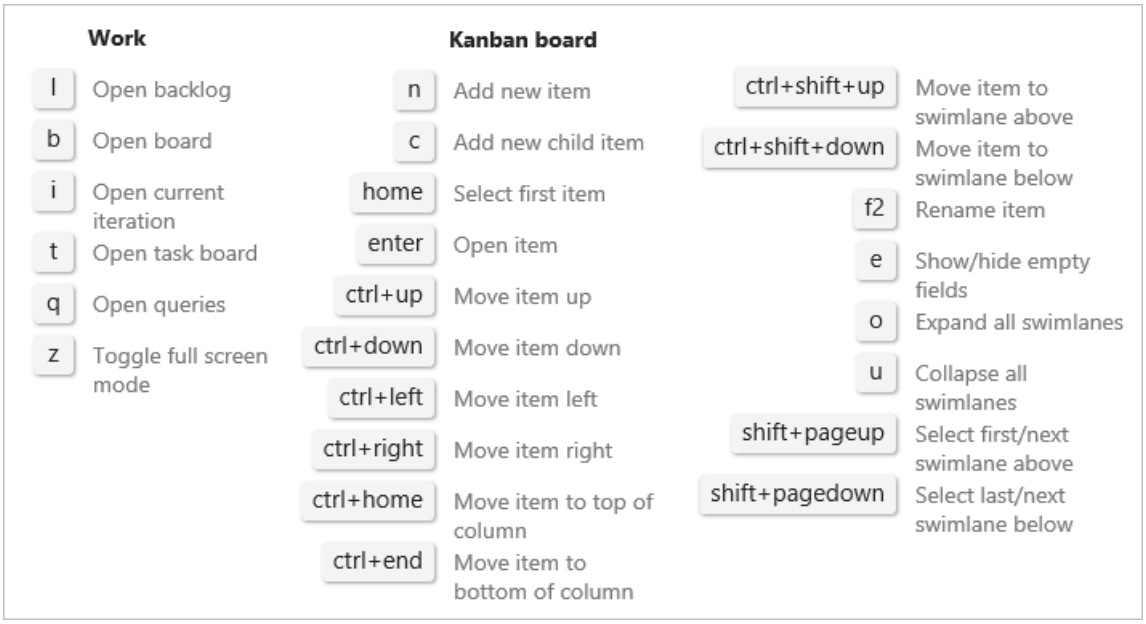
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can quickly switch from the backlog view to the board view using the **Backlog** and **Board** links. Use the following icons to enable other user interface features.

CONTROL	FUNCTION
Backlog	Switch to backlog view
Board	Switch to Kanban board view
	Filter by keywords, tags, or fields
	Enable live updates
	Customize the board and configure team settings: Cards | Card reordering | Columns | Swimlanes | CFD chart | Backlogs | Working days | Working with bugs
	Enter or exit full screen mode

Open keyboard shortcuts

Enter **?** to open the Kanban board keyboard shortcuts.

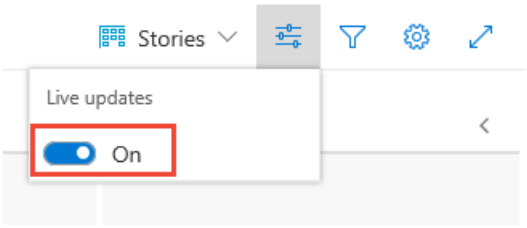


Enable live updates

Enable live updates to automatically refresh your Kanban board when changes occur. With live updates enabled,

you no longer have to press **F5** to see the latest changes.

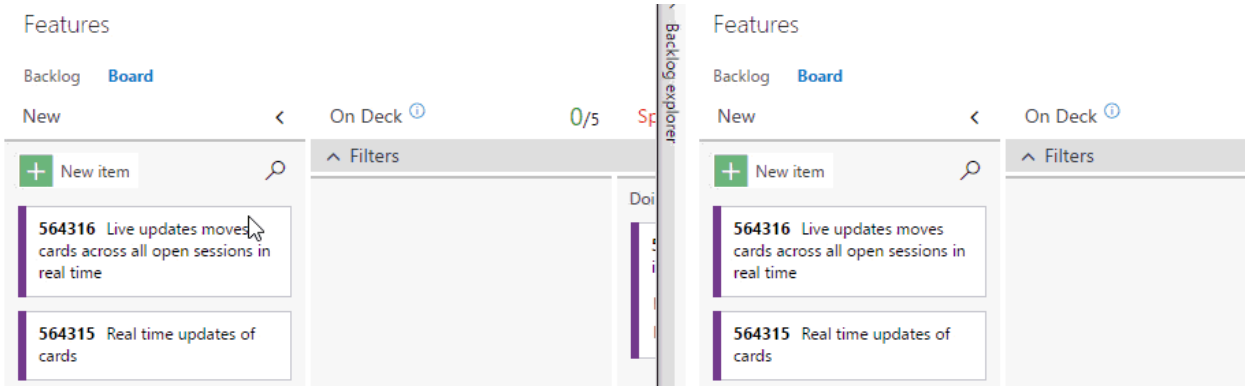
Choose the view options icon and move the slider for **Live updates** to On.



Choose the **Live updates** icon.



As one team member updates the status of a work item, other team members will see those updates in real time as they occur.



Enable live updates in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Enable live updates to automatically refresh your Kanban board when changes occur. As other team members move or reorder cards, your board will automatically update with the changes. With live updates enabled, you no longer have to press F5 to see the latest changes.

Prerequisites

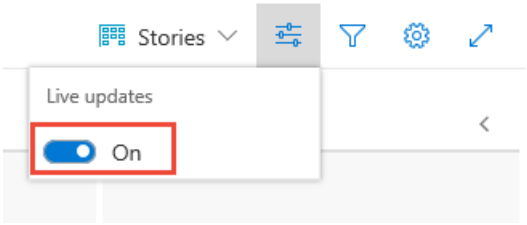
Boards are automatically created when you create a project or add a team. Each team has access to their own product and portfolio boards as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.
- Users with **Stakeholder** access for a public project have full access to board features just like users with **Basic** access.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access for a private project can add work items and update status through drag-and-drop, but cannot update fields displayed on cards. They can add tasks and change task status.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a team or project.
- To add work items and exercise all board features, you must be granted **Basic** access or higher.
- To view or modify work items, your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- Users with **Stakeholder** access can't exercise these board features: add work items, drag-and-drop work items to update status, or update fields displayed on cards. They can add tasks and change task status.

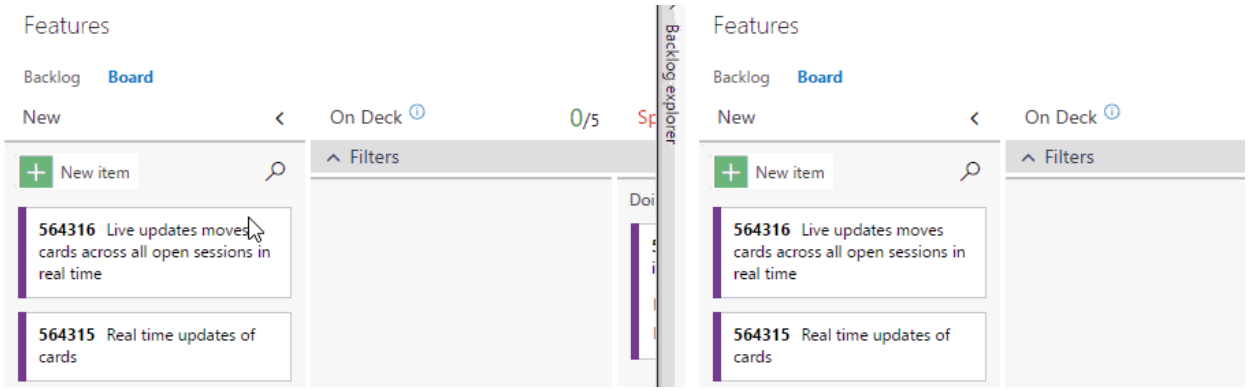
Choose the  view options icon and move the slider for **Live updates** to On.



From the Kanban board, choose the **Live updates** icon.



As one team member updates the status of a work item, other team members will see those updates in real time as they occur.



IMPORTANT

Work items that appear on more than one team's Kanban board can yield results that don't meet your expectations because each team can customize its Kanban board columns and swimlanes. The values assigned to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields might differ from what you expect when another team updates the work item from a different board. To learn more, see Add, review, and update work items in Azure Boards.

Related articles

- Filter your Kanban board
- Customize cards

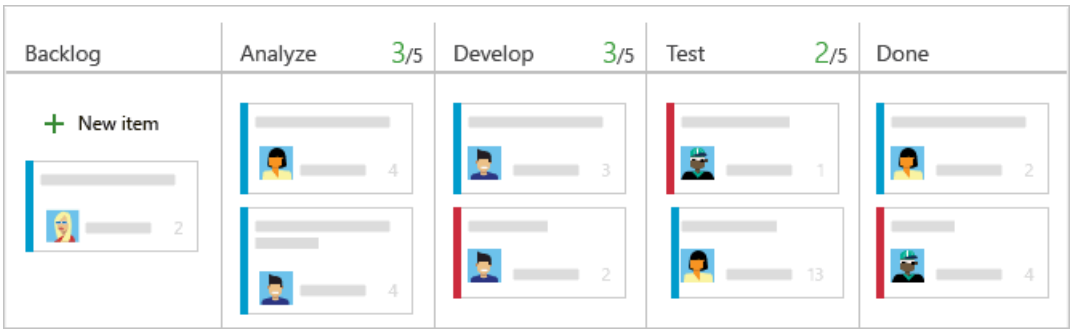
Add columns to your Kanban board to manage your workflow

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Kanban's number one practice is to visualize the flow of work. So, your number one task is to visualize your team's workflow. You do this task by identifying the types of work and handoffs that occur regularly as your team moves items off the backlog and into a shippable state.

For example, the main workflow stages for the following dev team are captured as Analyze, Develop, and Test. Each column corresponds to work that the team does before that stage is considered done.



After you identify your team's workflow stages, you're ready to configure your Kanban board to map to them. After you configure the Kanban board, you can use it to update status, reassign work, and reorder items to reflect changing priorities.

If you're just getting started, review Kanban basics to get an overview of how to access your board and implement Kanban.

> **NOTE**
> If you want to add columns to a sprint taskboard, see Customize a taskboard. To add columns to a backlog or query results, see Change column options.

> **NOTE**
> If you want to add columns to a taskboard, you need to customize the workflow. For details, see Add or modify a work item type. To add columns to a backlog or query results, see Change column options.
>
> For an overview of the features supported on each backlog and board, see Backlog, board, and plan views.

Why configure your Kanban board columns?

The main reason to configure your team's Kanban board columns is to support your team's workflow and Kanban processes. These processes might include triaging work before committing to it, managing handoff from one team member to another (such as development to test), managing work in progress (WIP), and more.

You can also use your Kanban board to do the following tasks:

- Visualize work that's in progress, nearing completion, and completed.

- Filter your board to focus on select work based on assignment to a team member or sprint, tags, or parent feature. For details, see Interactively filter backlogs, boards, queries, and plans.
- Update workflow status through drag-and-drop operations, as shown in Update status later in this article.
- Reorder cards to change priority of work items, as described in Change priorities later in this article.
- View and quickly assign values to key fields, as described in Customize cards.
- Create queries and charts based on board columns, as illustrated in Track board column status later in this article.
- Review a cumulative flow diagram based on column assignments, as described in View and configure a cumulative flow diagram.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.

- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Also, we recommend that you review the following articles:

- Configure and customize Azure Boards
- Set up your backlogs and boards
- Workflow states and state categories

Customization sequence for board columns

Before you configure your team's Kanban board columns, make sure that you complete the following tasks as much as possible. Otherwise, you'll find yourself revisiting your configuration. Revisiting a single team's configuration is one thing, but if you support many teams, you're adding extra work by skipping this step.

Process administrator:

1. Add custom work item types that you want to appear on your backlog or board. For details, see Add and manage work item types.
2. Customize your product and portfolio backlogs. Customization ensures that the chosen work item types appear on the backlogs and boards. For more information, see Customize backlogs and boards.
3. Customize workflow states. Each workflow state appears as a column on the Kanban board. For details, see Customize a workflow.

Team administrator:

1. Set the backlogs that you want to be active for your team. You only need to configure the Kanban boards of backlogs that your team will use. For details, see Select backlog navigation levels for your team.
2. Determine how to track bugs. Bugs might have different workflow states from other types of work items in the **Requirement** category. As a result, bugs must be mapped separately from other work item types. For details, see Show bugs on backlogs and boards.
3. Identify the columns that your team will use to support its workflow and Kanban processes. For more information, see Map the flow of work later in this article.
4. Add, remove, or rename columns and map workflow states to columns. For more information, see Manage columns on your Kanban board later in this article.

For example, if you add bugs or other work item types to appear on a Kanban board, you potentially introduce other workflow states. New workflow states request you to adjust the Kanban column-to-state mappings when one of the following customizations is made:

- Your team admin chooses to show bugs on backlogs and boards.
- Your project admin adds work item types (WITs) to backlogs and boards.
- Your project collection or project admin customizes the workflow for a WIT in the **Requirement** category: inherited process or on-premises XML process.

What is workflow state mapping?

One of the main configuration details to understand for correctly adding columns to your Kanban board is how workflow states map to column states. The Kanban board and other Azure Boards tools use two categories to group WITs that they want to treat the same:

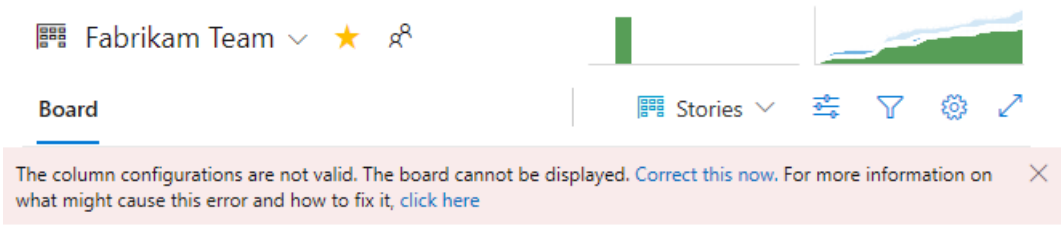
- **Work item type categories**:
  - Only work items whose work item types belong to the **Requirement** category appear on the product board.
  - Only work items whose work item types belong to the **Feature** category appear on the **Feature** portfolio board.
  - Only work items whose work item types belong to the **Epic** category appear on the **Epic** portfolio board.
  - Only work items whose work item types belong to a custom category appear on a custom portfolio board.
- **State categories**:
  - There are only four state categories: **Proposed**, **In Progress**, **Completed**, and **Removed**.
  - Workflow states defined for a work item type belong to one of the four state categories.
  - For each work item type and each Kanban board column, the workflow state must be specified (except workflow states in the **Removed** category).
  - The first Kanban board column is automatically mapped to the default state for each work item type's **Proposed** state category.
  - For middle Kanban board columns, you can map the column to any of the states that belong to the **Proposed** or **In Progress** state category. Also, you can only specify WIP limits, split columns, and definition of done for these columns.
  - The **Completed** state category can only map to the last Kanban board column.

TIP

If a workflow state isn't mapped to a Kanban column, that state doesn't appear on the Kanban board. It's allowed, but we don't recommend it.

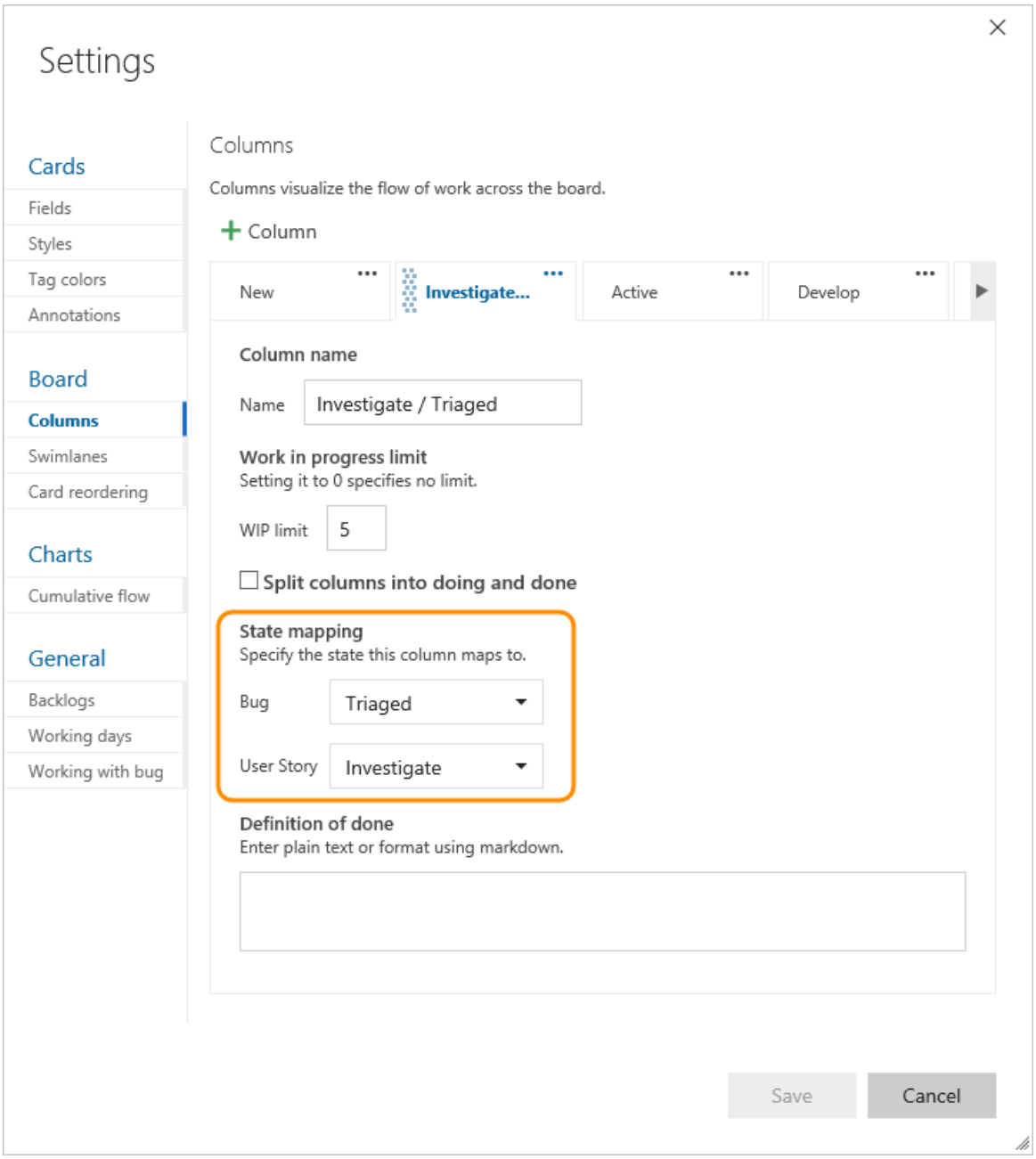
Correct your Kanban board configuration

If you see the following error when you open your Kanban board, you need to correct the configuration. The main reason for this error is that the workflow states of work item types that have been added to the **Requirement** category aren't mapped to the column.



Select **Correct this now** to open the **Settings** dialog.

In this example, two new states are added: **Triaged** for bug, and **Investigate** for user story. Each state is then mapped to an existing or new column. After each state is mapped to a column, the Kanban board displays the work items assigned to these states.



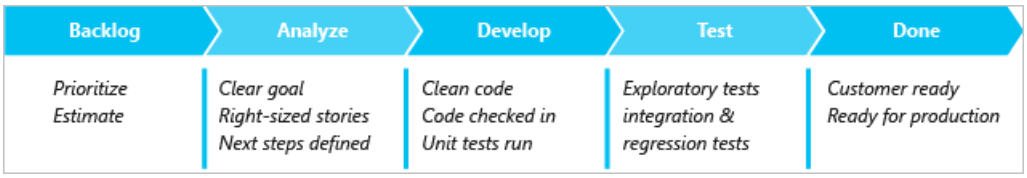
Map your workflow stages

It's best if you involve the entire team to identify an initial set of workflow stages. Each team member provides useful perspectives to capture and further deepen team understanding of the end-to-end processes.

To get started, ask your team these questions:

- *What types of activities do we regularly complete?*
- *What natural handoffs occur within our team or from our team to other teams?*
- *What activities will help reinforce our team policies, such as analysis, code review, or design acceptance?*
- *What work needs to occur at each stage?*
- *What limits do we want to set for each workflow state?*
- *Will a split column be useful for one or more columns?*

Our example development team came up with these stages as essential to their process:



- **Backlog**: Make a prioritized list of work items that the team isn't yet ready to work on.
- **Analyze**: Identify well-understood and shared acceptance criteria, along with overall work required to develop and test the item.
- **Develop**: Code and run unit tests for the item.
- **Test**: Run exploratory, automated, integration, and other tests.
- **Done**: Hand off to production because the item is ready.

You can always revisit these initial stages later and adjust.

Another idea is to capture the list of items that your team identifies as critical-to-complete for each stage. You can use that information later to fill out the definition of done for each column.

Manage columns on your Kanban board

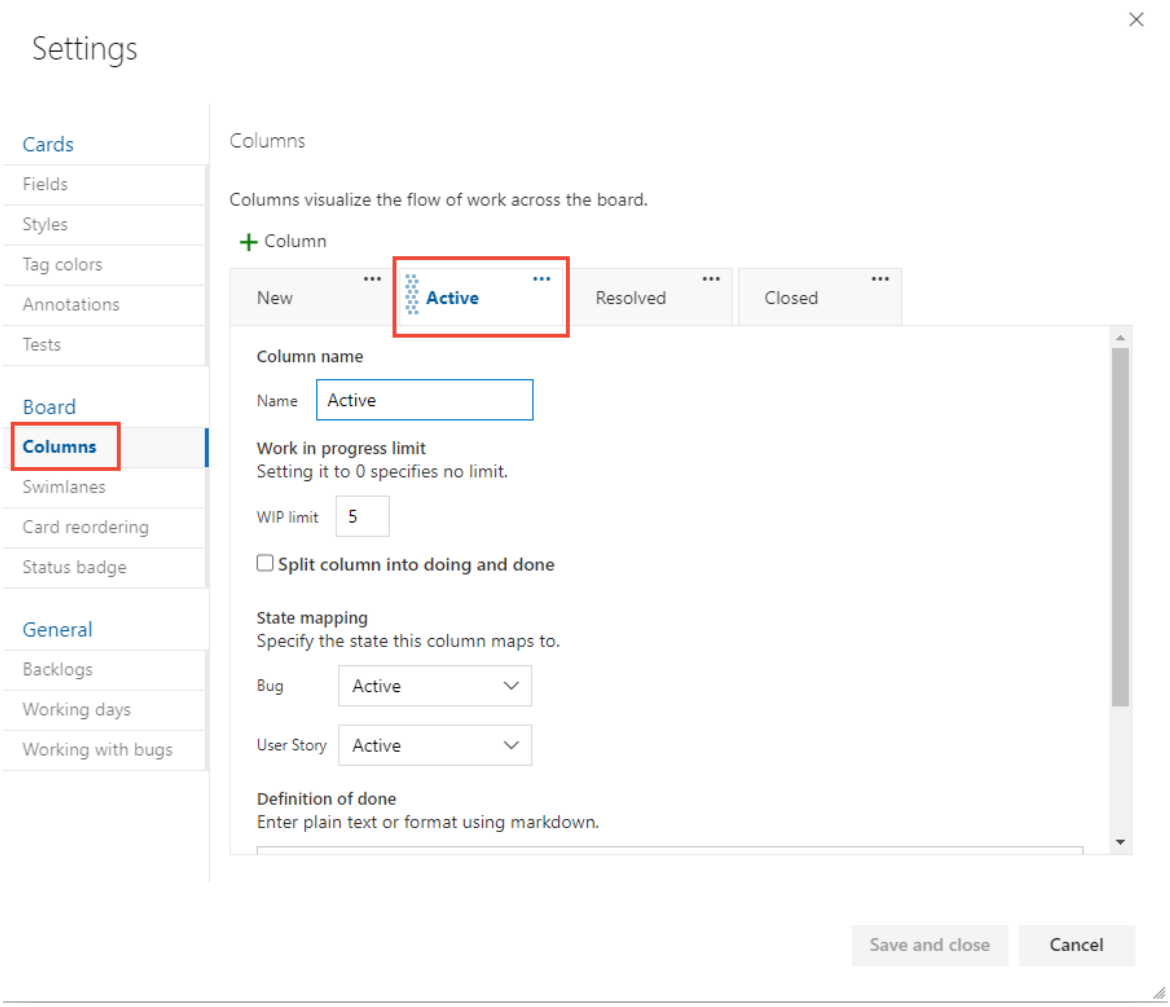
Now that you know the essentials of how to work with your Kanban board, here's how you get it to look the way you want.

Column titles and choices depend on the process that you used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Select **Configure team settings** to configure the board and set general team settings.



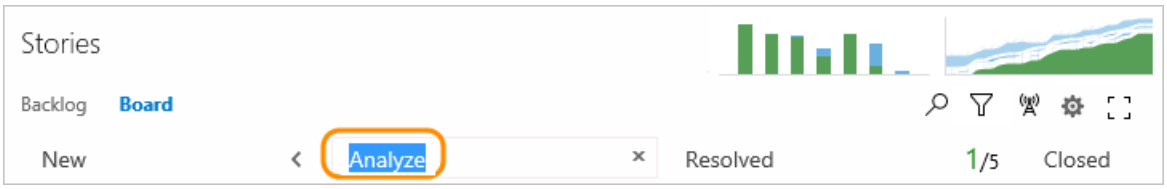
3. Select **Columns** and then a column tab to see all the settings that you can modify. Your initial column settings look similar to the settings shown in the following image.



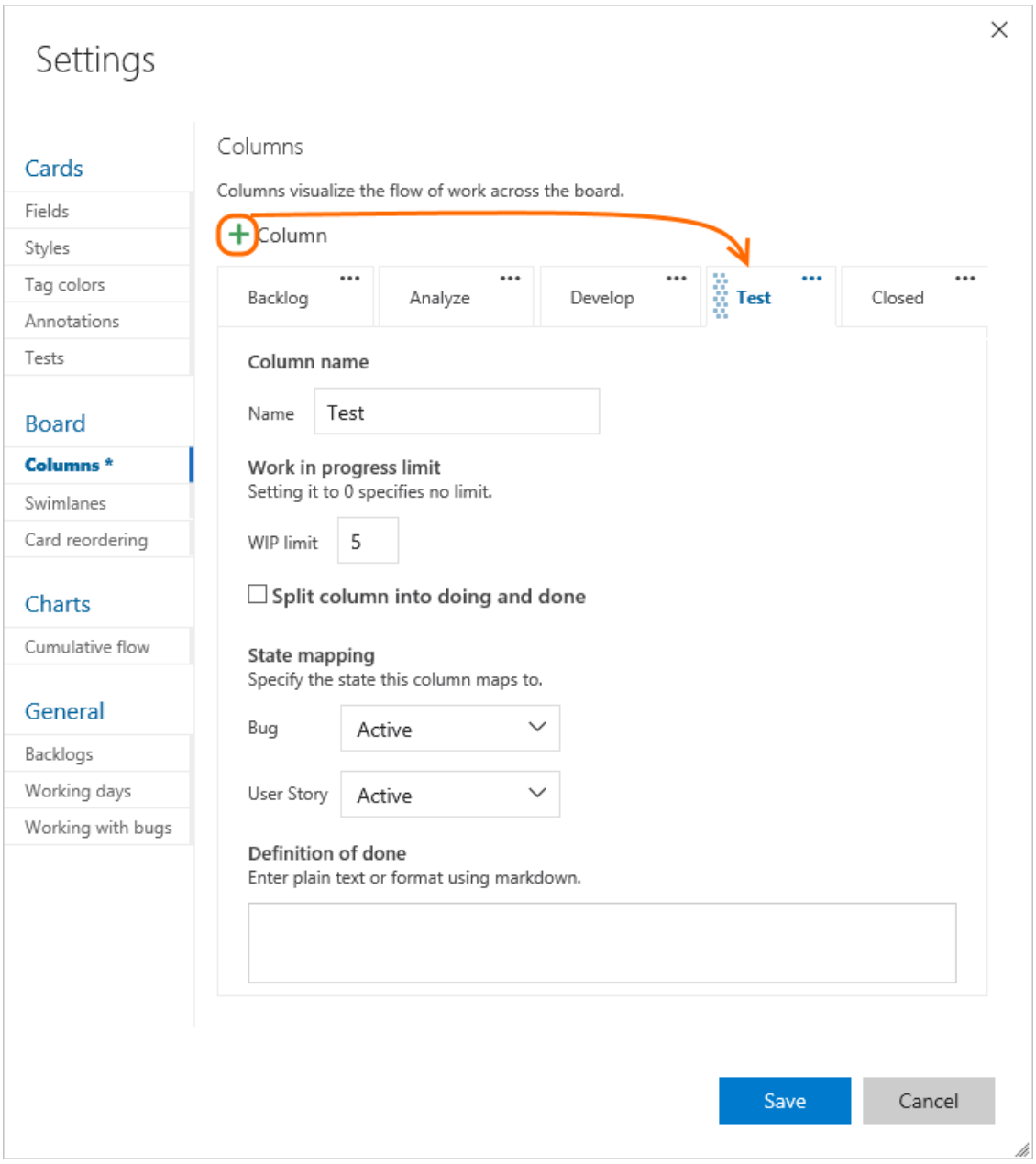
4. Change your column titles to map to your workflow stages. You can add, rename, and move columns to support more stages.

   Rename the first three columns to **Backlog**, **Analyze**, and **Develop**. Then, add a column and label it **Test**.

   You can rename a column directly from the Kanban board.



   Or, you can open the dialog and change one or more settings for a Kanban column.

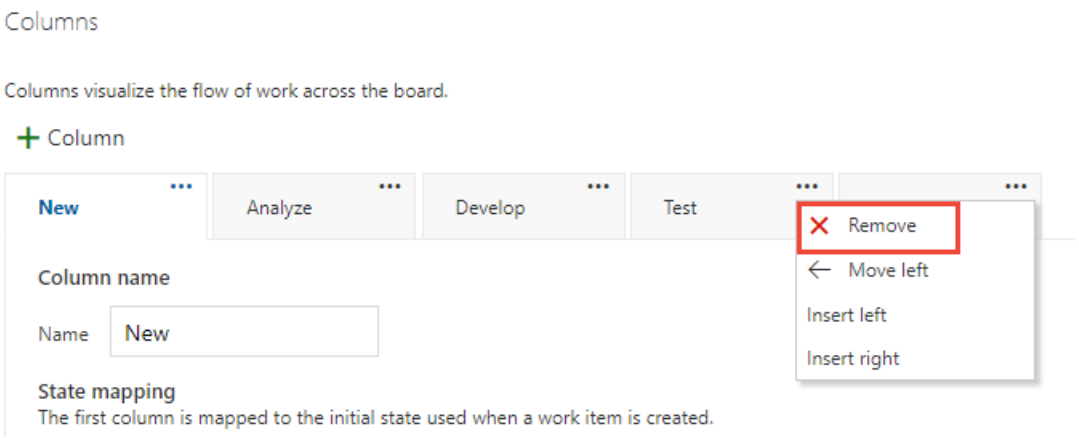


5. To change the column order, drag the column tab to the position that you want.

6. To delete a column, first make sure that the column doesn't contain any work items. If it does, move the items to another column. Then:

   a. Open **Settings**, select **Columns**, and select **Actions** ··· from the column tab.

   b. Select **Remove** from the menu.



7. Change state mappings as needed for added columns, added workflow states, or added WITs.

Usually, you need to update state mappings when you change the Working with bugs setting, add WITs to the Requirement category, or customize the workflow.

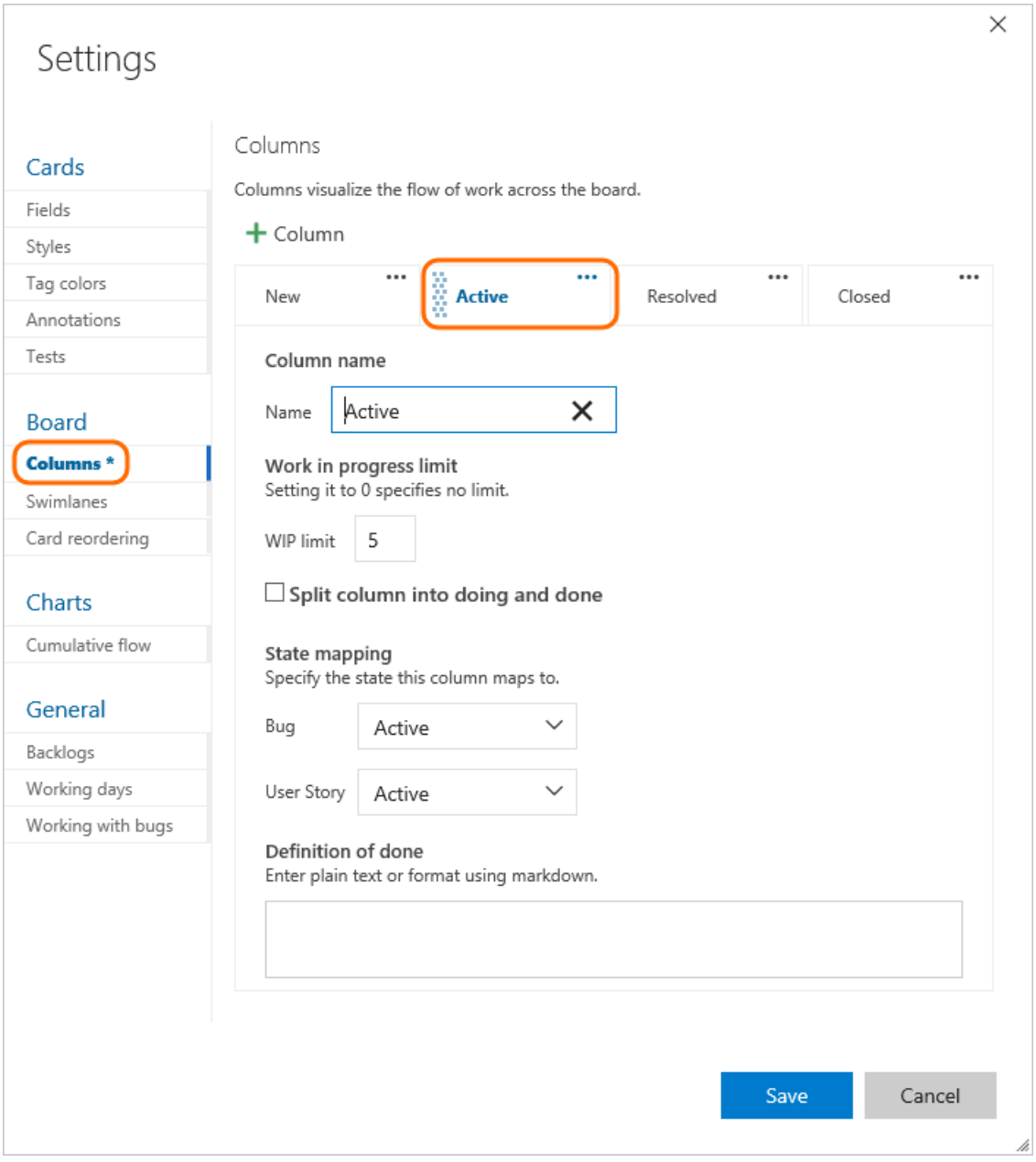
8. When you're done with your changes, select **Save**.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.

2. Select **Settings** to open the common configuration settings dialog for the Kanban board.



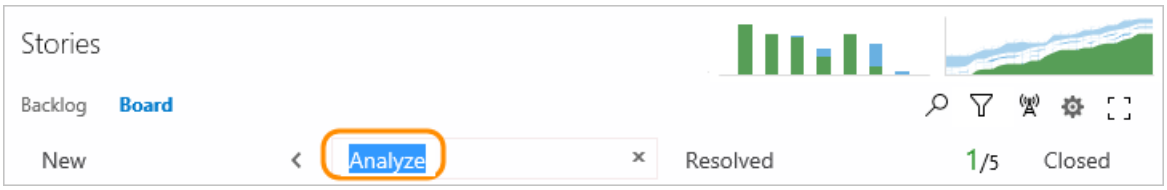
3. Select **Columns** and then a column tab to see all the settings that you can modify. Your initial column settings will look something like the following example.



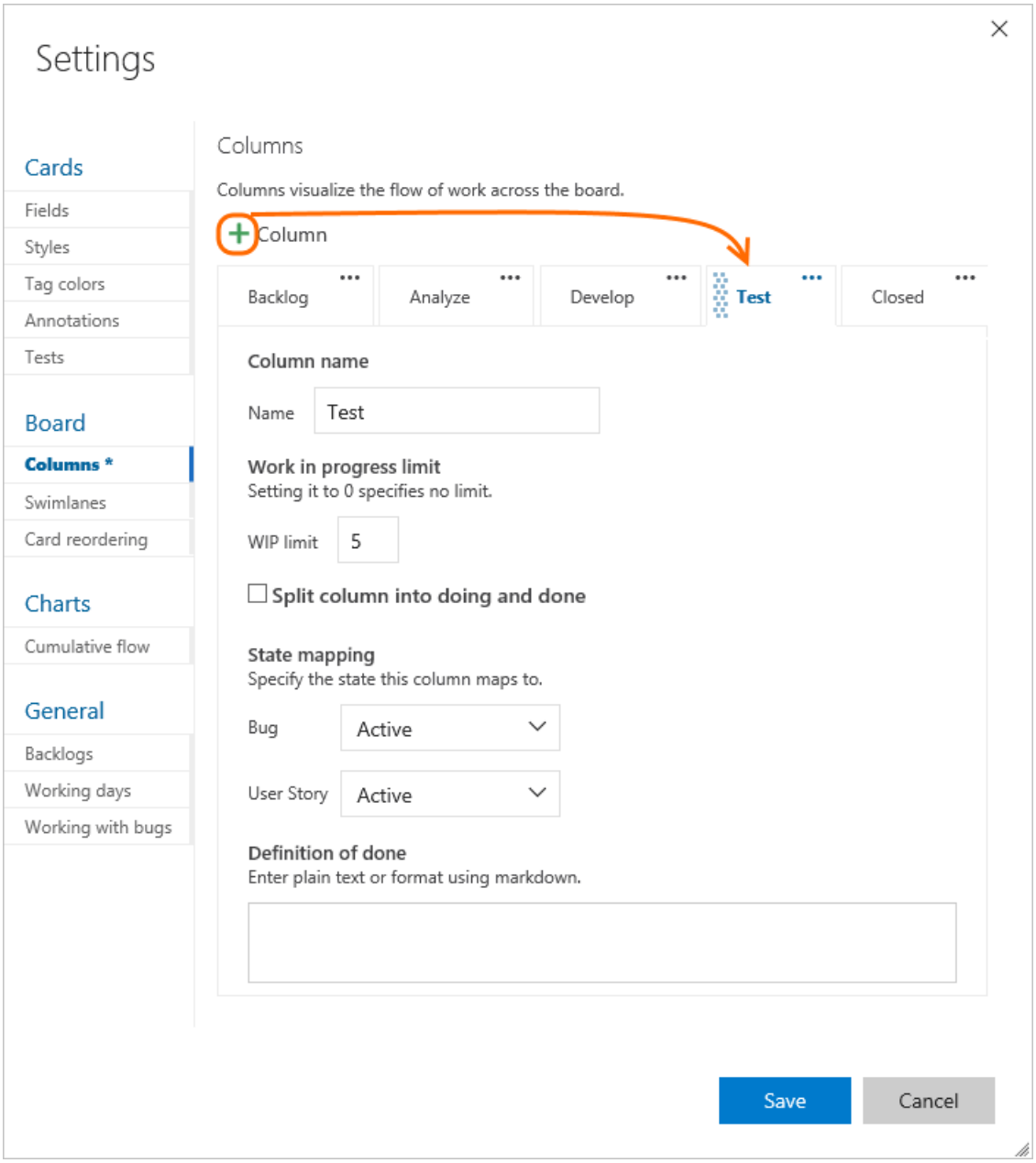
4. Change your column titles to map to your workflow stages. You can add, rename, and move columns to support more stages.

Rename the first, second, and third columns to **Backlog**, **Analyze**, and **Develop**. Then, add a column and label it **Test**.

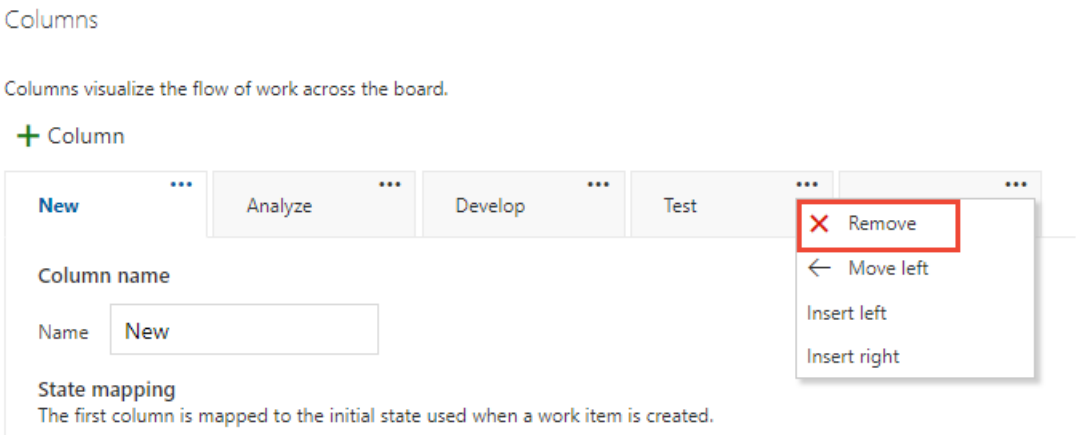
You can rename a column directly from the Kanban board.



Or, you can open the dialog and change one or more settings for a Kanban column.



5. To change the column order, drag the column tab to the position that you want.
6. To delete a column, first make sure that the column doesn't contain any work items. If it does, move the items to another column. Then, select **Actions** ··· on the column tab and select **Remove** from the menu.



7. Change state mappings as needed for added columns, added workflow states, or added WITs.

   Usually, you need to update state mappings when you change the Working with bugs setting, add WITs to the Requirement category, or customize the workflow.

8. When you're done with your changes, select **Save**.

Use your Kanban board

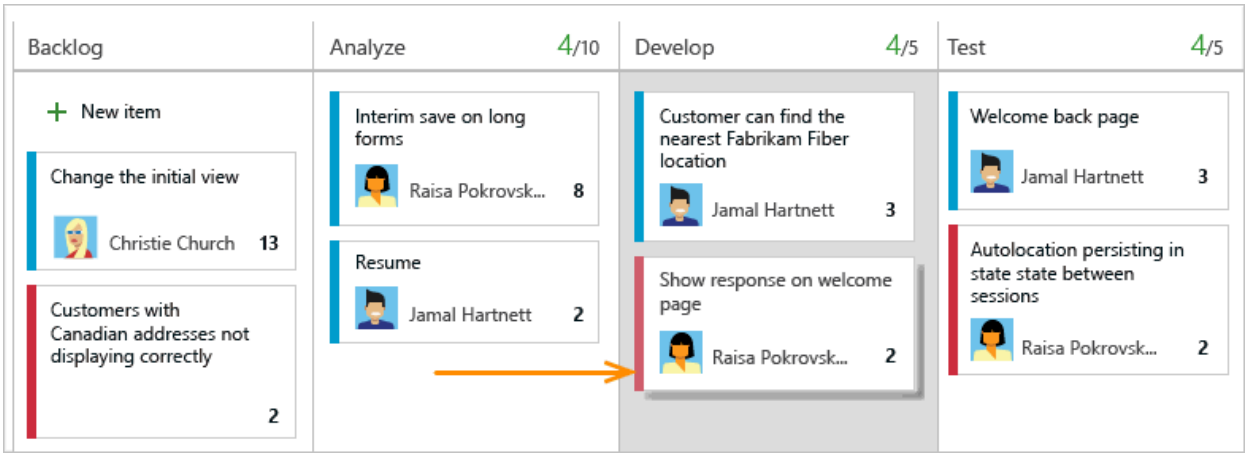
After you have a well-configured Kanban board, you can start using it. To learn how, see Start using your Kanban board.

> **IMPORTANT**
> Work items that appear on more than one team's Kanban board can yield results that don't meet your expectations because each team can customize its Kanban board columns and swimlanes. The values assigned to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields might differ from what you expect when another team updates the work item from a different board. To learn more, see Add, review, and update work items in Azure Boards.

Update status and handoff items

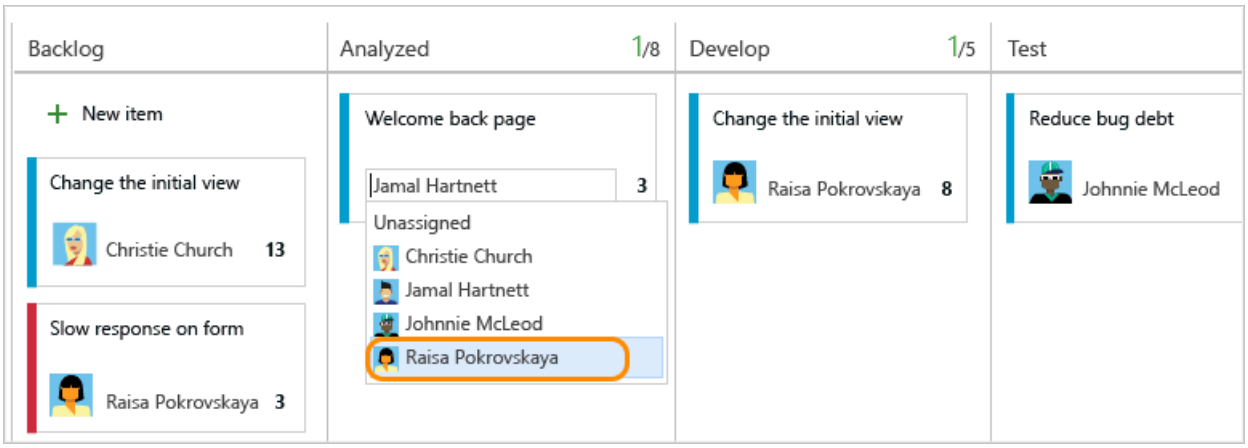
Using your Kanban board couldn't be simpler. You update the status or change priorities by using drag-and-drop operations.

For example, to signal when work can start in a downstream stage, drag items to the next column.



You can move an item from one column to any other column on the board. If you discover that more work is needed at an earlier stage, you can move the item backward; for example, from **Test** to **Analyze** or **Develop**.

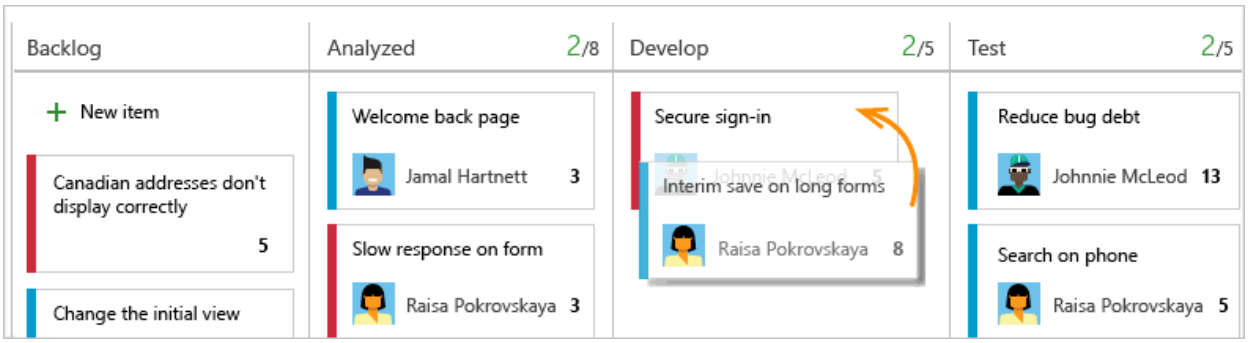
To hand off work to another team member, reassign it directly from the board.



Team members who receive the handoff can set alerts to get immediate email notifications of their newly assigned work.

Change your team's priorities

To keep teams working on the highest-priority items, you'll want to react quickly when a change in priority occurs, even after work starts. With your Kanban board, it's a snap. Simply drag an item up or down within a column.



Track Kanban column status

Your Kanban board is one of several tools that you have for tracking work. The query tool allows you to list a subset of work items for the purpose of review, triage, update, or chart generation. For example, you can create a query to list all active user stories (specify two clauses: `Work Item Type=User Story` and `State=Active`).

But what if you want to list items based on their Kanban column assignment? Can you do that? Yes, you can track column moves on a Kanban board by using the Board Column and Board Column Done fields.

FAQs

- Is there a way to widen columns on a Kanban board?
- Can I query based on Kanban board columns?
- Can I view a query as a Kanban board?
- Is there a way to copy a Kanban configuration to another team?

Supported Visual Studio Marketplace extensions

- Query Based Boards
- Azure Boards Kanban Tools
- Roll-up Board
- FlowViz

Related articles

That's about all you need to know about working with Kanban columns. Here are a few more options for customizing the look and feel of the board:

- Customize cards
- Accelerate work with swimlanes
- Show bugs on backlogs and boards
- Enable live updates

REST API resources

To interact programmatically with Kanban board and other team settings, see the Boards reference in the REST API.

Customize cards on a Kanban board

10/4/2022 • 15 minutes to read • Edit Online

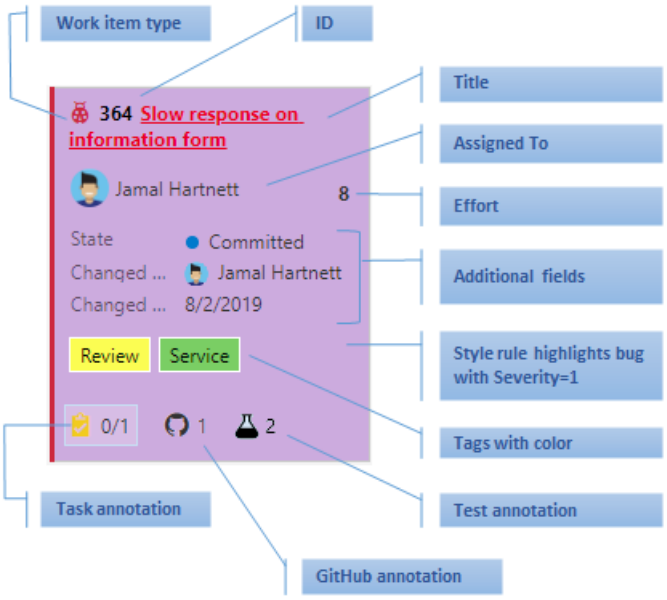
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With the Kanban board, you gain a rich set of tools and a rich set of customization options. Kanban boards display work items as cards. Each card corresponds to a work item that you use to share information, track status, and assign work. Information rich cards provide at-a-glance info of interest to you and your team and allow you to update a field without opening the work item. With style rules, you can highlight cards and tasks based on the criteria you set.

When you enable annotations, you gain quick access to linked work items and other features. If you're new to working with the Kanban board, see Kanban basics.

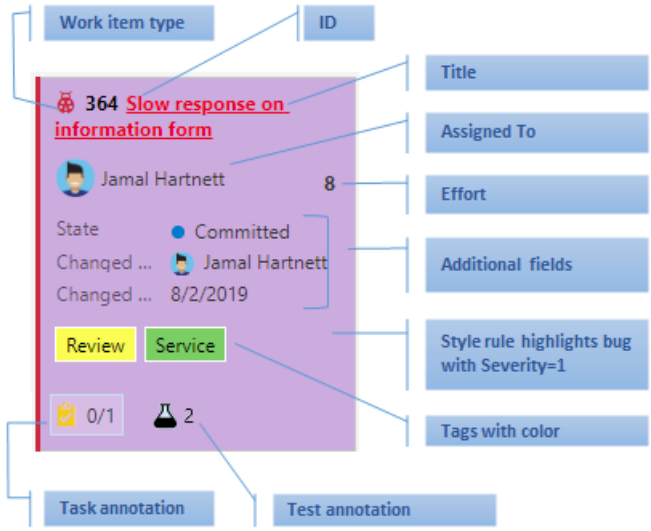
In the card shown below, the following options have been set for the bug work item type:

- Show all core fields: ID, Assigned To, Story Points, Tags
- Show three more fields: State, Changed By, and Changed Date
- Apply tag colors
- Apply styling rule to display bugs with Severity=1 as yellow and bold and underline the Title field
- Enable Task, GitHub, and Test annotations



In the card shown below, the following options have been set for the bug work item type:

- Show all core fields: ID, Assigned To, Story Points, Tags
- Show three more fields: State, Changed By, and Changed Date
- Apply tag colors
- Apply styling rule to display bugs with Severity=1 as yellow and bold and underline the Title field
- Enable Tasks and Test annotations



> **NOTE**
> This article addresses customization of a Kanban board. For information on customizing a Taskboard, see Customize sprint Taskboards.

You can either increase or simplify the information that displays on your cards. It all depends on what's of interest to your team.

- *Does your team like to refer to work items by their ID?*
- *Do they want to see estimates?*
- *Do they want to highlight work items according to set criteria?*
- *Or, will just the bare bones of title and assignment suffice?*

Your best bet is to show fields on cards based on what your team frequently refers to or updates when using the Kanban board. Also, add fields with information that you can use to filter the board.

> **NOTE**
> You can customize a work item type which is different than customizing the card displayed on the Kanban board. You customize a work item type by adding fields, changing the workflow, adding custom rules and more. You can also add custom work item types and custom backlog levels. For details, see Customize an inheritance process.

> **NOTE**
> You can customize a work item type which is different than customizing the card displayed on the Kanban board. You customize a work item type by adding fields, changing the workflow, adding custom rules and more. You can also add custom work item types and custom backlog levels. For details, see Customize the On-premises XML process model.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.

- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Card customization options

Once you open the Settings dialog for the Kanban board, choose one of the following tabs to set your customization options.

Settings tab

Use to...

Fields

Add or remove fields from cards.
Includes adding the **Parent** field to cards.

Fields

Add or remove fields from cards.

Styles

Add styling rules to change card color and title style based on field criteria.

Tag colors

Specify a tag color, and enable or disable a tag color.

Annotations

Enable or disable annotations to appear on cards.

Tests

Configure how you want tests to appear and behave on the cards.

Card reordering

Choose expected behavior when reordering cards on the board.

> **NOTE**
> Each team can customize the cards for their Kanban board. Board settings are not inherited from other teams that they may share portions of area paths.

Card customization sequence

Before you configure the cards, you'll want to make sure the following tasks are complete as possible. Otherwise, you'll find yourself revisiting your configuration.

Process Administrator:

1. Add custom work item types that you want to appear on your backlog or board. For details, see Add and manage work item types.
2. Customize your product and portfolio backlogs to ensure all work item types you want to have will appear on the backlogs and boards. For details see Customize backlogs & boards.
3. Customize each work item type to have any custom fields you want to show. For details, see Customize a workflow.

Team Administrator:

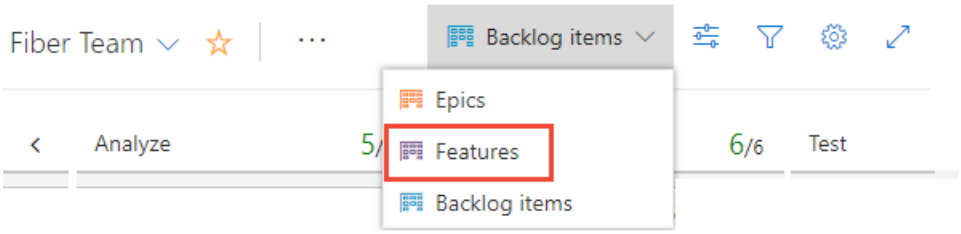
1. Meet with your team and determine how the team wants to manage bugs, similar to requirements or tasks.
2. Add any tags you want to customize on cards to work items.
3. Meet with your team and determine which annotations should appear on cards and how they want to configure inline tests.

Open your Kanban board settings

If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.

You can customize cards that appear on the Kanban board for your product backlog or portfolio backlog such as features and epics. You follow similar steps, however you start from the corresponding portfolio backlog.

1. Open your Kanban board..
2. Choose the backlog level you want to customize.



3. Choose the gear icon to configure the board and set general team settings.



4. When done with your changes, choose **Save**.

1. Open your Kanban board.
2. Choose the backlog level you want to customize.
3. Choose to open the common configuration settings dialog for the Kanban board.



4. When done with your changes, choose **Save**.

Choose which fields appear on cards

By showing a field on a card, you can edit that field directly on the board without having to open the card, unless it's a read-only field, such as the Change By and Changed Date fields. This quick update feature is useful when you need to update many work items at once. For example, you can add estimates for backlog items on the Kanban board.

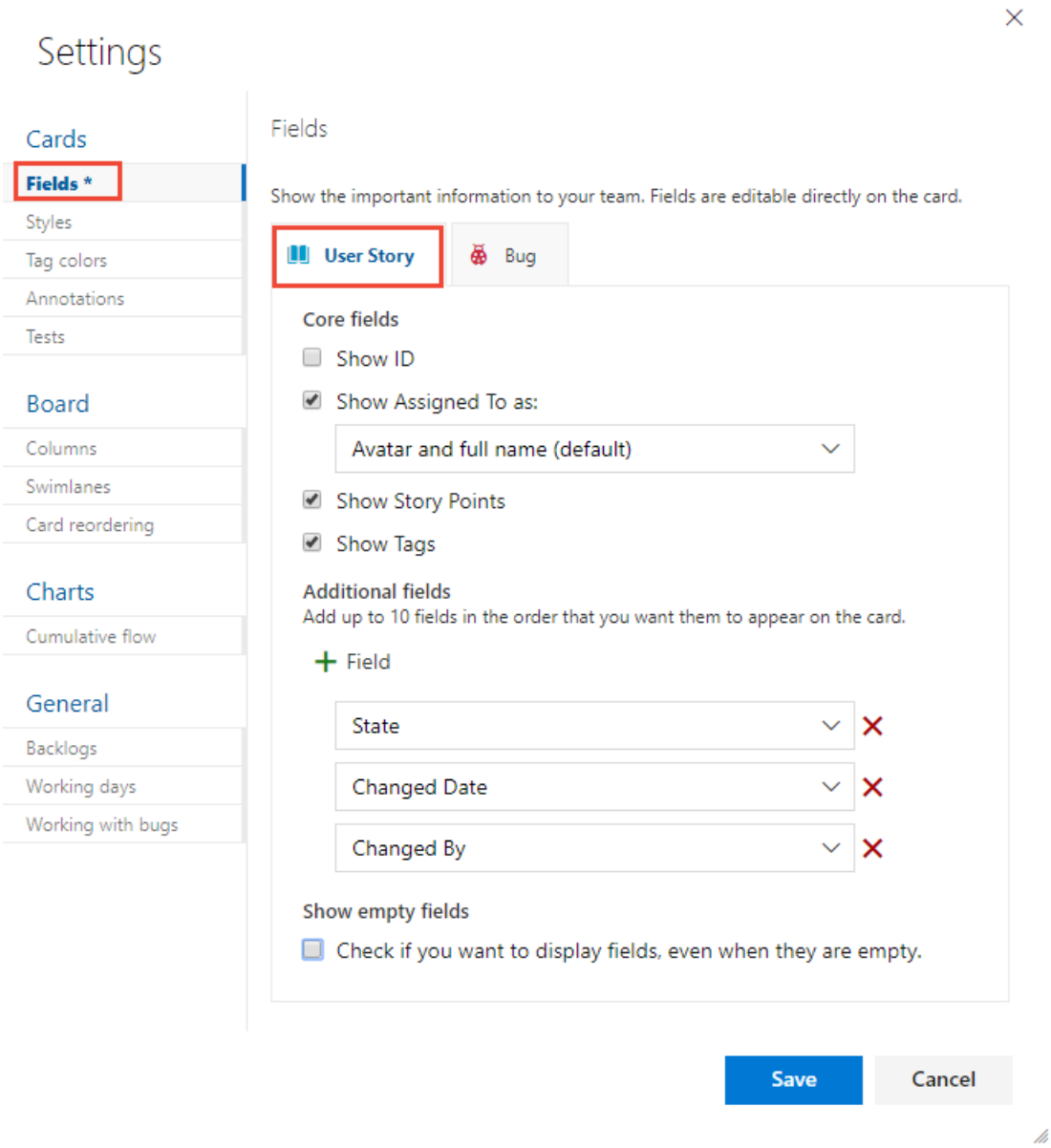
To change the Title, choose the pencil icon in the upper-right corner.

As a first step, you'll see that you can determine which fields appear on each card type. To add a custom field, you must first add it to the process used to customize the project.

As a first step, you'll see that you can determine which fields appear on each card type. To add a custom field, you must first add it to the work item type definition.

1. From the Settings dialog, choose **Fields** and then a work item type to see all the settings you can modify. Your initial column settings appear similar to the image below.

   Here we choose User Story. Your choices will vary based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.



2. Place a check mark in the check box for those fields you want to have appear on the board.

   If you want work estimates to show, check **Show Effort**. **Show Effort** corresponds to these fields: Effort (Scrum), Story Points (Agile), and Size (CMMI).

3. To add a field, choose the + plus icon and enter the name of a field you want to add.

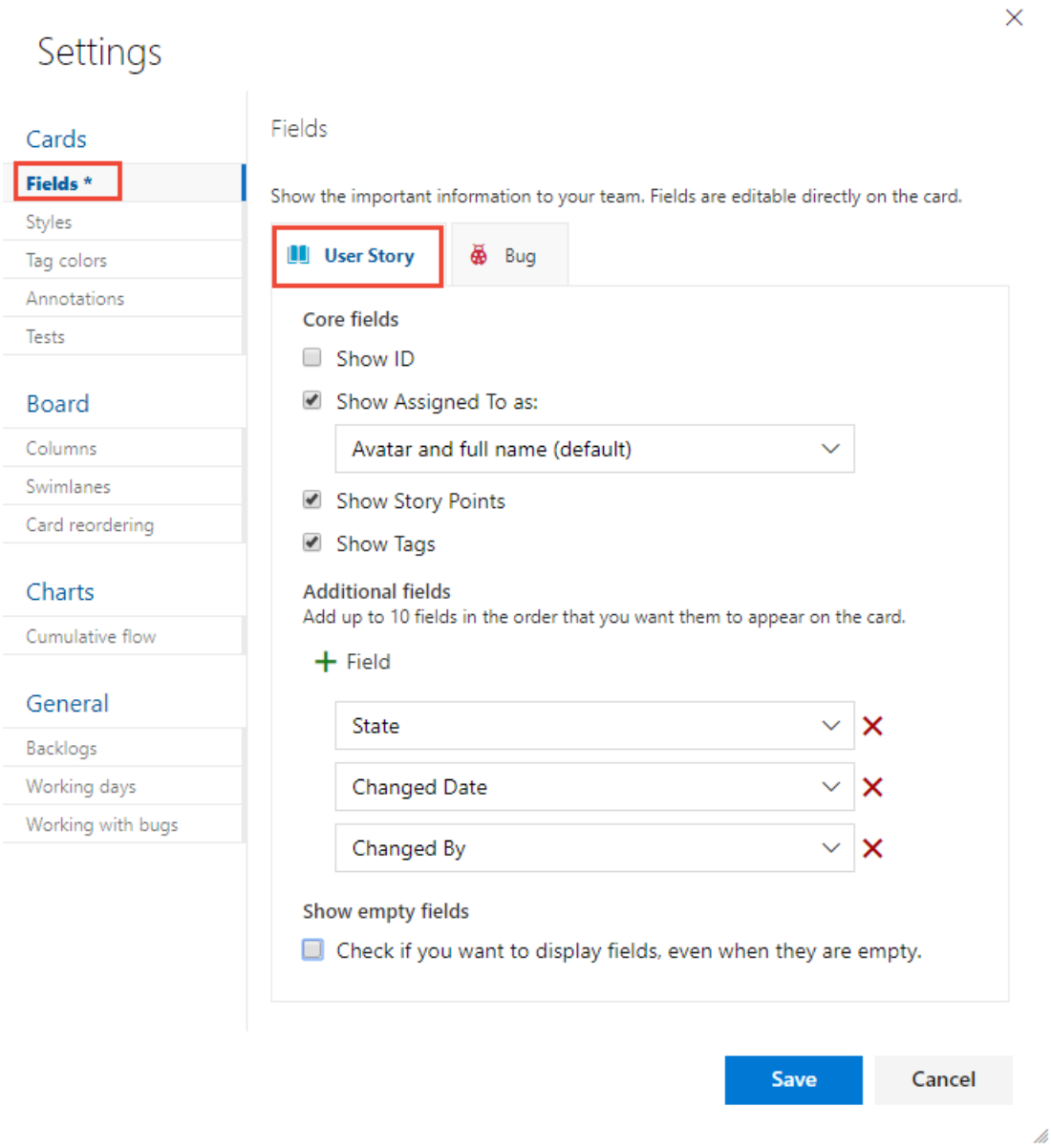
4. To remove a field, choose the ✕ delete icon next to the field.

5. When done with your changes, choose **Save**.

> **NOTE**
> To show the **Title** of the parent work item, choose the **Parent** field. Choosing the **Parent** title from a card opens the parent work item. To change the parent work item, open the child work item and remove the link and add a different parent work item. You can filter your board based on parent work items, whether the **Parent** field is added to cards or not.

1. From the Settings dialog, choose **Fields** and then a work item type to see all the settings you can modify. Your initial column settings will look something like this.

   Here we choose User Story. Your choices will vary based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.



2. Place a check mark in the check box for those fields you want to have appear on the board.

   If you want work estimates to show, check **Show Effort**. **Show Effort** corresponds to these fields: Effort (Scrum), Story Points (Agile), and Size (CMMI).

3. To add a field, choose the + plus icon and enter the name of a field you want to add.

4. To remove a field, choose the ✕ delete icon next to the field.

5. When done with your changes, choose **Save**.

Define style rules to highlight cards

With styling rules, you can cause cards to change color when their corresponding work items meet criteria that you set. Here, we highlight severity 1 bugs by having the cards display as yellow.



Examples of styling rules

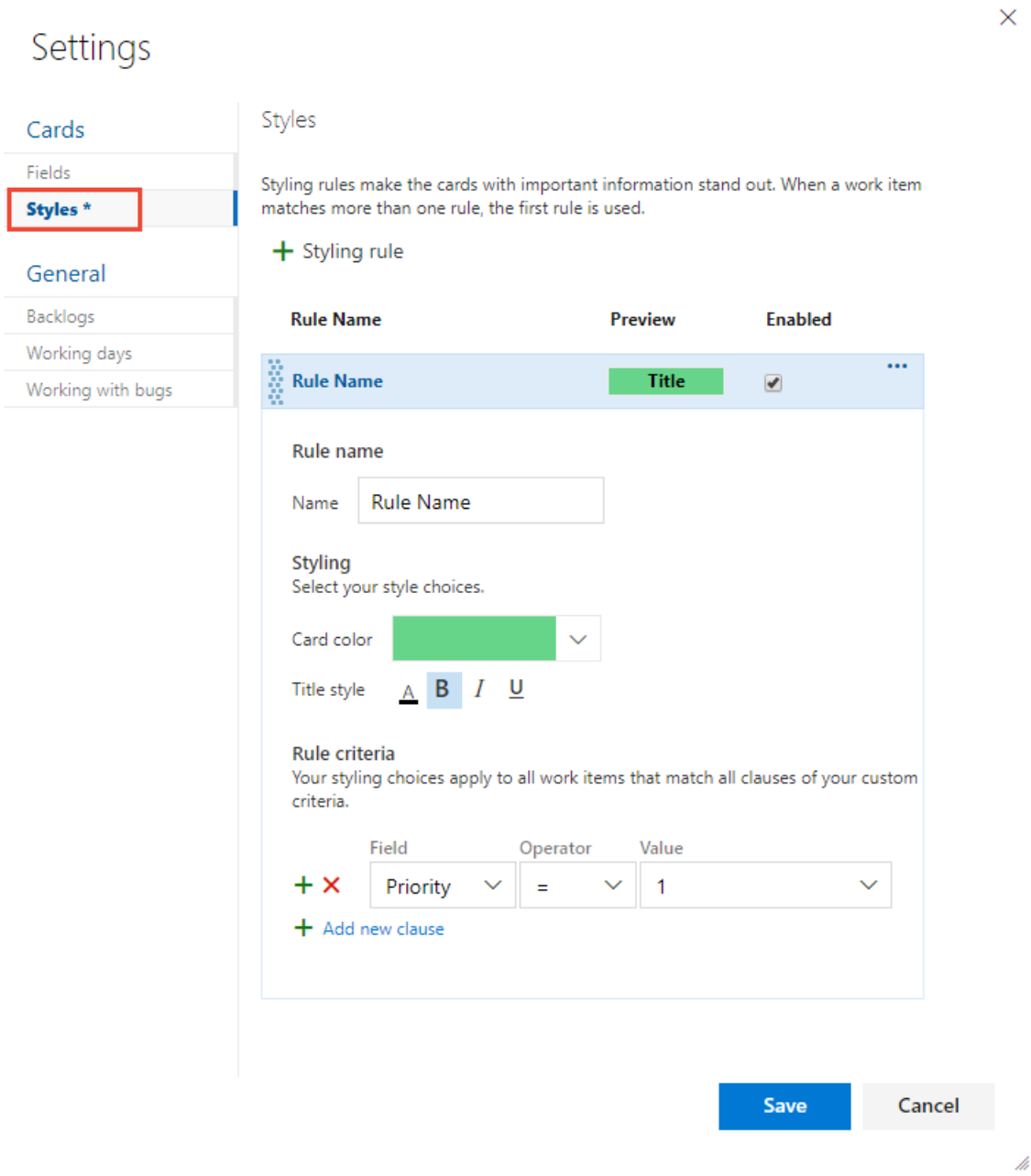
What rules should you apply to highlight work items? Here are a few examples and their associated criteria.

WORK ITEMS	CRITERIA
High priority items	`Priority = 1`
High effort items	`Effort 20` or `Story Points 20`
Stale items unchanged in the last 5 days	`Changed Date @Today-5`
Title contains a key word	`Title Contains Yes`
Severity 1 bugs	`Severity = 1 - Critical AND Work Item Type = Bug`
High value business items	`Business Value 50`
Items assigned to specific feature area	`Area Path Under Fabrikam Fiber\Phone`
Contains specific tag	`Tags Contain RTM`
Blocked tasks (Scrum process only)	`Blocked = Yes`

You can apply style rules to change the color of cards on Kanban boards and taskboards.

1. From the Settings dialog, choose **Styles** to specify a style rule. Choose the + plus icon to add a style. Select the color to apply to the card and define the criteria for the style rule.

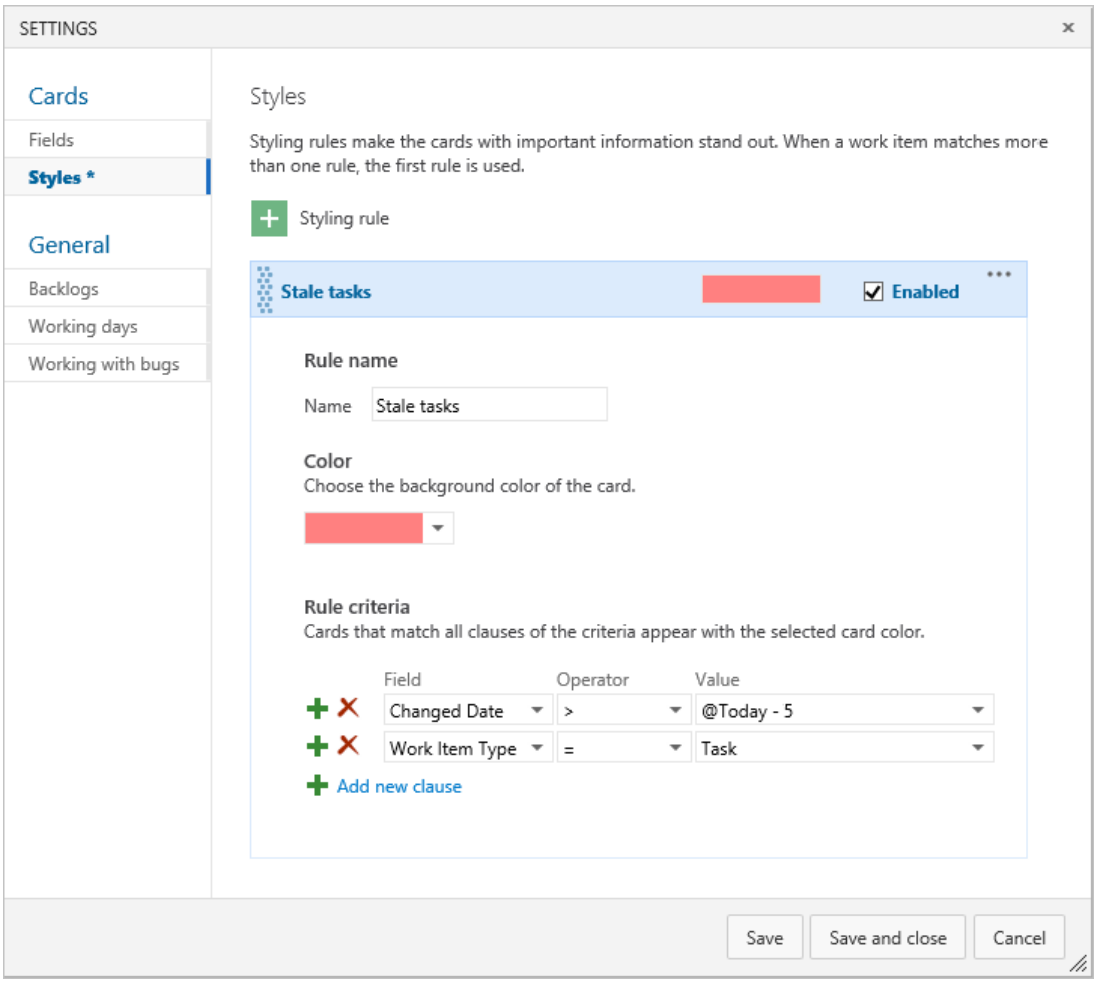
   In this example, we show the **Styles** dialog for the Dashboard.



Follow these rules when creating and ordering your style rules:

- The criteria you specify works in a similar fashion as when constructing a query.
- All clauses are considered AND clauses, grouping clauses isn't supported.
- Card rules apply to all work items that meet the rule criteria.
- Rule color applies to work items based on the order in which rules are listed. If you add more than one style rule, make sure that you move them in the order of most importance. Drag them into the order you want them applied.
- You can quickly enable and disable a style rule.

  Here we add a Stale tasks rule that highlights tasks that haven't changed in the last five days.

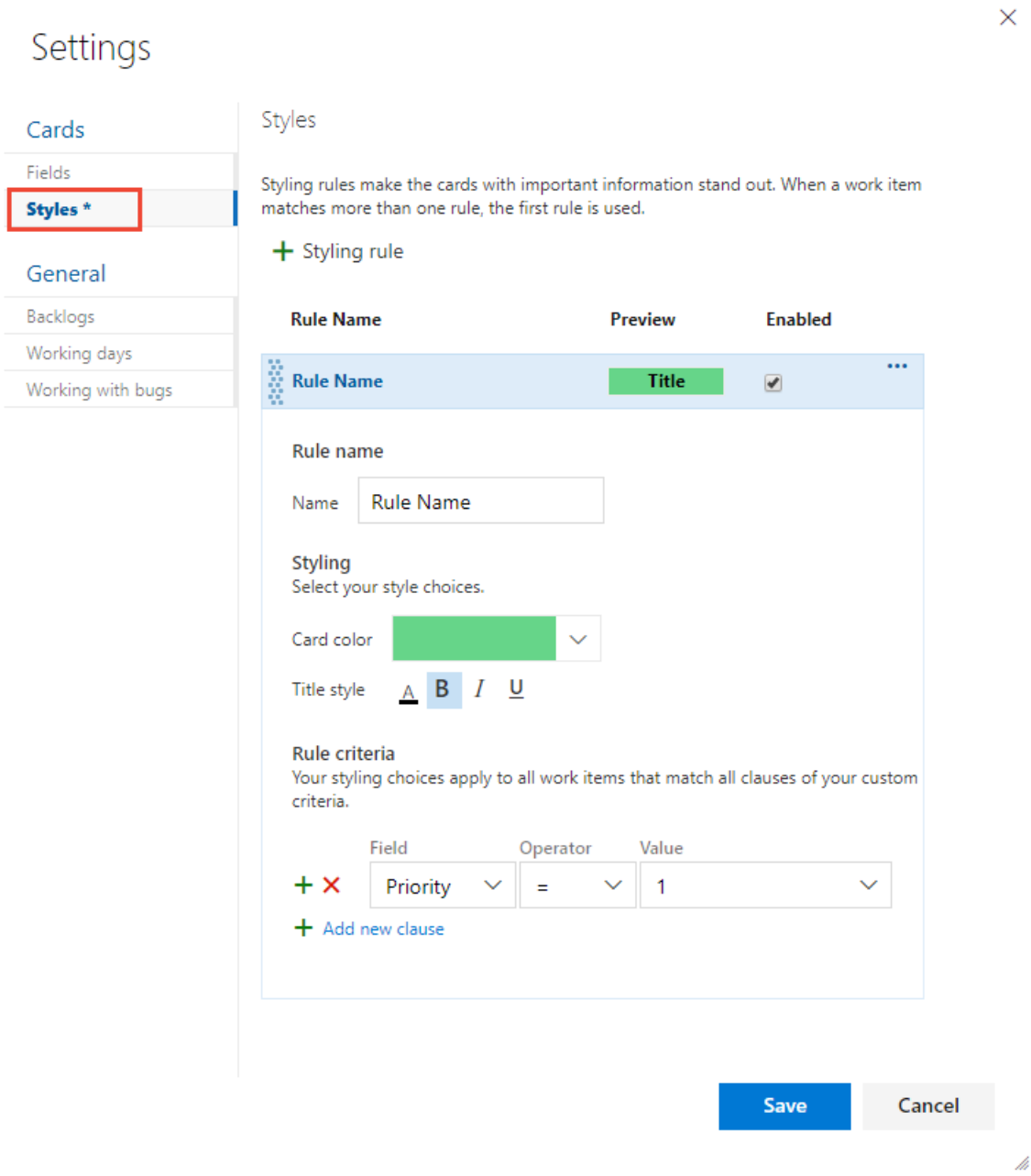


2. To copy or delete a style rule, choose the ••• actions icon and select **Clone** or **Delete**.

3. When done with your changes, choose **Save**.

1. From the Settings dialog, choose **Styles** to specify a style rule. Choose the + plus icon to add a style. Select the color to apply to the card and define the criteria for the style rule.

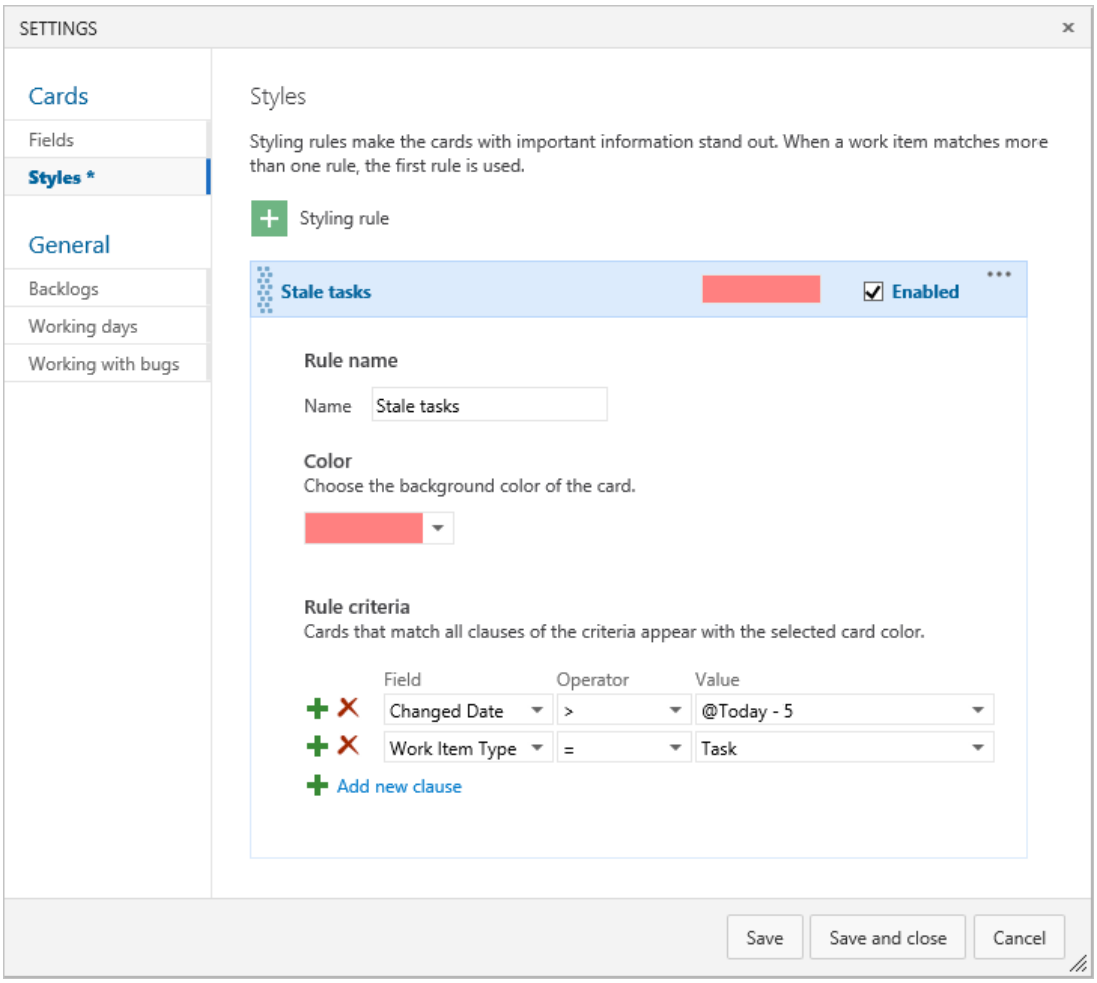
   In this example, we show the **Styles** dialog for the dashboard.



Follow these rules when creating and ordering your style rules:

- The criteria you specify works in a similar fashion as when constructing a query
- All clauses are considered AND clauses, grouping clauses isn't supported
- Card rules apply to all work items that meet the rule criteria
- Rule color applies to work items based on the order in which rules are listed. If you add more than one style rule, make sure that you move them in the order of most importance. Drag them into the order you want them applied.
- You can quickly enable and disable a style rule.

  Here we add a Stale tasks rule that highlights tasks that haven't changed in the last five days.



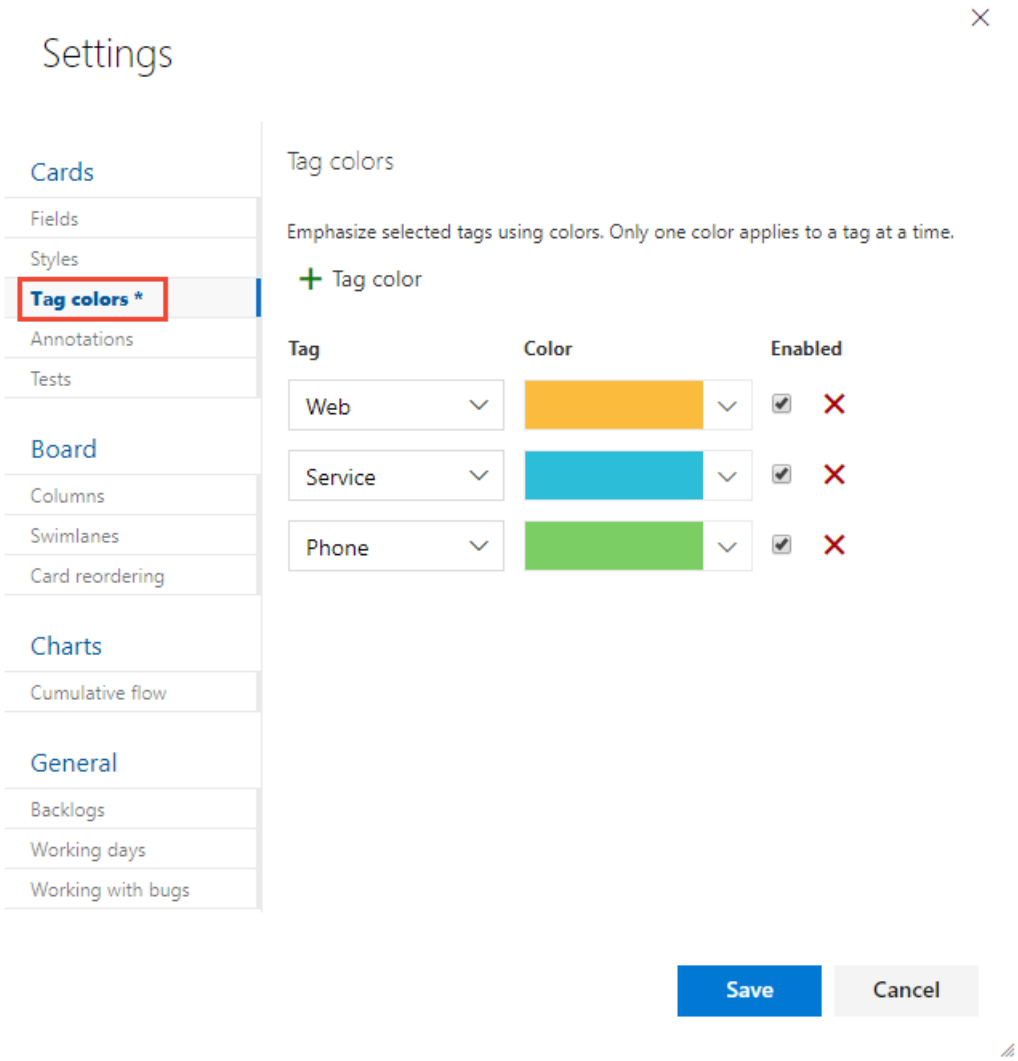
2. To copy or delete a style rule, choose the ··· actions icon and select **Clone** or **Delete**.

3. When done with your changes, choose **Save**.

Assign a color to a tag

Before setting tag colors, add tags to backlog items that you want to highlight with color.

1. From the Settings dialog, choose **Tag colors** and then choose the + plus icon to add a tag color. Then, select the tag and the color you want to appear on the cards.



TIP

If tags don't display on the cards, choose **Fields** and make sure that you've checked **Show Tags**.

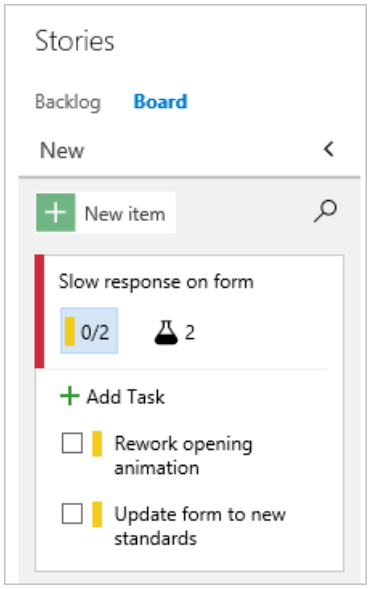
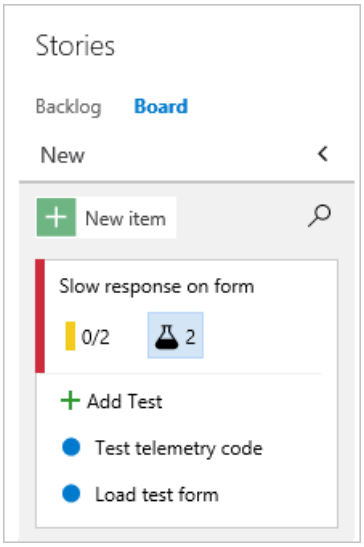
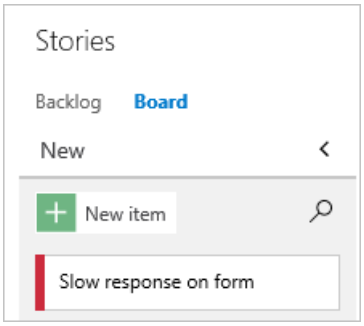
2. When done with your changes, choose **Save**.

Enable or disable an annotation

By default, all annotations are enabled that are applicable for the board you selected. These annotations include all work item types added to the next level backlog, GitHub, and Tests. Disable those annotations, which your team won't use or which you don't want teams to use for a particular backlog level.

When you disable an annotation, you also disable the feature to add the associated object from the Kanban board. For example, if you disable the Tests annotation, you disable the ability to add tests from the currently selected Kanban board.

As shown in the following examples, the **Task** and **Test** annotations indicate that two each of tasks and tests have been defined for the work item. Disable the annotations your team doesn't use from the **Annotations** tab.

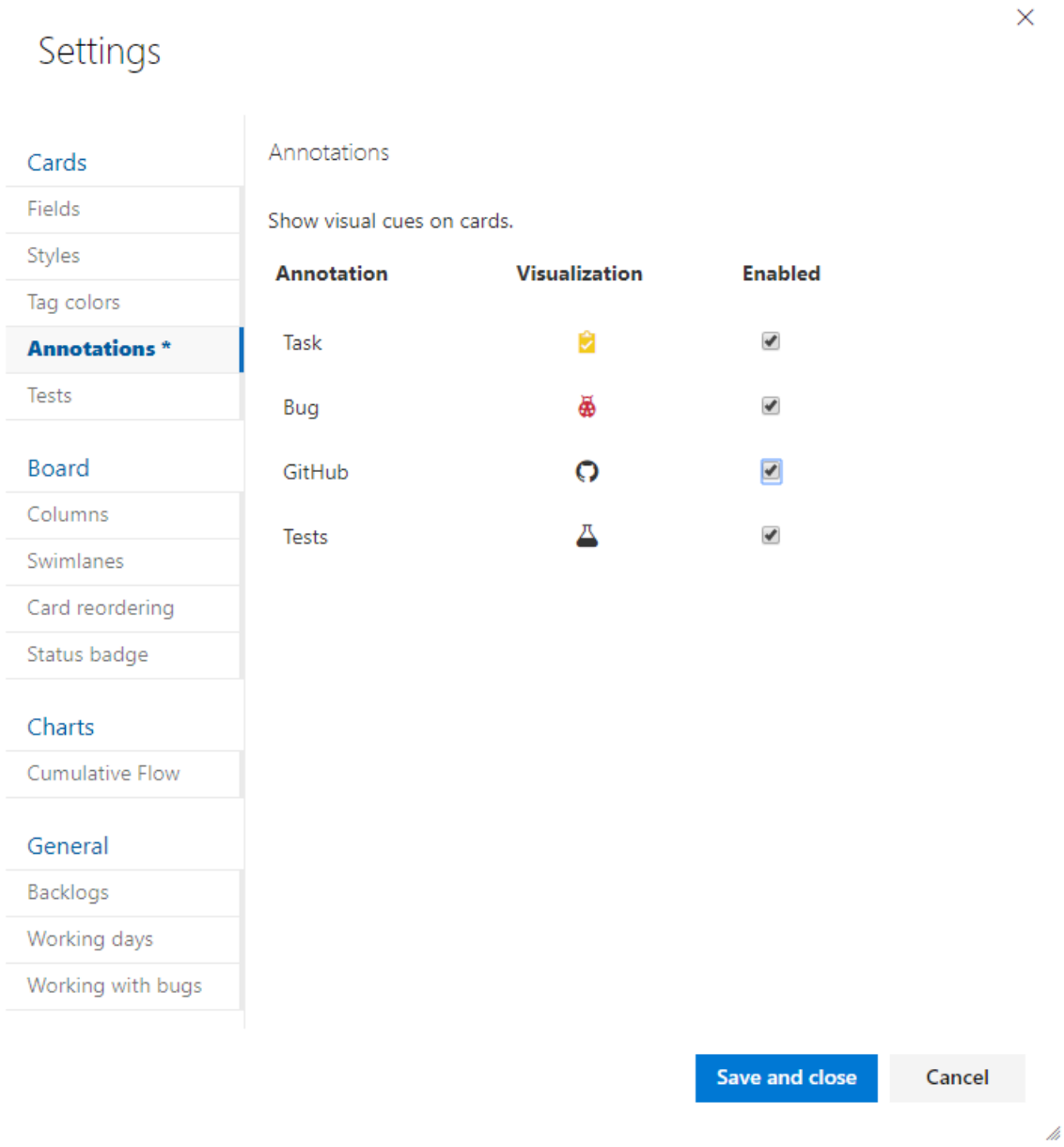
TASK ANNOTATIONS	TEST ANNOTATIONS	NO ANNOTATIONS
		

To learn more about using these features, see Add tasks or child items as checklists and Add, run, and update inline tests.

NOTE

If your project collection uses the On-premises XML process model to customize work tracking, you can enable work item types that you add to the Task Category to appear as a checklist on your product Kanban board. To learn how, see Set up your backlogs and boards, Customize your Kanban Board checklist items.

1. Open the Settings dialog for the Kanban board you want to customize and choose **Annotations** .
2. Check those annotations that you want enabled. For example, to enable tasks but disable tests, check the following boxes.

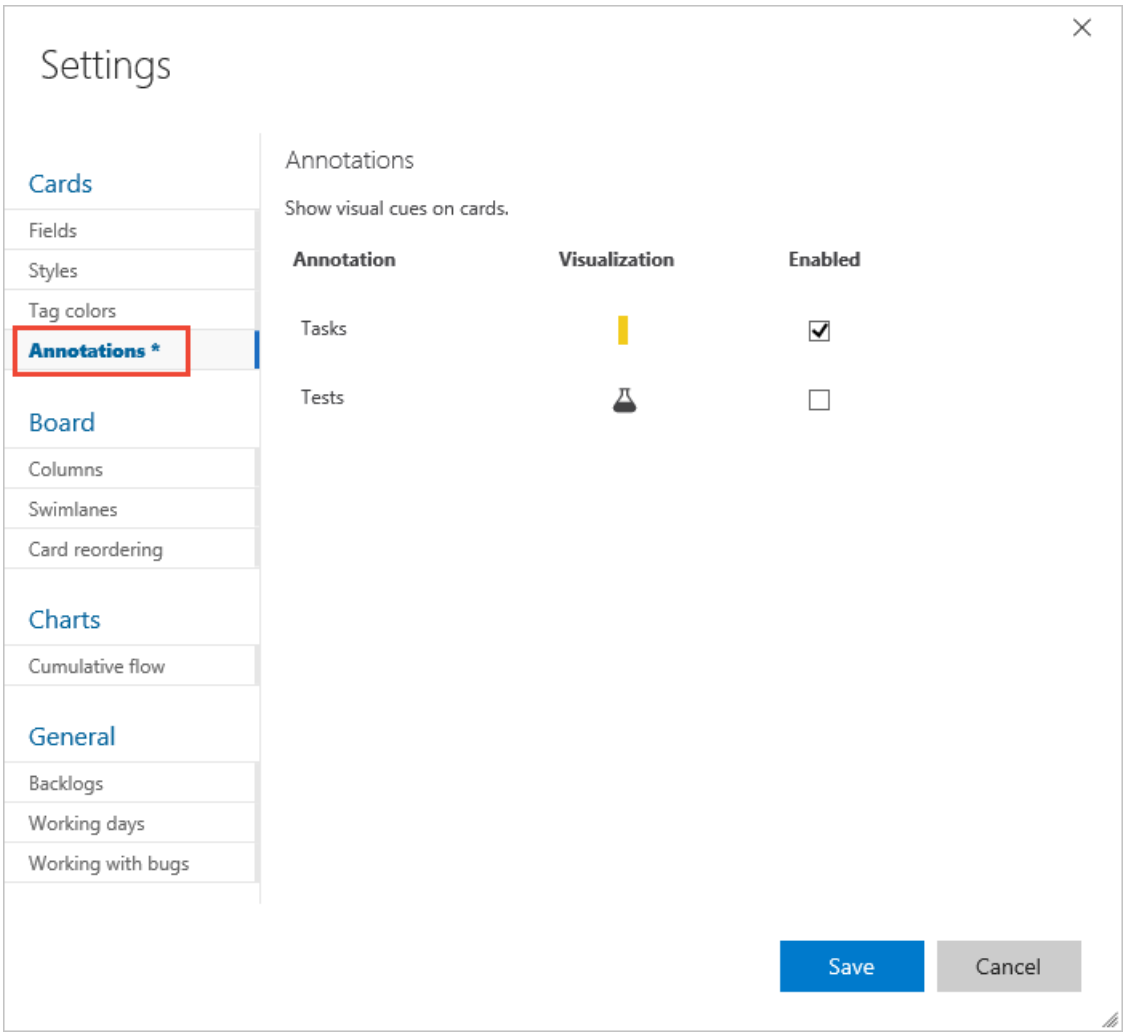


NOTE

GitHub annotations requires Azure DevOps Server 2019 Update 1 or later version. To learn more about linking Azure Boards work items to GitHub artifacts, see Link GitHub commits, pull requests, and issues to work items.

When done with your changes, choose **Save**.

1. Open the Settings dialog for the Kanban board you want to customize and choose **Annotations**
2. Check those annotations that you want enabled. For example, to enable tasks but disable tests, check the following boxes.

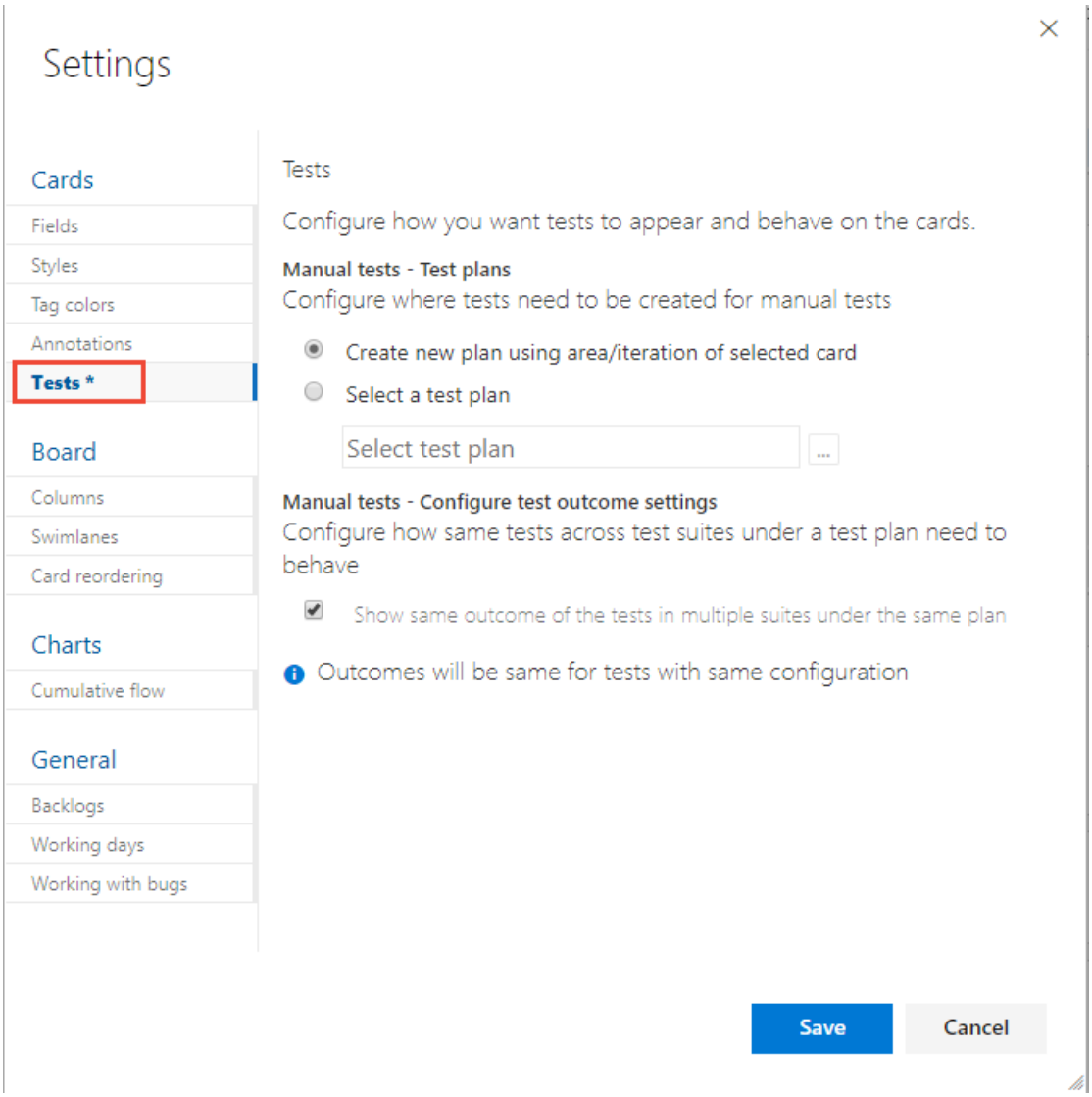


When done with your changes, choose **Save**.

Configure inline tests

You can control the test plan under which inline tests you add through the Kanban board are created. You can choose to create a new test plan for each new test that you add. Or, add all new tests to a selected test plan.

1. Open the Settings dialog for the Kanban board (product backlog only) you want to customize and choose **Annotations**. Make sure that **Test** annotation is enabled, a requirement to configure inline tests.
2. Then, choose **Tests**, and then choose the ··· actions icon and select one test plan from the query provided.

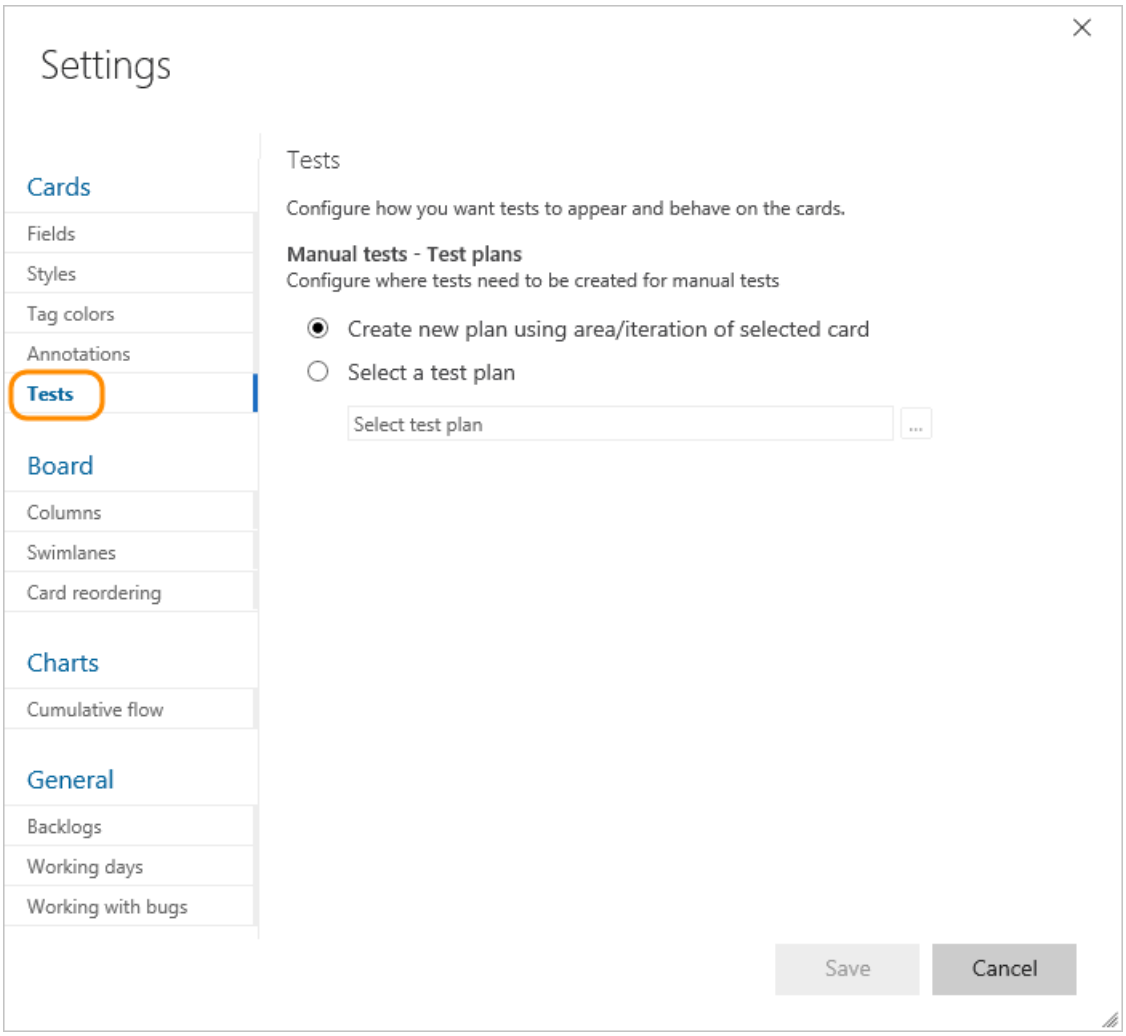


In a test plan, a test case can exist in multiple test suites. For example, you may define several test suites for each product feature and the test cases test scenarios across features. The test case might exist in both feature's test suite with the same configurations and test steps. Because of this setup, the tester might run the same test case multiple times for the same product version.

To avoid the redundancies that can occur under this scenario, you should choose **Show same outcome of the tests in multiple suites under the same plan** checkbox. When checked, the Test Points of the same Test Case and configuration, shows the same latest outcome. When the tester runs any one test, the output is synced with all other test points (which are of the same Test Case work item and same configuration) in the Test Plans. The tester can use the outcome and choose not to run the test again.

3. Save your changes.

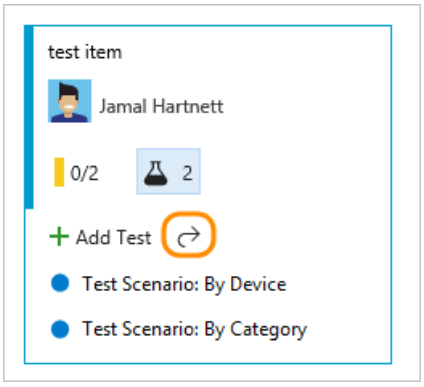
1. Open the Settings dialog for the Kanban board (product backlog only) you want to customize.
2. Choose **Annotations** and make sure that **Test** annotation is enabled. This is a requirement to configure inline tests.
3. Choose **Tests**, and then choose the options you want. To select a test plan, choose the ··· actions icon and select one test plan from the query provided.



4. Save your changes.

Open the test plan, test suite from a card

From a card on the Kanban board, you can easily navigate to the underlying test plan and test suite under which the tests are created. Choose the open icon to open another browser tab showing **Test** and associated test plan and test suite that controls the inline tests.



Related articles

- Card reordering
- Manage and configure team tools
- Setup your backlogs and boards
- Configure status badges
- Show bugs on backlogs and boards
- Select backlog navigation levels for your team

- Set working days

- Card reordering
- Manage and configure team tools
- Setup your backlogs and boards
- Show bugs on backlogs and boards
- Select backlog navigation levels for your team
- Set working days

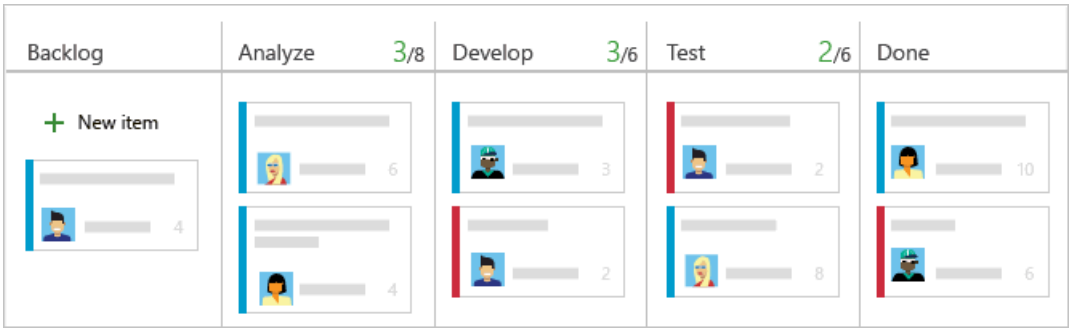
Set Work in Progress limits in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

An essential Kanban practice—Work in Progress limits, referred to as "WIP limits"—constrains the amount of work your team undertakes at each work stage. It's designed to focus your team on completing items before starting new work. While counter-intuitive at first, many teams find WIP limits helps them increase their productivity and improve their software quality.

You define WIP limits for each work stage, corresponding to each intermediate column. The limit sets a soft constraint on the number of items allowed within the column. Nothing actually prevents you from moving more items into the column and exceeding the limit. Your Kanban board shows the count of items at each stage next to each limit.



While setting WIP limits is simple, adhering to the limits takes a team commitment. Successful adoption of WIP limits involves a culture change. It moves teams from a focus on individual productivity to one of team productivity.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.

- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Determine initial WIP limits

To get started, have your team determine the initial WIP limits to set and how they'll use and monitor them. Beyond that, few rules apply to what numbers to set as they can vary based on several factors. Here are two guidelines to help you determine what limits to set:

- Set limits based on current works in progress. Count the items present in your existing Kanban columns.
- Set limits that don't exceed two or three items per team member that works within a stage. For example, if you have three team members and each team member can work on no more than two tasks at a time, the resulting WIP limit is 6 (= 3 developers X 2 tasks/developer).

Starting low may help your team discover bottlenecks more quickly and identify process issues to address.

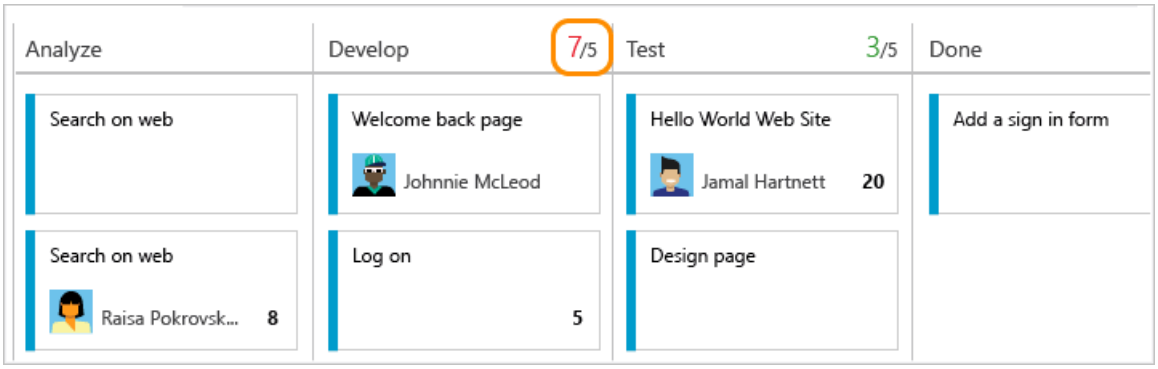
After you've defined an initial set of WIP limits, you'll likely want to fine tune them as your project progresses.

If you're new to Kanban, review Kanban basics to get an overview of how to access your board and implement Kanban.

Keep within WIP limits

After you've set your WIP limits, you'll want to track how well your team keeps within the limits.

Respecting WIP limits means teams don't pull items into a column if doing so causes the number of items in the column to exceed the column limit. When they do, your Kanban board provides immediate feedback. This feedback should act as a signal to the team to focus immediately on activities to reduce the number of items in the column.



Although simple in theory, keeping within WIP limits can force individuals, teams, and organizations out of their comfort zone. Team members who like to multitask might feel constrained. Others might find themselves without work as they wait for work to complete at an upstream stage.

To gain the advantages of constraining work-in-progress, have your team meet frequently to discuss the process changes taking place. As a starting point, consider hosting discussions around some of the challenges and solutions to support successful implementation of WIP limits provided below.

Identify bottlenecks

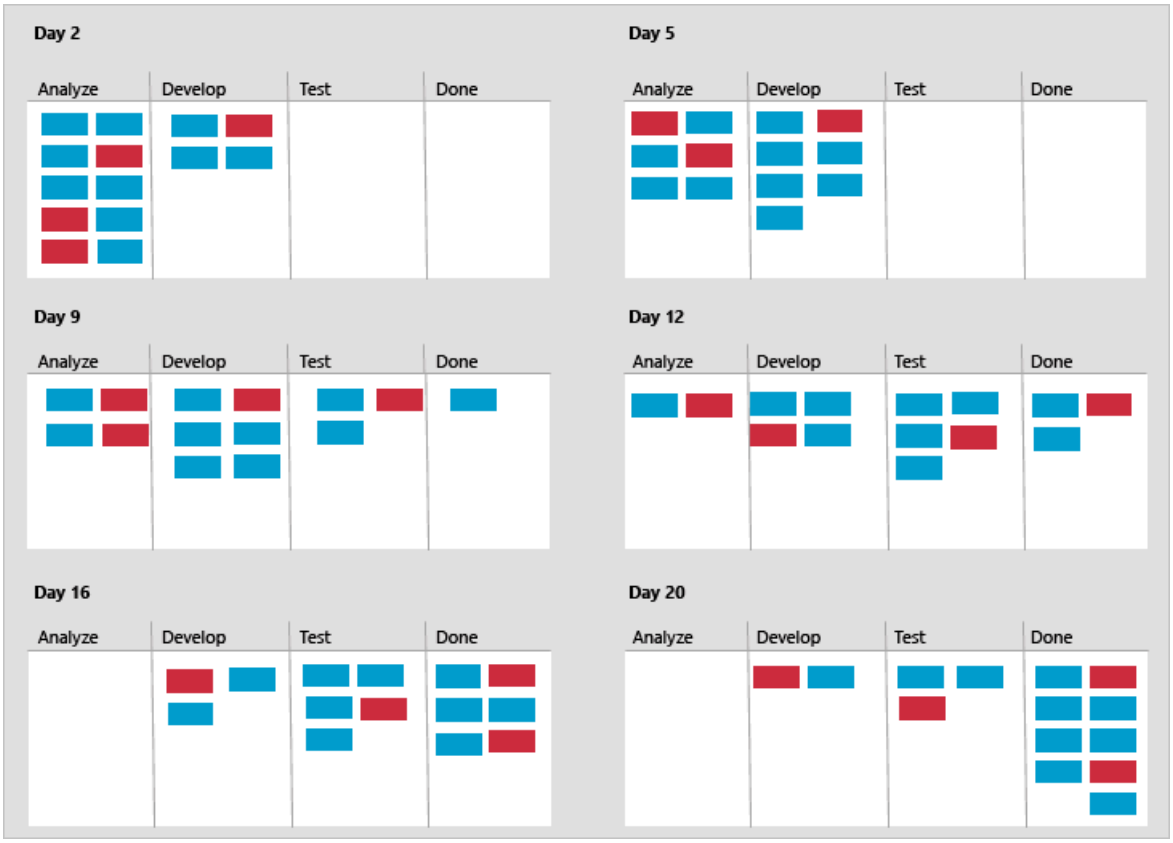
To optimize the flow of value, you naturally want to identify and eliminate bottlenecks. Bottlenecks indicate waste exists in the overall workflow process.

By monitoring your Kanban board over time, you can learn where bottlenecks occur. When several items sit in a column unworked for several days, a bottleneck has occurred. Bottlenecks typically occur when WIP limits are too high. However, no bottlenecks could indicate that WIP limits are too low.

The free eBook, Kanban and Scrum - making the most of both, provides this guidance:

Too low WIP limit => idle people => bad productivity Too high WIP limit => idle tasks => bad lead time

Taking periodic snapshots of your Kanban board can visually catalog where work flows smoothly and where bottlenecks appear.



Such snapshots can show your team:

- How many items on average exist within a workflow stage/column
- How many items are being worked versus team members who work within a workflow stage/column
- How many and which items remained in a workflow stage/column for long periods of time
- How many items did the team complete at the end of a one, two, or three week period?

Eliminate waste

Because bottlenecks signal waste in your workflow process, you'll want to identify the source of the waste. Kanban defines waste as anything not strictly needed to produce desired outcomes.

Common wastes in software development include:

- Unused code or features
- Defects that lead to rework
- Delays or time spent waiting for something
- Handoffs from one person, team, or business process to another
- Insufficient requirements
- Slow or poor communication

Eliminating waste calls for team discussions to identify causes and solutions acceptable to the team. Along with addressing the challenges and solutions posed by WIP limits, the team may decide to adjust their workflow process or WIP limits.

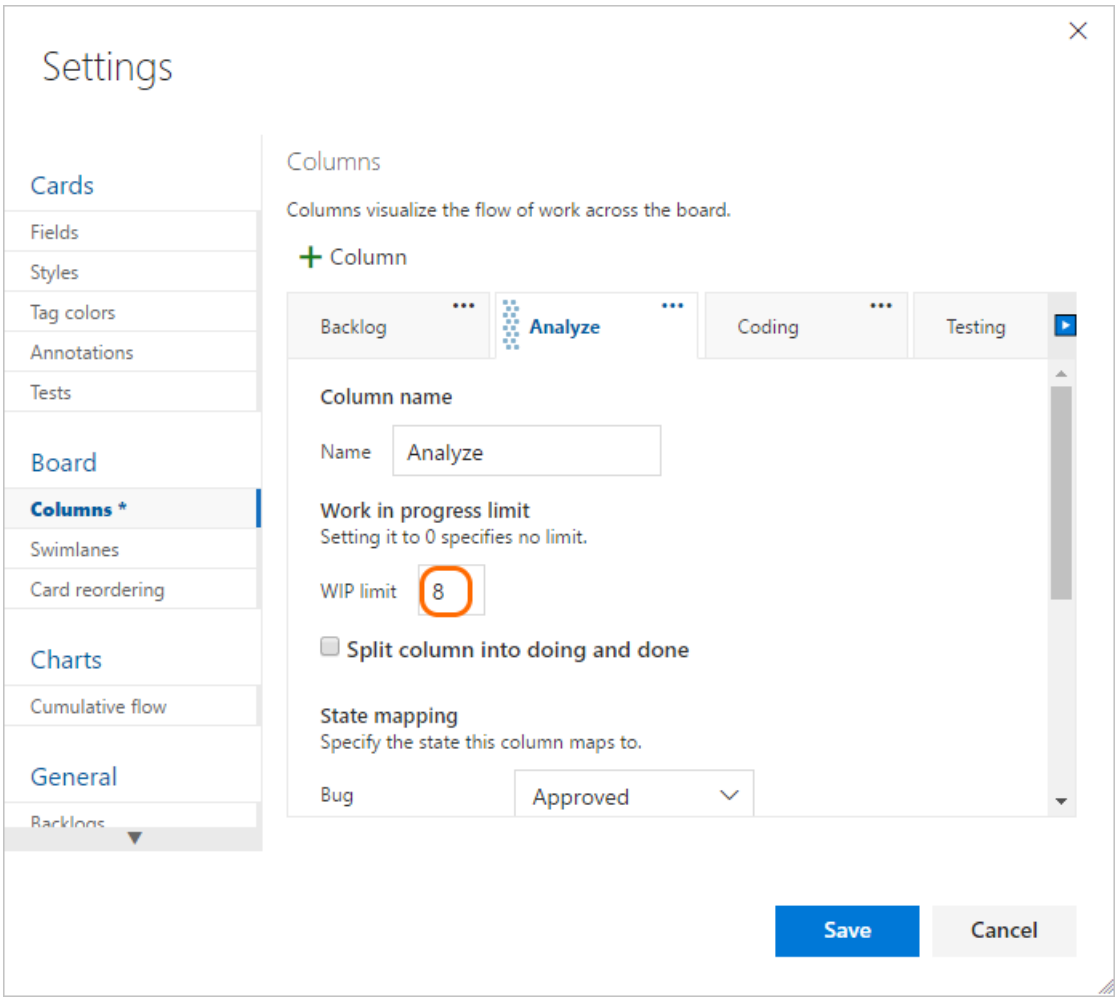
Set WIP limits

With an understanding of how you'll use WIP limits, here's how you set them. If you haven't yet mapped your team's work flow to Kanban columns, do that first. For information about accessing your Kanban board, see Kanban basics.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose the gear icon to configure the board and set general team settings.



3. Choose **Columns** and then a column tab to set the WIP limit for that column.



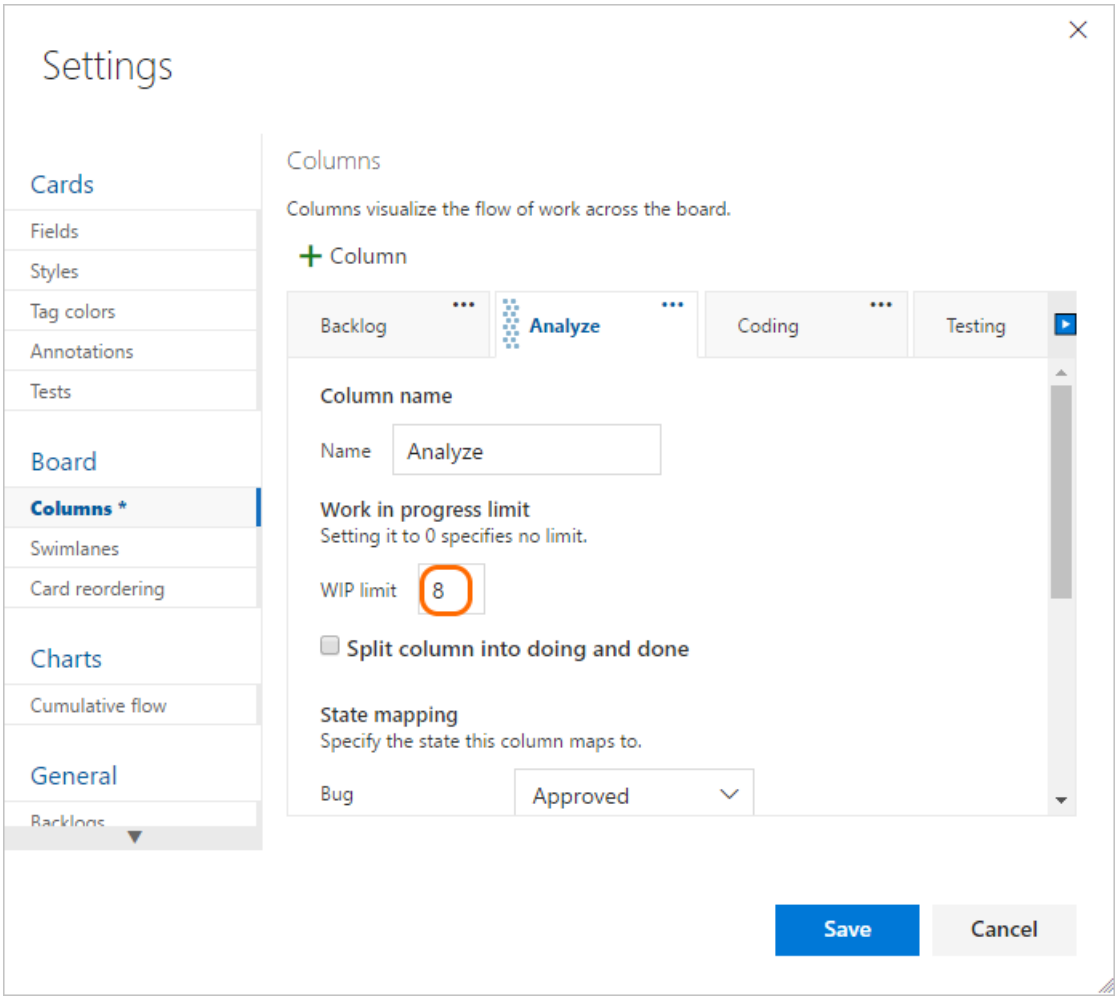
> **NOTE**
> You'll see different column titles and choices based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.

4. When done with your changes, choose **Save**.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose to open the common configuration settings dialog for the Kanban board.



3. Choose **Columns** and then a column tab to set the WIP limit for that column.



> **NOTE**
> You'll see different column titles and choices based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.

4. When done with your changes, choose **Save**.

Related articles

- Split columns
- Speed up work
- Definition of Done
- Customize cards
- Show bugs on backlogs and boards

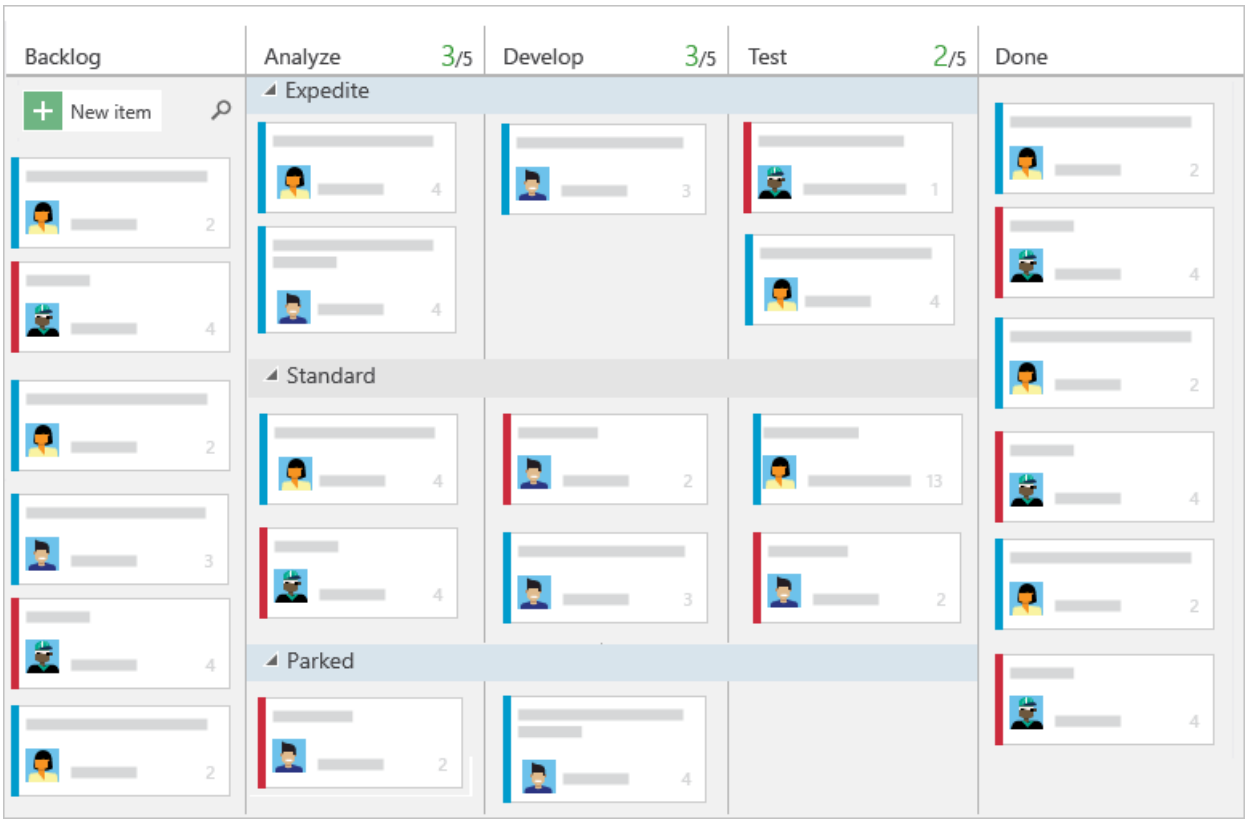
Speed up your team's work by using swimlanes in your Kanban board

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Your Kanban board helps you visualize the flow of work as it moves from defined to completed. When you add swimlanes, you can also visualize the status of work that supports different service-level classes. You can create a swimlane to represent any other dimension that supports your tracking needs.

For example, you can create three swimlanes—Expedite, Standard, and Parked—to track high-priority work, standard work, and work that's currently blocked.



TIP

Type **o** to expand all swimlanes and **u** to collapse all swimlanes. To move the focus up or down, enter the ↑↓ up/down arrows. For more tips, see Keyboard shortcuts.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit

work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Types of swimlanes

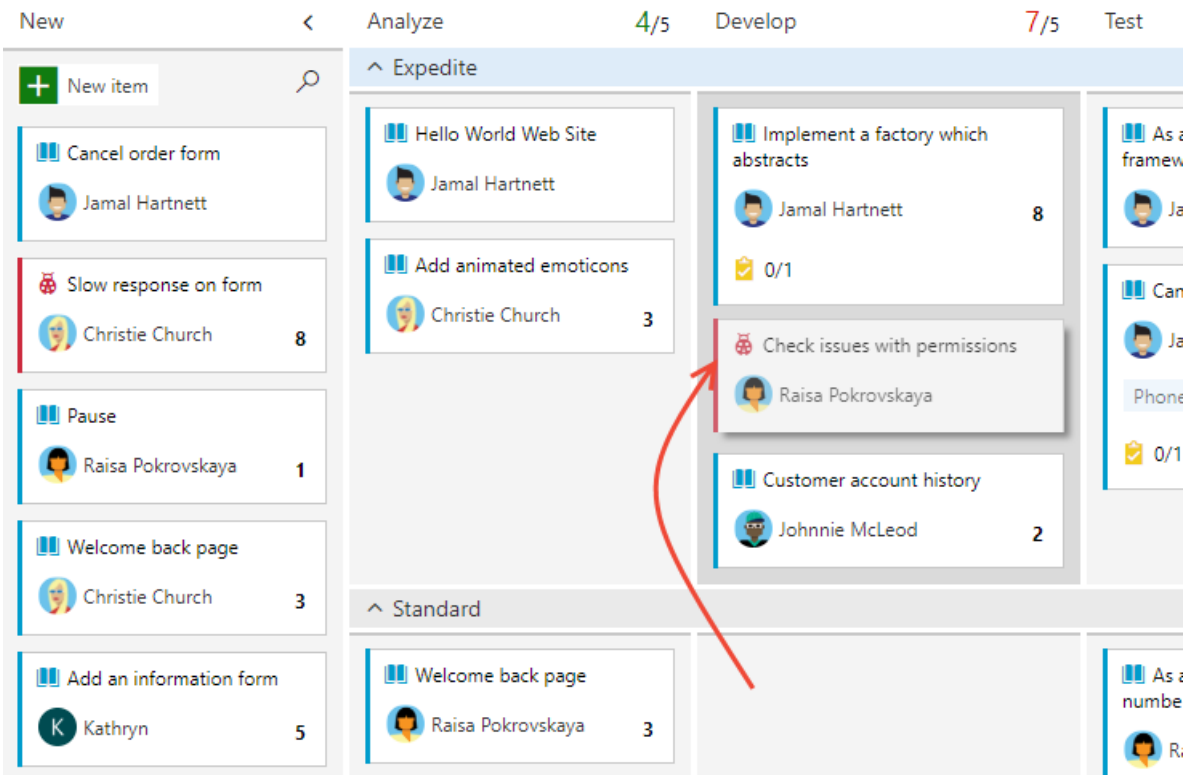
You can use swimlanes to sort work on your Kanban board to track items that you differentiate as follows:

- High priority items
- Service-level class
- Date-driven requirement
- Dependency for or from another team
- Blocked items
- Technical debt or other engineering work that's not a specific user story

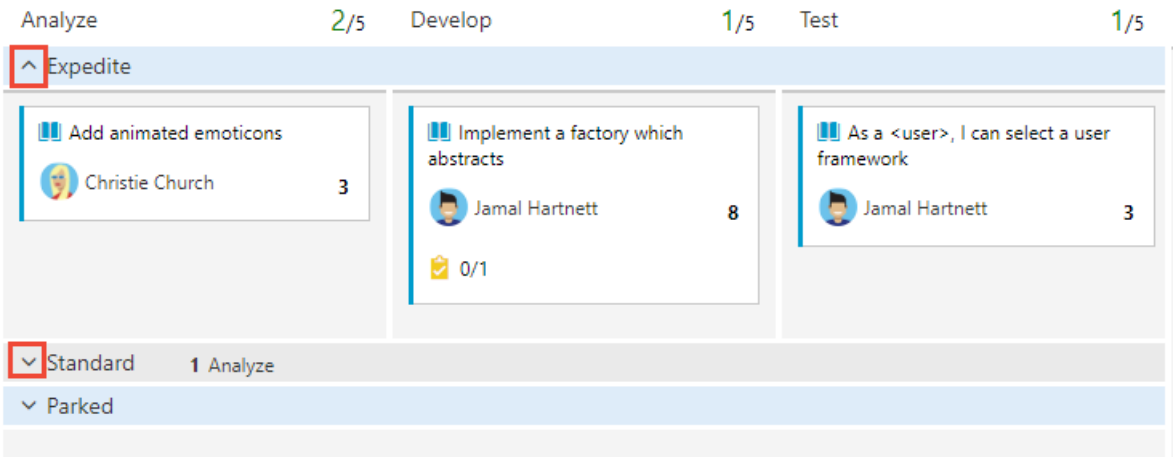
Track work in swimlanes

Once you've set up your swimlanes, you can drag items into a swimlane and also reorder them within the lane.

> **TIP**
> When you have many swimlanes or cards on your board, you may encounter slow performance when dragging a card. We recommend that you use swimlanes in conjunction with card styles, tags, and board filters to manage your work items. If you have a lot of cards in the default lane, place that lane lower on the board to enhance performance when dragging a card to another swimlane.



You can also focus on a single swimlane by collapsing all other lanes.



IMPORTANT

Work items that appear on more than one team's Kanban board can yield results that don't meet your expectations because each team can customize its Kanban board columns and swimlanes. The values assigned to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields might differ from what you expect when another team updates the work item from a different board. To learn more, see Add, review, and update work items in Azure Boards.

Add or remove a swimlane

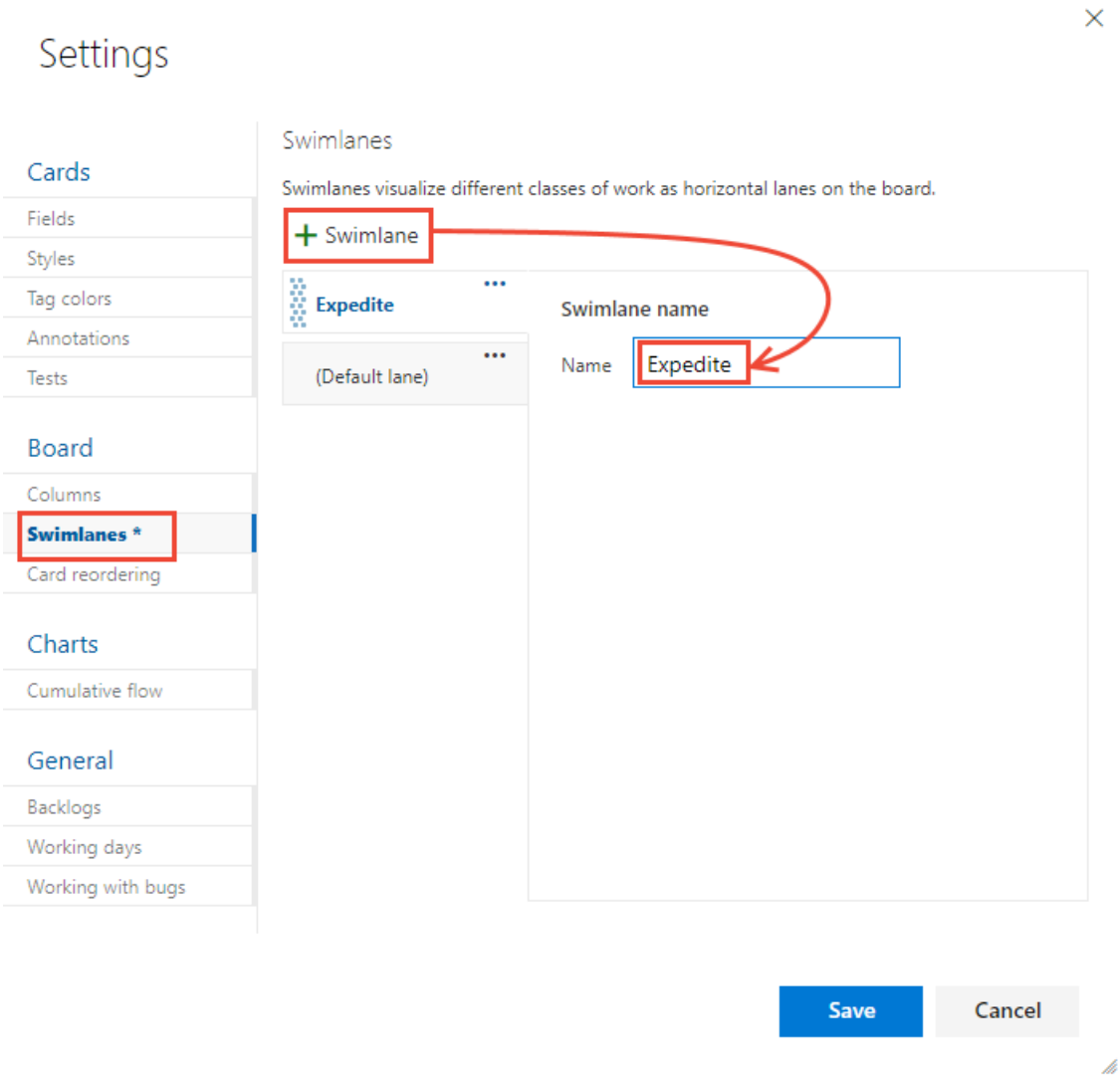
So, what swimlanes will support your tracking needs?

Once you've identified one or two, add them to your Kanban board.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose the gear icon to configure the board and set general team settings.

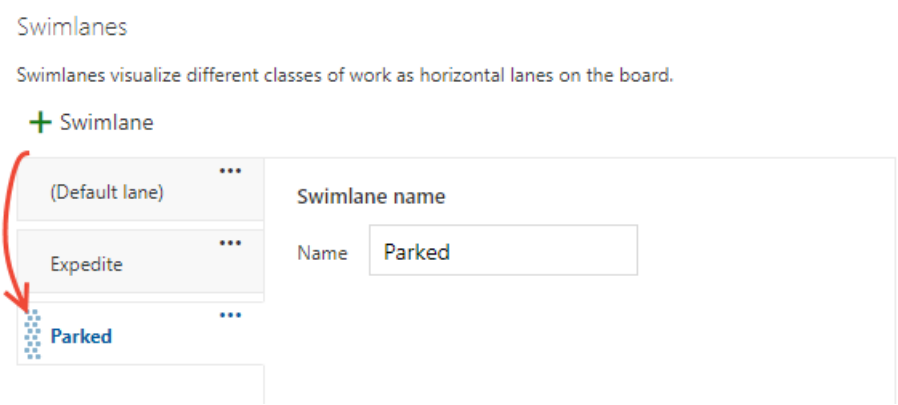


3. Choose **Swimlanes** and then choose the + plus icon and enter the name of the swimlane you want to add.



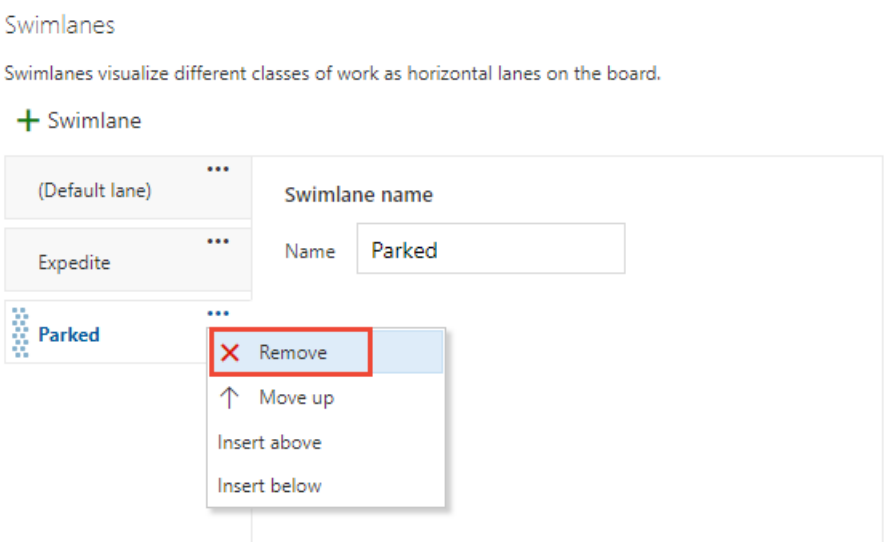
The default lane appears unlabeled on the Kanban board. You can rename it to anything you like, however, you can't delete it. Also, you can rename it directly from the Kanban board.

4. To reorder your swimlanes, grab the lane and move it up or down.



5. If you need to delete a swimlane, first move all items out of the lane. Then open the Settings dialog,

choose the ··· actions icon and select **Remove**.

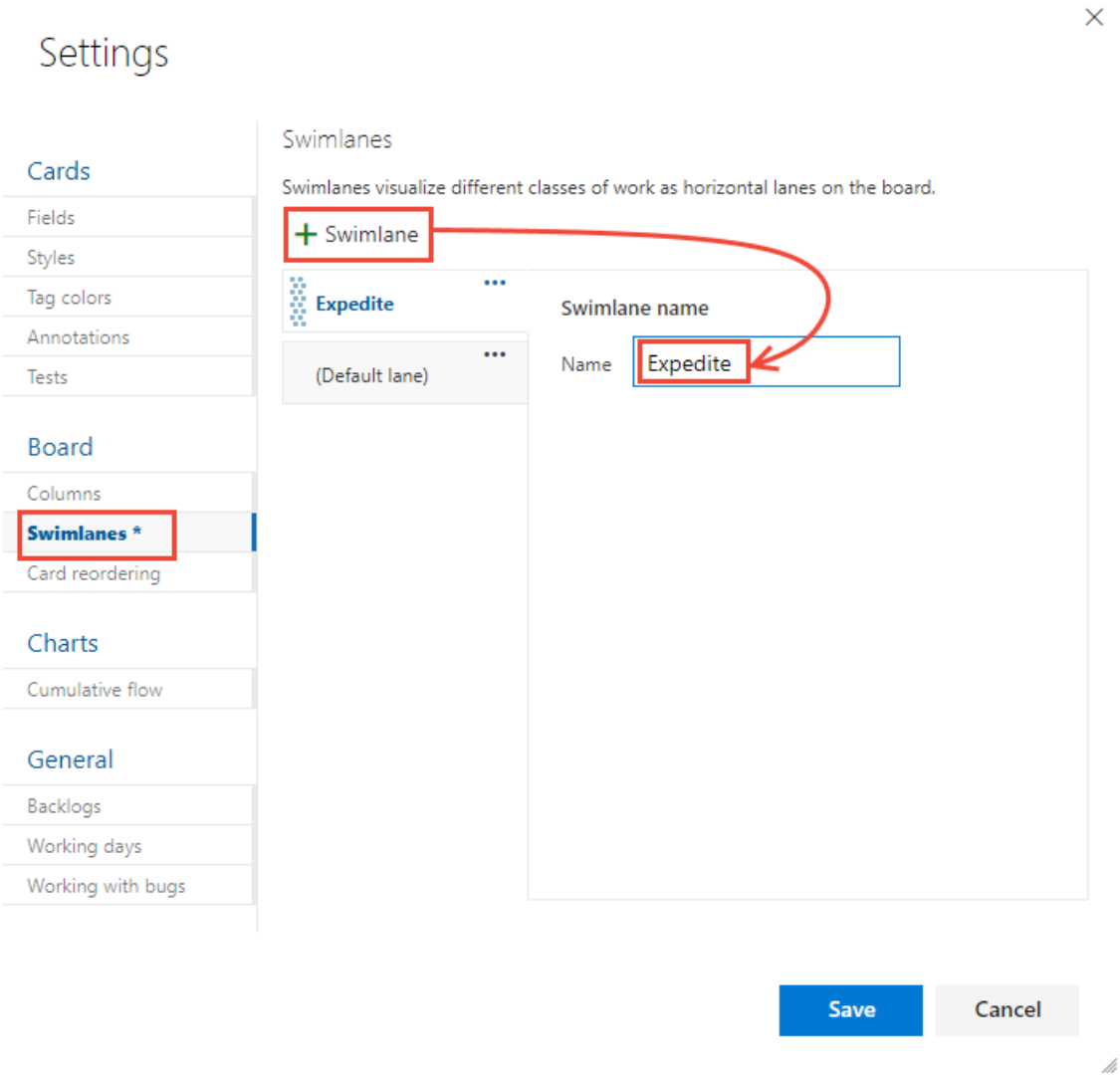


6. When done with your changes, choose **Save**.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose the ⚙ gear icon to open the common configuration settings dialog for the Kanban board.

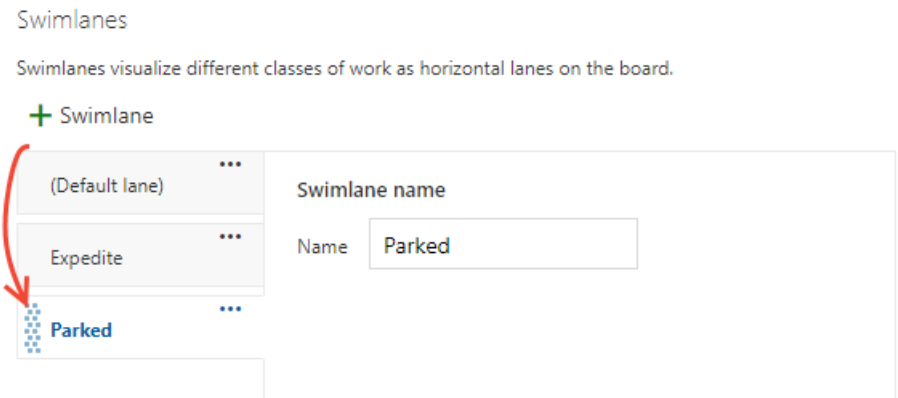


3. Choose **Swimlanes** and then choose the + plus icon and enter the name of the swimlane you want to add.

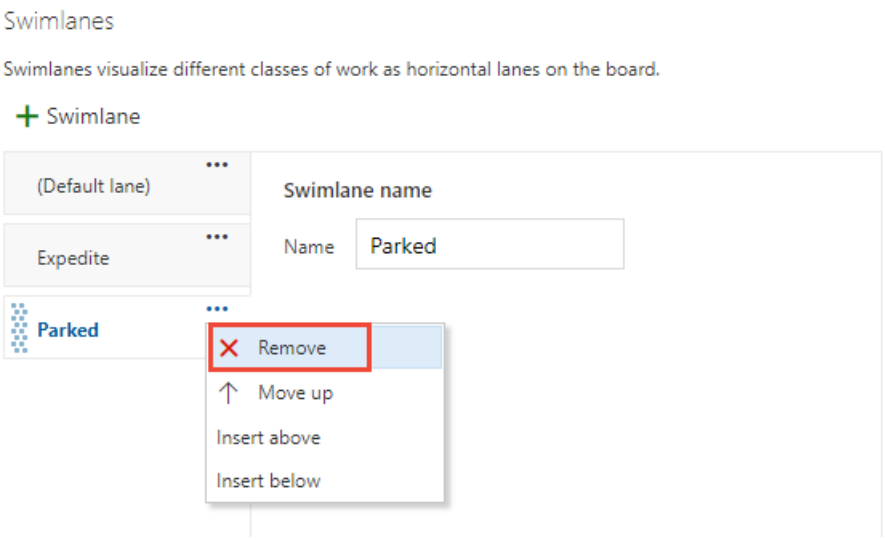


The default lane appears unlabeled on the Kanban board. You can rename it to anything you like, however, you can't delete it. Also, you can rename it directly from the Kanban board.

4. To reorder your swimlanes, grab the lane and move it up or down.



5. If you need to delete a swimlane, first move all items out of the lane. Then open the Settings dialog, choose the ··· actions icon and select **Remove**.



6. When done with your changes, choose **Save**.

Query for work items based on swimlane

You can track which work items have been added to a Kanban board swimlane by creating a query and using the Board Lane field.

Related articles

As you can see, swimlanes provides another way to organize and visualize the flow of work using Kanban. Here are a few more options you have for customizing the look and feel of your Kanban board.

- About teams and Agile tools
- Query by assignment or workflow changes
- Add columns
- Split columns
- Customize cards

REST API resources

To programmatically interact with the Kanban board and other team settings, see the REST API, Boards reference.

Split columns on your Kanban board to show work in progress

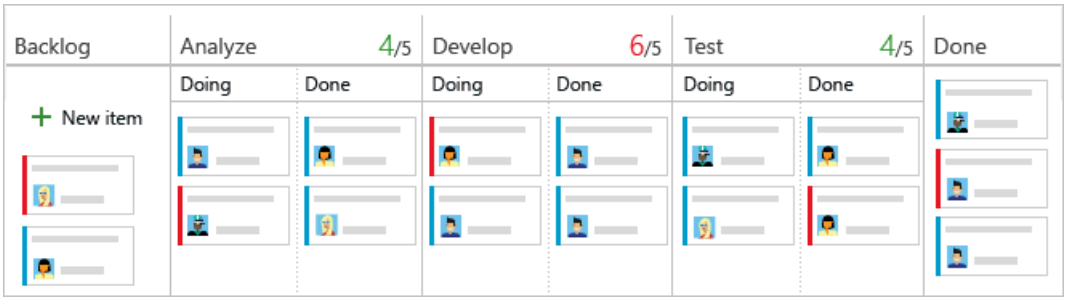
10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You use your Kanban board to visualize the flow of work, and monitor how items are or aren't progressing. Because each column corresponds to a stage of work, you can quickly see the number of items in progress at each stage.

However, a lag often exists between when work gets moved into a column and when work actually starts. To counter that lag and reveal the actual state of work in progress, you can turn on split columns.

When split, each column contains two subcolumns, **Doing** and **Done**.



Split columns let your team implement a pull mechanism within the workflow process. Without split columns, teams push work forward, to signal that they've completed their stage of work. However, pushing it to the next stage doesn't necessarily mean that a team member immediately starts work on that item.

By contrast, with split columns, your team knows exactly how many items sit idle, waiting for work to begin. You now have greater visibility into the quantity of items that sit idle at each stage throughout your workflow process.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

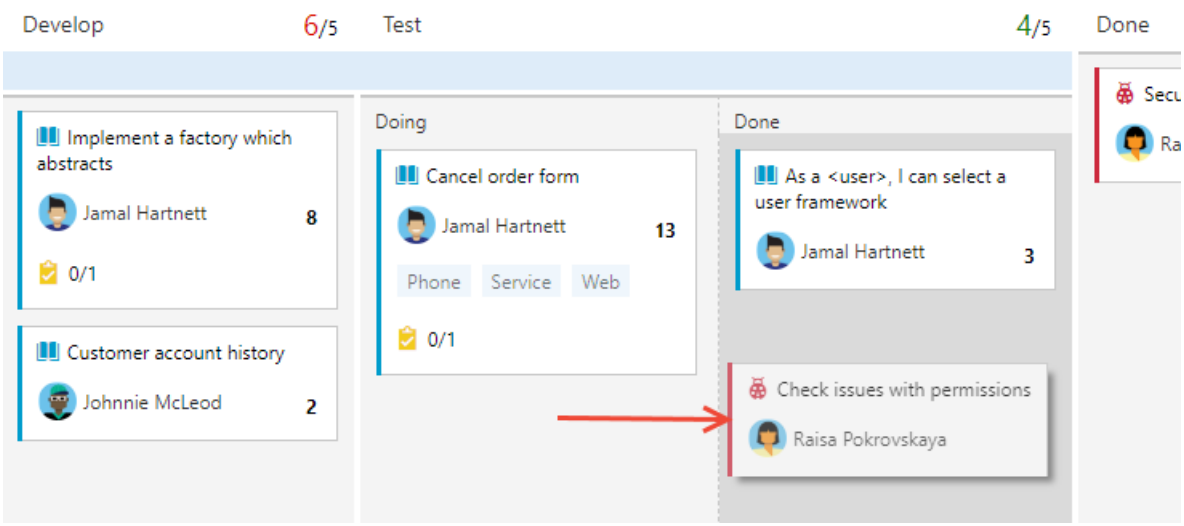
- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit

work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Move items into the Doing and Done columns

With split columns turned on, you update status of items on the Kanban board in the same way you have before. However, now when you've completed work on an item, you move it into Done, instead of a downstream column. When the next team member becomes free to work on the next high priority item, they pull it into Doing and reassigns it to themself.

For example, as a team member completes their coding task, they move the item into Done under the Develop column. When the tester is ready to test the item, they pull it into Doing under the Test column.

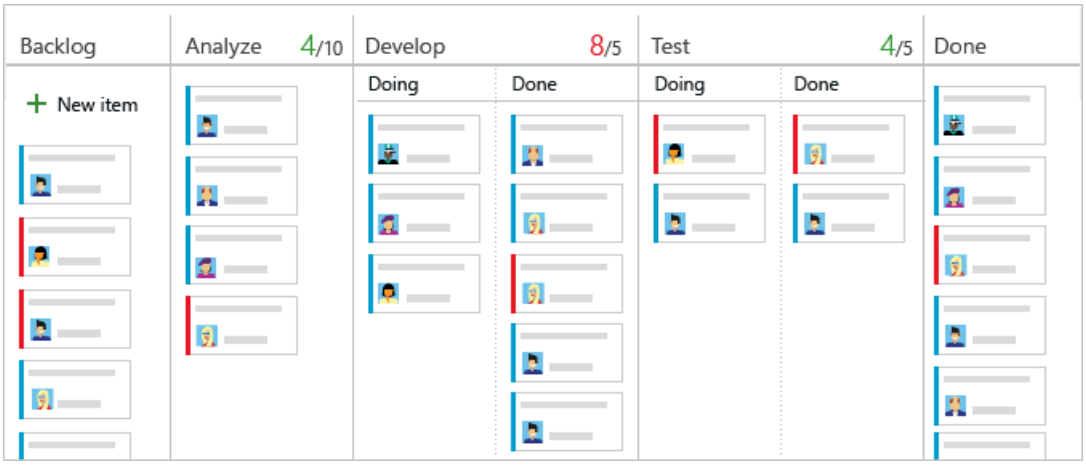


If you're new to Kanban, review Kanban basics to get an overview of how to access your board and implement Kanban.

Identify bottlenecks to drive toward a perfect flow scenario

How can you use split columns to improve workflow?

Split columns provide you even greater insight into how many items sit idle in a Done column. Your team can readily see when items pile up, which signal a potential bottleneck.



By reviewing the frequency of pile ups and where they occur, your team can adjust their processes to eliminate the bottlenecks. Workflow processes that incur no or few bottlenecks correspond to perfect flows. No item sits in a queue for any

Choose which columns you want to split

Now that you understand how your team can use split columns, here's how to turn them on. Before you split columns, you'll want to have mapped each stage of your team's process to a Kanban column.

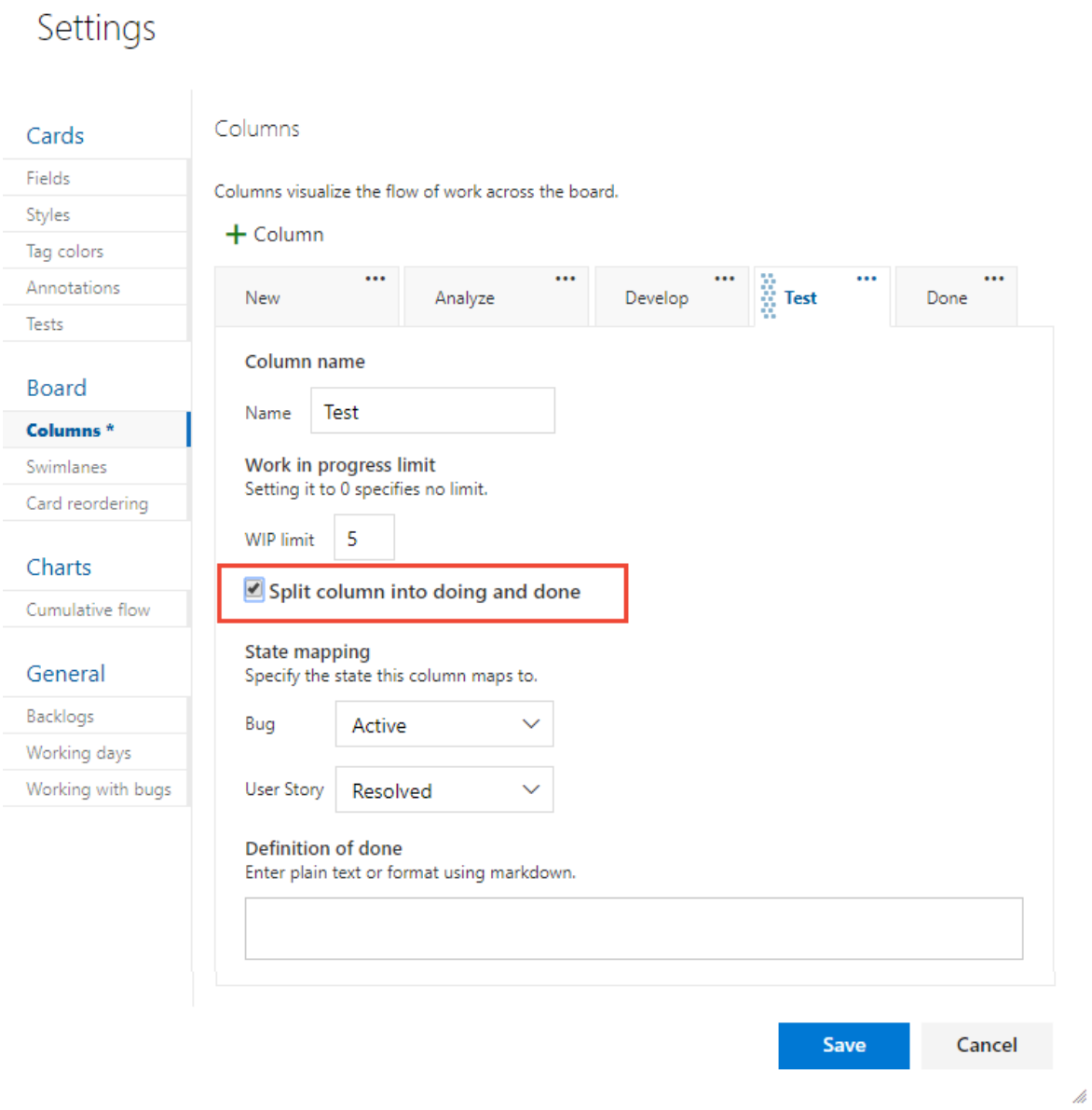
Only split columns where clear hand-offs exist and you want teams to pull the item into the next stage.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose the gear icon to configure the board and set general team settings.



3. Choose **Columns** and then choose the column tab that you want to split. Place a check in the checkbox to cause the column to split.

> **NOTE**
> You'll see different column titles and choices based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.



4. When done with your changes, choose **Save**.

> **TIP**
> You can filter queries and create charts using the Board Column Done field.

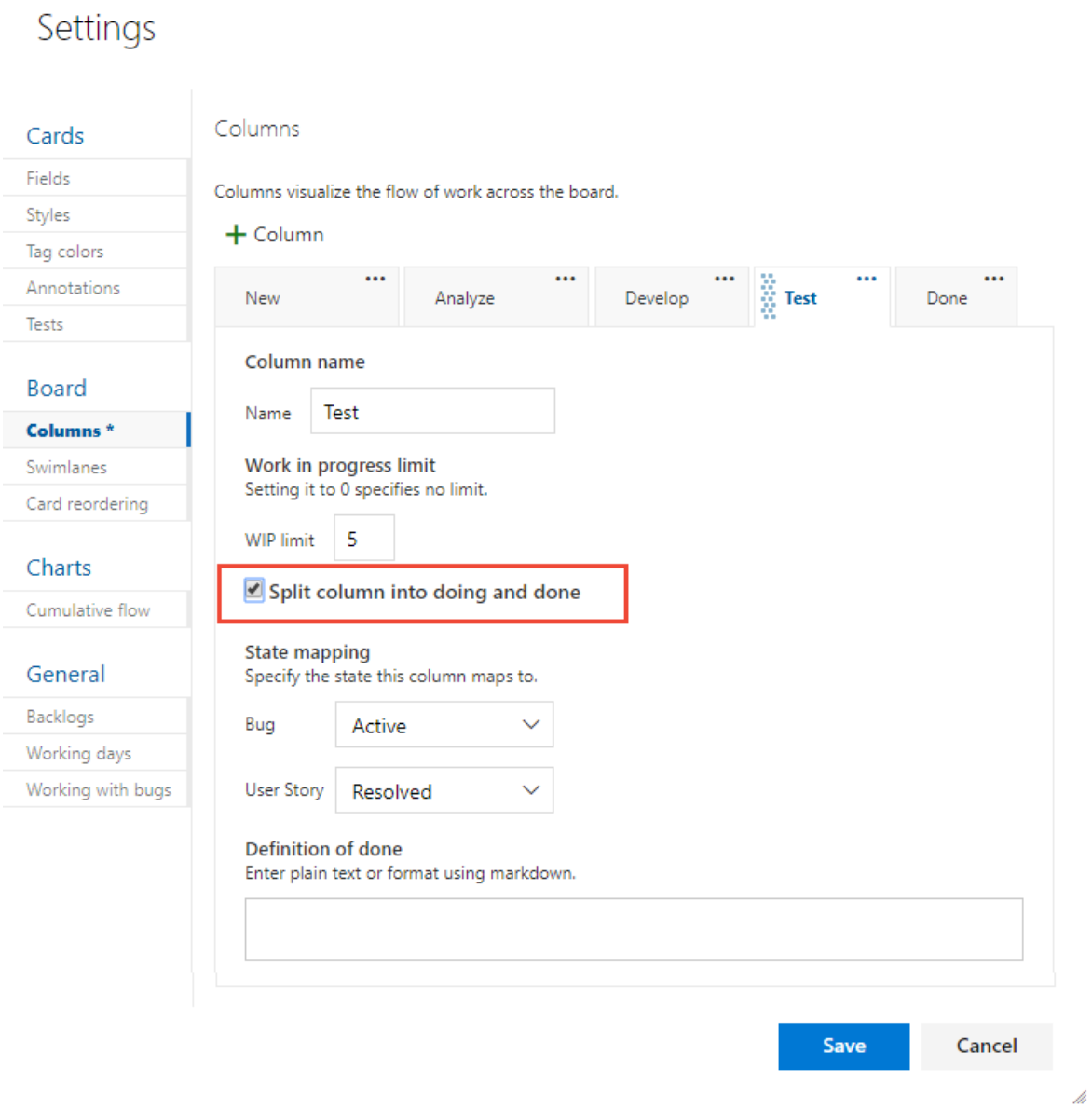
1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.

2. Choose gear icon to open the common configuration settings dialog for the Kanban board.



3. Choose **Columns** and then choose the column tab that you want to split. Place a check in the checkbox to cause the column to split.

> **NOTE**
> You'll see different column titles and choices based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.



4. When done with your changes, choose **Save**.

> **TIP**
> You can filter queries and create charts using the Board Column Done field.

List work items in a Doing or Done column

You can query for work items in a split column using the **Board Column Done** field. This field takes of a value of False when in the Doing column and True when in the Done column.

For examples on querying Board columns, see Query by assignment or workflow changes

Related articles

- Add columns

- Query by assignment or workflow changes
- Work in Progress limits
- Add swimlanes, expedite work
- Definition of Done
- Customize cards

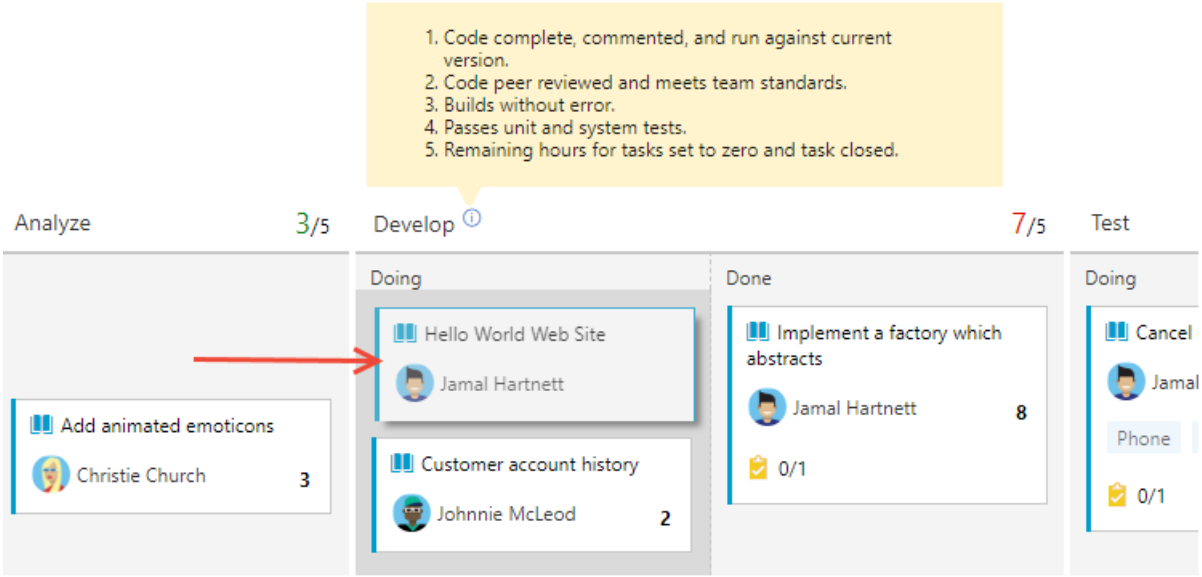
Specify the Definition of Done criteria for your Kanban columns

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As your team updates the status of work as it progresses from one stage to the next, it helps that they agree on what "done" means. Help your team agree what "done" means by specifying the Definition of Done criteria for each Kanban column. Defining what "done" means helps you identify the essential tasks to complete before moving an item into a downstream stage. Also, you'll have implemented one of the core Kanban tenets: **make processes and policies explicit.**

When set, team members can quickly double-check the done criteria.



If you're just getting started, review Kanban basics to get an overview of how to implement Kanban.

Prerequisites

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.
- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board and can't update fields displayed on cards. For details, see About access levels.

- You must have a Kanban board you want to configure. When you add a team, you add a Kanban board for that team. To learn more, see About teams and Agile tools.
- To configure team settings, you must be added to the team administrator role or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.

- Users assigned **Basic** access or higher can exercise all backlog and board features.
- Users assigned **Stakeholder** access have limited access to backlog and board features. Stakeholders can edit work items on the board and add existing tags to a work item. They can't add work items to a board, can't drag-and-drop work items to update status or reorder cards, and can't update fields displayed on cards. For details, see About access levels.

Add the Definition of Done to a column

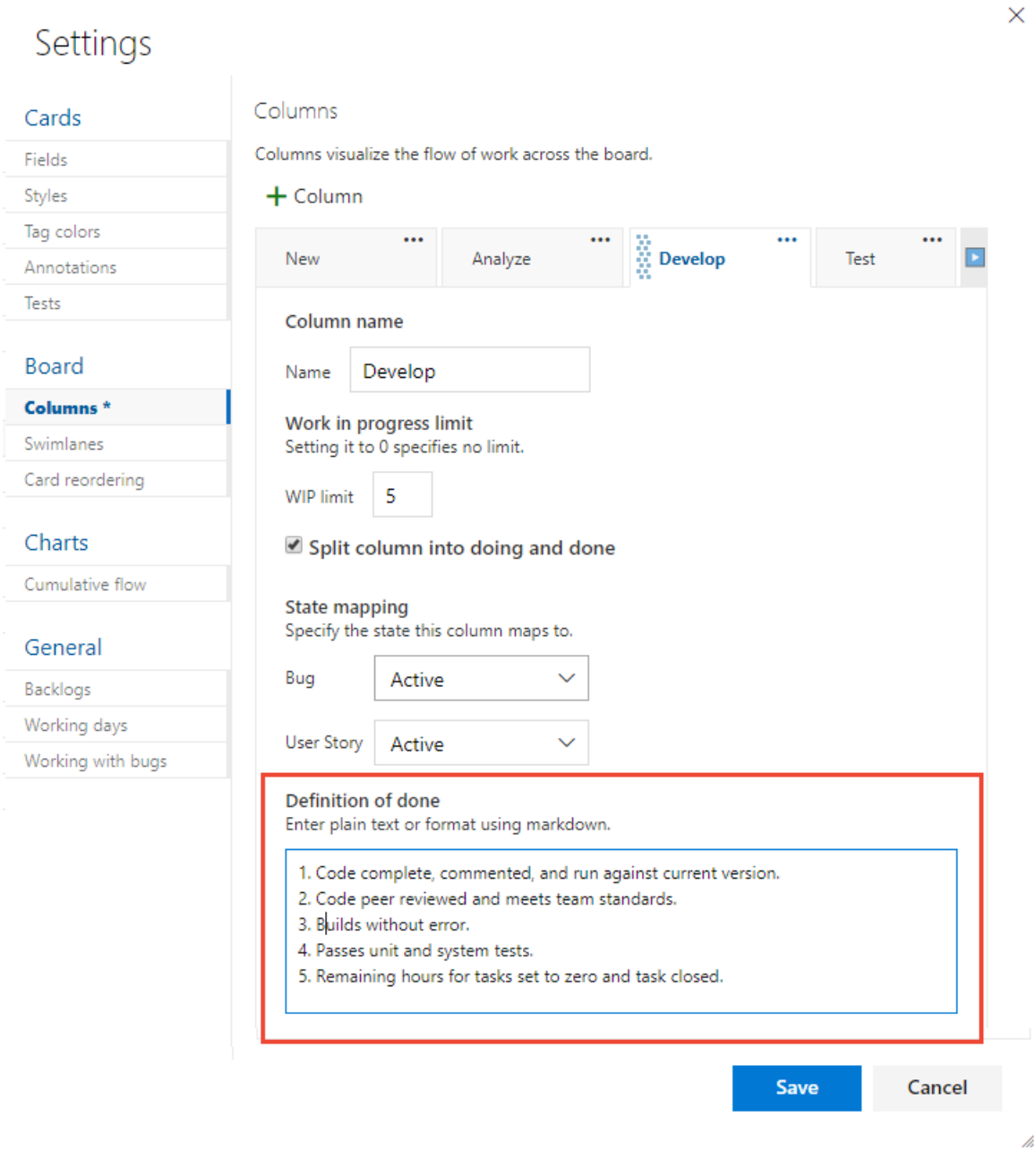
1. Open your Kanban board.

   If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.

2. Choose the gear icon to configure the board and set general team settings.



3. Choose **Columns** and then a column tab to configure the Definition of Done for that column.

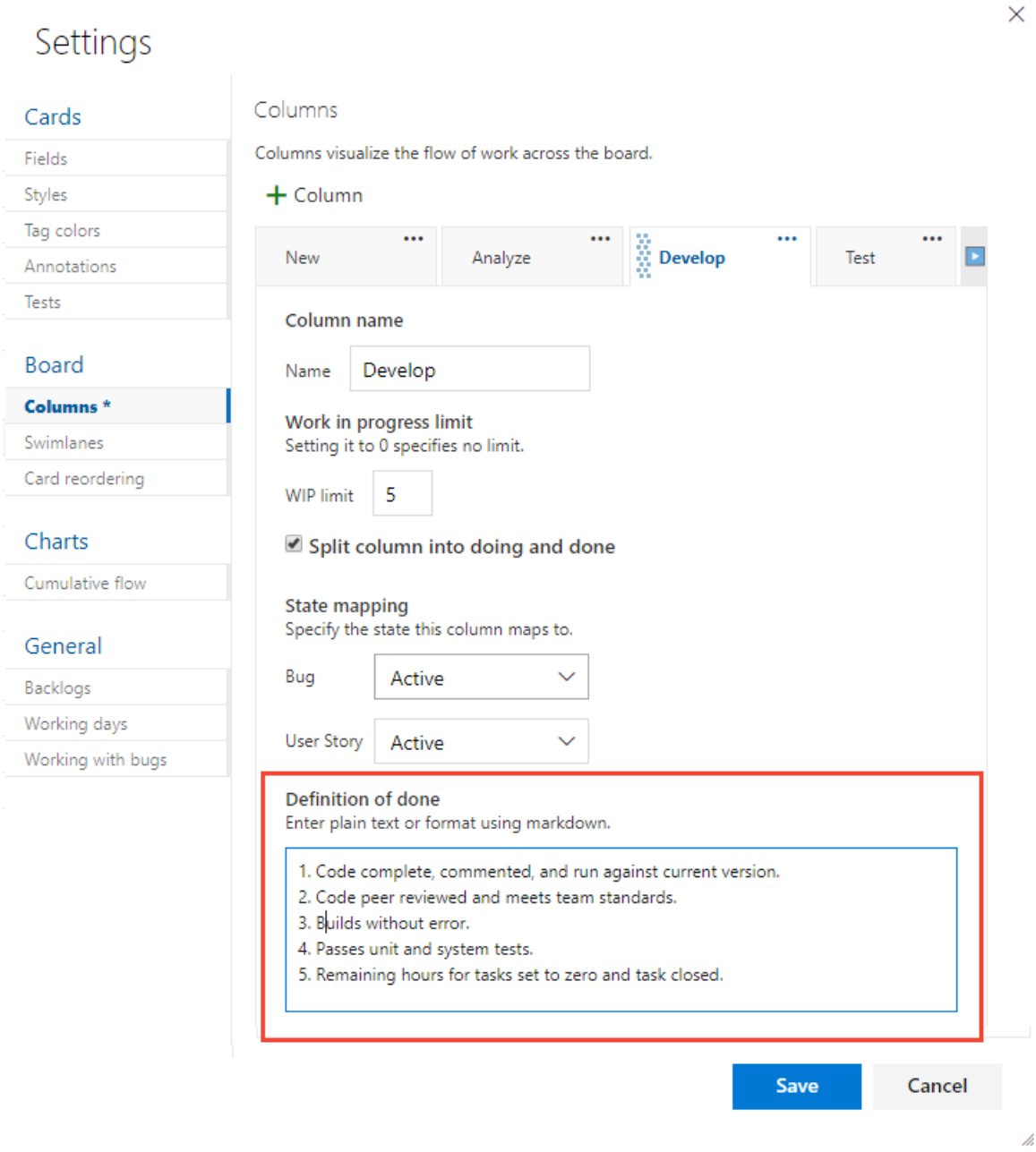


4. When done with your changes, choose **Save**.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose ⚙ to open the common configuration settings dialog for the Kanban board.



3. Choose **Columns** and then a column tab to configure the Definition of Done for that column. You can specify the Definition of Done for each intermediate column on your team's Kanban board.



4. When done with your changes, choose **Save**.

Team members can quickly check that they've met the criteria by choosing the Information tooltip ⓘ info icon.

Related articles

- Add, rename, move, and delete columns
- Work in Progress limits
- Add swimlanes, speed up work
- Split columns
- Customize cards

Show bugs on backlogs and boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As your team identifies code defects or bugs, they can add them to the backlog and track them similar to tracking requirements. Or, they can schedule bugs to be fixed within a sprint along with other tasks. When you *track bugs as requirements*, they appear on the product Backlogs and Kanban Boards. When you *track bugs as tasks*, the bugs appear on the Sprint Backlogs and Taskboards. For more information about other work item types, see Add other work item types to backlogs or boards.

You can define this team setting for the Agile, Scrum, and CMMI processes. The Bug work item type isn't defined for the Basic process, so there isn't a team setting for Basic. Instead, you should track bugs and code defects using the Issue work item type.

> **NOTE**
> *Requirements* specify expectations of users for a software product. In Azure Boards, requirements are defined by work items that appear on your product backlog. They correspond to User Stories (Agile), Product Backlog Items (Scrum), Issues (Basic), or Requirements (CMMI) based on the process selected for your project. They also belong to the Requirements Category which manages which work item types appear on the product backlog.

Prerequisites

- To configure team settings, you must be added as a team administrator or be a member of the **Project Administrators** group. See Change project-level permissions.

Options your team has to track bugs

The following table summarizes the three options teams have to track bugs. Before making your choice, we recommend you review the information provided in Define, capture, triage, and manage bugs. That article provides an overview of the Bug work item type and supported tools for managing bugs.

Option	Choose when you want to...
Track bugs as Requirements	• Prioritize (stack rank) bugs along with requirements • Estimate Bug effort for forecasting • Update bug status on Kanban board • Include Bugs in Velocity charts and Cumulative Flow Diagrams • Can use the **Forecast** tool to support sprint planning • Can drag and drop bugs onto **Planning** pane to assign bugs to a sprint • Can view Bugs on Delivery Plans

NOTE

- Bugs are assigned to the Requirements Category

Track bugs as Tasks

- Estimate work for bugs similar to tasks
- Update bug status on sprint Taskboards
- Link bugs to requirements as child items
- Can drag and drop bugs onto Planning pane to assign bugs to a sprint

NOTE

- Bugs are assigned to the Task Category
- User Stories (Agile), Product Backlog Items (Scrum), or Requirements (CMMI) are the natural parent work item type for Bugs
- Bugs won't be visible on Delivery Plans

Bugs don't appear on backlogs or boards

- Manage bugs using queries

NOTE

- Bugs are associated with the Bugs Category and won't appear on either backlogs or boards
- Bugs won't be visible on Backlogs, Boards, Sprint Backlogs, Taskboards, or Delivery Plans
- Can't drag and drop bugs onto Planning pane to assign bugs to a sprint

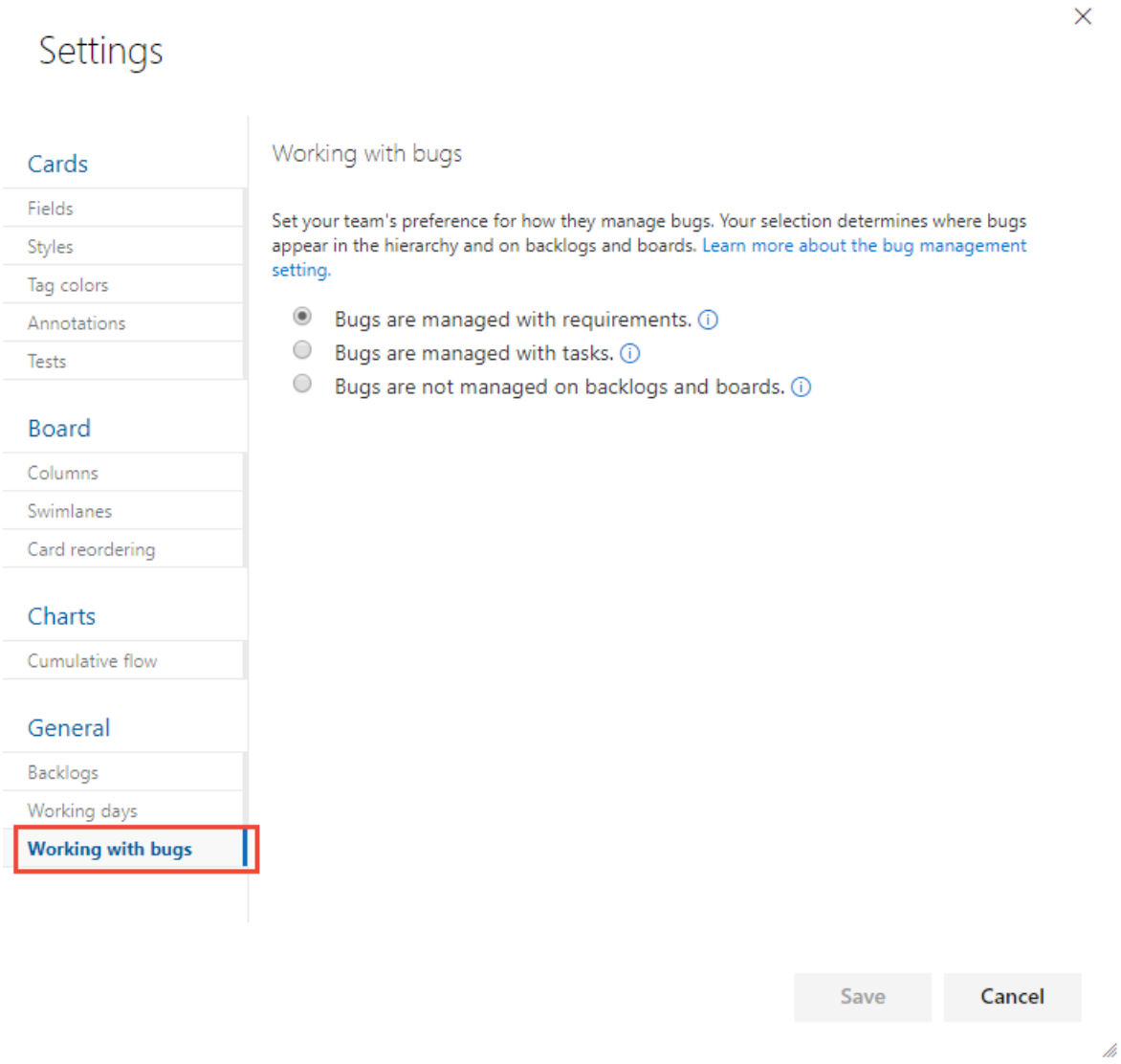
Set your team's preference for bug tracking

You can change the setting from a backlog or board view, or from **Project Settings>Team Settings**. Here we show how to change it from the board view.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose **Board settings** to configure the board and set general team settings.



3. Choose **Working with bugs** and then choose the option that best meets your team's way of working.

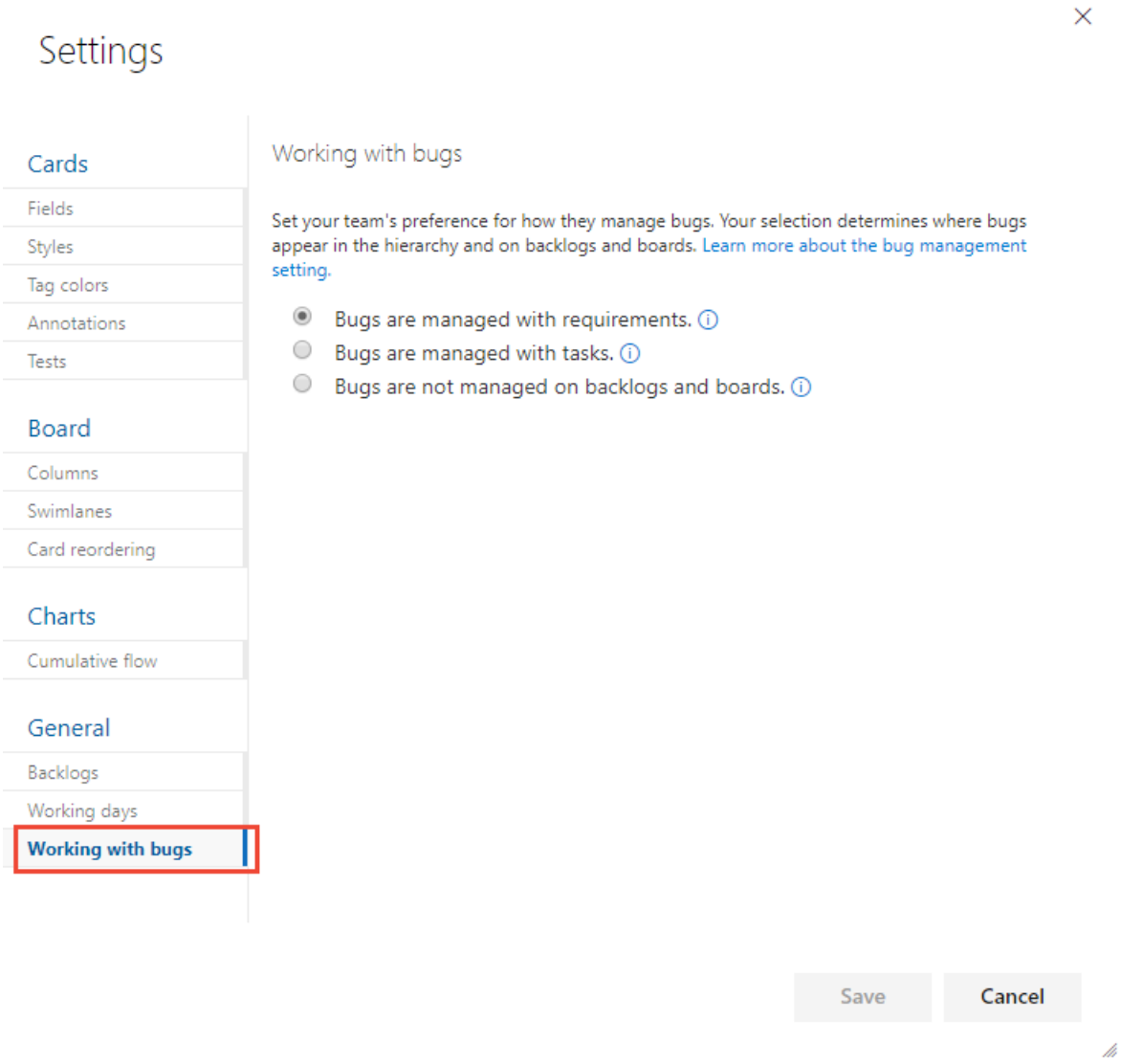


4. When you're done with your changes, choose **Save**.
5. To see the changes, open or refresh the team's backlog or Kanban board.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose **Board settings** to open the settings dialog.



3. Choose **Working with bugs** and then choose the option that best meets your team's way of working.



4. When done with your changes, choose **Save**.

5. To see the changes, open or refresh your team's backlog or Kanban board.

Nested items

> **TIP**
> If, after refreshing a backlog or board, you don't see bugs where you expect to see them, review How backlogs and boards display hierarchical (nested) items. Only leaf nodes of nested items appear on the Kanban or task boards.

When you manage bugs with requirements or tasks, they appear on one or more of your Agile tool backlogs and boards. However, if you nest items—create parent-child links of items that belong in either the Requirements or Task categories—not all items may appear on your backlogs and boards. To learn more about how nested items are treated, see How backlogs and boards display hierarchical (nested) items.

Add other work item types to your backlogs or boards

Bugs are a common item that teams want to track, and choose how they track them. For more information, see Manage bugs.

However, what if you want to track other work item types on your backlogs and boards?

You can add other work item types—such as change requests, issues, or impediments—by customizing your

process or project, based on the process model you use. For details,

- For the Inheritance process model, see Customize your backlogs or boards for a process.
- For Hosted XML and On-premises XML process models, see Add a work item type to a backlog and board.

You can add other work item types—such as change requests, issues, or impediments—by customizing your process or project, based on the process model you use. For details, see Add a work item type to a backlog and board.

For an overview of process models, see Customize your work tracking experience.

Create, list, and manage bugs

Bugs that are managed with requirements can be added through the product backlog or Kanban board. When bugs are managed along with tasks, you can add them to a sprint backlog or task board. Or, capture them using other tools. For more information, see Define, triage, and manage bugs.

> **TIP**
> Effort should automatically be part of a bug, but if you don't see it, customize the bug work item type for it to appear.

You can review bugs defined for your project by creating a query and specifying the **Work Item Type=Bug**. Or, open a predefined query, **Active Bugs** (Agile and CMMI), or **Work in Progress** (Scrum).

Related articles

- Define, capture, triage, and manage bugs
- Enable backlog levels of interest to your team
- Manage teams and configure team tools
- View, run, or email a work item query
- Triage work items
- Query by assignment or workflow changes

Use best practices when implementing Kanban in Azure Boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Having worked through the four configuration steps that were provided in Kanban basics, you're well on your way to implementing most of Kanban's six core practices.

1. **Visualize your workflow**. Teams track their work using a Kanban board that maps to how they work. Teams discuss how to best focus their resources to deliver the most important work.
2. **Limit work in progress**. Teams set and adhere to work in progress (WIP) limits they set for each stage of work. They use WIP limits to maintain focus on completing what they started and to identify bottlenecks occurring in their processes.
3. **Manage flow**. Teams monitor the overall work in progress and lead time, which gives them an idea of the speed of their delivery.
4. **Make policies explicit**. Teams spell out the standards and processes they agree to follow and make them readily accessible. For example, by making the team's Definition of Done for each work stage explicit, they can avoid wasted time and effort.
5. **Create opportunities for feedback**. Teams meet periodically to reflect on what's working and what needs improvement.
6. **Improve collaboratively, evolve experimentally**. Teams determine how to improve the continuous flow of delivery over time based of key metrics. They involve the entire team to gather insights and ideas. And, when persistent bottlenecks arise, they determine the changes that will help resolve them.

Over time, Kanban can provide your team insight as to how well their current processes work end-to-end and how to improve them. Incremental adoption of Kanban practices tends to yield greater success and builds on the sixth practice, to evolve experimentally. These practices arose from principles of Lean Manufacturing and Systems Thinking.

WIP limits, challenges, and solutions

Teams occasionally exceed WIP limits by one or two items. However, if your team frequently exceeds the limits by three or more items, they should review processes or adjust the limits.

After a team has worked with WIP limits for several weeks, discuss the challenges team members have. Then, decide which solutions they'd like to use and adjust the limits as needed. The following list, although not exhaustive, indicates some of the common challenges teams come across and proven solutions to overcome them.

WIP challenges

- **Social dynamics.** When it comes to following rules, team members can feel challenged. Some naturally want to rebel. Others don't see that the rule applies to them or don't see what they do as breaking the rules. Some team members may take on extra work that's outside the scope of what's been agreed to. And, still others don't want to give up multitasking as they believe it's the key to their productivity and individual achievement.
- **Variability of work in progress.** Wide variability in the size of work items—users stories and bugs—can negatively influence the overall workflow. For example, items with estimates that vary in size from 4

hours to 14 days, or 2 to 55 story points, can't be counted the same when it comes to constraining work in progress.

- **Ignoring systemic problems.** Instead of addressing workflow problems when bottlenecks occur, teams soldier on, putting in more time to overcome the bottleneck.
- **Culture change.** Adopting WIP limits introduces changes to the system, culture, and team.

Solutions for managing WIP

- **Build a culture of team productivity.** Address the natural tension that exists between individual productivity versus team productivity. Identify ways in which team members can enhance the overall productivity of the team and workflow process.
- **Size work to minimize variability.** Before work starts on any item, the team should discuss the overall size of work required and determine if it can be broken down into smaller tasks.
- **Focus on the flow of high priority items.** When idle, team members ask how they can help move an upstream item forward. When blocked or challenged to deliver an item on time, team members ask for help with completing an item.
- **Resource team capacity for each work stage.** Bottlenecks can occur when there aren't enough specialists who work in a particular stage. Determine ways to either increase team skills within each work stage, or add resources as needed to meet an understaffed work stage.
- **Build shared understanding.** Continuously strive to increase the team's understanding of how to work using Kanban practices. Take actions that allow team members to contribute to process changes. Consider scheduling regular retrospectives or team meetings to discuss what works well and what needs changing. Document team policies to limit ambiguity.
- **Use metrics to adjust processes.** Periodically check Kanban metrics of work in progress and lead time to determine when changes need to be made.
- **Manage culture changes mindfully.** People want to do their best work—a core tenet underlying Kanban and its associated disciplines. Apply change management principles as you adopt new practices. Create greater ownership within the team for the success of implementing WIP limits.

Help your agile team deliver working software

One of the 12 principles of Agile software development is to "deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."

> *All agile teams must establish what they mean when they say "working software," which is frequently known as the definition of done. At a high level, a piece of functionality is complete only when its features pass all tests and can be operated by an end user. At a minimum, teams must go beyond the unit test level and test at the system level. The best teams also include integration testing, performance testing, and customer acceptance testing in their definition of what it means to be done with a piece of functionality.* — Jeff Sutherland

One of the major causes of teams failing to implement Agile is they lack good definitions of done.

Each stage indicates a handoff to someone else who will do work. What information does the next person in the flow sequence need to quickly succeed. Incomplete work or uncommunicated information can lead to delays and wasted effort.

As a starting point, consider some of the following criteria as you work with your team to decide what done means throughout the development process.

Stage

Done criteria

Before work starts on a feature, user story, or requirement

1. User story is properly scoped and estimated.
2. Acceptance criteria is well defined.
3. Customer needs are understood by the team.
4. Dependencies have been identified and are tracked.

Bug filing

1. Bug title identifies the issue clearly.
2. Repro steps are clear and minimal.
3. Bug specifies a single issue.
4. Related issues are linked to as related.
5. Terms used are clearly understood within the team.

Code complete, ready for testing

1. Code complete, commented, and run against current version.
2. Code peer reviewed and meets team standards.
3. Builds without error.
4. Passes unit and system tests.
5. Remaining hours for tasks set to zero and task closed.

Test complete, ready for release

1. Unit tests implemented for all new features or functions.
2. Unit tests are all passing.
3. Acceptance/story tests are written and passing.
4. Regression tests are green with known failures.
5. Sufficient exploratory testing has been done.
6. Feature/function works correctly as expected.
7. Unsolved defects have been logged as bugs.
8. Code coverage is stable or improving.

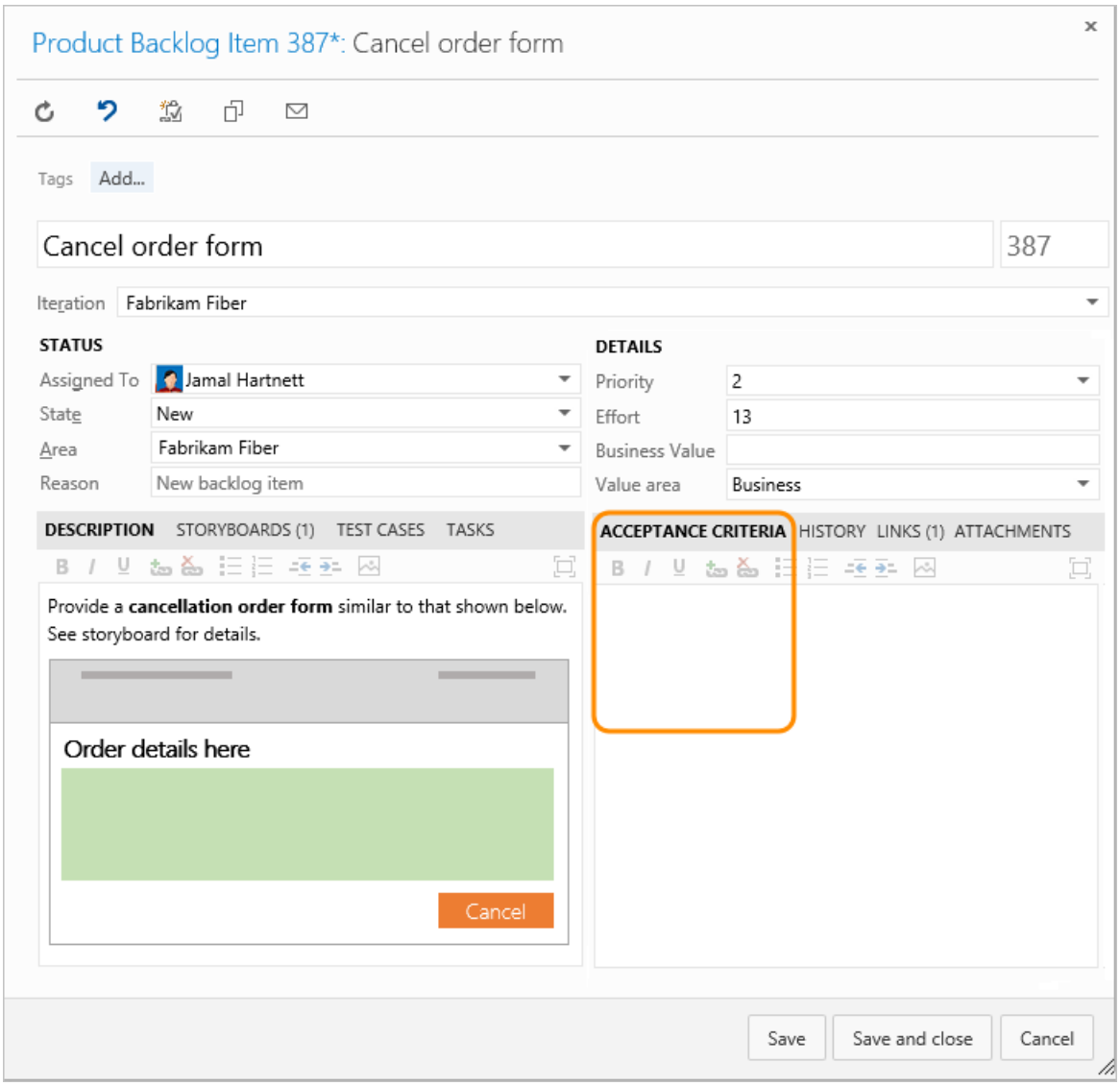
As your team makes progress, revisit your Definition of Done criteria.

> *A development team's Definition of Done is meant to expand over time. A newly formed team will invariably have a less stringent and smaller Definition of Done than a more mature team with a shared history of improving. Expanding a team's Definition of Done lies at the very core of Kaizen, a Japanese term meaning a mindful and constant focus on improvement. While a team may initially require only that code build before being checked in, over time they should evolve more exacting standards like the need for unit tests to accompany new code.* — David Starr

Acceptance Criteria versus Definition of Done

Acceptance criteria corresponds to what a customer should expect when a user story, feature, or requirement has been implemented. Conversations between the team and customers to determine the acceptance criteria helps ensure a common understanding within the team to meet customers' expectations. The acceptance criteria can be used as the basis for acceptance tests so that the team can more effectively evaluate whether an item has been satisfactorily completed.

Acceptance criteria defines when a feature is shippable. Capture the criteria for each backlog item in the Acceptance Criteria field (for Scrum product backlog items) or the Description field (for Agile user stories and CMMI requirements).



The Definition of Done, though, is about delivering an incremental piece of a feature as it moves from not started to complete. Agile teams meet with greater success when each handoff made is in a ready state for the recipient to begin their work.

> *Agility requires delivering done, ready-to-use increments of working software each Sprint. Yet most Scrum and agile teams generate partially done, incomplete Increments. When a Scrum Team is asked why Product Backlog requirements were not completely done in a Sprint, team members often reply, "We didn't have time."* — Ken Schwaber and David Starr

Other resources

- DoD Goes Agile
- Walking Through a Definition of Done
- Agile Culture
- What is Kanban?
- Kanban: Successful Evolutionary Change for Your Technology Business by David J. Anderson
- Agile Project Management with Kanban by Eric Brechner

Key Kanban concepts and terms in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This article provides a short dictionary of terms and available tools used in tracking work using Kanban boards and Kanban methods. See also:

- Agile glossary
- Work item field index
- Project management and navigation glossary

Blocker

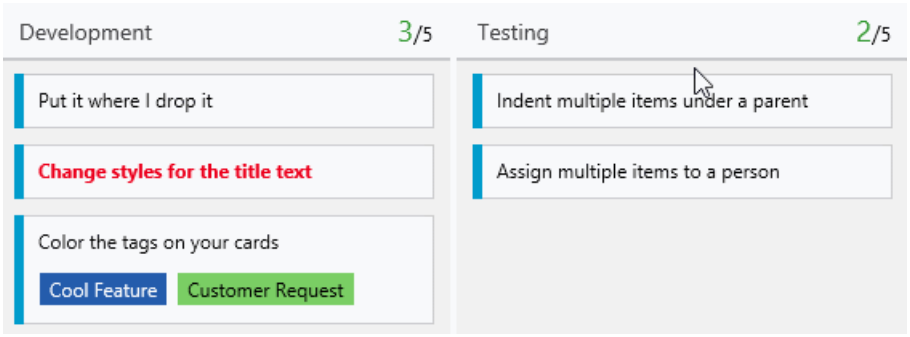
An issue that prevents work from progressing. You can highlight work that is blocked by using tags and changing the card color. Learn more: Customize cards, Define style rules to highlight cards.

Bottleneck

A constraint in the system that limits the flow of work. Identifying bottlenecks makes it easier to reduce their impact and provides a mechanism for controlling work flowing through the process. Learn more: Split columns, Identify bottlenecks.

Card reordering

Card reordering is a configurable setting for a team's Kanban board that either forces cards to maintain the backlog priority when dragged and dropped on the board, or allows the priority order to change. Learn more: Reorder cards.



Cumulative flow diagram (CFD)

The in-context CFD report shows the count of items in each Kanban column for the past 30 weeks or less. From this chart, you can gain an idea of the amount of work in progress and lead time. Work in progress counts unfinished requirements. Learn more: Cumulative flow, lead time, and cycle time guidance.

Cycle time

Cycle time is the time calculated for a work item from first entering an *In Progress* category state to entering a *Completed* state category. Learn more: Cumulative flow, lead time, and cycle time guidance.

You can gain valuable metrics and visualize the cycle time for a team and a configurable time period by adding the Cycle Time widget to the dashboard.

Definition of Done

Criteria that a team specifies for each stage of work to share and standardize on what makes up work being done at that stage. Learn more: Kanban best practices, working software and the Definition of Done.

Kanban board

An interactive, electronic sign board that supports visualization of the flow of work from concept to completion and lean methods. Azure DevOps provides a Kanban board for each product and portfolio backlog. Learn more: Kanban basics and Kanban board features and epics.

> **IMPORTANT**
> Work items that appear on more than one team's Kanban board can yield results that don't meet your expectations because each team can customize its Kanban board columns and swimlanes. The values assigned to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields might differ from what you expect when another team updates the work item from a different board. To learn more, see Add, review, and update work items in Azure Boards.

Kanban columns

A Kanban column maps to a stage of work. The default columns map to the workflow states of the work item types that appear on the Kanban board. You configure the columns to map workflow states of your team. Learn more: Kanban basics, Map the flow of work.

Lead time

Lead time is the time calculated for a work item from first entering a *Proposed* category state to entering a *Completed* state category. Learn more: Cumulative flow, lead time, and cycle time guidance.

You can gain valuable metrics and visualize the lead time for a team and a configurable time period by adding the Lead Time widget to the dashboard.

Live updates

Live updates is a Kanban board view option that when enabled automatically refreshes the Kanban board as other team members move or reorder cards. Learn more: Enable live updates.

Product backlog

An interactive list of work items that corresponds to a team's project plan or roadmap for what the team plans to deliver. The product backlog supports prioritizing work, forecasting work by sprints, and quickly linking work to portfolio backlog items. You can define your backlog items and then manage their status using the Kanban board.

Each product backlog can be customized by a team. Learn more: Create your backlog.

Product backlog item

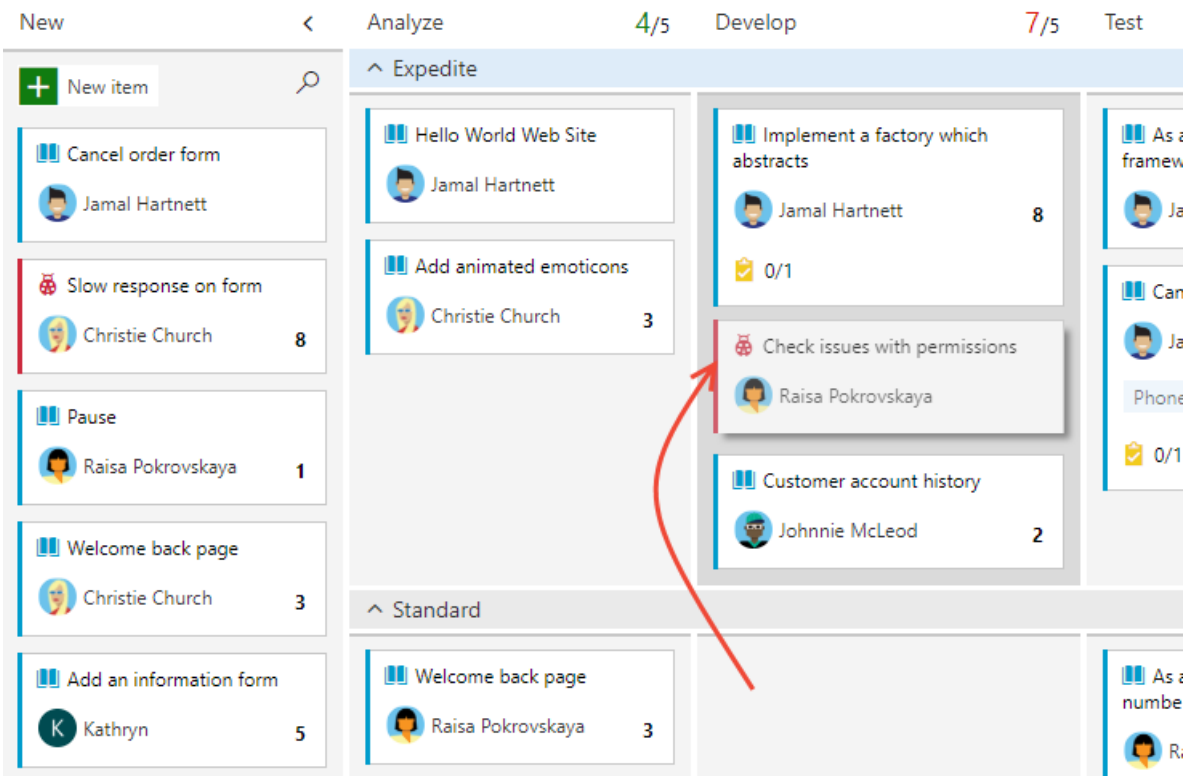
A type of work item that defines the applications, requirements, and elements that teams plan to create. Product owners typically define and stack rank product backlog items which are defined with the Scrum process. Learn more: Scrum process work item types and workflow.

Portfolio backlog

An interactive list of work items, similar to the product backlog, that supports organizing or grouping work under features, epics, or scenarios. Portfolio backlogs work similarly to product backlogs in that you can prioritize work and view the tree hierarchy of work. Learn more: Define features and epics.

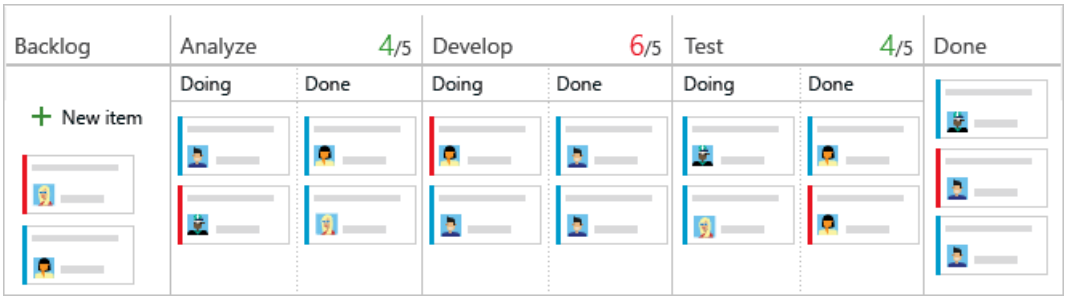
Swimlanes

A swimlane is a configurable row on a Kanban board, used to support different service class levels of work. Learn more: Speed up work with swimlanes.



Split columns

The Split columns feature lets your team implement a pull mechanism within the workflow process. Without split columns, teams push work forward, to signal that they've completed their stage of work. However, pushing it to the next stage doesn't necessarily mean that a team member immediately starts work on that item. With split columns, your team knows exactly how many items sit idle, waiting for work to begin. Learn more: Split columns.



Task checklists

A task is a type of work item used to track work required to complete a user story or product backlog item. You can add tasks from your Kanban board that appear as a checklist of work to be done. As you complete a task, you can update its status by checking the checkbox for the task. Learn more: Add tasks or child items as

checklists.

Task switching

Task switching, also referred to as *context switching* or *multitasking*, is when a team member shifts their attention among different tasks. Limiting task switching can allow a person to work more efficiently by minimizing the amount of time required to redirect cognitive function to a new activity.

User story

A type of work item that defines the applications, requirements, and elements that teams plan to create. Product owners typically define and stack rank user stories. User story is defined with the Agile process. Learn more: Agile process work item types and workflow.

WIP limit

A WIP limit is a constraint that a team applies to one or more workflow stages to help prevent potential bottlenecks that hinder the continuous flow of work in the system. Learn more: Work in Progress limits.

Work in Process (WIP)

Work that has been started but isn't done or completed.

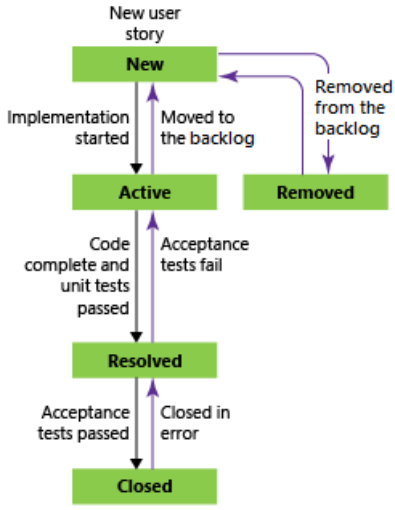
Workflow states

Workflow states are defined for each work item type to support tracking the status of a work item, from its creation to it's completion. These states define the workflow process: actions, steps, or stages that a piece of work goes through from inception to completion.

The State and Reason fields differ depending on the work item type and process selected for the project.

- Agile process
- Basic process
- Scrum process
- CMMI process

Workflow states: User Story, Agile process



You can customize your workflow states, adding states or renaming states. Learn more: Customize the workflow.

You can customize your workflow states, adding states, renaming states, and changing state transitions and reasons. Learn more: Customize the workflow.

Workflow state categories

State categories determine how the Kanban board treats each workflow state. The state categories used by the backlogs are *Proposed*, *In Progress*, *Resolved*, and *Completed*. To learn more, see Workflow states and state categories.

Related articles

- Refine your backlog
- Kanban best practices
- About boards and Kanban

View and configure a Cumulative Flow Diagram

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You use cumulative flow diagrams (CFD) to monitor the flow of work through a system. There are two CFD charts: the in-context report you can view from a team backlog or Kanban board and the CFD widget you can add to a dashboard.

CFDs help teams monitor the count of work items as they progressively move through various workflow states. These diagrams can show the flow of epics, features, user stories, issues, product backlog items, or requirements, depending on the process selected for your project:

- Agile
- Basic
- Scrum
- Capability Maturity Model Integration (CMMI)

CFDs help teams monitor the count of work items as they progressively move through various workflow states. These diagrams can show the flow of epics, features, user stories, product backlog items, or requirements, depending on the process selected for your project:

- Agile
- Scrum
- Capability Maturity Model Integration (CMMI)

Use this article to learn how to:

- Configure the Cumulative Flow Diagram widget (Analytics)
- View and configure the CFD in-context report (Analytics)

Use this article to learn how to:

- Configure the Cumulative Flow Diagram widget (Analytics)
- View and configure the CFD in-context report (work tracking data store)

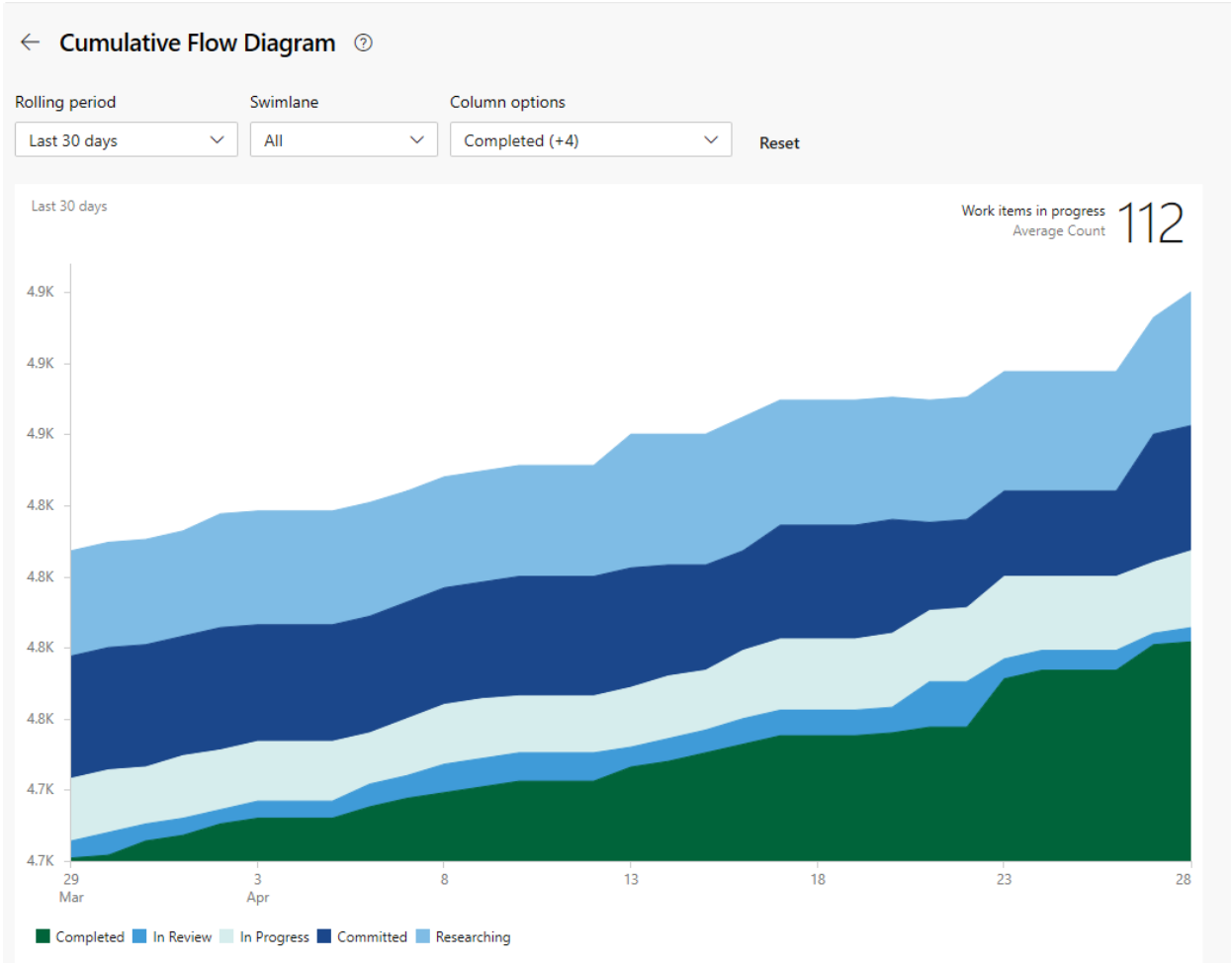
You use cumulative flow diagrams (CFD) to monitor the flow of work through a system. CFDs help teams monitor the count of work items as they progressively move through various workflow states. These diagrams can show the flow of epics, features, user stories, product backlog items, or requirements, depending on the process selected for your project:

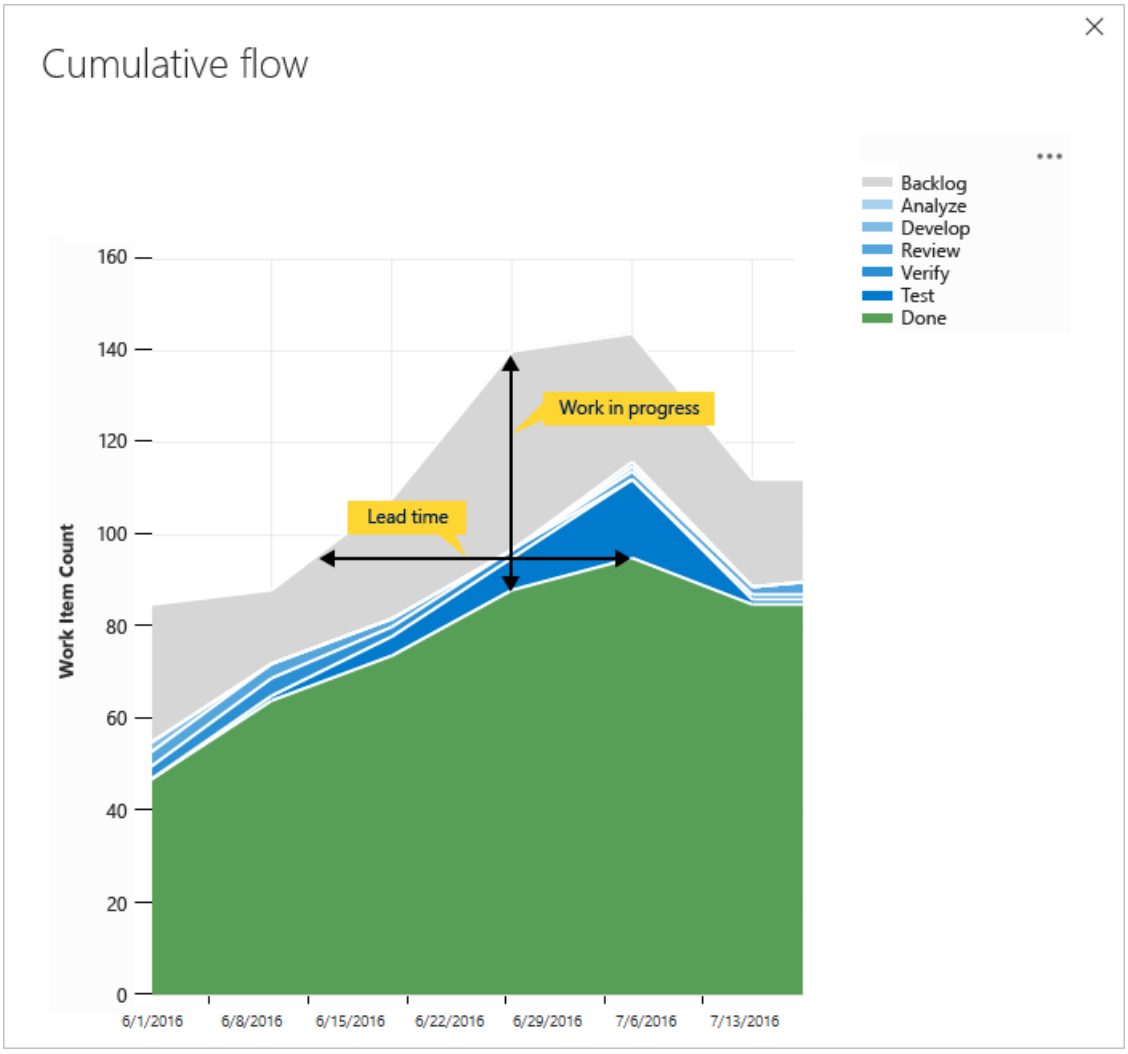
- Agile
- Basic
- Scrum
- Capability Maturity Model Integration (CMMI)

Use this article to learn how to:

- View and configure the CFD in-context report (work tracking data store)

The CFD shows the count of items in each Kanban column for the selected time period. From this chart, you can gain an idea of the amount of work in progress and lead time. Work in progress counts unfinished requirements. Lead time indicates the amount of time it takes to complete a requirement once work has started.


Cumulative Flow Diagram
Rolling period
Last 30 days
Swimlane
All
Column options
Completed (+4)
Reset
Last 30 days
Work items in progress
Average Count
112
4.9K
4.9K
4.9K
4.8K
4.8K
4.8K
4.8K
4.7K
4.7K
29
Mar
3
Apr
8
13
18
23
28
Completed
In Review
In Progress
Committed
Researching


Cumulative flow
Backlog
Analyze
Develop
Review
Verify
Test
Done
Work in progress
Lead time
Work Item Count
160
140
120
100
80
60
40
20
0
6/1/2016
6/8/2016
6/15/2016
6/22/2016
6/29/2016
7/6/2016
7/13/2016

> **NOTE**
> The in-context report always uses the blue-green color theme. However, the Analytics-based Cumulative flow diagram widget provides support for choosing different color themes.

For the CFD to provide useful information, you'll want to update the status of work items to reflect progress as it occurs. You can quickly make these updates through your Kanban board.

For usage guidance, see Cumulative flow, lead time, and cycle time guidance.

Prerequisites

- You must be a member of a project. If you don't have a team project yet, create one.
- If you haven't been added as a project member, get added now.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Have enabled or installed Analytics. You must be an account owner or a member of the **Project Collection Administrators** group to add extensions or enable the service.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

Open your backlog from the web portal

1. Check that you selected the right project, and select **Boards** > **Backlogs**. Then select the correct team from the team selector menu.

   To select another backlog, open the selector and then select a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.

2. To view the in-context reports for the product backlog, check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level. Or

1. Check that you selected the right project, and select **Boards** > **Backlogs**. Then select the correct team from the team selector menu.

   To select another backlog, open the selector and then select a different team or select the **Browse all backlogs** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.

2. To view the in-context reports for the product backlog, check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level. Or

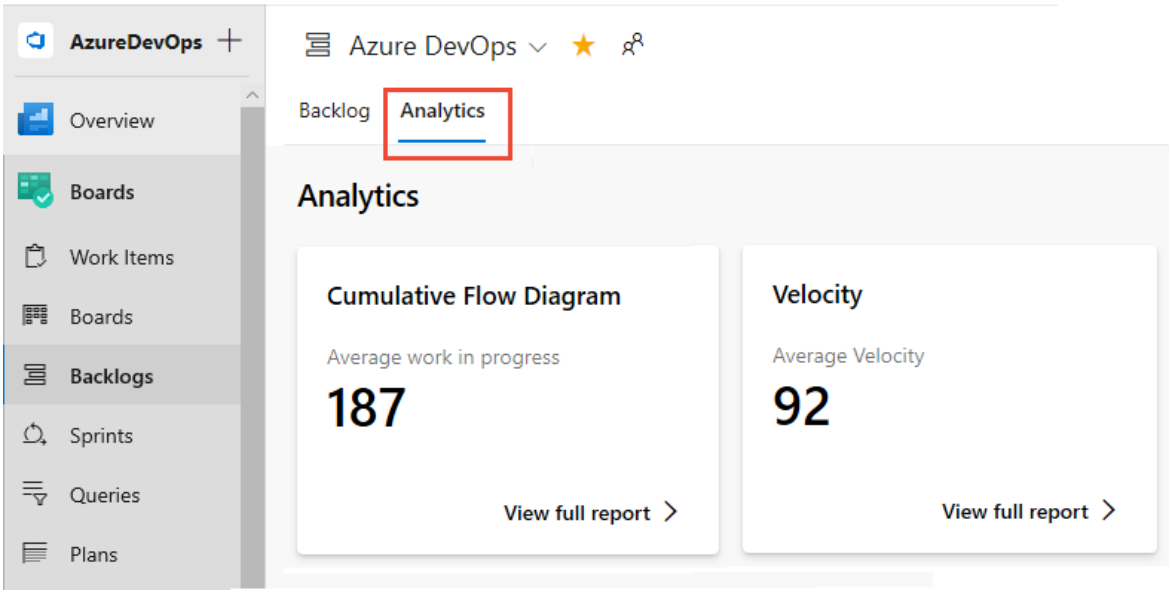
On your web browser, open your team's product backlog and select the team from the project and team selector. Then select **Work** > **Backlogs**. Select the product backlog, which is **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI.

To select another team, open the project and team selector. Select a different team, or select the **Browse** option.

View the CFD in-context report

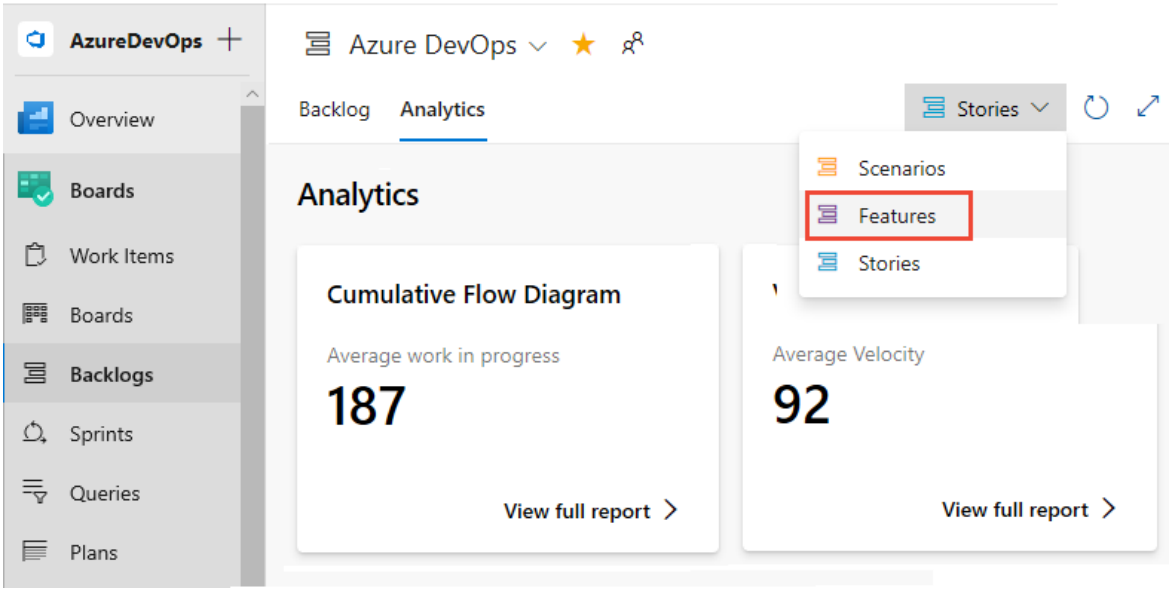
CFD reports are available for each backlog level, both product and portfolio backlogs. Each report provides interactive controls to provide each user the view of interest to them.

1. You open the CFD for your product or portfolio backlog by choosing **Analytics**.



   The Average work in progress value excludes completed work items.

2. To select a portfolio backlog, select it from the backlog selector menu.

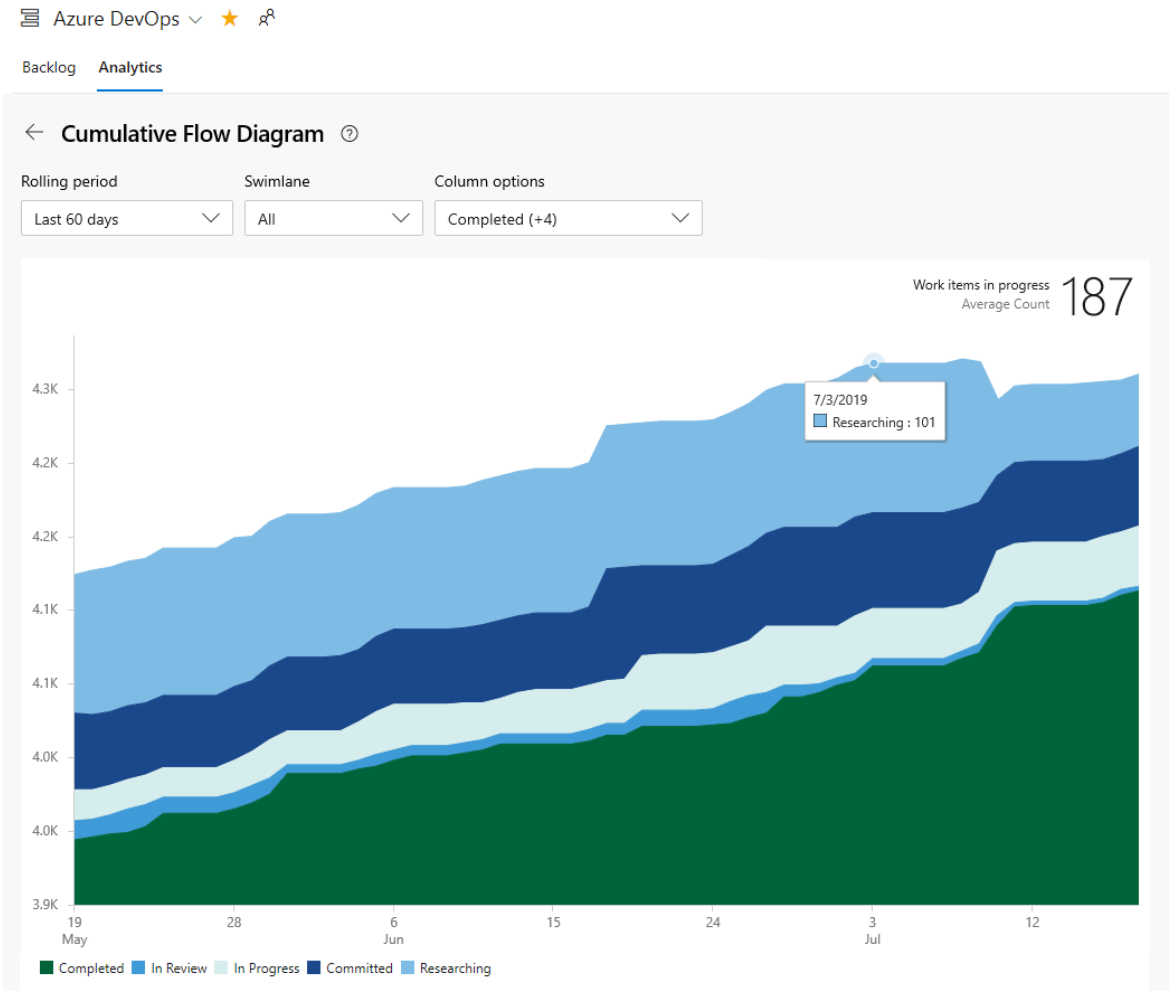


3. Next, select **View full report** for the Cumulative Flow Diagram.

4. Use the interactive controls to select the time frame, swimlanes, and workflow states or Kanban board columns. You can select a rolling period of 14 days or up to 180 days.

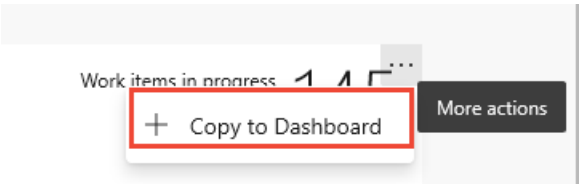
   Hover over a point in time to show how many work items are in a particular state. The default setting for the Cumulative Flow Diagram-Average work in progress includes completed work items since the team started tracking work.

   For example, On July 3, 101 items were in a Research state.



   The selections you make are only set for you, and persist across sessions until you change them.

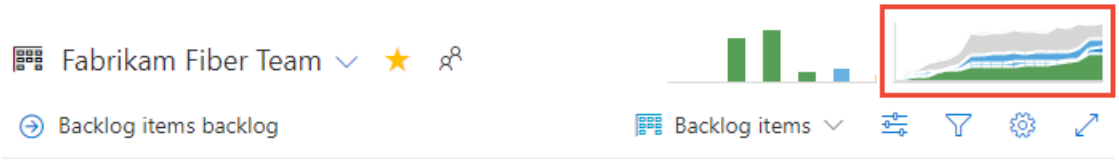
5. To add the report to a dashboard, select the ··· actions icon and select **Copy to Dashboard**.



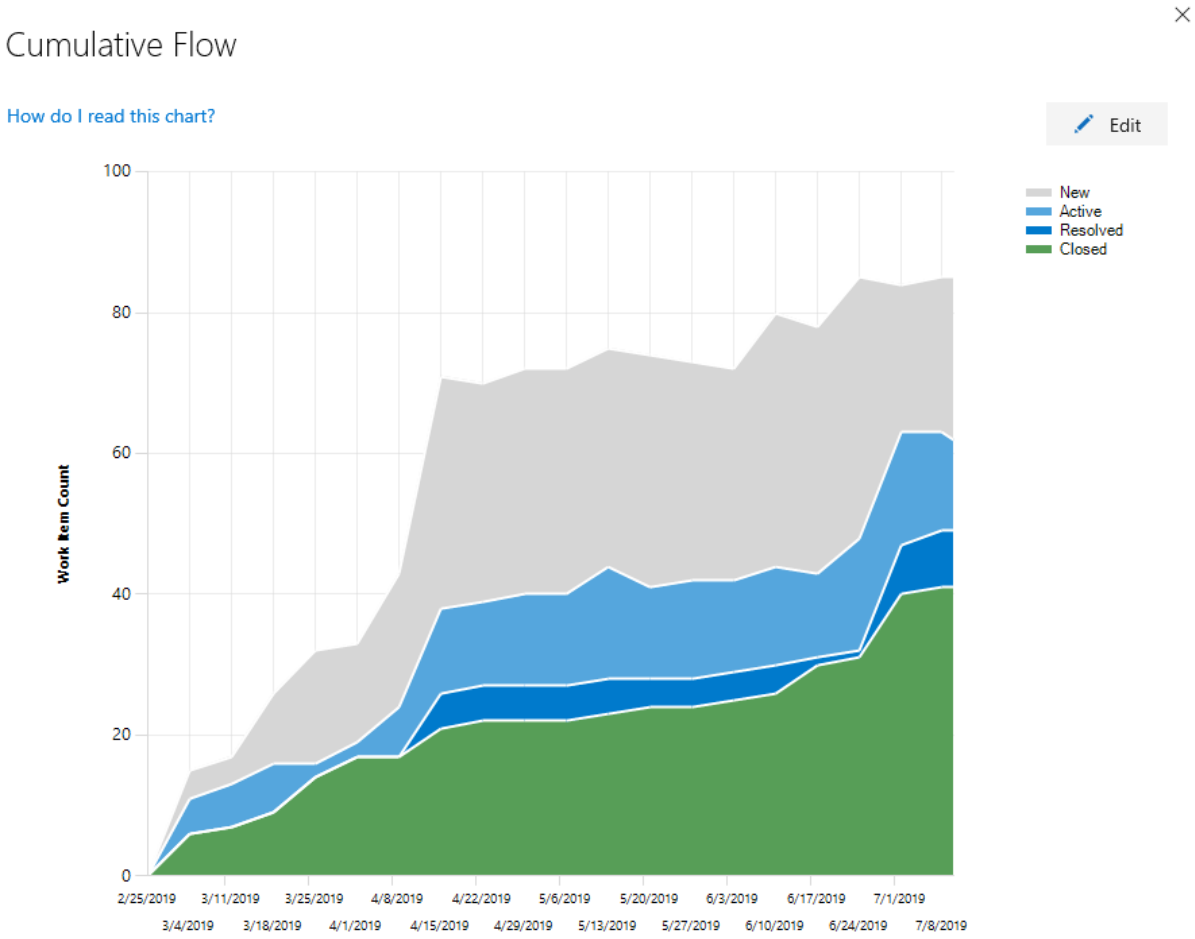
Select the dashboard and select **OK**.

6. To return to the Analytics summary, select the back arrow.

To open the CFD in-context report for your product or portfolio backlog, select the image in the upper-right corner of your **Boards > Boards** page.



The image opens to display a larger view of the CFD.



To open the CFD in-context report for your product or portfolio backlog, select the image in the upper-right corner of your **Work > Backlogs** page.

Configure the CFD in-context report

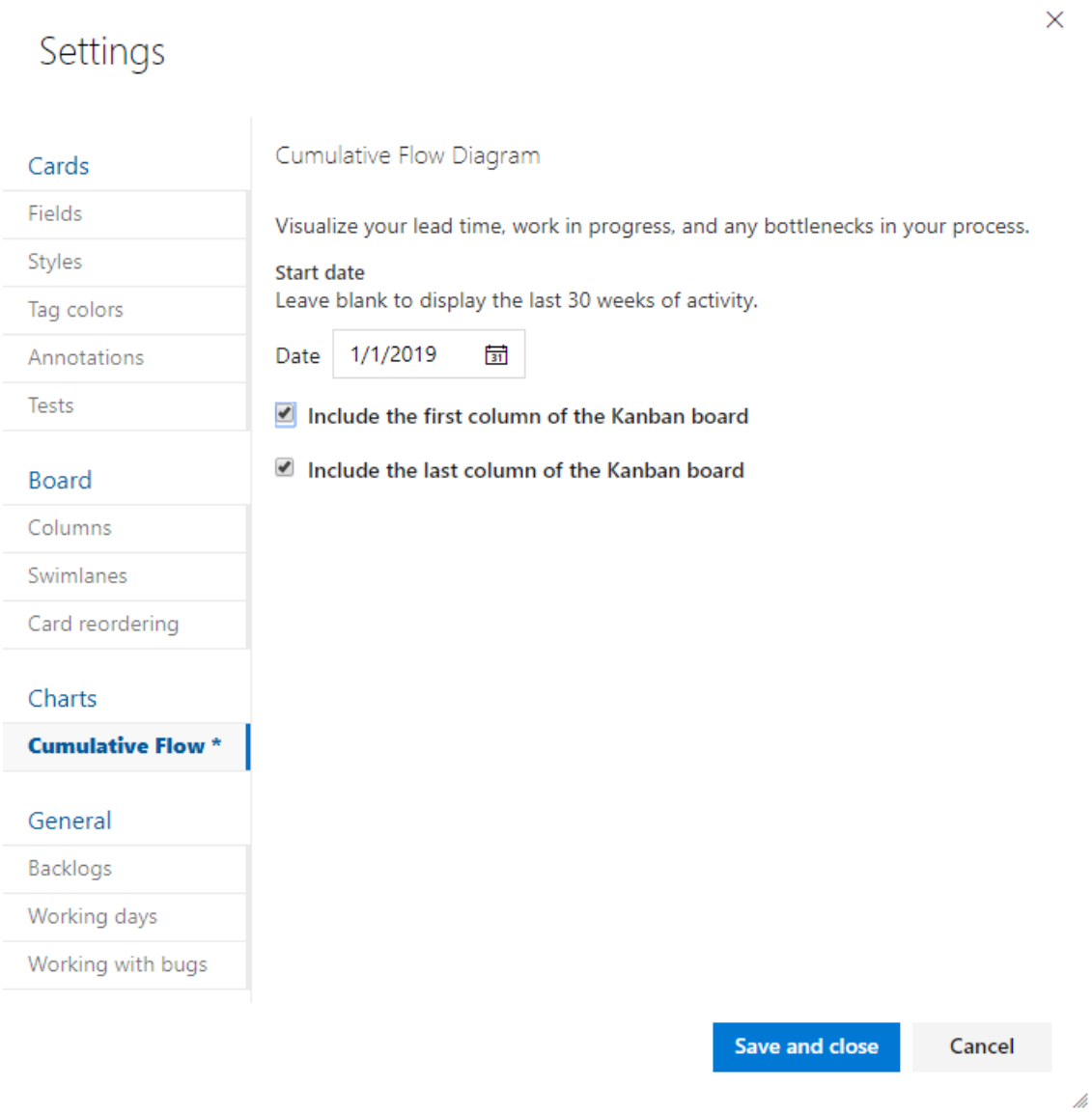
Each team can set their preferences for the in-context cumulative flow charts.

1. Open the backlog level for which you want to configure and then open the common configuration dialog. Select the gear icon.



   If you're not a team admin, get added as one. Only team and project admins can customize the Kanban boards and CFD charts.

2. Select **Cumulative flow** and specify the team's preferences.

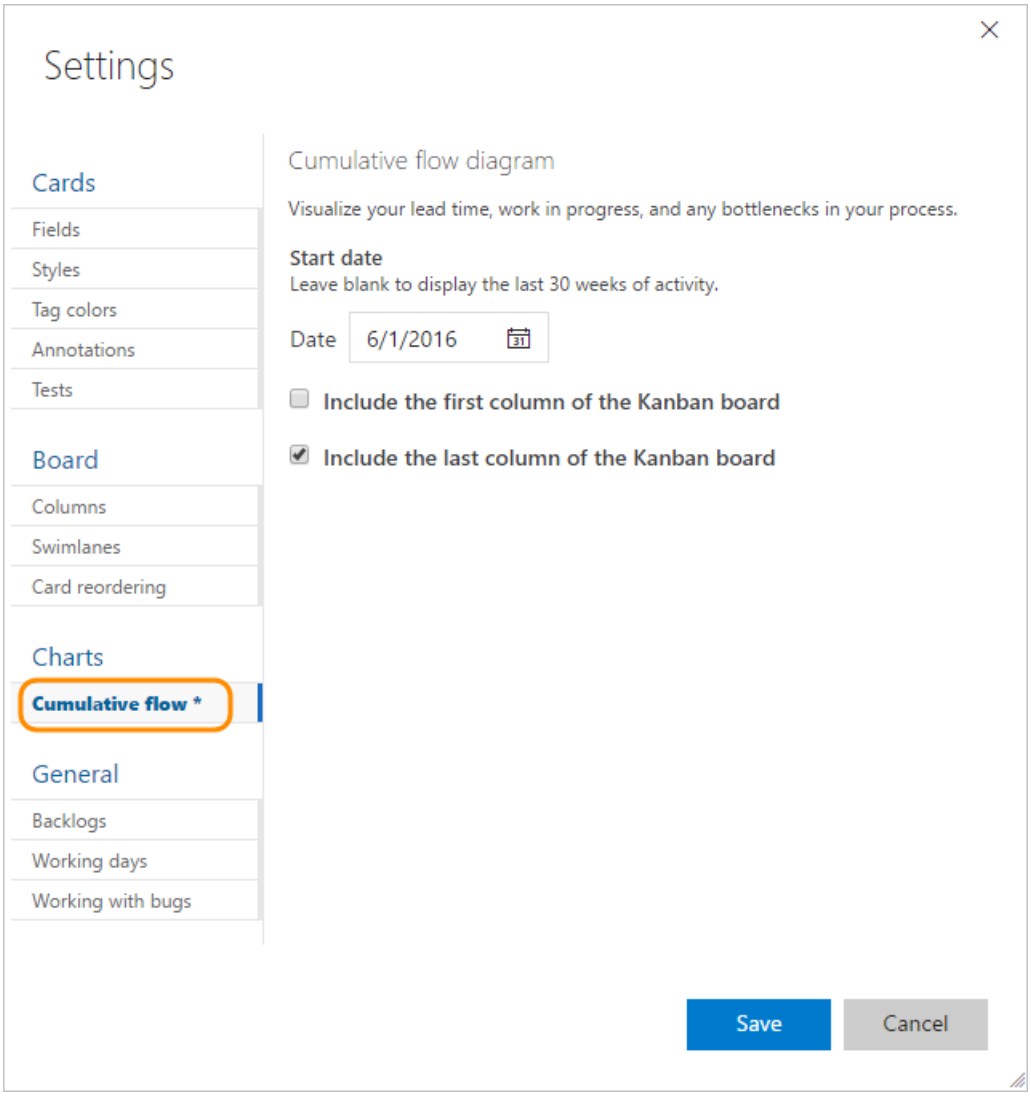


1. Open the backlog level for which you want to configure and then open the common configuration dialog. Select the gear icon.



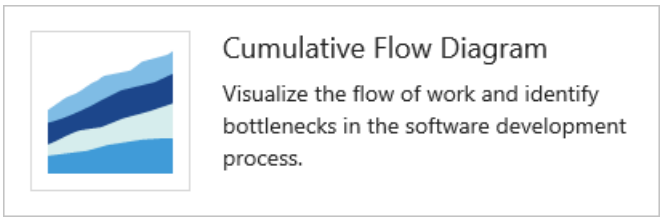
If you're not a team admin, get added as one. Only team and project admins can customize the team Kanban boards and CFD charts.

2. Select **Cumulative flow** and specify the team's preferences.



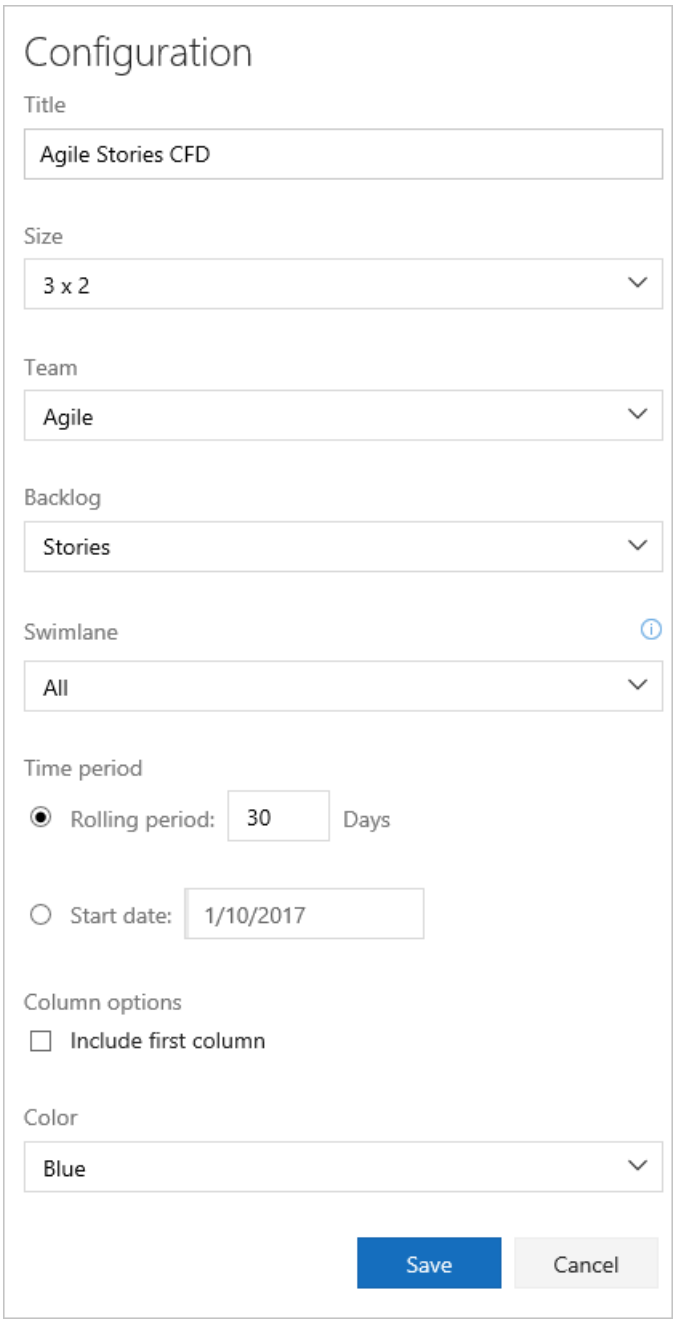
Add the Cumulative Flow Diagram widget to your dashboard

1. If you haven't yet configured your Kanban board, do that now. Define the columns and swimlanes that support your workflow processes.
2. If you want fixed scope CFD charts, make sure that you've defined the sprint iterations for those sprints of interest.
3. To add a CFD chart to your team dashboard, see Add a widget to a dashboard. Add the Cumulative Flow Diagram widget.



4. Select the ··· actions icon and select the **Configure** option to open the configuration dialog. Modify the title, and then select the values you want to monitor:

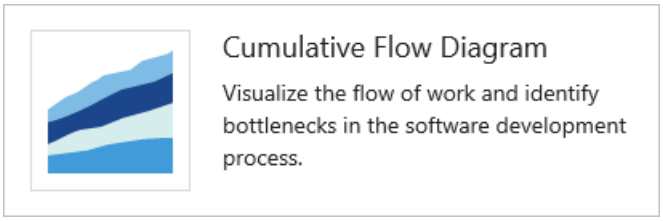
   - Team
   - Backlog level
   - Swimlanes
   - Time period



1. If you haven't yet enabled or installed Analytics, do that now.
2. If you haven't yet configured your Kanban board, do that now. Define the columns and swimlanes that support your workflow processes.
3. If you want fixed scope CFD charts, make sure that you've defined the sprint iterations for those sprints of

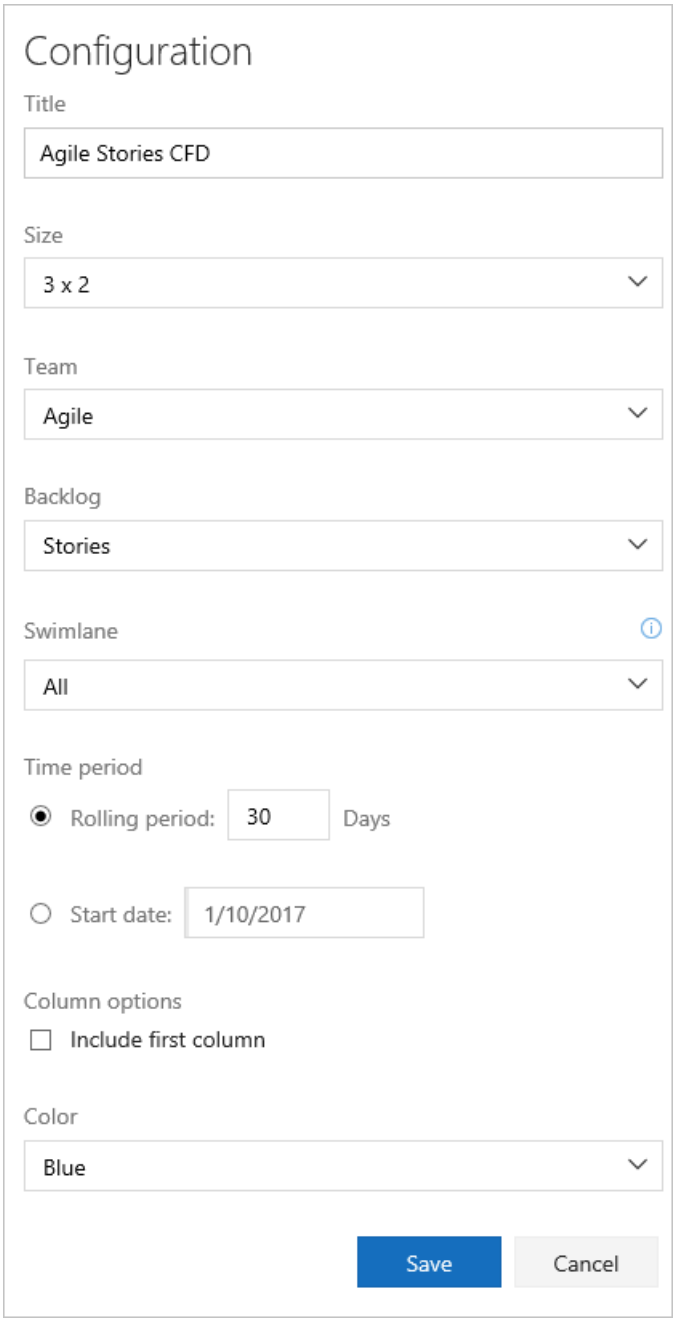
interest.

4. To add a CFD chart to your team dashboard, see Add a widget to a dashboard. Add the Cumulative Flow Diagram widget.



5. Select the ··· actions icon and select the **Configure** option to open the configuration dialog. Modify the title, and then select the values you want to monitor:

   - Team
   - Backlog level
   - Swimlanes
   - Time period



Configure the CFD widget

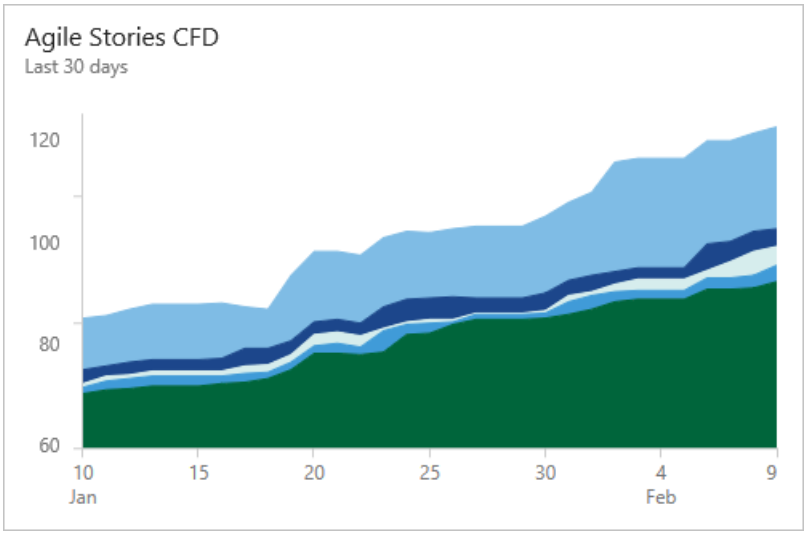
1. For a continuous flow diagram, select Rolling period and specify the number of days you want to view on the chart.

   Or, for a fixed scope view, select and specify the Start date. Select this view if your team employs a Scrumban process or follows a standard sprint process.

   The main difference between these two types of CFD charts is that the fixed scope CFD will provide information (in most cases) of scope change.

2. Select the color. You can distinguish the CFD for different teams by choosing different colors.

3. Select **Save** when done. The following image shows an example CFD chart showing 30 days of data.



Next steps

Cumulative flow, lead time, and cycle time guidance or Kanban basics

Related articles

- Add columns
- Add swimlanes
- Widget catalog
- Marketplace widgets

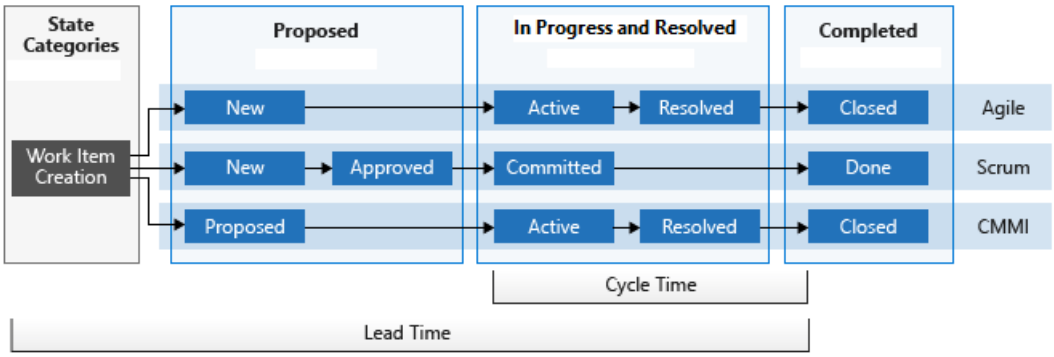
Lead Time and Cycle Time widgets

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Both lead time and cycle time widgets are useful to teams. They both indicate how long it takes for work to flow through their development pipeline. Lead time measures the total time elapsed from the creation of work items to their completion. Cycle time measures the time it takes for your team to complete work items once they begin actively working on them.

The following diagram illustrates how lead time differs from cycle time. Lead time is calculated from work item creation to entering a completed state. Cycle time is calculated from first entering an In Progress or Resolved state category to entering a Completed state category. To understand how workflow states map to state categories, see How workflow states and state categories are used in Backlogs and Boards.



These measures help teams plan, spot variations in efficiency, and identify potential process issues. The lower the lead and cycle times, the faster the throughput your team has.

In this article you'll learn:

- How to install and configure the Lead Time and Cycle Time widgets (Analytics)
- How to interpret the scatter-plot control charts
- How moving average and standard deviation are calculated in the charts

To learn more, see Cumulative flow, lead time, and cycle time guidance.

Prerequisites

- You must be a member of a project. If you don't have a team project yet, create one.
- If you haven't been added as a project member, get added now.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Have enabled or installed Analytics. You must be an account owner or a member of the **Project Collection**

Administrators group to add extensions or enable the service.

- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

Add the widget to your dashboard

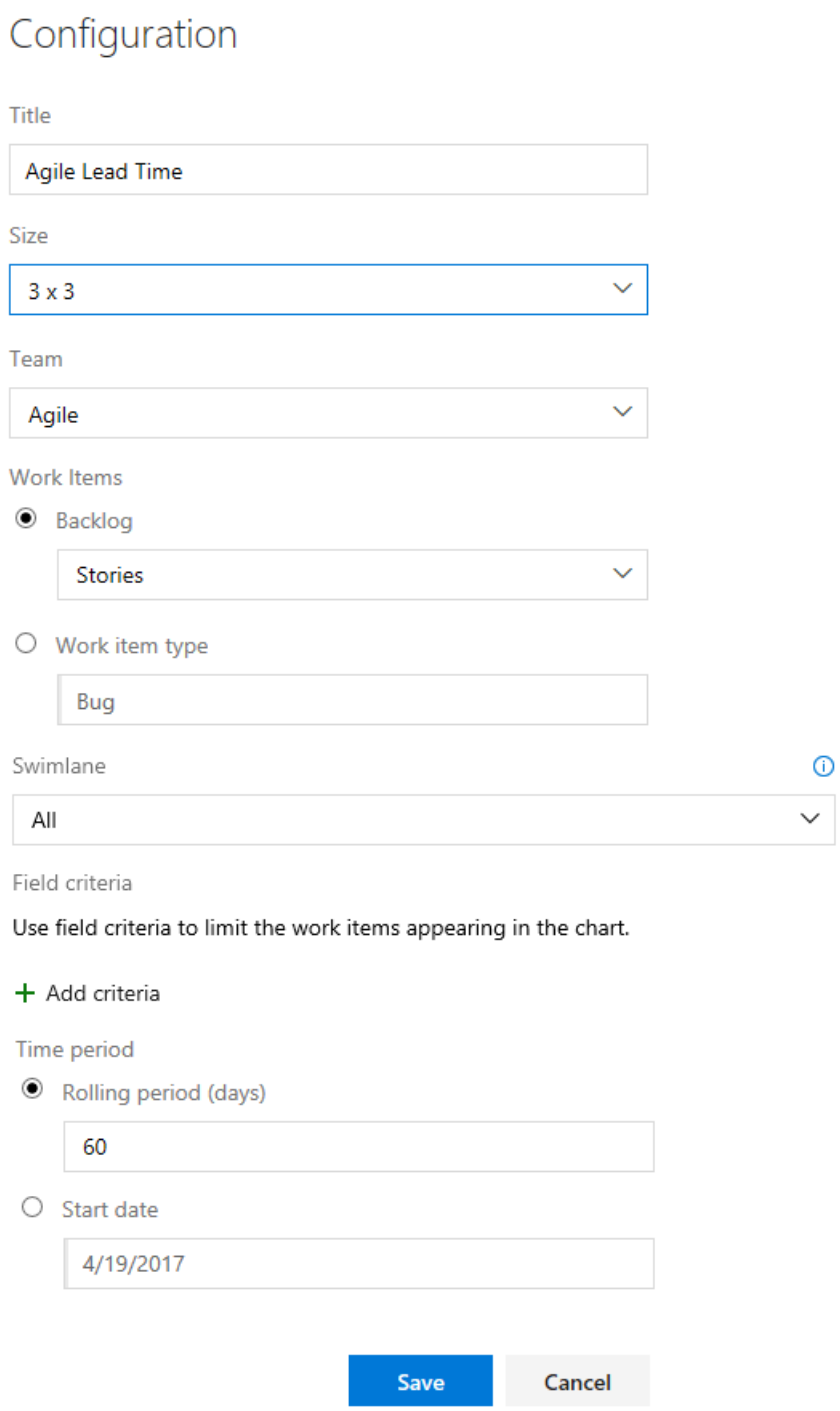
1. (Optional) If you haven't yet configured your team's Kanban board, do that now. Define the columns and swimlanes that support your workflow processes.
2. If you haven't yet added the widget to your dashboard, do that now. There are two widgets: Cycle Time and Lead Time. Select the one you want to display and configure.

Configure the Cycle Time and Lead Time widgets

The Configuration dialog is the same for the Cycle Time and Lead Time widgets. You configure these widgets for a team. To learn more about teams, see Add teams.

1. Select the ••• context menu icon and select **Configure** to open the configuration dialog. Modify the title, and then select the values you want to monitor:

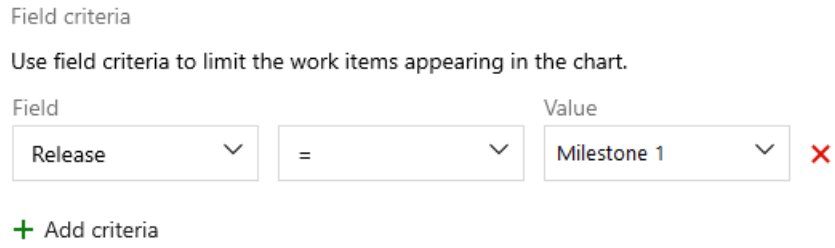
   - Team
   - Backlog level
   - Swimlane
   - Field criteria
   - Time period



To select a **Swimlane**, you must select a **Backlog**.

> **NOTE**
>
> You can only select work item types that have been added to a backlog. To add work item types to a backlog, see Customize your backlogs or boards (Inheritance process). For On-premises XML process, see Process configuration XML element reference.

2. To further filter the work items used to calculate the lead or cycle time, specify the **Field Criteria**. For example, all the work items whose Release field is set to Milestone 1.



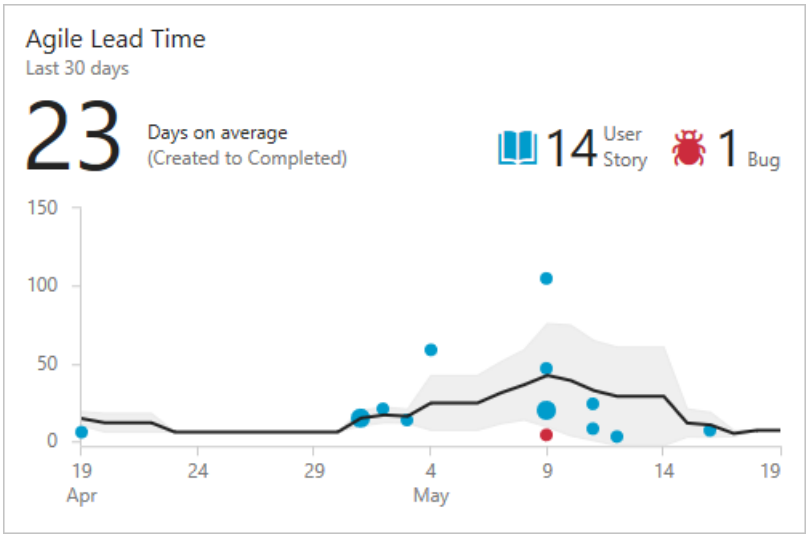
> **NOTE**
> Supplying no values to the filter may lead to selection of all workitems, or may be an invalid filter argument depending on type of filter criteria.

3. For a continuous flow, select **Rolling period** and specify the number of days you want to view on the chart.

   Or, for a fixed scope view, select and specify the Start date. Select this view if your team employs a Scrumban process or follows a standard sprint process.

   The main difference between these two types of charts is that the fixed scope chart will provide information (in most cases) of scope change.

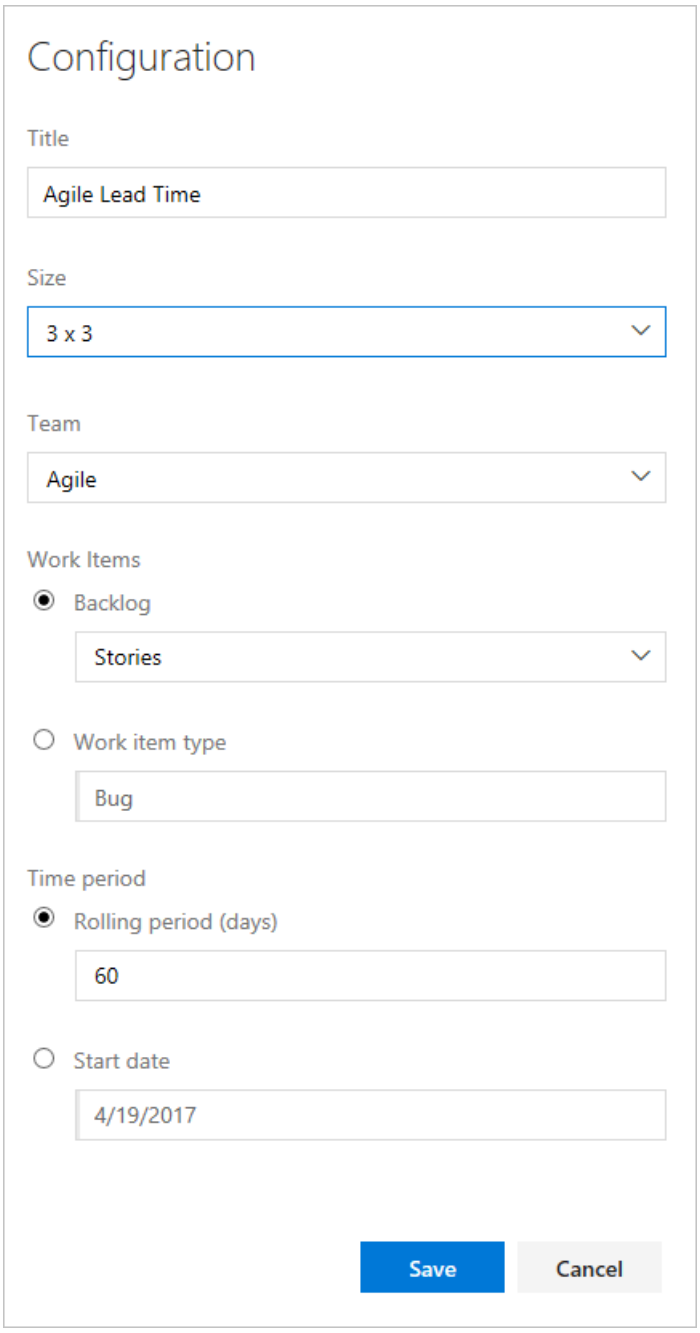
4. Select **Save** when done. The following image shows an example Lead Time chart showing 60 days of data.



For your lead/cycle time charts to provide useful data, your team must quickly update the status of those work items that the widgets track.

1. Select the ··· context menu icon and select **Configure** to open the configuration dialog. Modify the title and then select the values you want to monitor:

   - Team
   - Backlog level
   - Swimlane
   - Time period



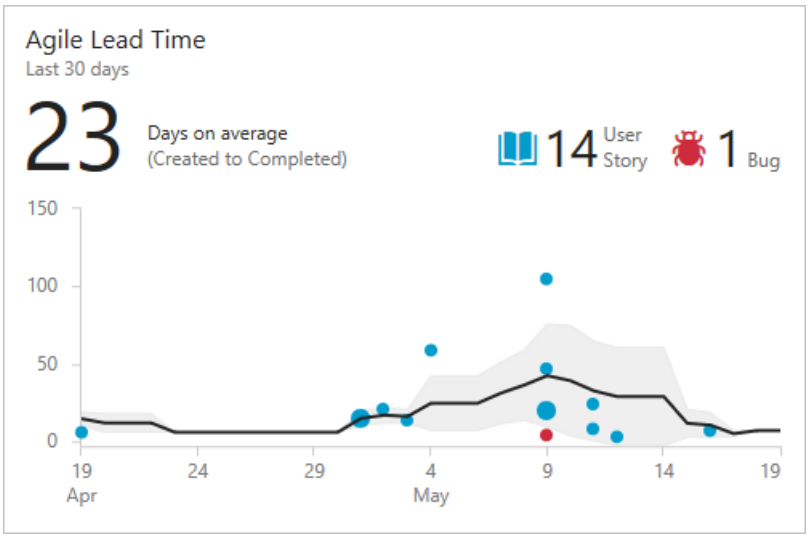
To select a **Swimlane**, you must select a **Backlog**.

2. For a continuous flow, select **Rolling period** and specify the number of days you want to view on the chart.

   Or, for a fixed scope view, select and specify the Start date. Select this view if your team employs a Scrumban process or follows a standard sprint process.

   The main difference between these two types of charts is that the fixed scope chart will provide information (in most cases) of scope change.

3. Select **Save** when done. The following image shows an example Lead Time chart showing 60 days of data.

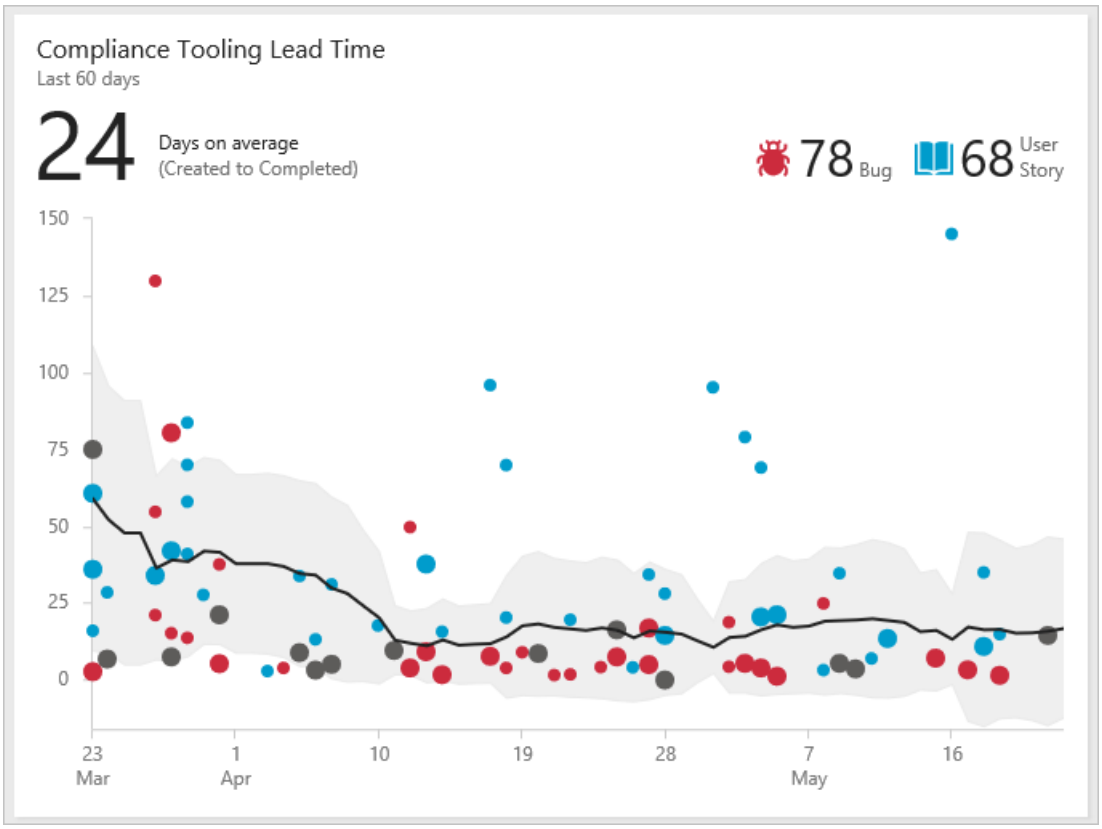


For your lead/cycle time charts to provide useful data, your team must quickly update the status of those work items that the widgets track.

Interpret the scatter-plot control charts

Both Lead Time and Cycle Time widgets display as scatter-plot control charts. They display summary information and provide several interactive elements.

Example Lead Time widget



The chart dots represent completed work items where their position on the horizontal axis represents the date the team completed them. Their position on the vertical axis represents the calculated lead time or cycle time.

- Larger dots represent multiple work items with the same lead/cycle time
- Dot color corresponds to the work item type displayed in the legend
- Dark gray dots correspond to a mix of work item types.

Summary elements

- Days on average (average lead time or cycle time) for the main work item types configured for the chart. This

number may not be equal to the average cycle/lead time of all work items. It depends on configurations used for widgets. The average number is calculated based on each day the team takes time for work item.

- The number of backlog work items used in the chart calculations; if there are more than three types of work items, you'll see a summary for **Other**
- The black trend line indicates the moving average
- The band around the trend line shows the standard deviation.

Interactive elements

- Hover over any dot to see which work items contributed to the data point and the lead/cycle time for those items
- Select a dot to open the work item or query that lists the work items
- To filter the chart, select a work item type in the legend (, , or other icon) to filter on that type; to return to the original chart, refresh the dashboard.

Moving average and standard deviation calculations

The daily moving average value corresponds to the average of data points that fall within the moving average window. The time-based moving average window is calculated based on the current day and previous *N* days. *N* corresponds to 20% of the number of days the chart displays, rounded down to the nearest odd number.

Here's an example. If the chart displays the last 30 days, then *N* = 5 days:

> 20% of 30 days is 6 days rounded down to the nearest odd number which is 5.

The moving average window for April 10 corresponds to the previous 5 days. So, the April 10 moving average is the average of all data points that fall on April 5 through April 10.

If you don't have data points that fall within the moving average window, the chart doesn't show a moving average line. This scenario can occur if you're starting out and there aren't enough days to calculate a moving average.

The standard deviation appears as a band that encompasses the moving average. Standard deviation is calculated based on all data points falling within the same moving average window. Like moving average, if no data points fall within the moving average window, the chart doesn't plot standard deviation.

Related articles

We recommend your team review the lead/cycle time charts before or during each retrospective. Use lead time to help estimate delivery times and track service level agreements (SLAs). Use cycle time to identify potential process issues, spot variations in trends, and help with planning.

- Cumulative flow, lead time, and cycle time guidance
- Kanban basics
- Cumulative flow diagram
- Workflow states and state categories
- Agile, Scrum, and CMMI processes

Use backlogs for effective project management in Azure Boards

10/4/2022 • 12 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With **Backlogs**, you can quickly plan your project by adding user stories or requirements to your product backlog. Once you have your plan in place, you can start driving code development efforts.

If you're a project administrator just getting started, review the Configure settings and manage your Azure Boards project. Review the settings to learn more about defining area and iteration paths and customizing your work item types. Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

Use backlogs

You plan and track your project using the suite of Agile tools you access from the web portal. Agile tools support the core Agile methods—Scrum and Kanban—used by software development teams today. Scrum tools support defining and managing work within sprints, setting capacity, and tracking tasks. Kanban tools allow you to manage a continuous flow of work via an interactive sign board.

If you're new to Agile, see What is Agile? for an overview.

In a nutshell, you use backlogs to:

- Quickly define the work your team is tasked with by defining user stories, product backlog items, or requirements
- Reorder your backlog to make sure you're working on the highest priority items first
- Add details and estimates to your backlog items
- Quickly assign backlog items to team members and to sprints. You can use either bulk update or drag to a sprint
- Group or organize backlog items by mapping them within a hierarchy
- Review the hierarchy or portfolio of work assigned to multiple teams
- Forecast work to estimate what can be delivered within a sprint
- Display rollup progress, counts, or totals to show completion of work or amount of work still to do.

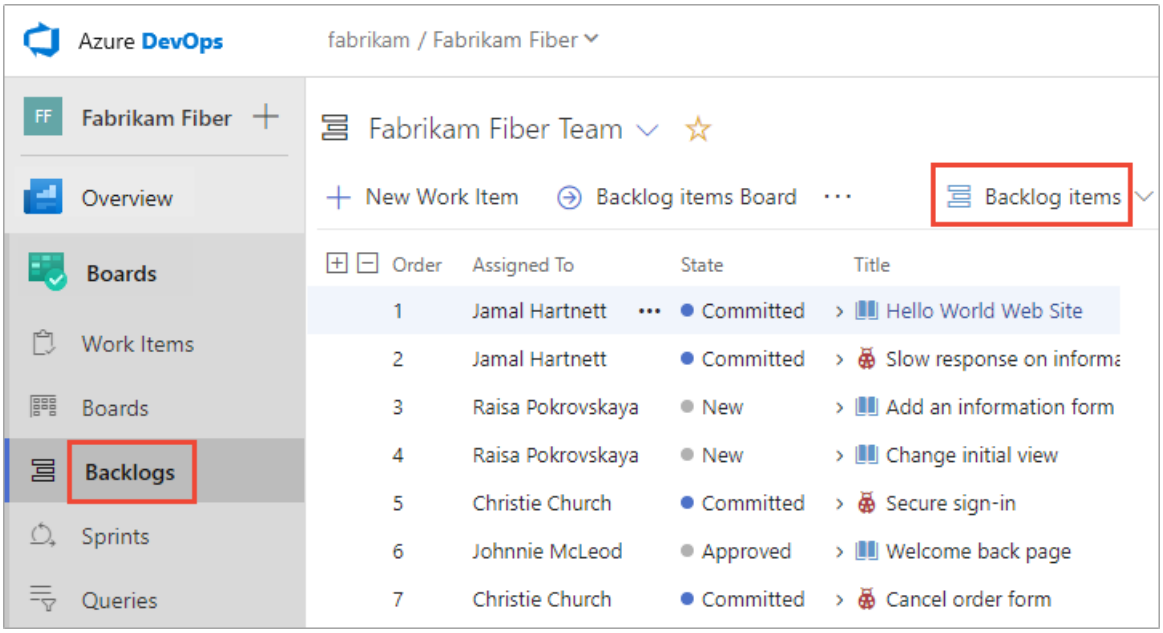
- Quickly define the work your team is tasked with by defining user stories, product backlog items, or requirements
- Reorder your backlog to make sure you're working on the highest priority items first
- Add details and estimates to your backlog items
- Quickly assign backlog items to team members and to sprints. You can use either bulk update or drag to a sprint
- Group or organize backlog items by mapping them within a hierarchy
- Review the hierarchy or portfolio of work assigned to multiple teams
- Forecast work to estimate what can be delivered within a sprint.

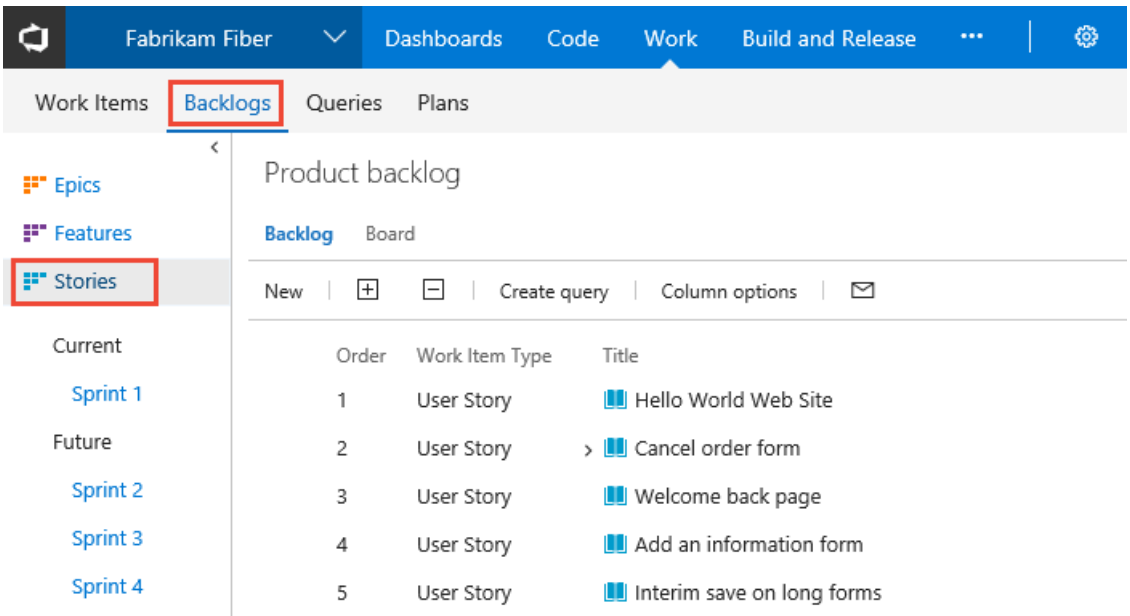
NOTE

To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans.
If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

Product and portfolio backlogs

Backlogs present work items as lists. A product backlog represents your project plan, the roadmap for what your team plans to deliver. Your backlog also provides a repository of all the information you need to track and share with your team. Portfolio backlogs allow you to group and organize your backlog into a hierarchy.





Backlog configuration

NOTE

How do I add a backlog or board? You don't add backlogs or boards. You add a team which is automatically configured with it's own set of backlogs and boards as described in About teams and Agile tools.

Each backlog is associated with a team. Team configuration settings determine the work items that will appear

on the team backlog. Specifically, the team administrator defines the following for their team:

- Selects the Area Paths that are active for the team, only work items assigned to these area paths appear on the team's backlog
- Sets the default Area Path and Iteration Path used when defining work items from the team backlog
- Selects the Iteration Paths that are active for the team
- Determines which backlog levels are active for the team
- Defines how bugs will be treated, as requirements or as tasks.

For details, see the following articles:

- Define iteration (sprint) paths and configure team iterations
- Define area paths and assign to a team
- Select backlog levels
- Show bugs on backlogs or boards

> **TIP**
> Each team member has several tools to configure their backlog view: **Expand/Collapse one level**, **Column Options**, **Backlog level selector**, **View options**, and **Filter** toolbar. Options set for each backlog level are distinct and persist until changed. To learn more, see Configure your backlog view.

Common backlog configurations for multiple teams

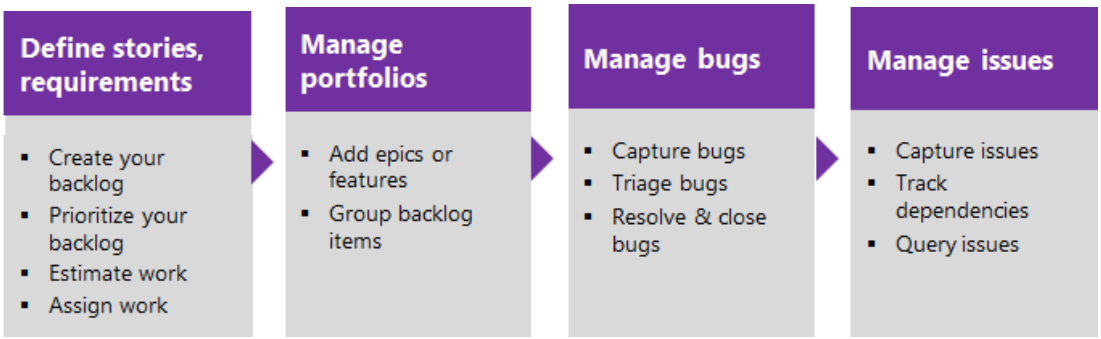
Question: *Can you define a backlog configuration that multiple teams can subscribe to?* **Answer**: *No. Each team controls their own team settings and backlog configurations.*

Because each user can configure their own **Column Options** and **View Options**, there's no way to configure a common backlog view for all teams. Also, there's no default column options that can be set for each team.

Because each user can configure their own **Column Options** and **View Options**, there's no way to configure a common backlog view for all teams. You can, however, define the default column options for all team members by editing the process configuration. To learn how, see Process configuration XML element reference, Set default columns.

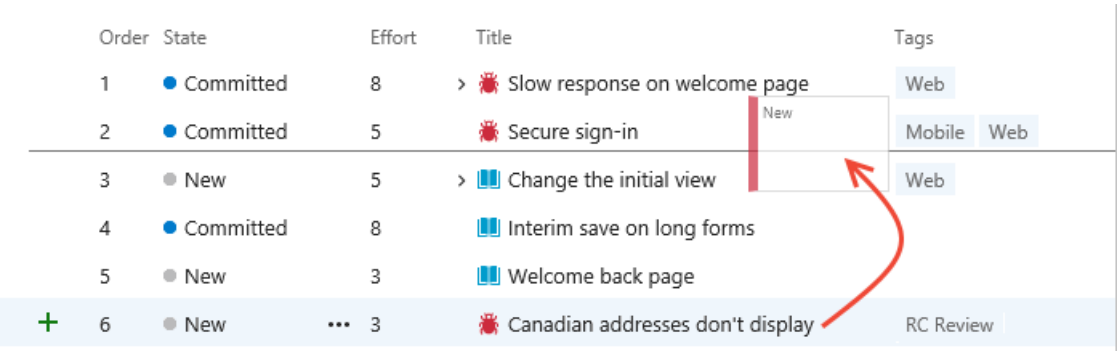
Define work items and create your backlog

You build your project plan by creating a backlog of work items. These items represent the features, requirements, user stories, or other work to complete. Portfolio backlogs provide support for organizing work in a hierarchical fashion. They help track major product initiatives or scenarios that rely on many stories or requirements. Different types of work items help you track different types of work, such as user stories, tasks, bugs, issues, and more.



Backlog priority or stack rank order

The sequence of items on each backlog is determined according to where you've added the items or moved the items on the page. As you drag items within the backlog list, a background process updates the **Stack Rank** (Agile and CMMI processes) or **Backlog Priority** (Scrum process) fields. These fields are used by the system to track the relative ranking of items on the product, feature, epic, or other portfolio backlog. By default, these fields don't appear on the work item form.



Refrain from using the bulk modify function to change the value of the backlog priority field. While you can assign a value to these fields, you'll be assigning the same value to all items you've selected for bulk edit.

The preferred method for bulk edit is to use multi-select to move items to the top, bottom, or specific position within the page. If you must edit one of the backlog order fields in bulk to get a large number of work items in the priority order you want, use Excel. You can export a query containing the backlog items, update either the Backlog Priority or Stack Rank fields, and then publish your changes.

In Progress items and work listed on the backlog

Backlogs are designed to display work that corresponds to a Proposed, In Progress, or Resolved category state. Once you've completed work and its state enters a *Done*, or *Closed* state, then it falls off the backlog view. You can always create a query to view completed work, or view the **Recently completed** pivot from the **Work Items** page.

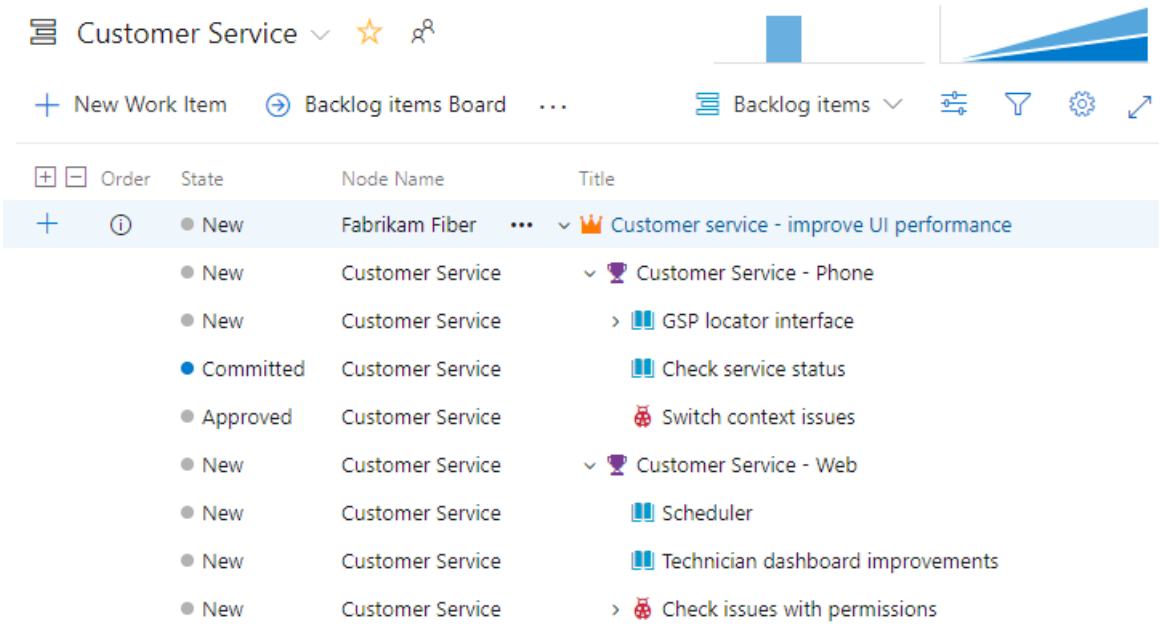
In general, you'll want to display all items that are in the **In Progress** category state, which corresponds to the *Active* and *Committed* states. To focus on work that is proposed but not in progress, you can toggle the backlog view to turn off **In Progress**. This toggle is useful when forecasting your product backlog.

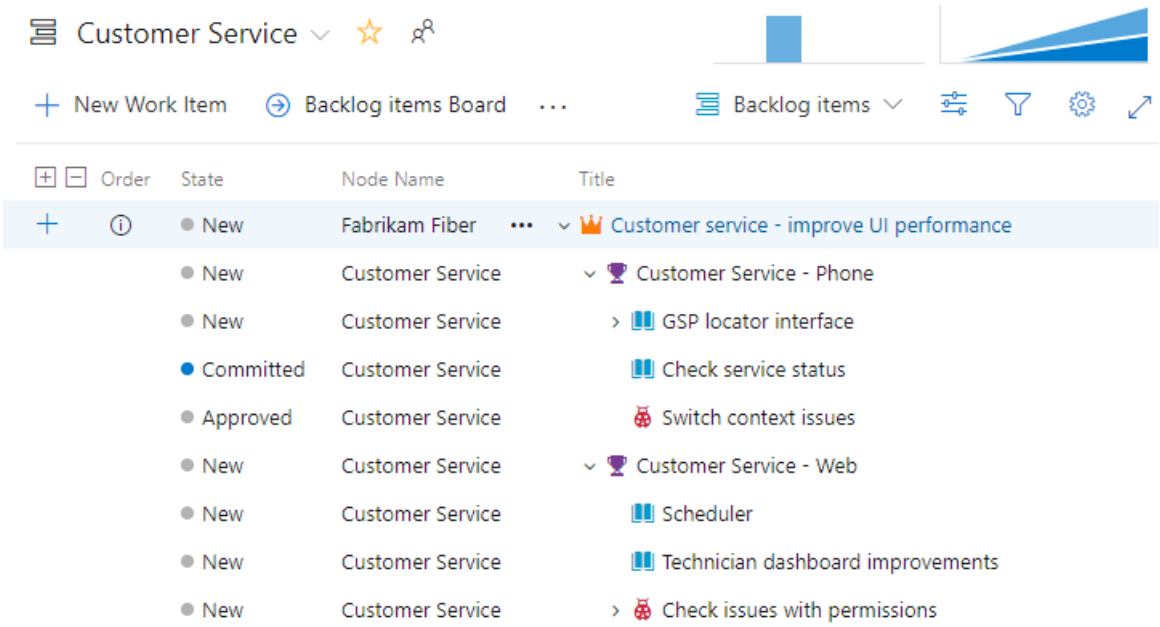
If your backlog is missing items, you might check if the **In Progress** view has been turned off. For more information, see Workflow states and state categories.

Organize your backlog by mapping and reparenting backlog items

When you have many initiatives your teams are working on, you may want to group the work according to these initiatives. By defining features and epics, you can group your work into a three-tiered hierarchy consisting of epics, features, and backlog items.

For example, here the Customer Service team has organized several backlog items under two features and one epic.





Velocity

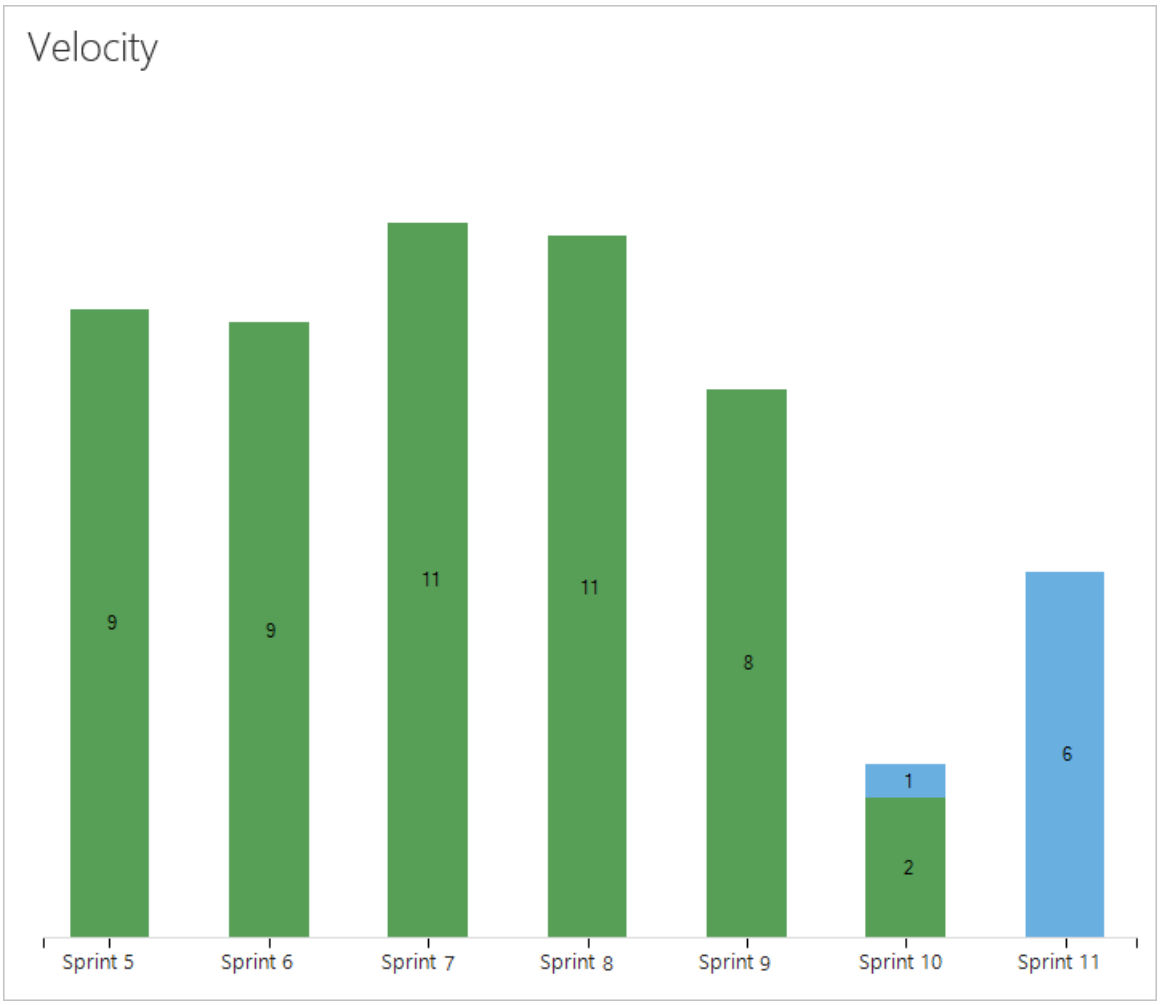
When you assign your backlog items to sprints, you'll gain access to an in-context velocity report for both product and portfolio backlogs. Velocity helps teams determine how much work they can perform sprint-over-sprint.

You can quickly configure the Velocity report to count work items or sum Story Points, Effort, Remaining Work or other numeric field.



When you assign your product backlog items to sprints, you'll gain access to an in-context velocity report for your product backlog. Velocity helps teams determine how much work they can perform sprint-over-sprint.

The report tracks the team's estimated backlog work—sum of Effort (Basic or Scrum processes), Story Points (Agile process), or Size (CMMI process—that your team has completed (green) in the previous sprints, or that are still in progress (blue).



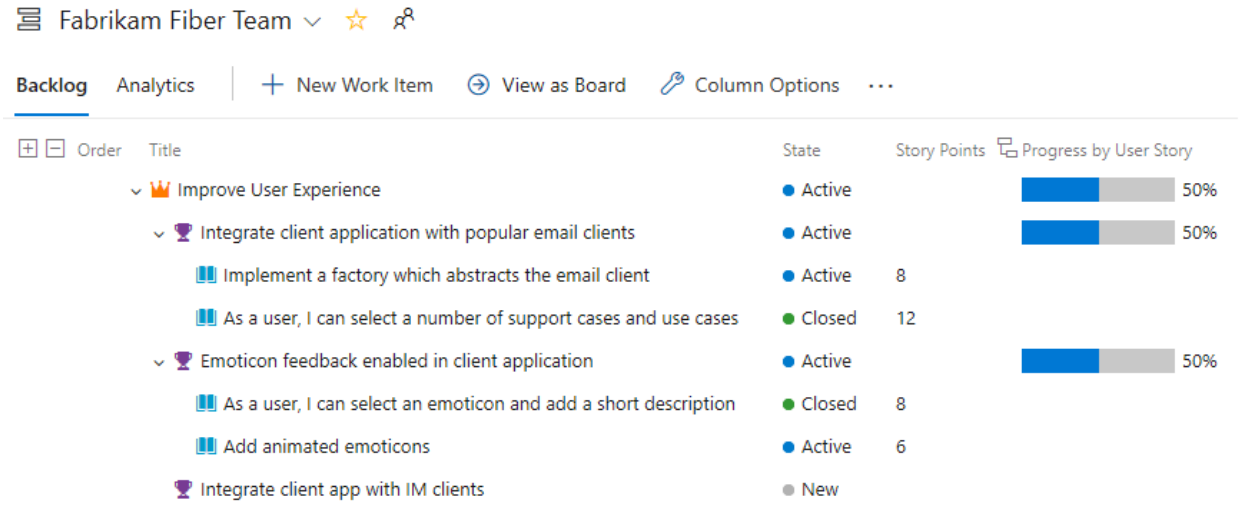
To learn more, see the following articles:

- View or configure team velocity
- Velocity metrics and usage guidance

Display rollup progress, counts, or totals

Product and portfolio backlogs support the display of rollup columns. You can add one or more rollup columns to display progress bars, counts of descendant work items, or totals of select numeric fields. Your settings persist for each page you customize and are only valid for your views.

Progress bars indicate the percentage of descendant items within a hierarchy that are closed or completed. *Counts* display the total number of descendant items. And, *Totals* provide a sum of numeric fields, such as Effort, Story Points, Completed Work, or Remaining Work of descendant items.
For example, progress bars are shown here for a portfolio backlog.



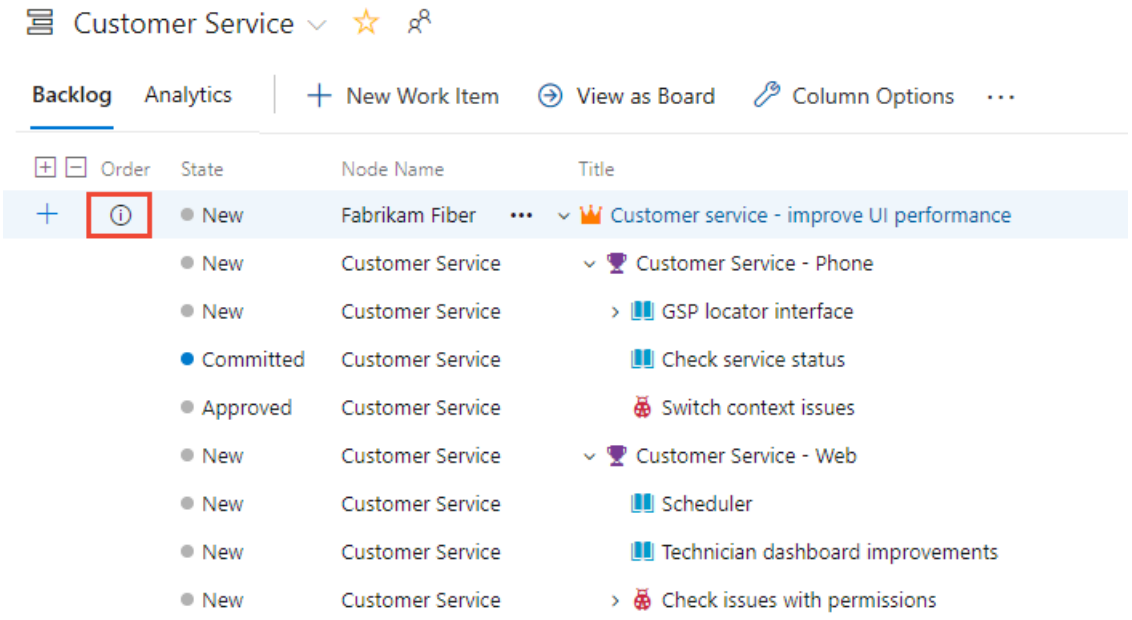
Work with multi-team ownership of backlog items

When you have several teams, your hierarchical views may show items that belong to other teams.

View backlog items and parent items owned by other teams

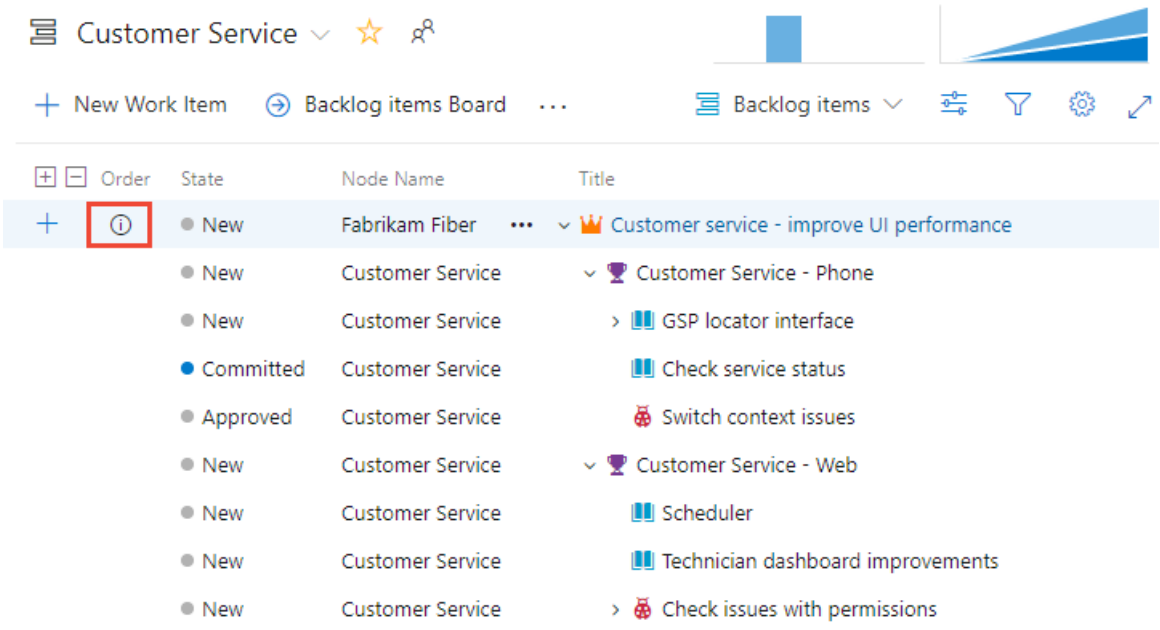
Your team's product backlog lists only those items whose area path matches items assigned to your team. However, if you show parents, you'll see the parent epic of the features and backlog items, even if the epic or feature is owned by another team.

Items that are owned by other teams appear with an information icon ⓘ .



> **TIP**
> Add the **Node Name** field as a column to identify the area path/team associated with the work items.

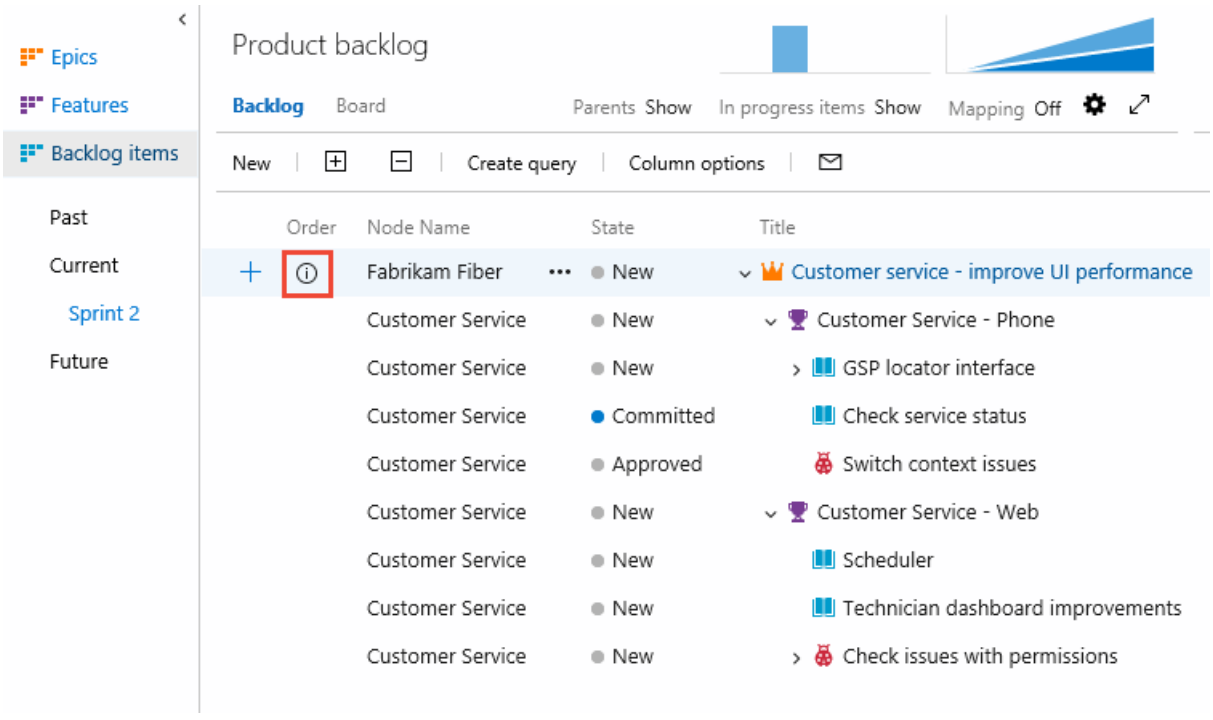
Items that are owned by other teams appear with an information icon ⓘ .



TIP

Add the **Node Name** field as a column to identify the area path/team associated with the work items.

Items that are owned by other teams appear with an information icon ⓘ .



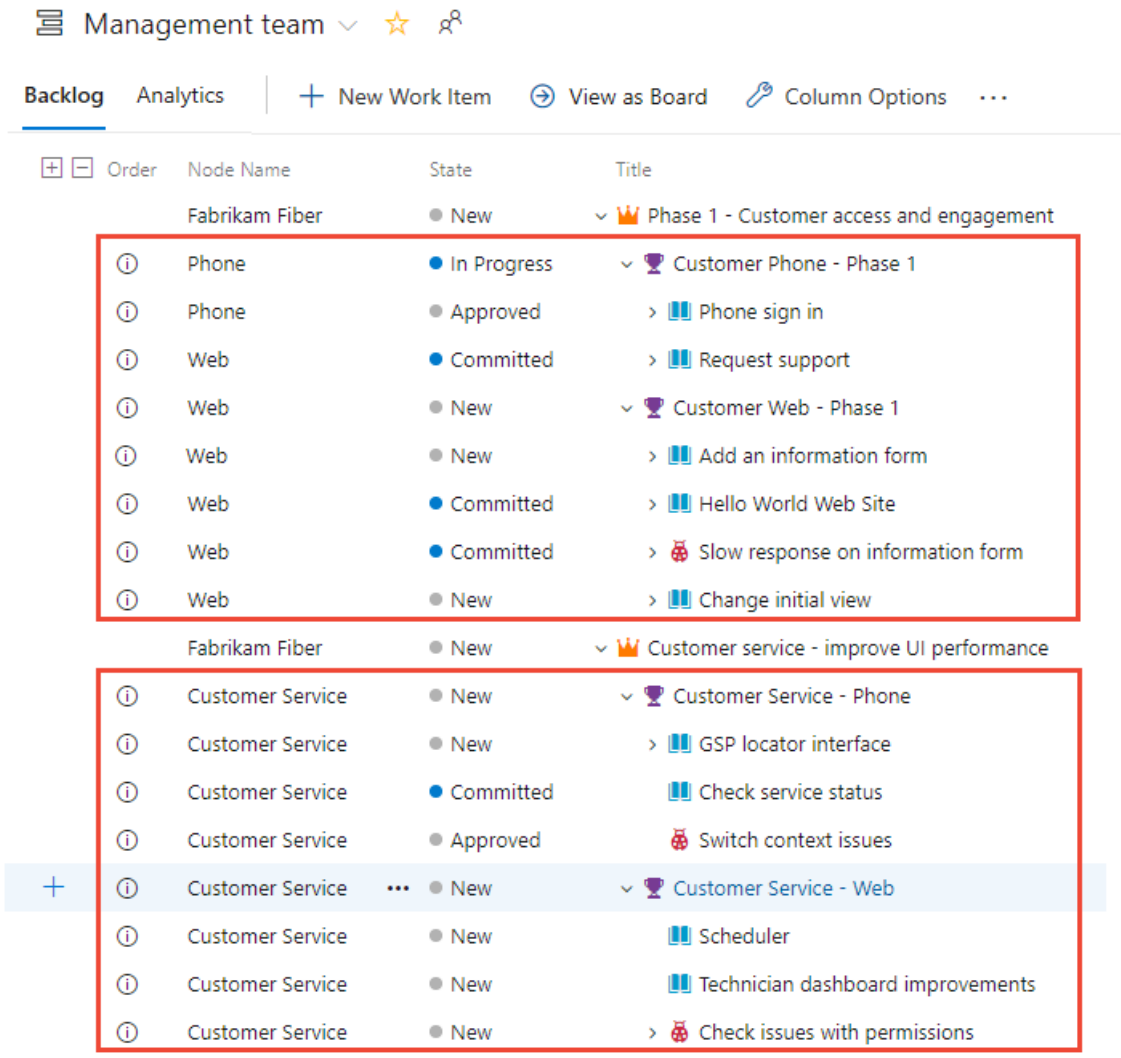
TIP

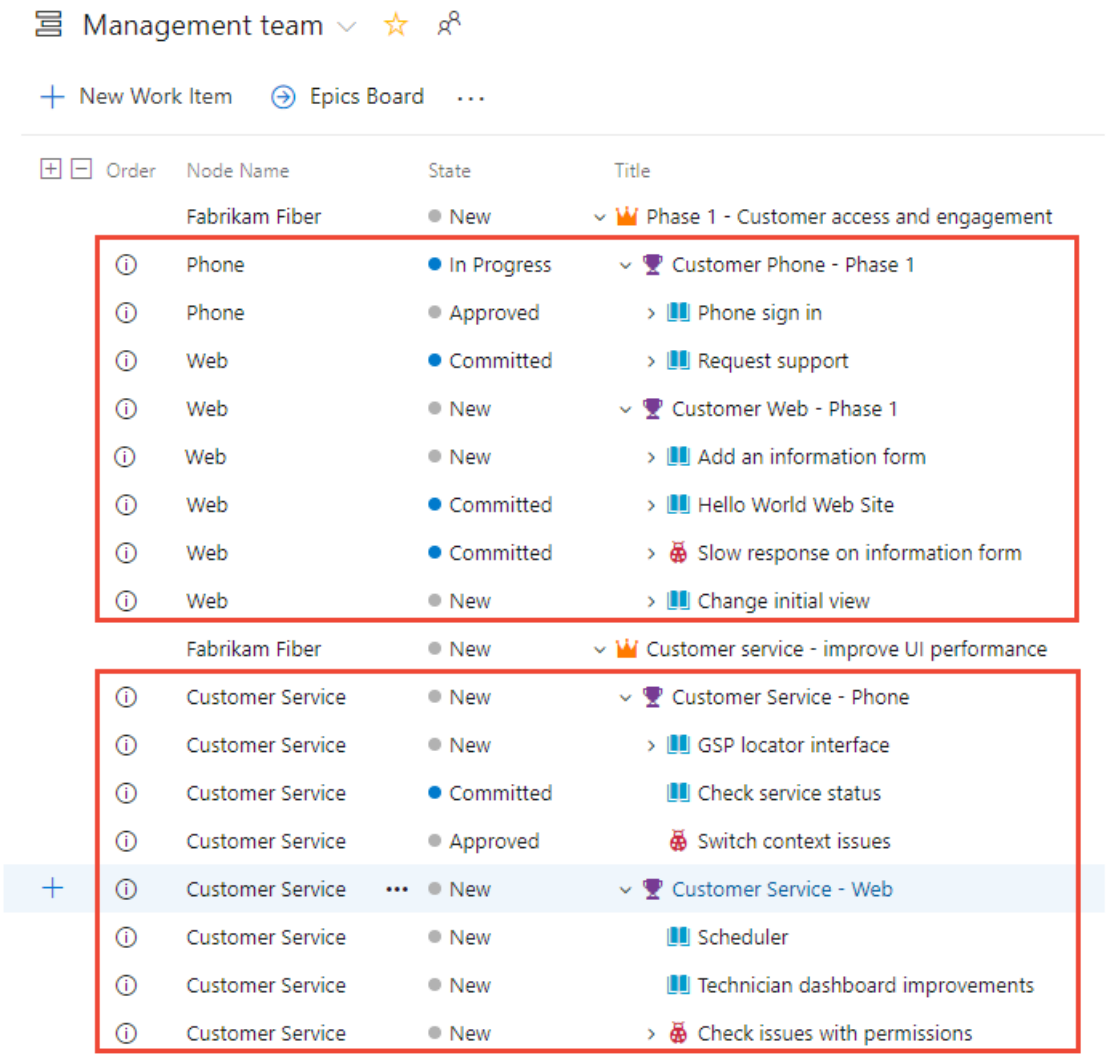
Add the **Node Name** field as a column to identify the area path/team associated with the work items.

For details, see Define area paths and assign to a team.

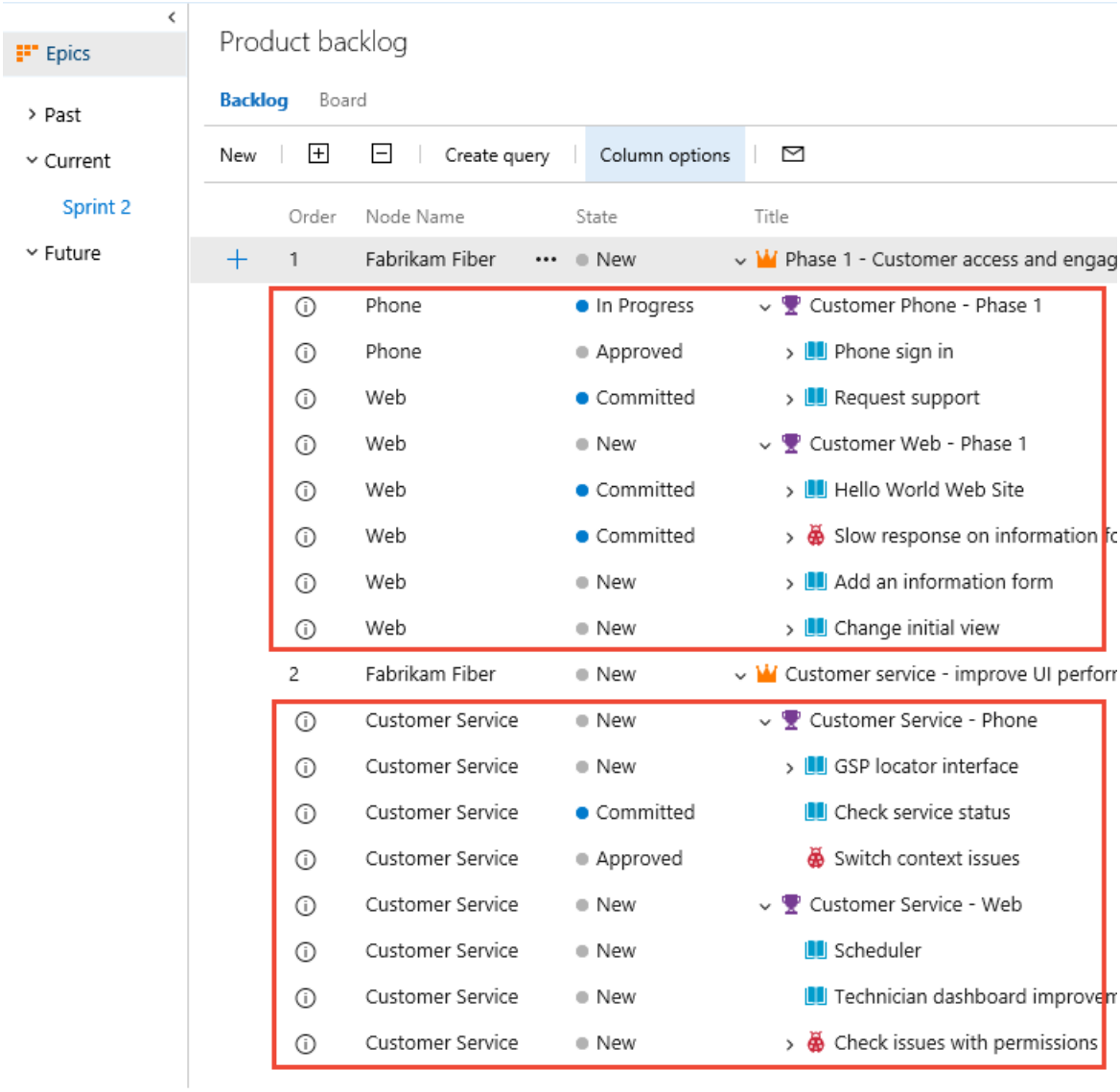
View epics and child items owned by other teams

Here's another example that shows the **Epics** backlog for the **Management** team. Drilling down, you can see all the backlog items and features, even though they belong to one of three different teams: Customer Service, Phone, and Web.


Management team
Backlog
Analytics
New Work Item
View as Board
Column Options
Order
Node Name
State
Title
Fabrikam Fiber
New
Phase 1 - Customer access and engagement
Phone
In Progress
Customer Phone - Phase 1
Phone
Approved
Phone sign in
Web
Committed
Request support
Web
New
Customer Web - Phase 1
Web
New
Add an information form
Web
Committed
Hello World Web Site
Web
Committed
Slow response on information form
Web
New
Change initial view
Fabrikam Fiber
New
Customer service - improve UI performance
Customer Service
New
Customer Service - Phone
Customer Service
New
GSP locator interface
Customer Service
Committed
Check service status
Customer Service
Approved
Switch context issues
Customer Service
New
Customer Service - Web
Customer Service
New
Scheduler
Customer Service
New
Technician dashboard improvements
Customer Service
New
Check issues with permissions



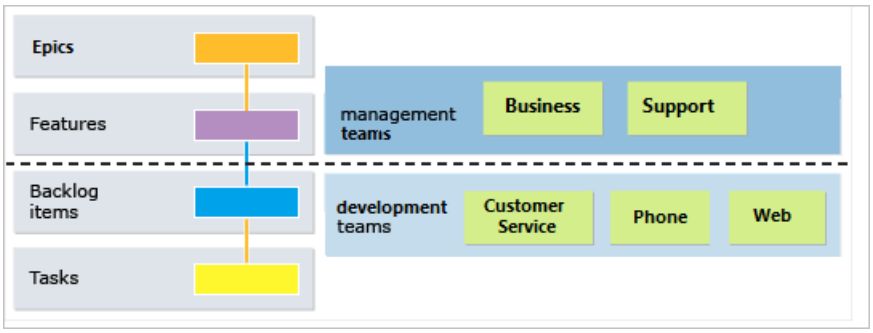
Here's another example that shows the **Epics** backlog for the **Management** team. Drilling down, you can see all the backlog items and features, even though they belong to one of three different teams: Customer Service, Phone, and Web.



From these views, you can re-parent items, both those that you own and those owned by other teams. However, you can't reorder items that another team owns.

This organization enables management teams to focus on high-level features and epics, and development teams to focus on just the backlog items they're responsible to deliver.

To make this work for you, you'll need to add teams and set their area paths. For example, you can create a team structure similar to this one with two management and three development teams.

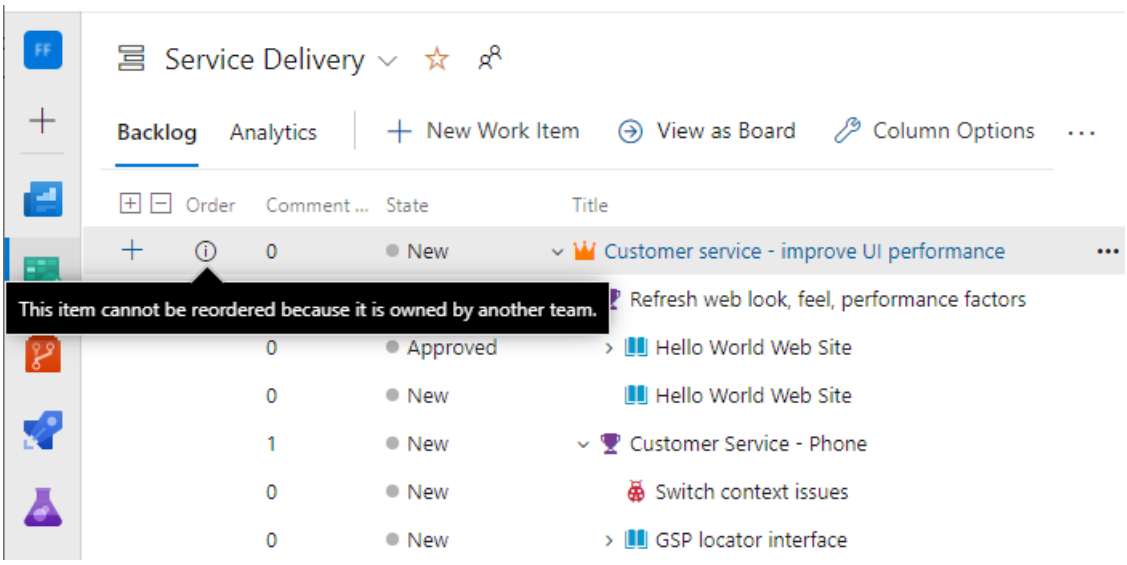


To learn more about hierarchical team and backlog structures, see Portfolio management.

Reordering and reparenting work items

All backlogs and boards support dragging to reorder and reparent work items. Updates made to one team's backlogs and boards are reflected in other team backlogs and boards that share the same area path. You may need to refresh the page to view the changes.

You can only use dragging to reorder or reparent work items assigned to area paths selected for your team. When the **Parents** view option is enabled, work items may appear on your backlog that your team doesn't own. Anything that appears with the ⓘ information icon can't be reordered nor reparented as it's owned by another team.



Display leaf node work items

For TFS 2018 and earlier versions, the Kanban board only shows the leaf node with nested items of a same-category hierarchy. For all versions, sprint backlogs and taskboards only show the last node in a same-category hierarchy, called the leaf node.

While you can create a hierarchy of backlog items, tasks, and bugs—we don't recommend that you create same-category hierarchies. That is, don't create parent-child links among work items of the same type, such as story-story, bug-bug, task-task. The reason is that the display of the leaf node, the last node in a same-category hierarchy, may only appear on Kanban boards, sprint backlogs, and taskboards. For example, if you link items within a same-category hierarchy that is four levels deep, only the items at the fourth level appear on the Kanban board, sprint backlog, and taskboard.

Instead of nesting requirements, bugs, and tasks, we recommend that you maintain a flat list. In other words, only create parent-child links one level deep between items that belong to a different category. To learn more, see Fix re-ordering and nesting issues, How backlogs and boards display hierarchical (nested) items.

Permissions and access

As a member added to the Contributors group of a project, you can use most features provided under **Boards** or **Work**. Users with Basic access have full access to all features. Users with Stakeholder access are limited to certain features. For details, see Stakeholder access quick reference.

To learn more about permissions and access, see Permissions and access for work tracking and Stakeholder access quick reference.

To add users to a project, see Add users to a project or team.

Add portfolio backlog levels and boards

If you need more than two portfolio backlogs, you can add up to two more for a total of five backlog levels.

You can add them by customizing your process, adding new work item types, and then configuring your backlogs and boards. You can also add or modify the fields defined for a work item type (WIT) or add a custom

WIT. For details, see Customize an inheritance process and Customize your backlogs or boards (Inheritance process).

You can add them by either customizing your inherited process or modifying your XML definition files. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. To learn how, see the following articles based on the process model used to update your project:

Inheritance process model:

- Customize your backlogs or boards for a process.
- Customize an inheritance process.

On-premises XML process model:

- Customize work tracking
- Add portfolio backlogs
- Customize the On-premises XML process model

You can add them by defining additional work item types and then customizing your process configuration. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. To learn more, see Customize the On-premises XML process model and Add a portfolio backlog level.

Try this next

If you're just getting started, see Start using Azure Boards.

Related articles

- Web portal navigation
- About Kanban and Agile project management
- About work items
- What is Agile?
- What is Agile development?
- Agile culture

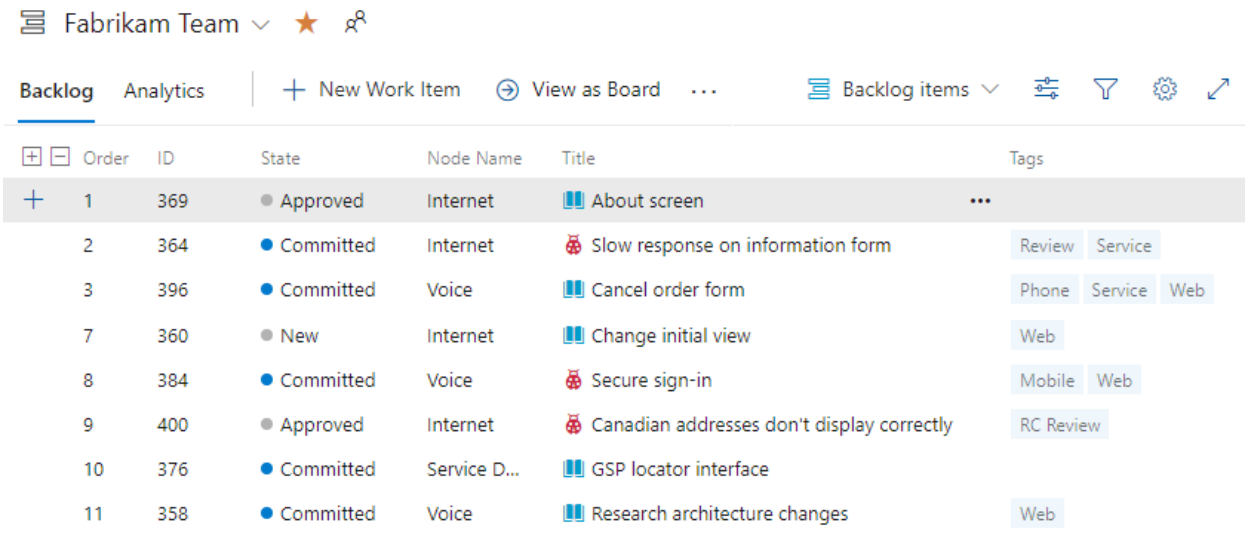
Create your product backlog in Azure Boards

10/4/2022 • 14 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Your product backlog corresponds to your project plan, the roadmap for what your team plans to deliver. You create your product backlog by adding user stories, backlog items, or requirements. As shown in the following image, your backlog consists of a flat list of work items.

> **NOTE**
> The following image illustrates the product backlog image for a Scrum process for Azure DevOps Services. For the Agile, Basic, and CMMI process models, the **Backlog items** selection appears as **Stories**, **Issues**, and **Requirements**.



After you define it, you have a prioritized list of features and requirements to build. Your backlog also provides a repository of the information you need to track and share with your team. And, you can interactively filter the backlog to focus on a subset of work items.

> **NOTE**
> For guidance on configuring and customizing your project and teams to support your business needs, review Configuration and customization of Azure Boards.

Your backlog consists of a list of work items. You use work items to share information, assign work to team members, track dependencies, organize work, and more. Because the most important work appears at the top of the list, your team always knows what to work on next.

> **NOTE**
> Your product backlog is one of three classes of backlogs available to you. For an overview of the features supported on each backlog and the two types of boards, see Backlogs, boards, and plans. If you're not seeing the work items you expect on your backlog, review Setup your backlogs and boards.

Add a backlog

If you have a project, you have a backlog. Each project defines a default team and set of backlogs for that team. You only need to add a backlog when you want to support a new team. When you add a team, you add various team assets. A team admin can configure the assets to support the way the team works. To add a set of backlogs to support a new team, see Add a team.

Each team's set of backlogs are associated with one or more work item types. The work item type associated with a backlog depends on the:

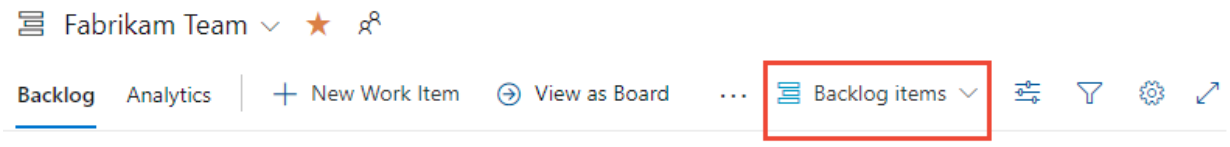
- Process selected at project creation
- Team configurations
- Process customizations

The backlogs defined for each default process are:

- **Agile**: **Stories**, **Features**, and **Epics**
- **Basic**: **Issues** and **Epics**
- **Scrum**: **Backlog items**, **Features**, and **Epics**
- **CMMI**: **Requirements**, **Features**, and **Epics**

- **Agile**: **Stories**, **Features**, and **Epics**
- **Scrum**: **Backlog items**, **Features**, and **Epics**
- **CMMI**: **Requirements**, **Features**, and **Epics**

You choose the backlog level from the backlog selector as shown in the following image.



To customize your backlogs with custom work item types, add portfolio backlogs or other supported options. See the following articles, depending on the process your project uses:

- Inherited process model
- On-premises XML process model.

To customize your backlogs to add custom work item types, add portfolio backlogs, or other supported options, see On-premises XML process model.

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your

team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

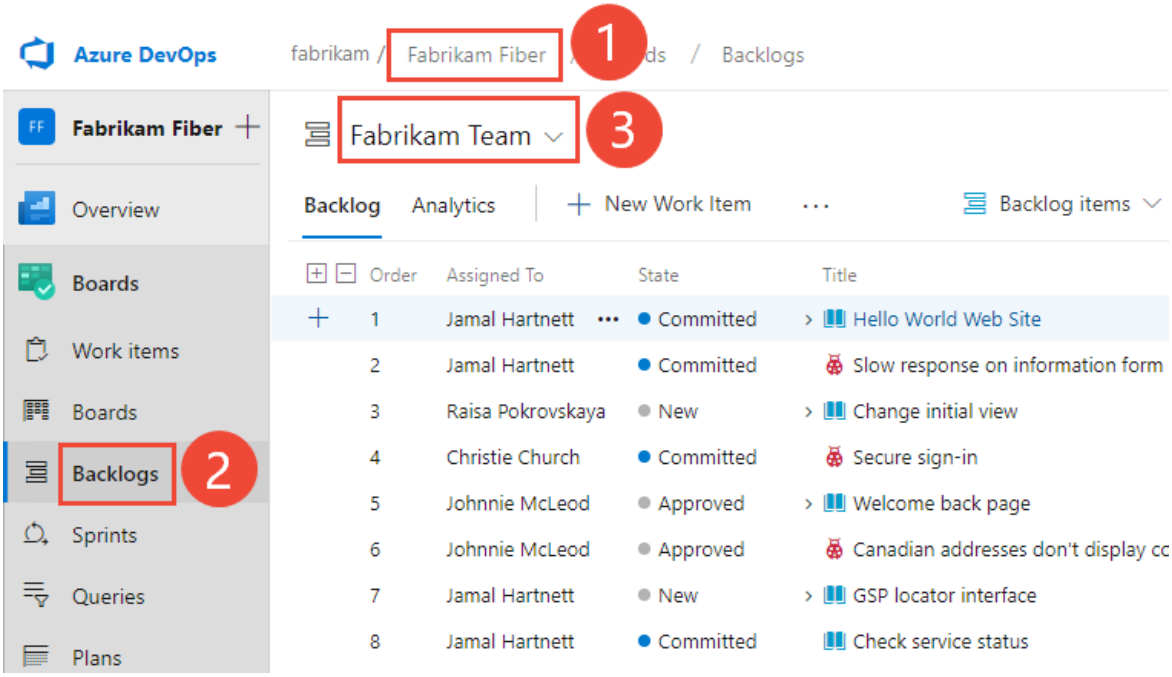
> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

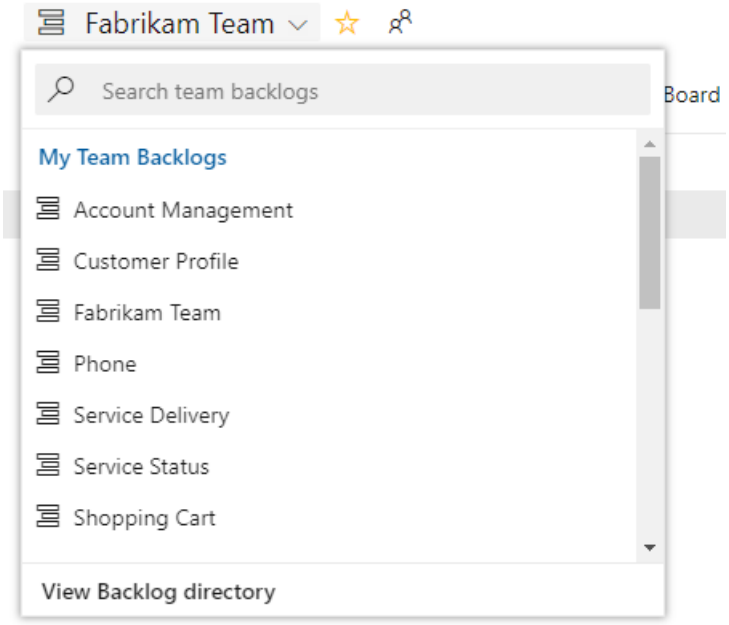
Open your backlog

From your web browser, open your product backlog.

1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



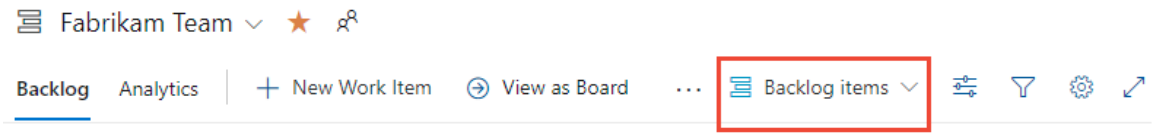
To select another backlog, open the selector and then choose a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.



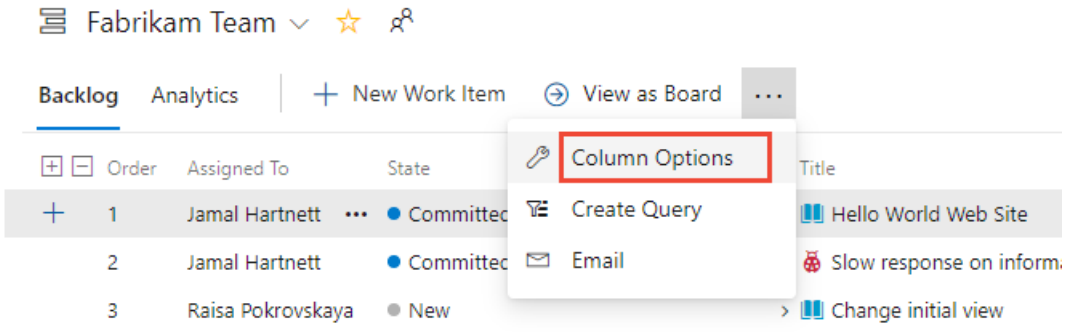
TIP

Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

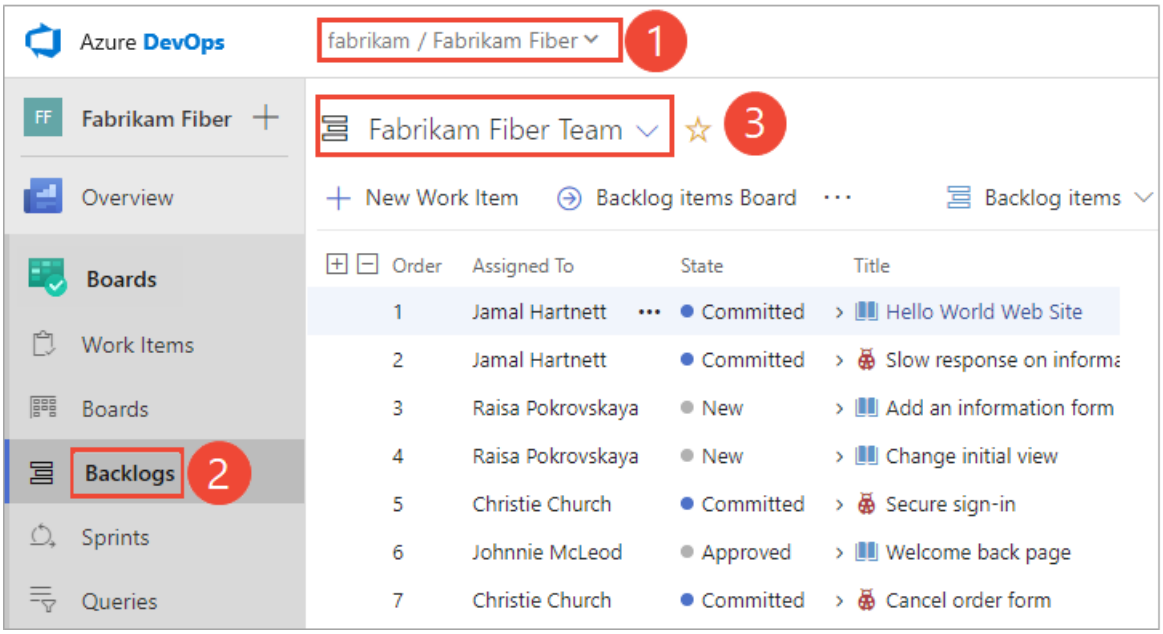
2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



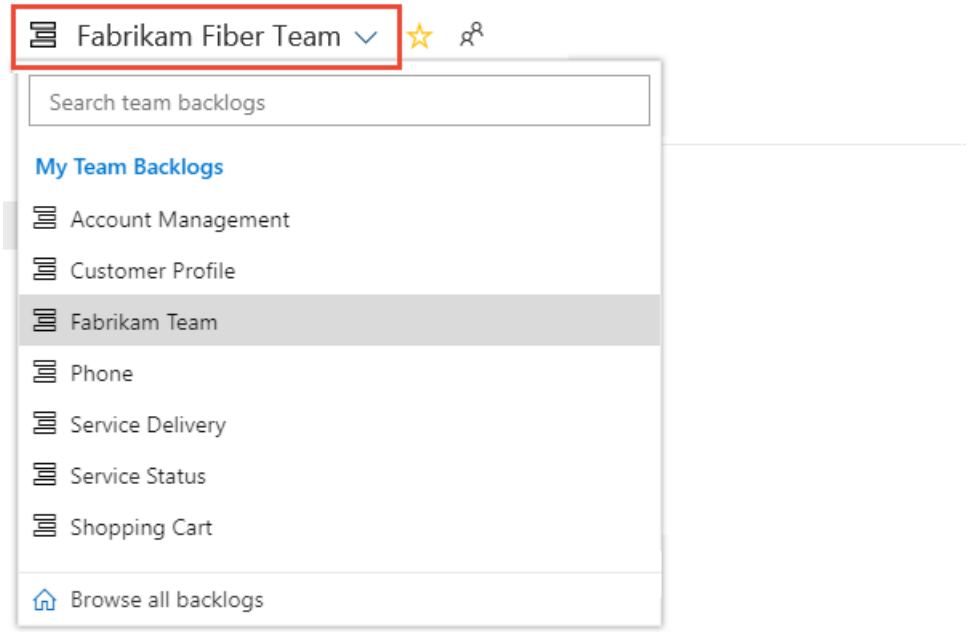
3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.



1. Check that you selected the right project, and select **Boards** > **Backlogs**. Then select the correct team from the team selector menu.



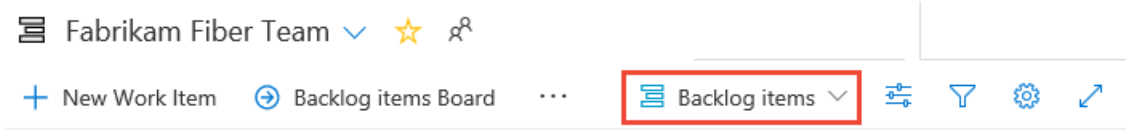
To select another backlog, open the selector and then choose a different team or select the **Browse all backlogs** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.



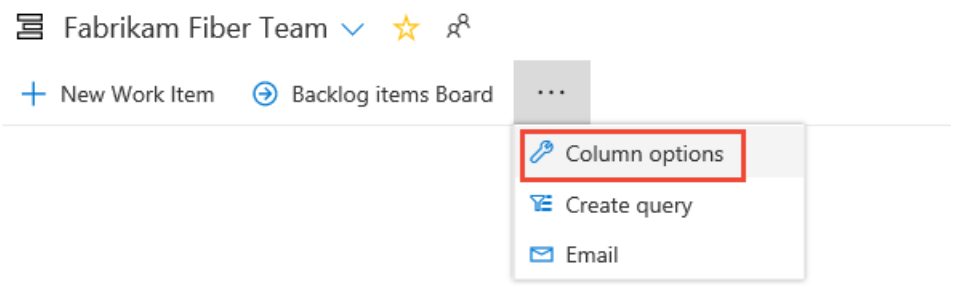
TIP

Select the ☆ star icon to make a team backlog a favorite. Favorite artifacts (★ favorite icon) appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level.



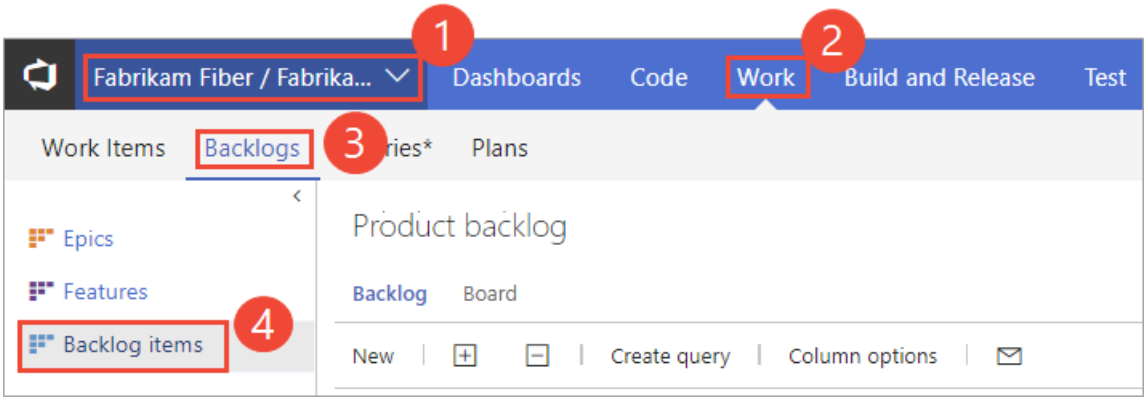
3. (Optional) To select which columns display and in what order, select the ··· actions icon and select **Column options**. To learn more, see Change column options.



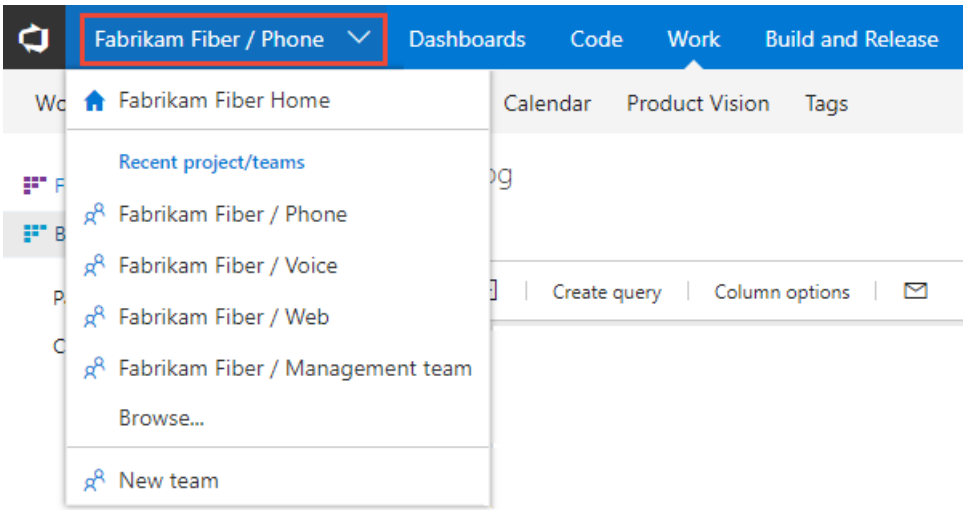
TIP

Each team member has several tools to configure their backlog view: **Expand/Collapse one level**, **Column Options**, **Backlog level selector**, **View options**, and **Filter** toolbar. Options set for each backlog level are distinct and persist until changed. To learn more, see Configure your backlog view.

On your web browser, open your team's product backlog and select the team from the project and team selector. Then select **Work** > **Backlogs**. Select the product backlog, which is **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI.



To select another team, open the project and team selector. Select a different team, or select the **Browse** option.



Track bugs on your backlog

You can choose how you want to manage bugs. Some teams like to track bugs along with requirements on the backlog. Other teams like to track bugs as tasks completed in support of a requirement. The bugs then appear on their taskboard.

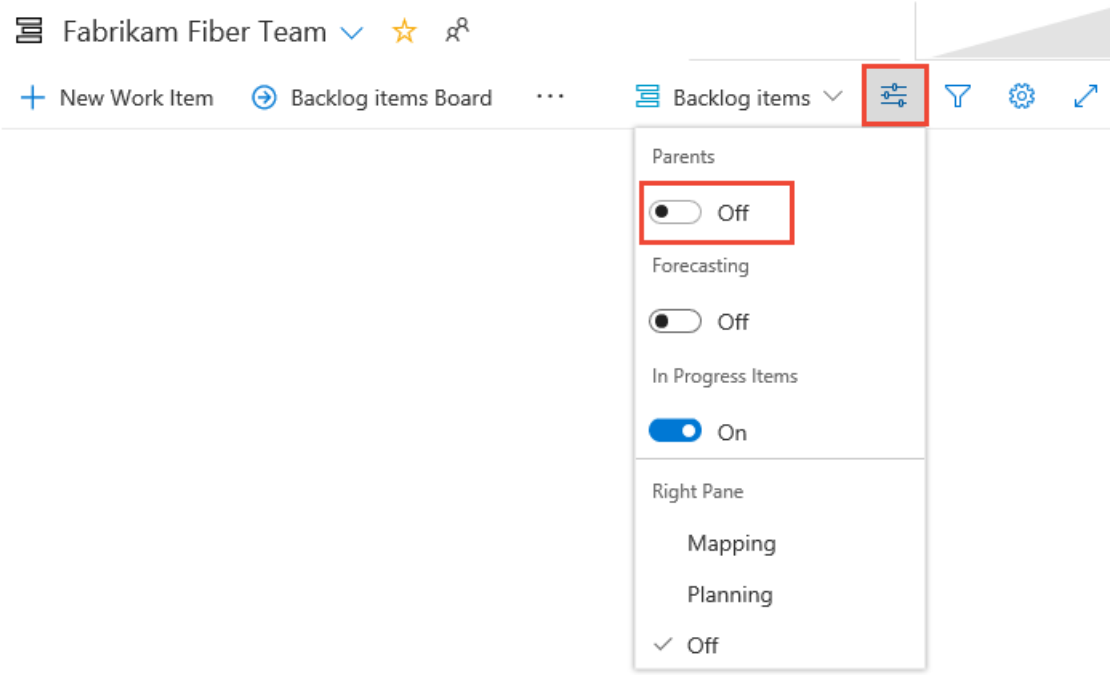
Before deciding, review Configure and customize, Treat bugs as requirements or tasks for guidance. Or, go directly to Show bugs on backlogs and boards.

Convert ideas into backlog items

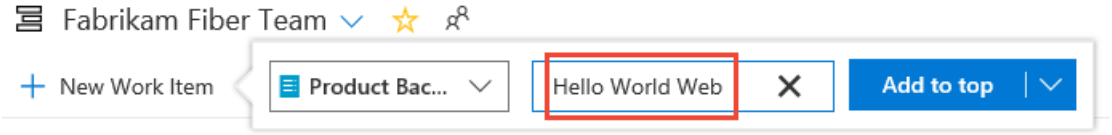
Your backlog shows work that you plan to do or have started to work on. As soon as the **State** of a work item is set to *Done* or *Completed*, the work item no longer shows up on your backlog. You can use the backlog controls to filter or change your view.

> **TIP**
> If you already have defined a long list of items, you don't have to reenter them one at a time. Instead, use Import or update work items in bulk by using CSV files or Microsoft Excel to quickly import them to your backlog.

1. Before you add work items, select **View options** and turn the slider for **Parents** and **Forecasting** to **Off**. Optionally, turn **In Progress Items** on or off.



2. To add a work item, select **New Work Item** and enter a title. Then press Enter or select **Add to top**. Work items are automatically assigned the default **Area Path** and **Iteration Path** selected for the team. To learn more, see Configure team settings.

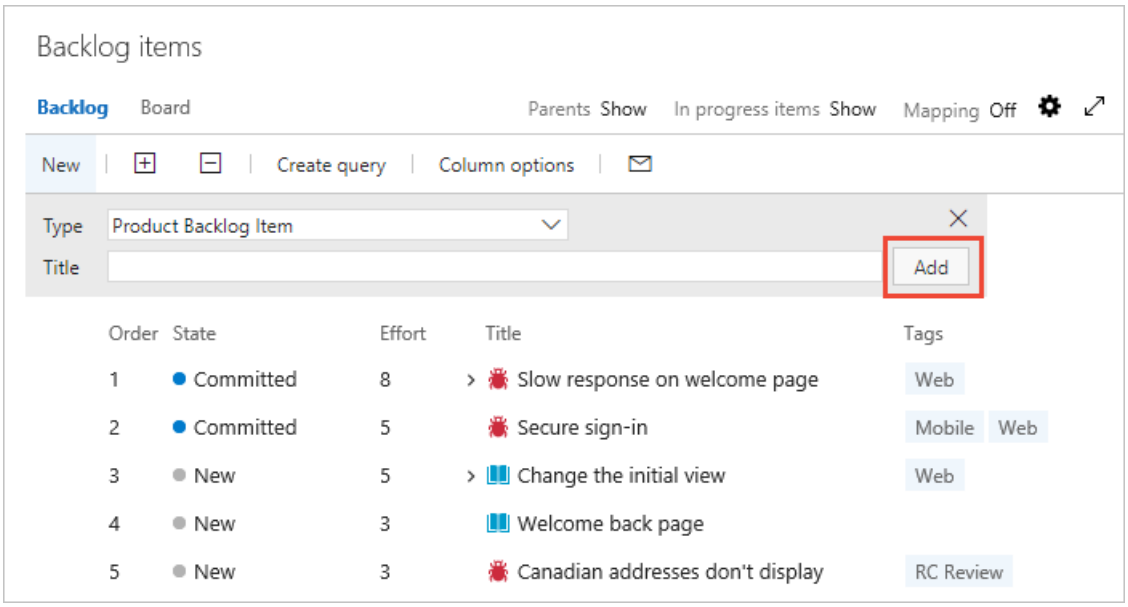


> **NOTE**
> If you have **Stakeholder** access , you can only add work items to the bottom of the backlog. For details, see Stakeholder access quick reference.

3. Repeat this step to capture all your ideas as work items.

To build your backlog, enter a title and select **Add**. If you don't see the **Add** link, select **New** to open the quick

add panel. Optionally, set **In progress items** to **Show** or **Hide**. Work items are automatically assigned the default **Area Path** and **Iteration Path** selected for the team. To learn more, see Configure team settings.



> **NOTE**
> If you have **Stakeholder** access , you can only add work items to the bottom of the backlog. For details, see Stakeholder access quick reference.

Repeat this step until you capture all your main ideas.

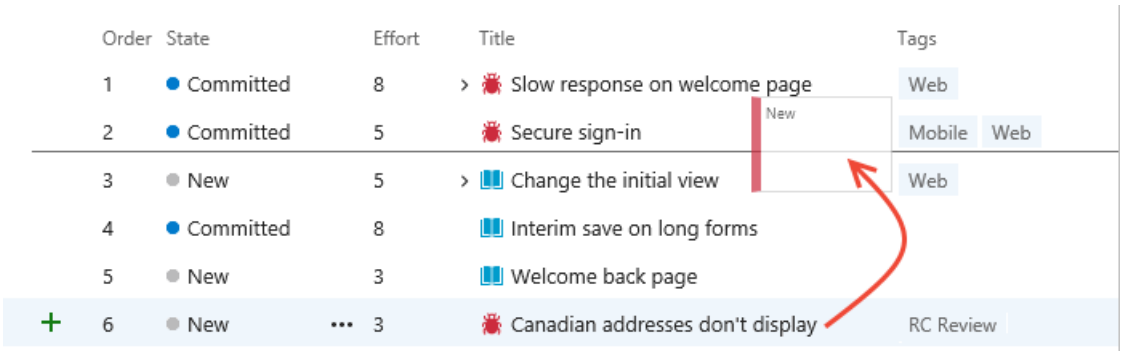
> **NOTE**
> Depending on whether you create your project with Basic, Agile, Scrum, or CMMI, the items in your backlog might be called issues, user stories, PBIs, or requirements. All three are similar. They describe the customer value to be delivered and the work to be performed.
>
> By default, user stories appear on Agile backlogs, issues on Basic backlogs, PBIs and bugs appear on Scrum backlogs, and requirements appear on CMMI backlogs.

Reorder your backlog

After you have some items in your backlog, reorder them to create a prioritized list of work. Review and prioritize your backlog frequently to help your team know what's most important to deliver next.

> **TIP**
> You can't sort your backlog on a column. To view a sorted listed, select **Create query**. Save and open the query, and then sort the query results. To learn more about queries, see Use the query editor to list and manage queries.

To reorder your backlog, drag the work items. Or, if you prefer to use the keyboard, hold down the Alt key and use the up and down arrows.



> **NOTE**
> To reorder a backlog, you must have Basic or higher level access. If you have Stakeholder access, you can't reorder backlog items. For more information, see Stakeholder access quick reference.

Backlogs that participate in portfolio management or that contain nested same-type child items might not allow you to reorder the items. For more information, see these articles:

- Backlogs, portfolios, and Agile project management, Work with multi-team ownership of backlog items
- Fix reordering and nesting issues

Add details and estimates to backlog items

Building and prioritizing your backlog provides a high-level roadmap. Before your team can start work on any item, however, they need more details. Capture the details within the work item form.

To open each item, double-click or press Enter. Then add all the information you want to track. Change one or more field values, add a description, or make a note in the **Discussion** section. You can also choose the **Attachments** tab and drag a file onto it to share the file with others.

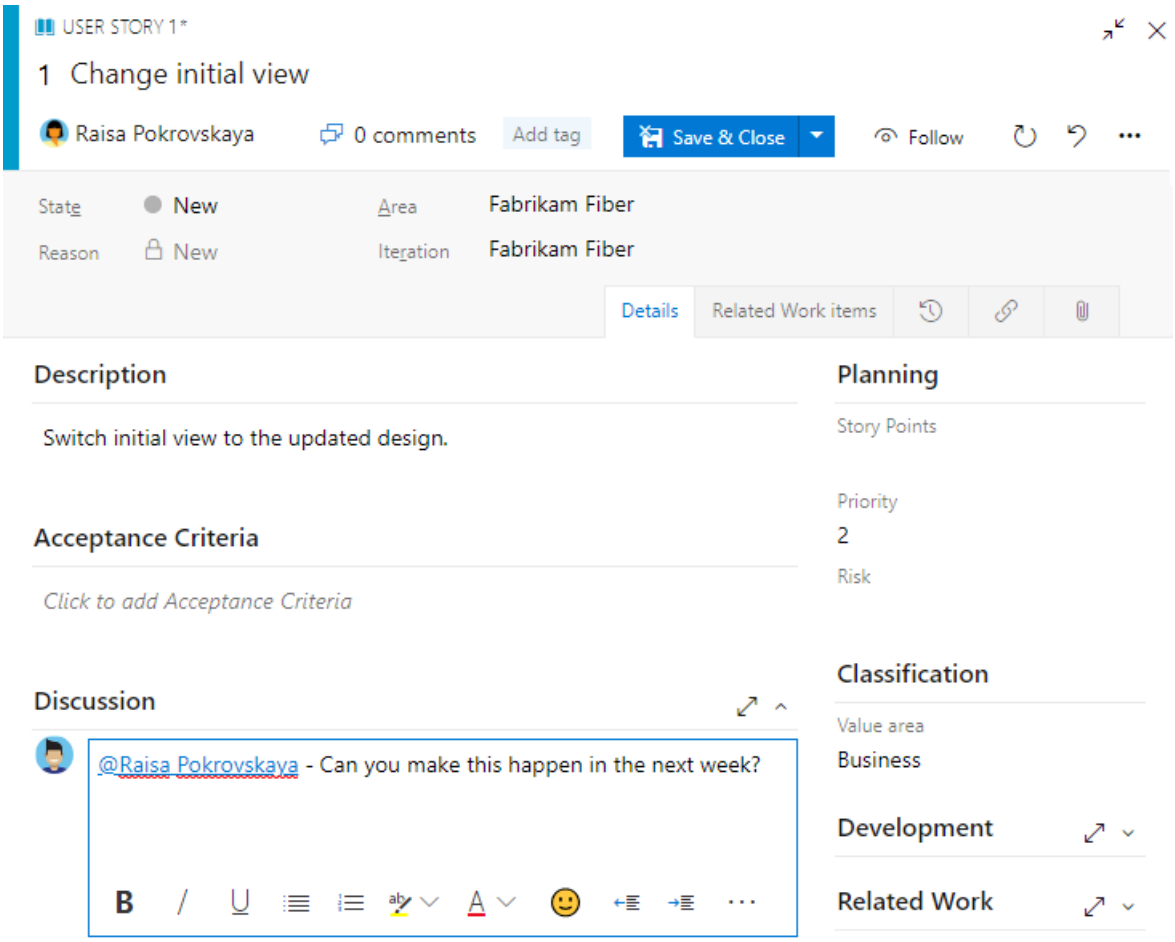
Enter as much detail as the team needs to:

- Understand the scope.
- Estimate the work required.
- Develop tests.
- Ensure that the end product meets acceptance criteria.

> **NOTE**
> You can only assign work to a single user. If you need to assign work to more than one user, add a work item for each user and distinguish the work to be done by title and description. The Assigned To field only accepts user accounts that have been added to a project or team.

- Agile process
- Basic process
- Scrum process
- CMMI process

For example, here we assign the story to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa.



Choose **Save & Close** when done.

> **TIP**
> To plan a sprint, at a minimum, estimate the effort involved to implement each backlog item. To capture effort in the work item form, use **Effort** for Basic or Scrum, **Story Points** for Agile, or **Size** for CMMI.

Field	Usage
Effort, Story Points, Size	Provide a relative estimate of the amount of work required to complete a PBI. For user stories and requirements, you capture estimates in **Story Points** and **Size**. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, and so on). Use any numeric unit of measurement your team prefers. The estimates you set for **Effort**, **Size**, or **Story Points** are used to calculate velocity and forecast sprints.
Business Value	Specify a priority that captures the relative value of a PBI compared to other PBIs. The higher the number, the greater the business value. Use this field when you want to capture a priority separate from the changeable backlog stack ranking.
Description	Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the

user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

Acceptance Criteria

Define what "Done" means by describing the criteria for the team to use to verify whether the PBI or the bug fix is fully implemented.

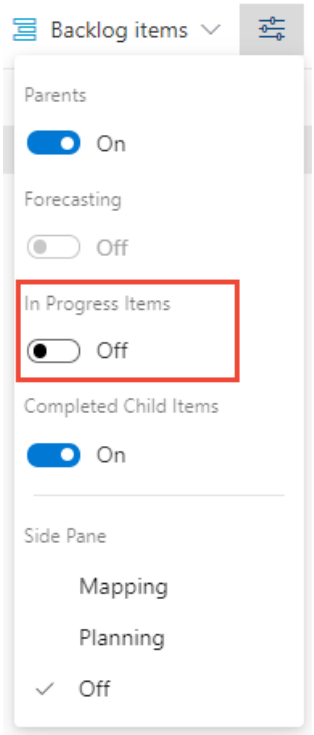
Before work begins on a PBI or bug, describe the criteria for customer acceptance as clearly as possible. Have conversations between the team and customers to determine the acceptance criteria. These criteria help ensure a common understanding within the team to meet customers' expectations. Also, this information provides the basis for acceptance testing.

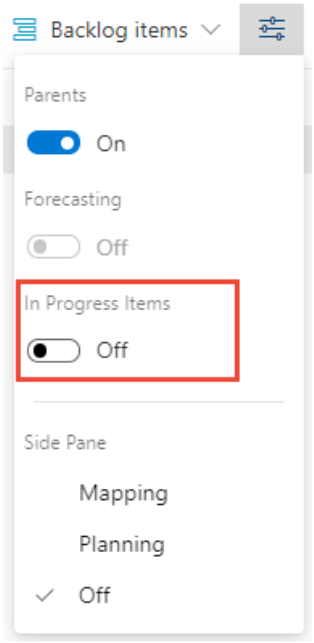
Impact Assessment (CMMI only)

Describes the customer impact of not implementing the requirement. You might include details from the Kano model about whether this requirement is in the surprise, required, or obvious categories.

Show/hide items that are in progress

From the **View options** selector, you can choose to show or hide **In Progress items**. If you turn the **In Progress** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or states that map to the In Progress category state won't appear in the backlog.



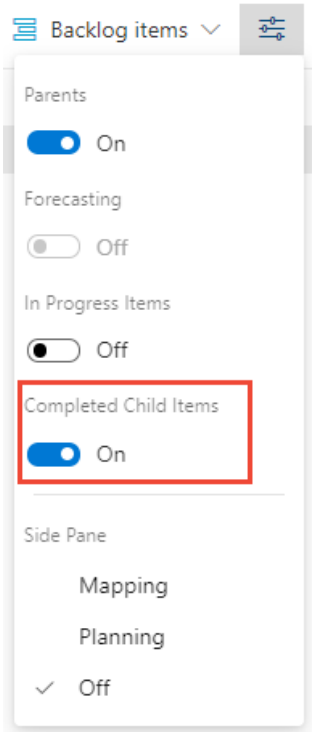


Choose **In progress items** show or hide **In Progress** backlog items. If you turn the **In Progress items** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or states that map to the In Progress category state won't appear in the backlog.

You usually choose to hide **In Progress items** when you want to forecast work. To learn more, see Forecast your product backlog.

Show/hide child items that are complete

From the **View options** selector, you can choose to show or hide **Completed Child items**.



You usually choose to show Completed child items when you want to view rollup columns.

You usually choose to hide Completed child items when you want to forecast work. To learn more, see Forecast your product backlog.

> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than 183 days (about a half a year). You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

> **NOTE**
> Completed or closed work items don't display on the backlogs and boards once their **Changed Date** is greater than a year old. You can still list these items using a query. If you want them to show up on a backlog or board, then you can make a minor change to them which resets the clock.

Try this next

Now that you have a working backlog in place, your team can begin work on the top-priority items. From here, it's time to decide how you want to work as a team. Do you want to use Scrum or Kanban? You can use these methods independently or together.

Scrum: Schedule sprints or Kanban

Teams that want the least overhead for tracking and estimating might prefer Kanban. Teams that like to work at a steady cadence and plot the details of their sprint plan might prefer Scrum and sprint planning.

Related articles

- Configure and customize Azure Boards
- Bulk modify work items
- Copy or clone work items
- Refine your backlog
- Display rollup progress bars or counts
- Product backlog controls
- Interactively filter backlogs, boards, queries, and plans
- Backlog priority or stack rank order
- Keyboard shortcuts

- Bulk modify work items
- Copy or clone work items
- Refine your backlog
- Product backlog controls
- Filter product and portfolio backlogs
- Backlog priority or stack rank order
- Keyboard shortcuts

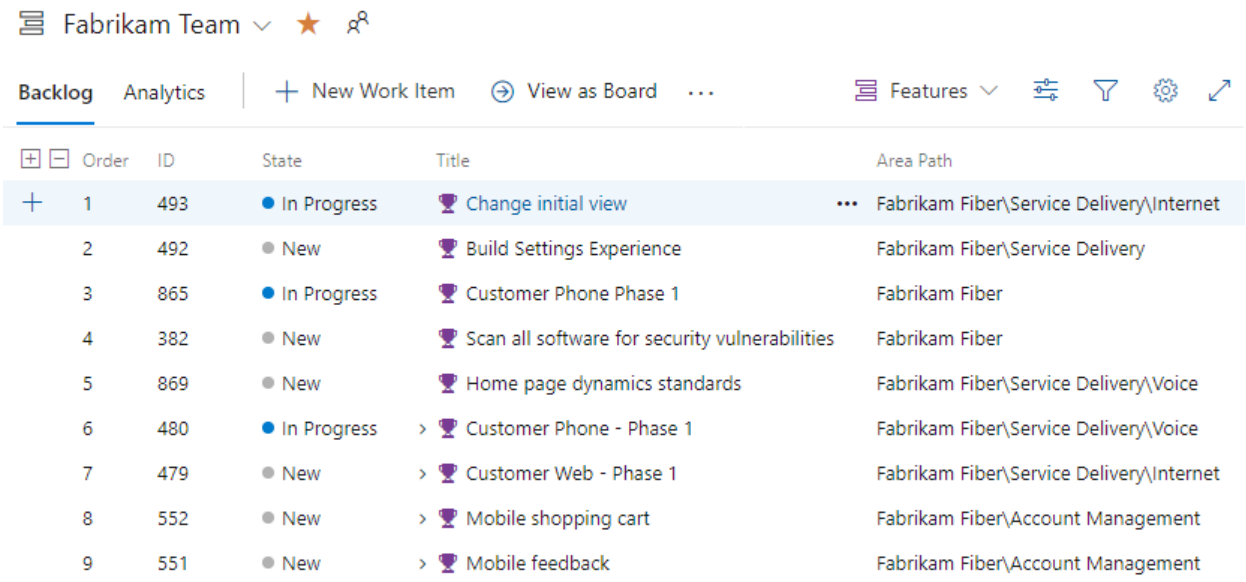
Define features and epics, organize your product and portfolio backlogs in Azure Boards

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

While many teams can work with a flat list of items, sometimes it helps to group related items into a hierarchical structure. Perhaps you like to start with several major features or scenarios and break them down into smaller deliverables. Or, you've got an existing backlog and now need to organize it.

The following image shows a Features portfolio backlog that consists of a flat list of Feature work items.



No matter your starting point, you can use portfolio backlogs to bring more order to your backlog. Use your backlogs to plan your project and to:

- Manage a portfolio of features that are supported by different development and management teams
- Group items into a release train
- Minimize size variability of your deliverables by breaking down a large feature into smaller backlog items

Use this article to learn how to perform these tasks:

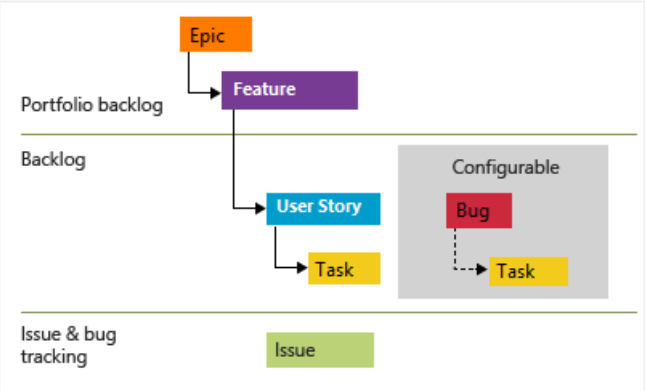
- Determine what is a good feature or epic
- View a backlog or portfolio backlog
- Add features and epics
- Add child items

With portfolio backlogs, you can quickly add and group items into a hierarchy. You can also drill up or down within the hierarchy, reorder and reparent items, and filter hierarchical views. Portfolio backlogs are one of three classes of backlogs available to you. For an overview of the features supported on backlogs and the types of boards, see Backlogs, boards, and plans. To learn how to track progress across teams, see Visibility across teams.

- Agile process
- Basic process
- Scrum process

- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

What makes a feature or epic?

The epics and features that you create should reflect your business focus. As user stories or product backlog items roll up into features, and features roll up into epics—you'll want to name your features and epics with that in mind.

Typically, a feature is a shippable software component. An epic corresponds to a collection of features that are related to one another and combine to form a larger value proposition. As you define your features and epics, consider the time required to complete them. In general, backlog items such as user stories or product backlog items, should be accomplished within a sprint. Features and epics may take one or more sprints to complete. Here are a few examples of each.

Features

- View option enhancements
- Shopping cart support on mobile devices
- Text alert support for login and account changes
- Web portal user interface improvements

Epics

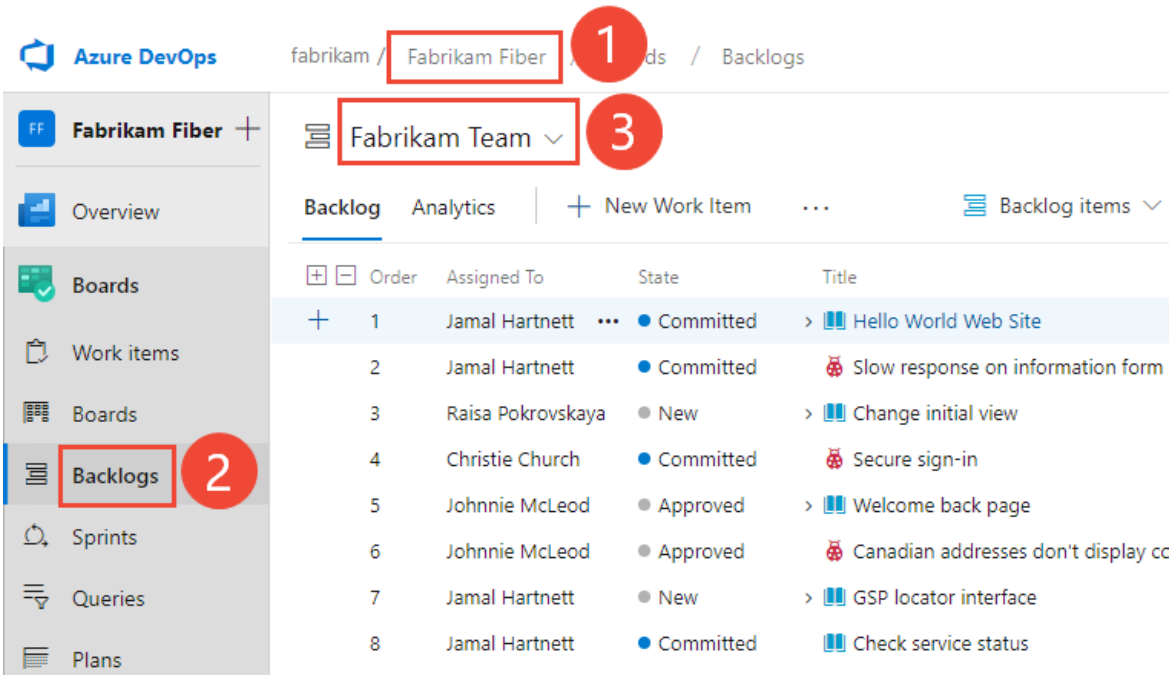
- New user experience complete
- Performance improvements
- Integration with communication apps
- First phase of mobile apps complete

View a backlog or portfolio backlog

To focus on one level of a backlog at a time, choose the name of the backlog you want to view. If you don't see all three backlog levels— Epics, Features, and Backlog items— you can enable them for your team.

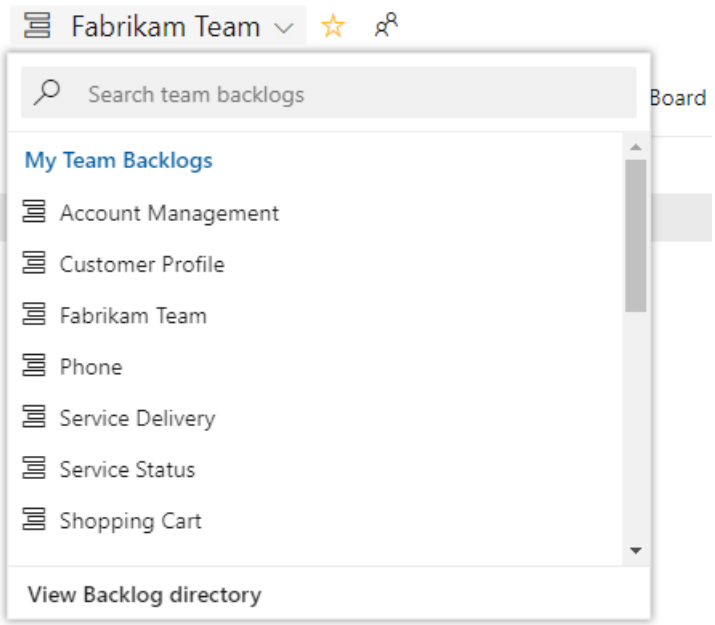
For example, when you choose **Epics**, you'll see a list of all Epics in your team's active area paths. From there, you can drill down to see child features and backlog items.

1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



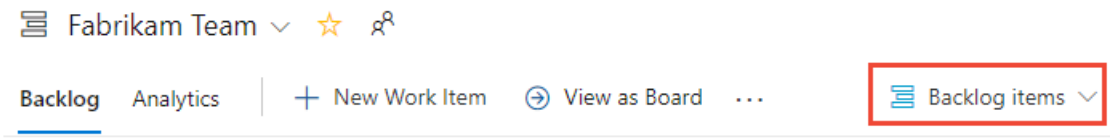
To select another backlog, open the selector and then choose a different team or select the **View Backlog**

directory option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.

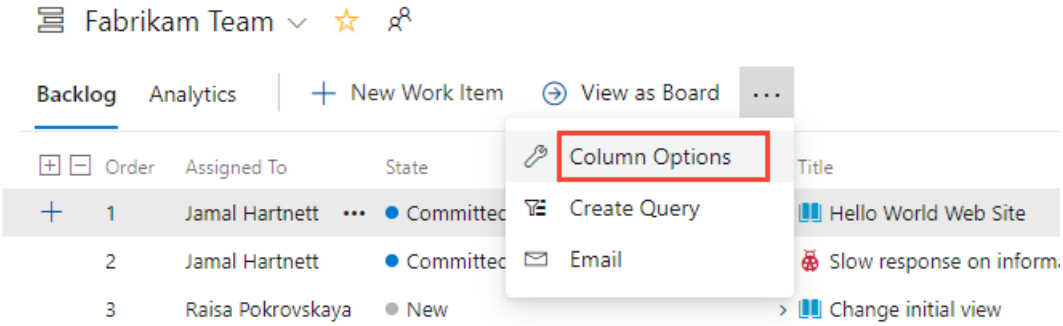


> **TIP**
> Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

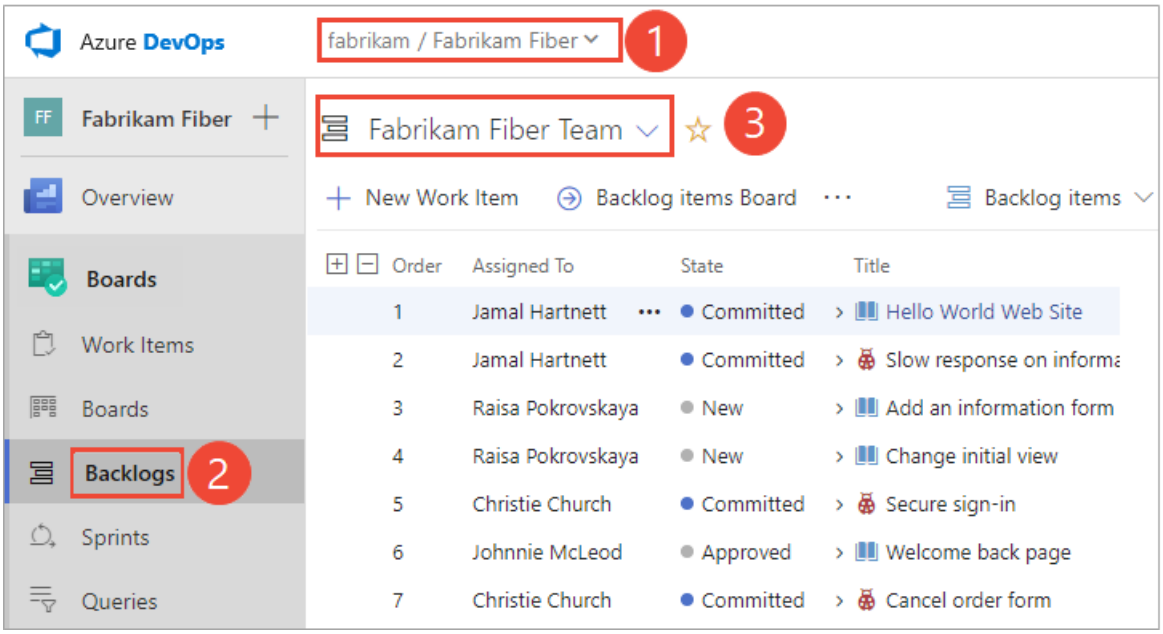
2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



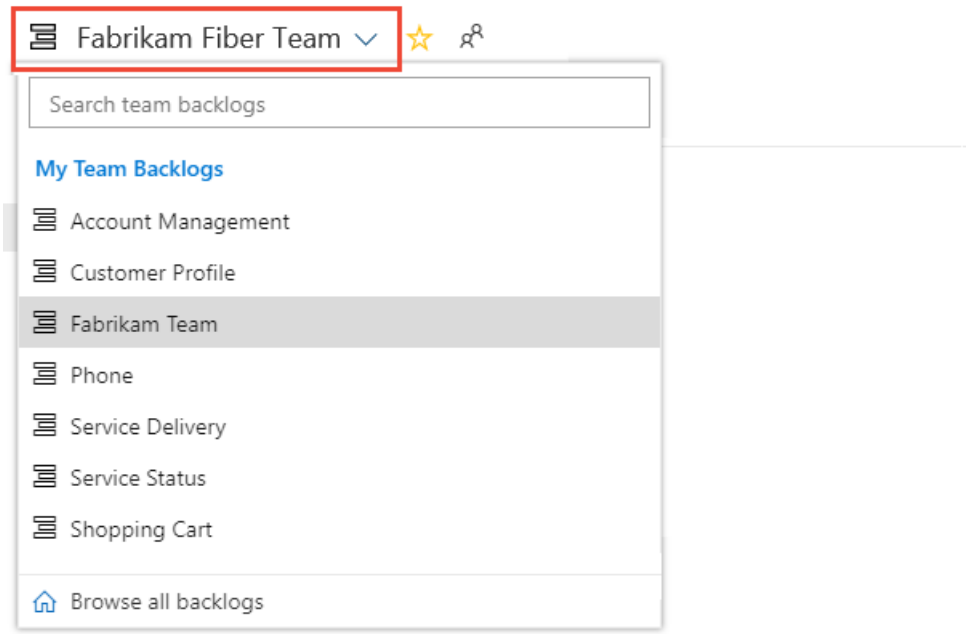
3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.



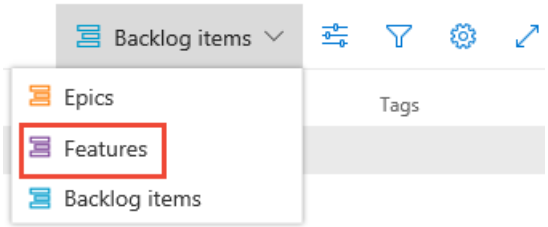
1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



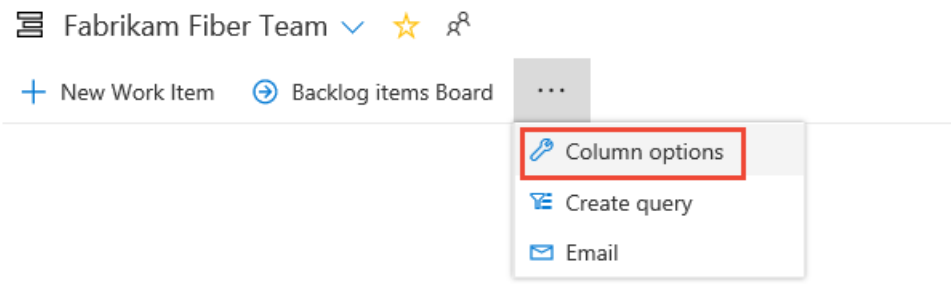
To choose another team, open the selector and select a different team or choose the **Browse all backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



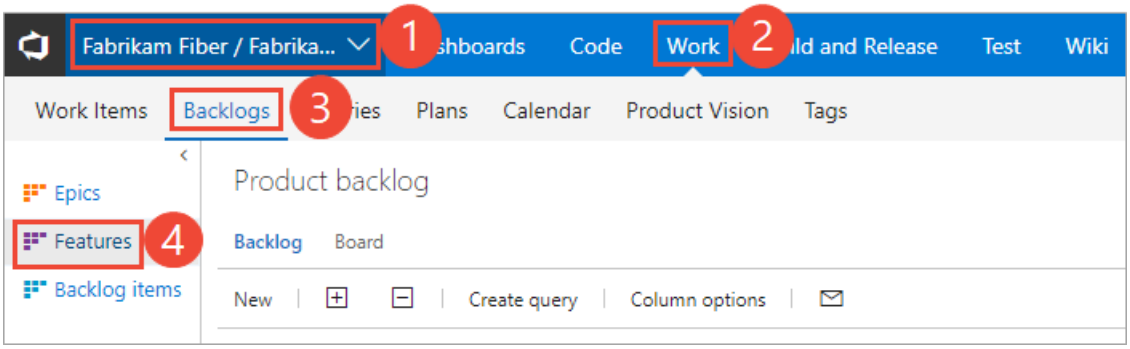
2. Choose the portfolio backlog that you want, **Features**. or **Epics**.



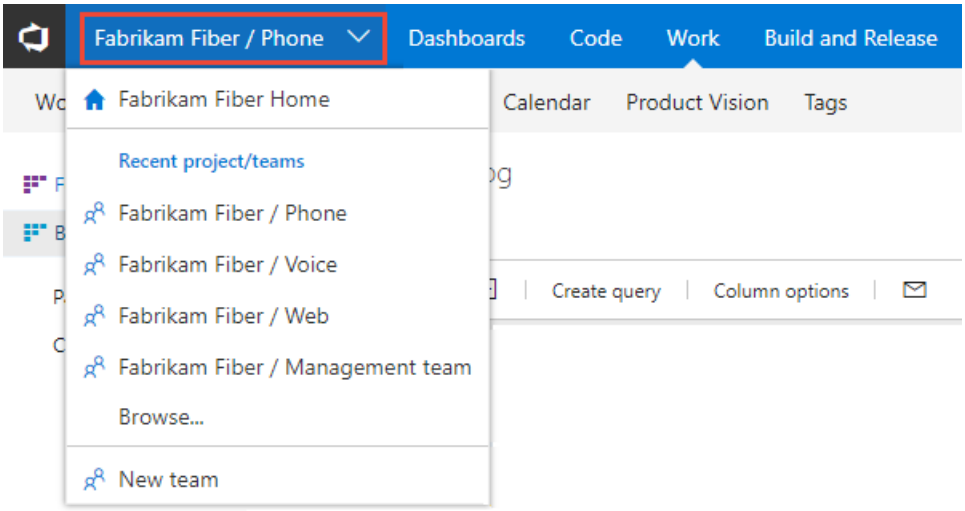
3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.



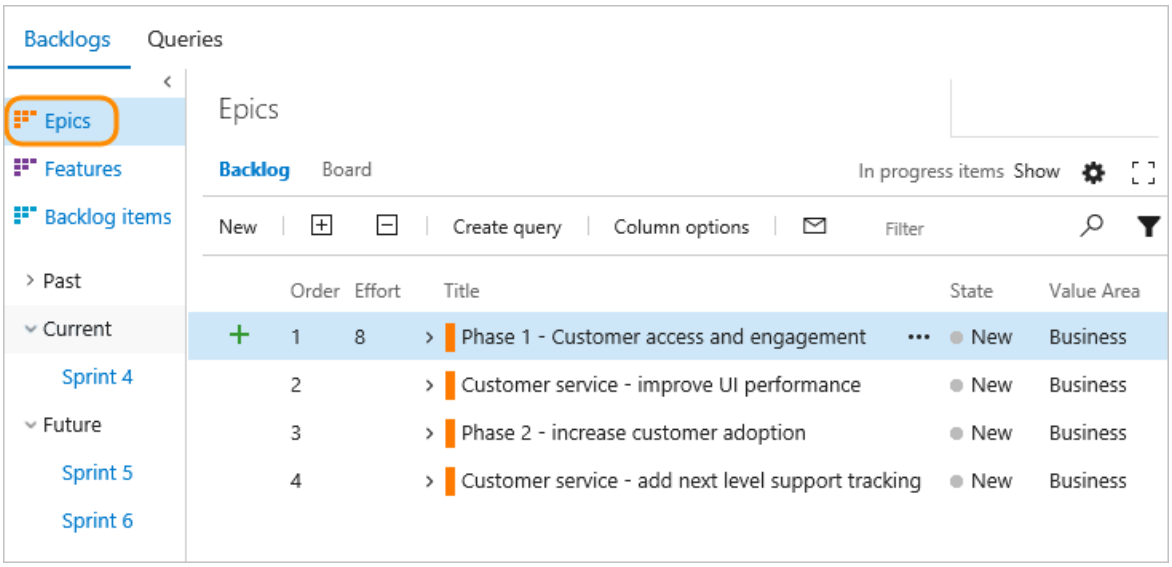
1. From your web browser, open your team's backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the portfolio backlog of interest, which is **Features** or **Epics**.



To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



2. Choose **Epics** to see a list of all epics defined in your team's active area paths.



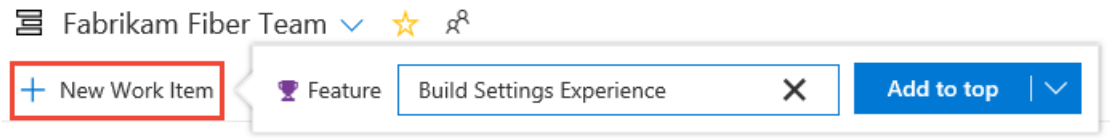
TIP

Each team can choose the backlog levels that are active as described in Select backlog navigation levels for your team.

Add features and epics to your backlog

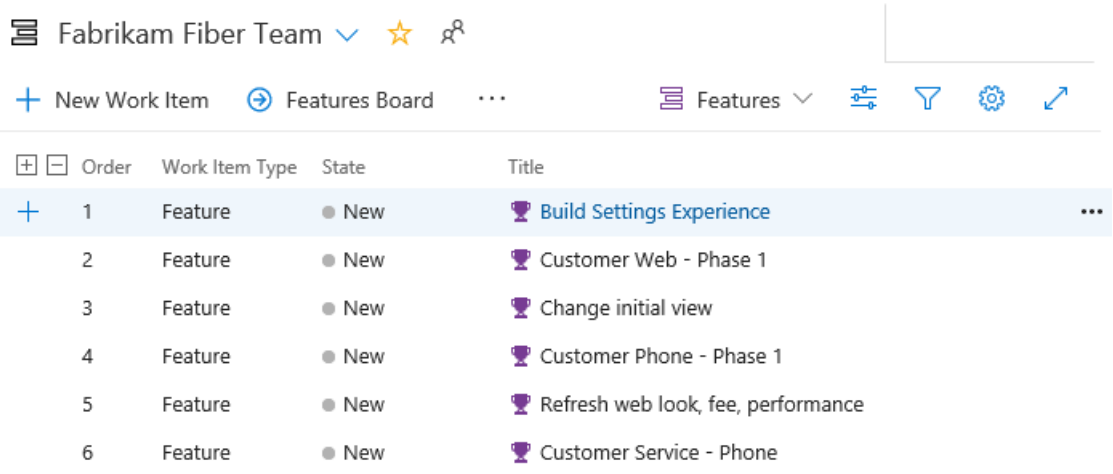
Just as you can add items to your product backlog, you can add items to your features and epics backlogs.

1. To add a feature, choose the **New Work Item**, enter a title, and then press **Enter** or choose **Add to top**.



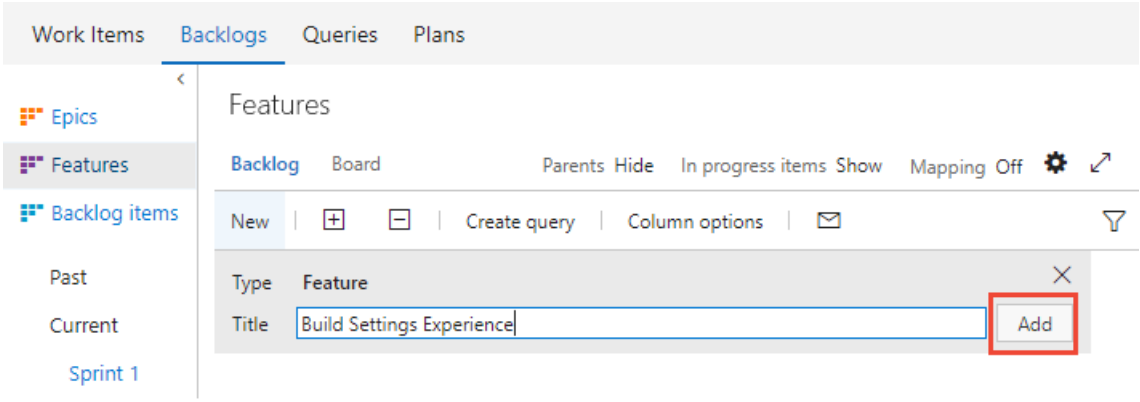
2. Repeat this step to capture all your ideas as work items.

   Here, we've added six features.



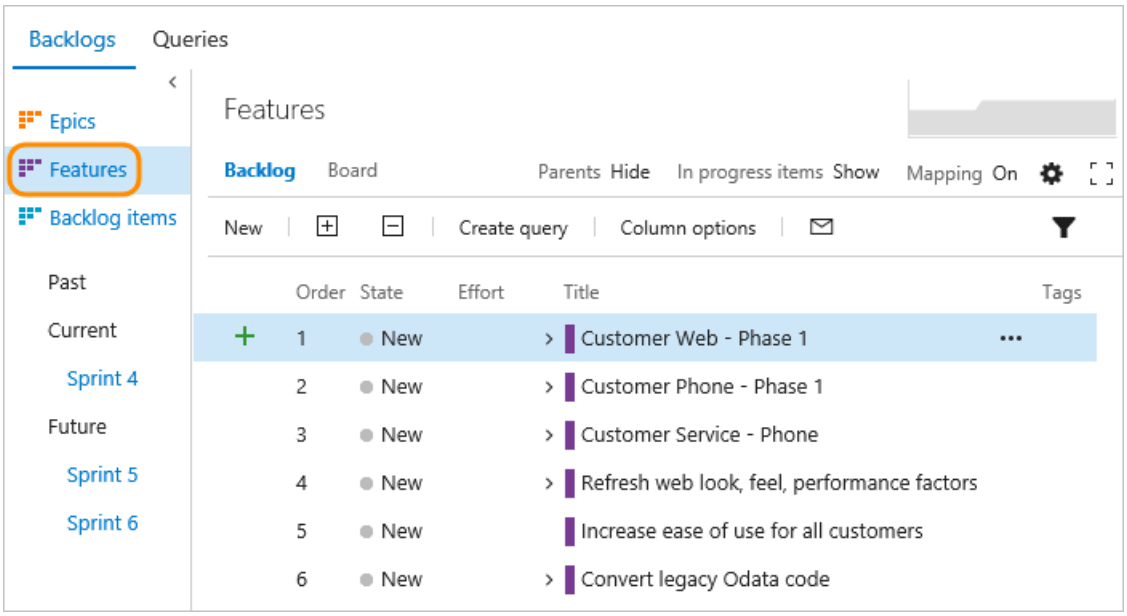
You can add epics in the same way. Open the **Epics** backlog from the backlogs selector.

1. To add a feature, enter a title and choose **Add**. If you don't see the **Add** link, choose **New** to open the quick add panel.



2. Repeat this step until you've captured all your main ideas.

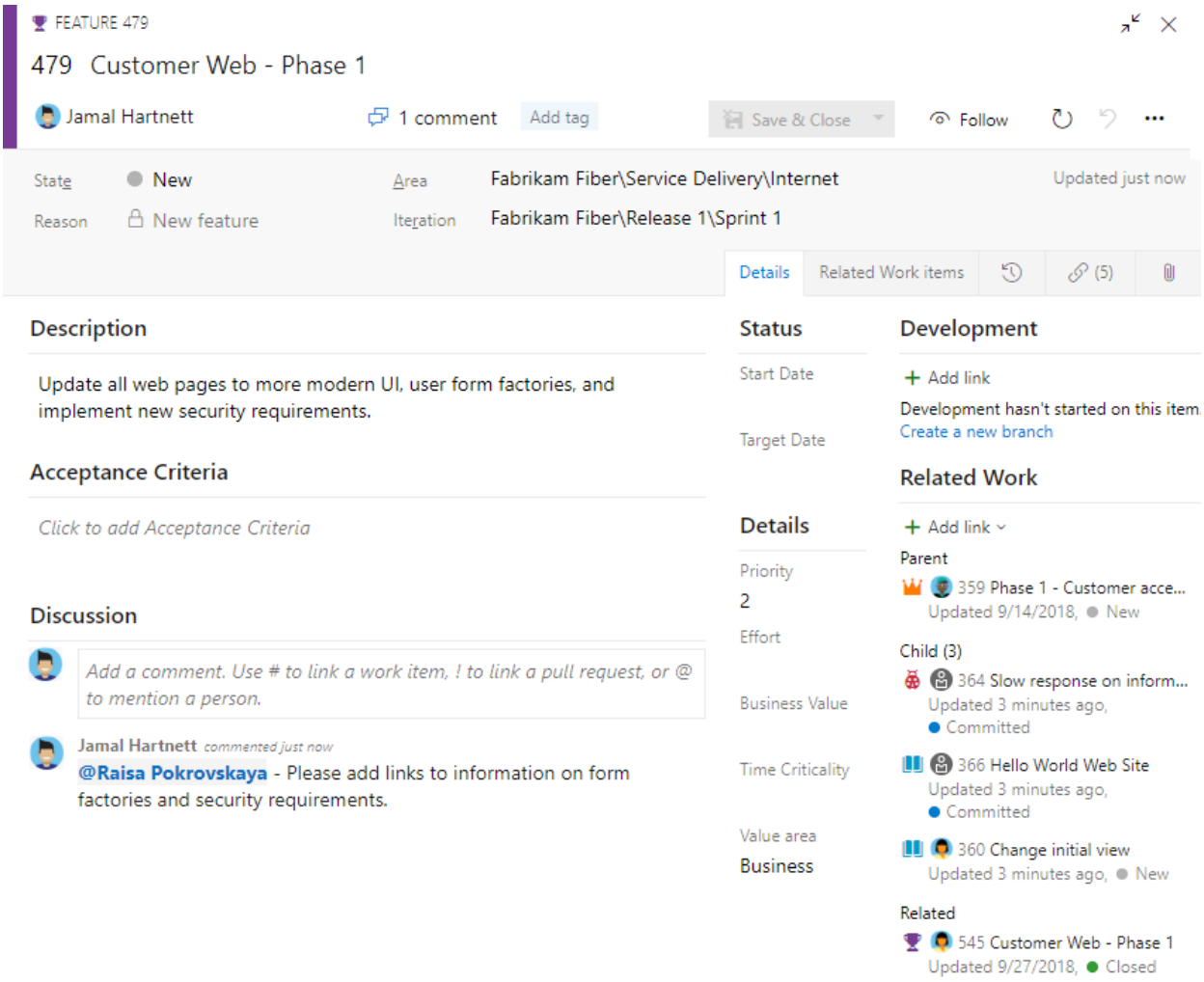
   Here, we've added six features.



Add details to a feature or epic

Open each item by double-clicking, or press Enter to open a selected item. Then, add the info you want to track. Enter as much detail as possible. The team will need to understand the scope, estimate the work, develop tests, and ensure that the end product meets acceptance criteria.

> **NOTE**
>
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your web app, options that you or your admin have enabled, and which process was chosen when creating your project—Agile, Basic, Scrum, or CMMI. The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.



Field

Usage

Value Area

The area of customer value addressed by the epic, feature, or backlog item. Values include:

- **Architectural** —technical services to implement business features that deliver solution
- **Business** (Default) —services that fulfill customers or stakeholder needs that directly deliver customer value to support the business

Effort
Story Points
Size

Provide a relative estimate of the amount of work required to complete a Feature or Epic. Use any numeric unit of measurement your team prefers. Some options are story points, time, or other relative unit.

Business Value

Specify a priority that captures the relative value of an Epic, Feature, or backlog item compared to other items of the same type. The higher the number, the greater the business value. Use this field when you want to capture a priority separate from the changeable backlog stack ranking.

Time Criticality

A subjective unit of measure that captures how the business value decreases over time. Higher values indicate that the Epic or Feature is inherently more time critical than those items with lower values.

Target Date

Specify the date by which the feature should be implemented.

Add child items

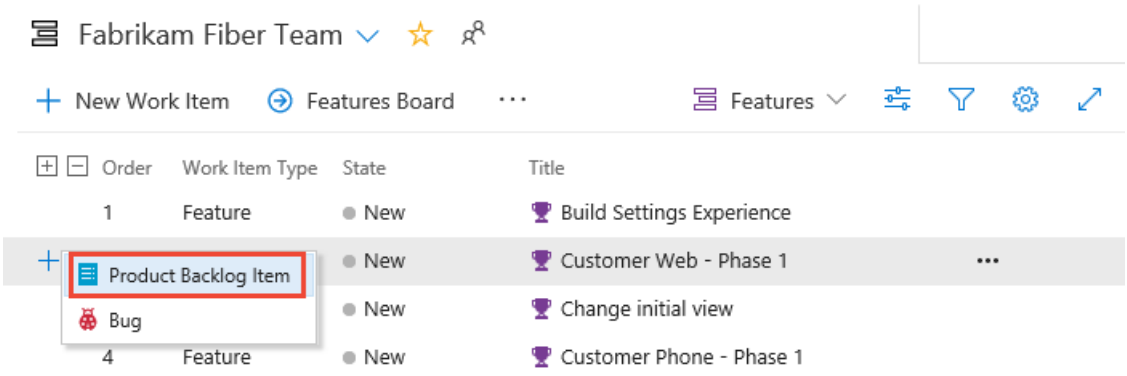
With your features defined, add child items to them. You can add child items from any backlog. And you can add features to epics and backlog items to features.

> **TIP**
> You can also add child user stories (Agile), or product backlog items (Scrum) or requirements (CMMI) from the Kanban board for Features. And, you can add child features from the Epic board. For details, see Kanban board features and epics. Also, you can quickly parent or reparent children from a backlog using the mapping pane as described in Organize your backlog, map child work items to parents.

Each team member has several tools to configure their backlog view: **Expand/Collapse one level**, **Column Options**, **Backlog level selector**, **View options**, and **Filter** toolbar. Options set for each backlog level are distinct and persist until changed. For tips on setting these view options and how to prioritize child items of portfolio backlog items, see Configure your backlog view.

To add a work item, choose + **Add**, and choose from the options provided.

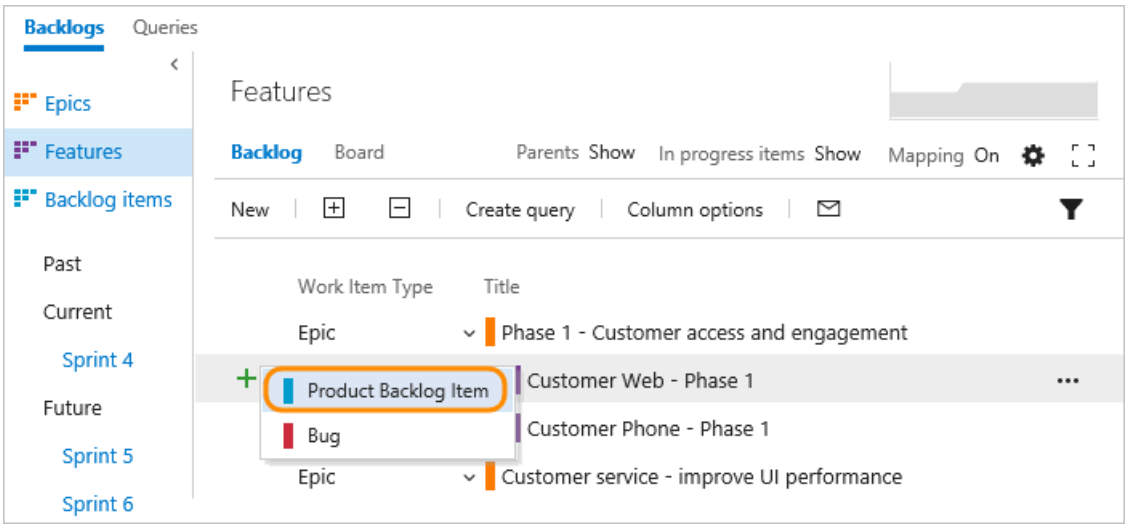
Here we add a product backlog item as a child to the Customer Web - Phase 1 feature.



Whenever you see the + **Add** icon, you can add a child item. The work item always corresponds to the hierarchy of work item types that are defined for your project.

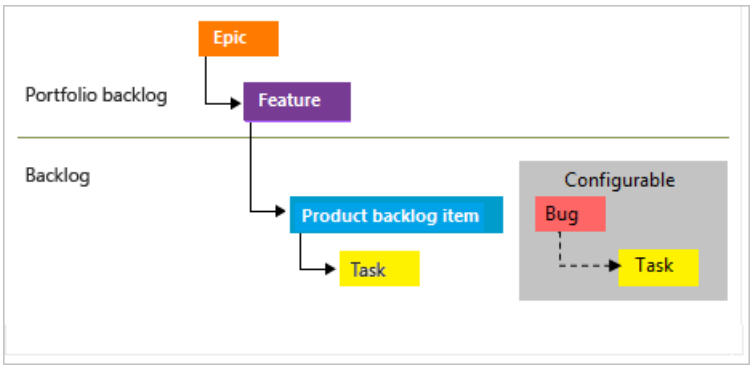
To add a work item, choose + **Add**, and choose from the options provided.

Here we add a product backlog item as a child to the Customer Web - Phase 1 feature.



Whenever you see the + **Add** icon, you can add a child item. The work item(s) always corresponds to the hierarchy of work item types that are defined for your project.

For Scrum projects, your hierarchy is as shown:



Because teams can also set bugs as tasks, bugs can be added as children of PBIs.

The work item types you'll see depends on the process you selected to create your project.

If you want bugs to show up on your backlog and you're not seeing them, enable them for your team.

Add portfolio backlog levels and boards

If you need more than two portfolio backlogs, you can add up to two more for a total of five backlog levels.

You can add them by customizing your process, adding new work item types, and then configuring your backlogs and boards. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. For details, see Customize an inheritance process and Customize your backlogs or boards (Inheritance process).

You can add them by either customizing your inherited process or modifying your XML definition files. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. To learn how, see the following articles based on the process model used to update your project:

Inheritance process model:

- Customize your backlogs or boards for a process.
- Customize an inheritance process.

On-premises XML process model:

- Customize work tracking
- Add portfolio backlogs

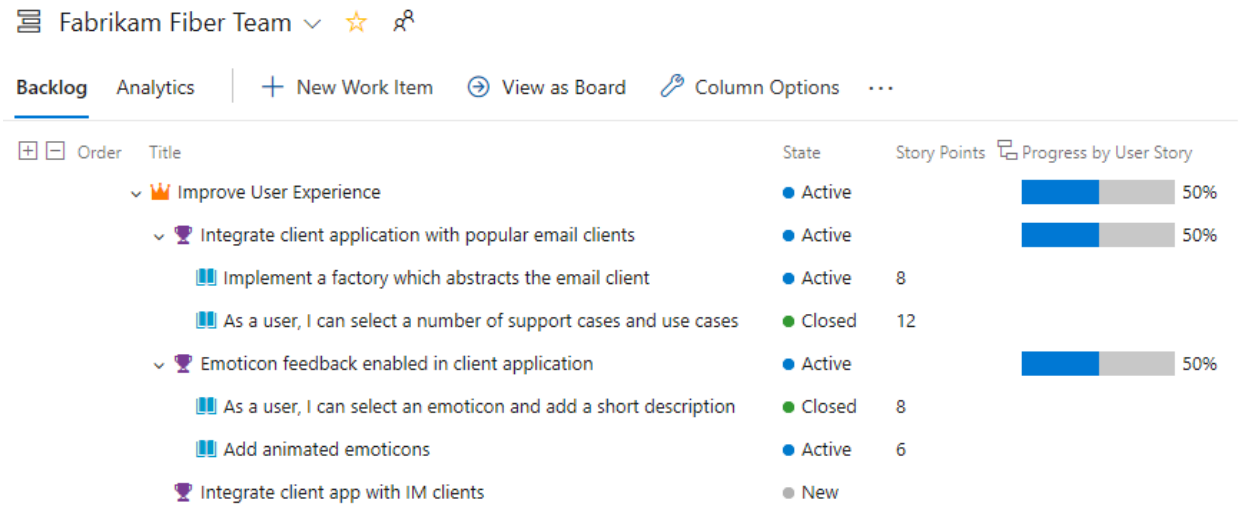
- Customize the On-premises XML process model

You can add them by defining additional work item types and then customizing your process configuration. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. To learn more, see Customize the On-premises XML process model and Add a portfolio backlog level.

Display rollup progress, counts, or totals

Product and portfolio backlogs support the display of rollup columns. You can add one or more rollup columns to display progress bars, counts of descendant work items, or totals of select numeric fields. Your settings persist for each page you customize and are only valid for your views.

Progress bars indicate the percentage of descendant items within a hierarchy that are closed or completed. *Counts* display the total number of descendant items. And, *Totals* provide a sum of numeric fields, such as Effort, Story Points, Completed Work, or Remaining Work of descendant items.
For example, progress bars are shown here for a portfolio backlog.



Try this next

Portfolio backlogs are a great way to organize your project plan and provide visibility of project plans across enterprise teams. With portfolio backlogs, management teams gain insight into project status across their development teams.

Organize your backlog

Related articles

- Configure your backlog view
- Azure Boards FAQs
- Filter product and portfolio backlogs
- Select backlog navigation levels for your team
- Work with multi-team ownership of backlog items
- Product backlog controls
- Keyboard shortcuts

Organize your backlog and map child work items to parents in Azure Boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

After you've added features or epics to your portfolio backlog, organize your backlog by mapping backlog items. You can quickly add and group items into a hierarchy. And also drill up or down within the hierarchy, reorder and reparent items, and filter hierarchical views.

In this article you'll learn how to:

- Open your product backlog or portfolio backlog
- View the tree hierarchy
- Group backlog items using the Mapping pane
- Reparent items through dragging or the **Change parent** option

> **NOTE**
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans.
> If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.

- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

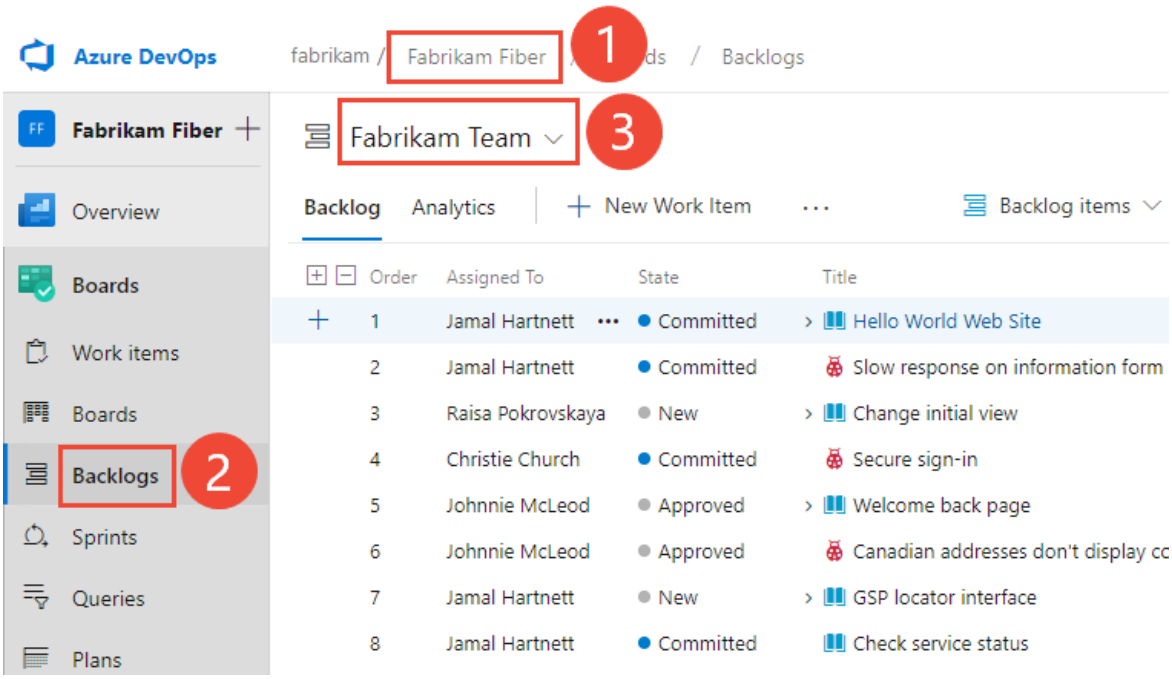
> **NOTE**
> **Stakeholder** access users for a private project can't drag items to map or reparent them or to assign their sprint.

> **NOTE**
> **Stakeholder** access users can't drag items to map or reparent them or to assign their sprint.

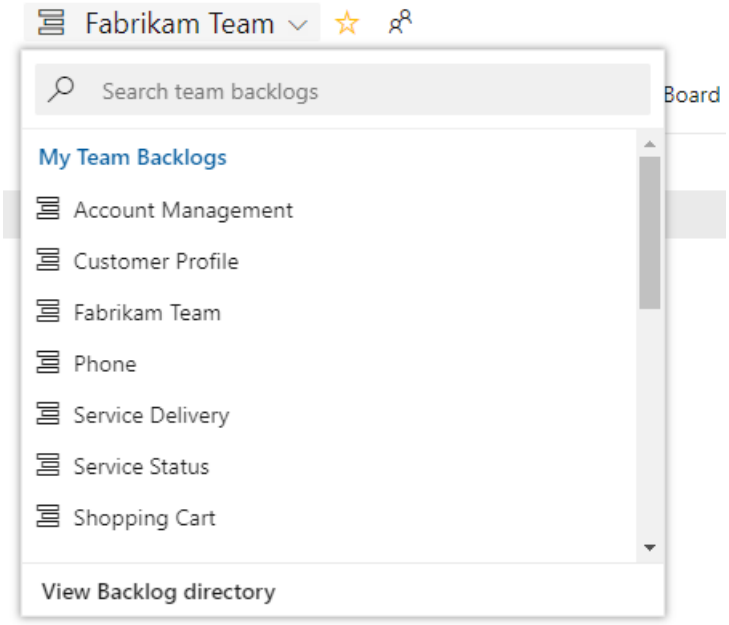
Open your backlog from the web portal

From your web browser, open your product backlog.

1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



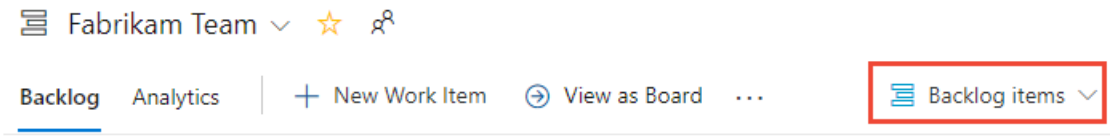
   To select another backlog, open the selector and then choose a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.



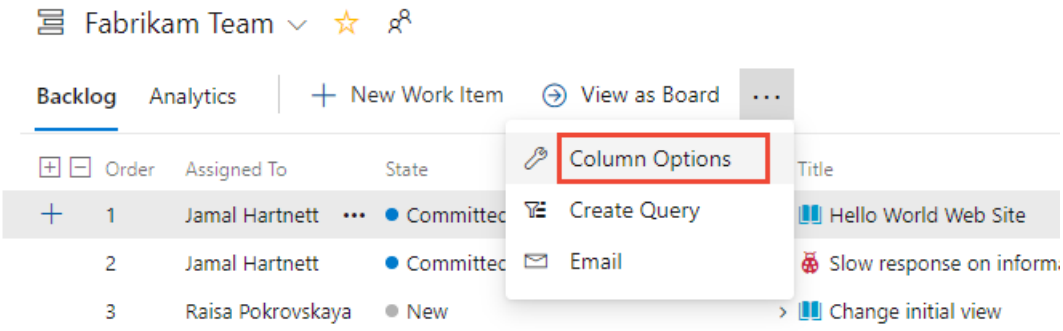
TIP

Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

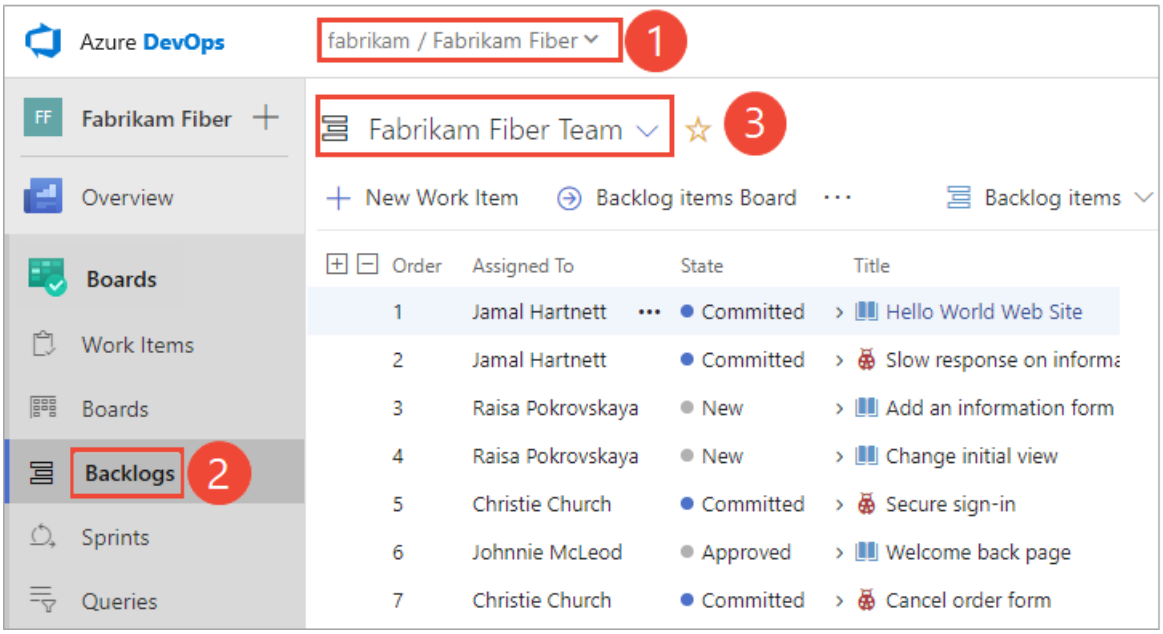
2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



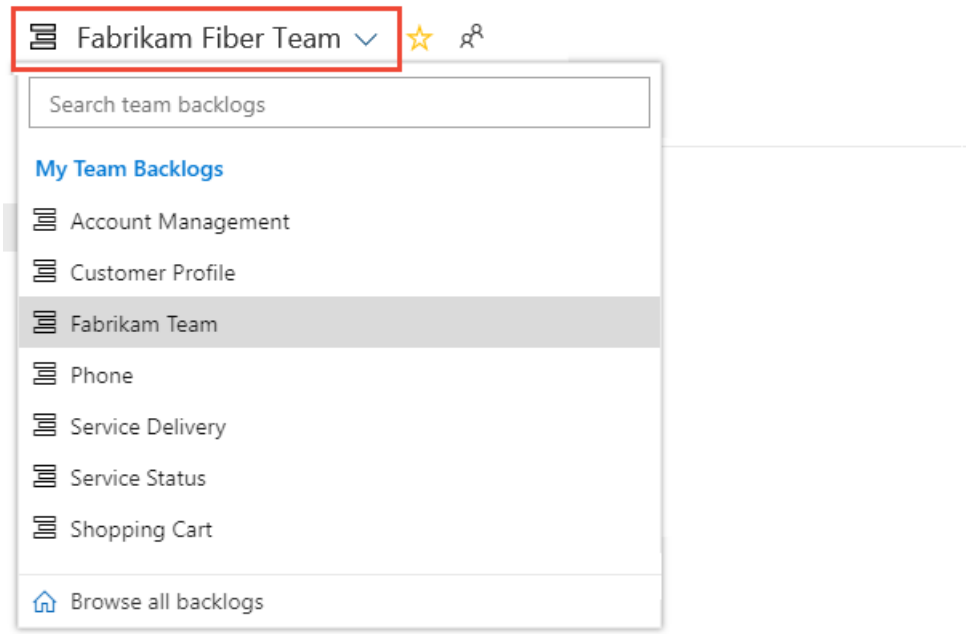
3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.



1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



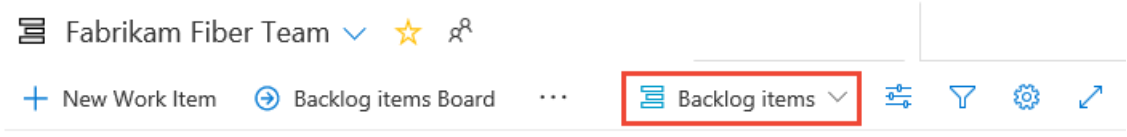
To choose another team, open the selector and select a different team or choose the **Browse all backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



TIP

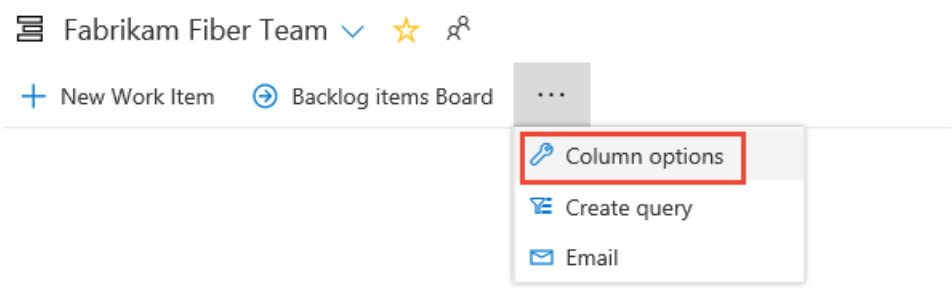
Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

2. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level.

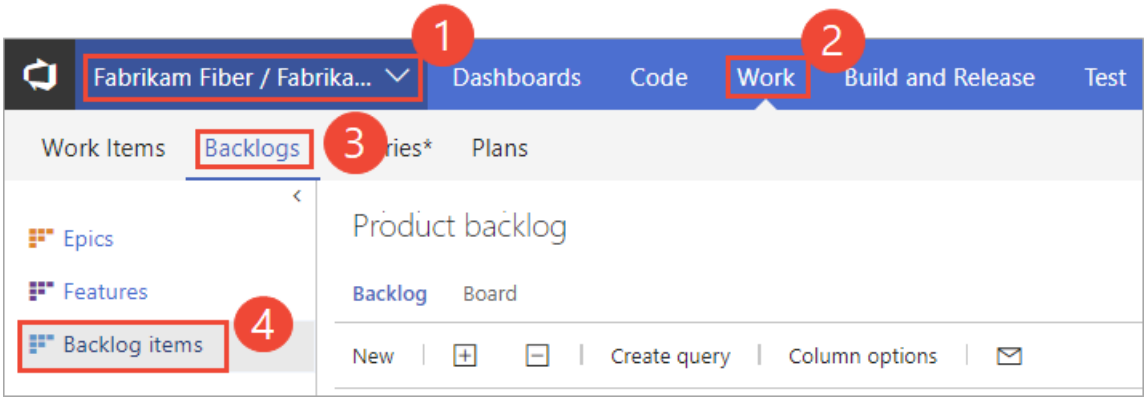


3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and

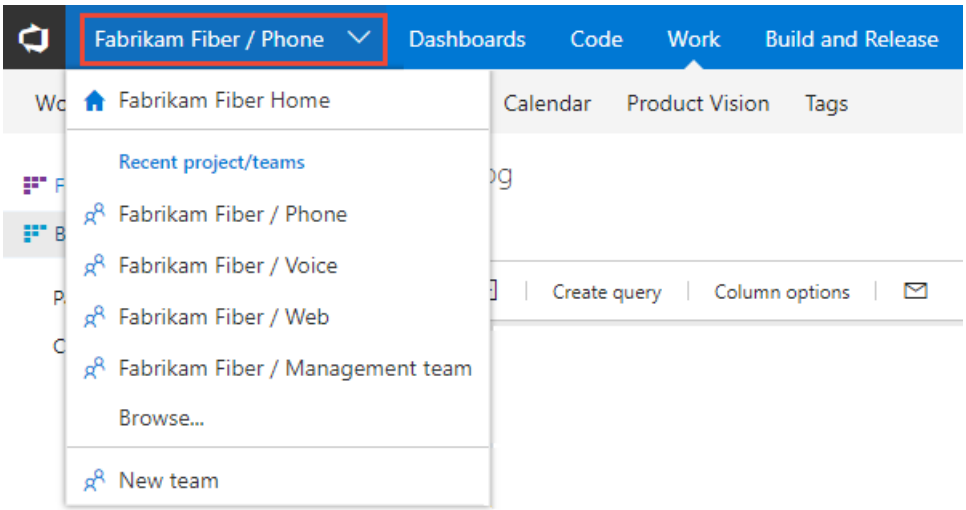
select **Column options**. To learn more, see Change column options.



From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).



To choose another team, open the project/team selector and select a different team or choose the **Browse** option.

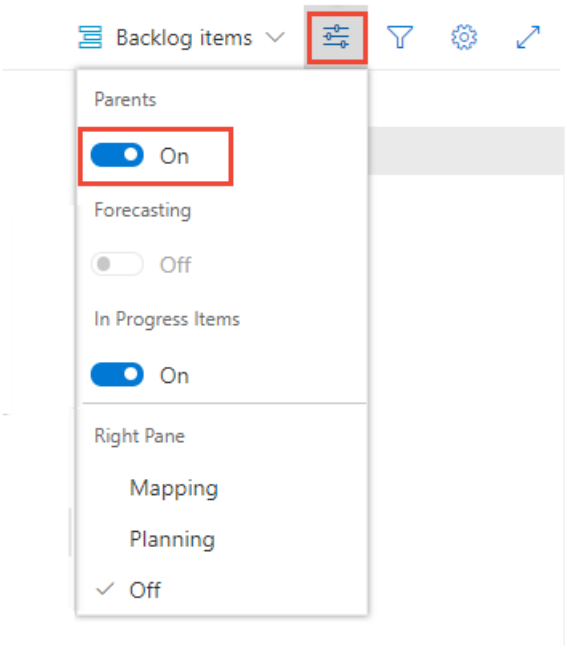


NOTE

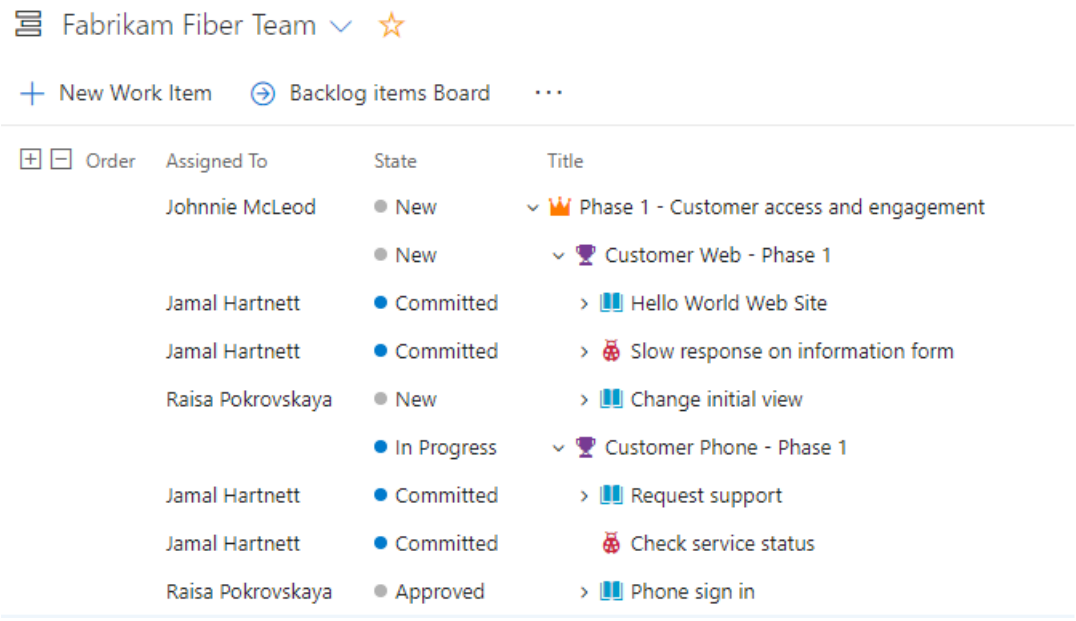
The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your web app, options that you or your admin have enabled, and which process was chosen when creating your project—Agile, Basic, Scrum, or CMMI. The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.

Show parent tasks and expand the tree hierarchy

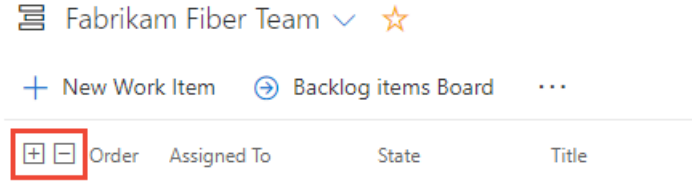
1. To view Parents or a tree hierarchy, choose the view options icon and slide **Parents** to **On**.



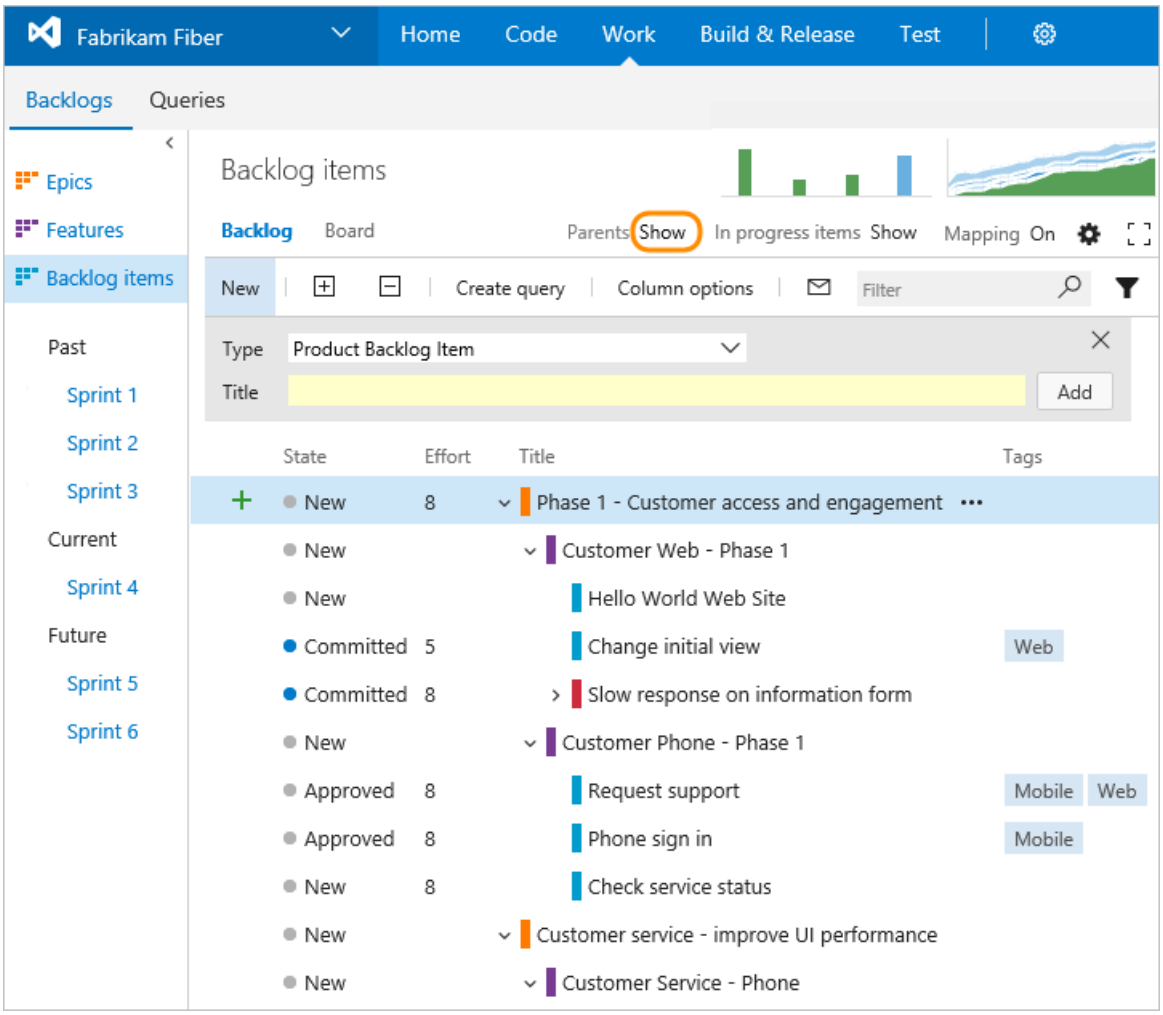
The hierarchical view displays. From this view, you can reparent items by dragging a child item to a new parent.



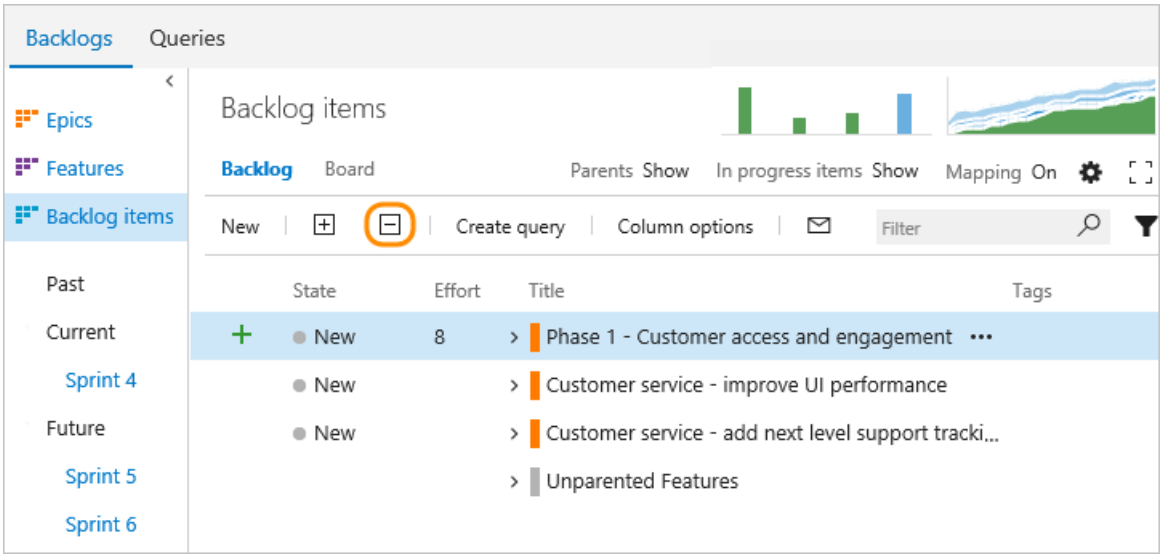
2. Use the expand ⊞ and collapse ⊟ icons to expand or collapse one level of the hierarchy.



From the product backlog page, set **Parents** to **Show** when you want to drill up or down within the hierarchy. You can also drag items to reparent items from this view.



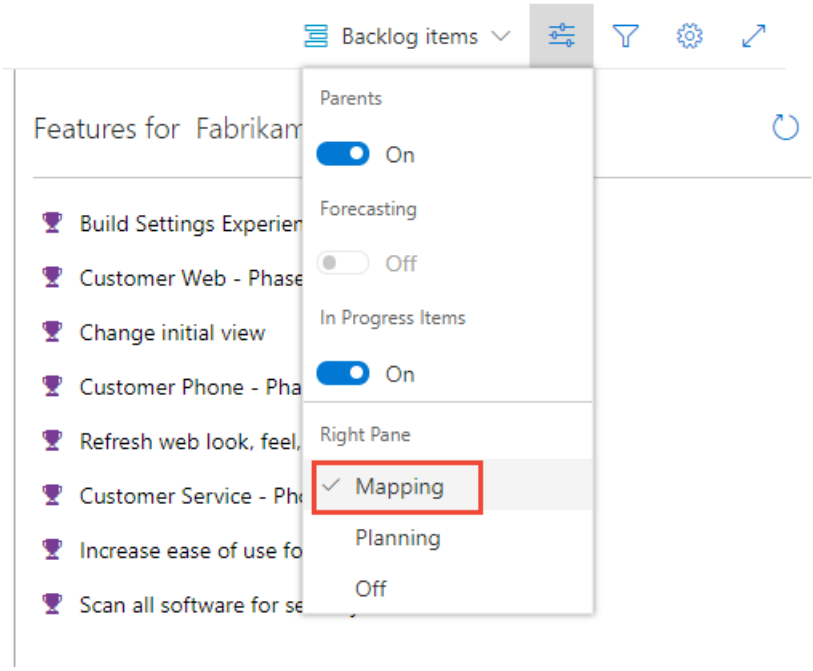
Use the expand ⊞ and collapse ⊟ icons to expand or collapse one level of the hierarchy.



Map items to group them under a feature or epic

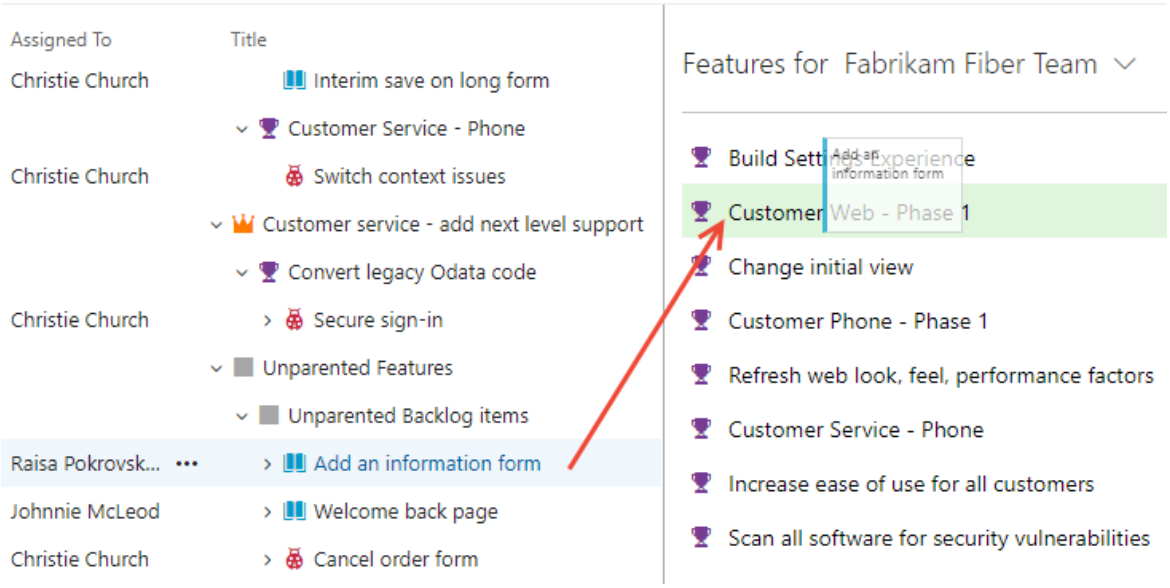
If you've already created your backlog, and now you want to organize it, you can do that most easily by mapping child items to parents.

1. Choose the view options icon and select **Mapping**.



The Mapping pane displays immediately.

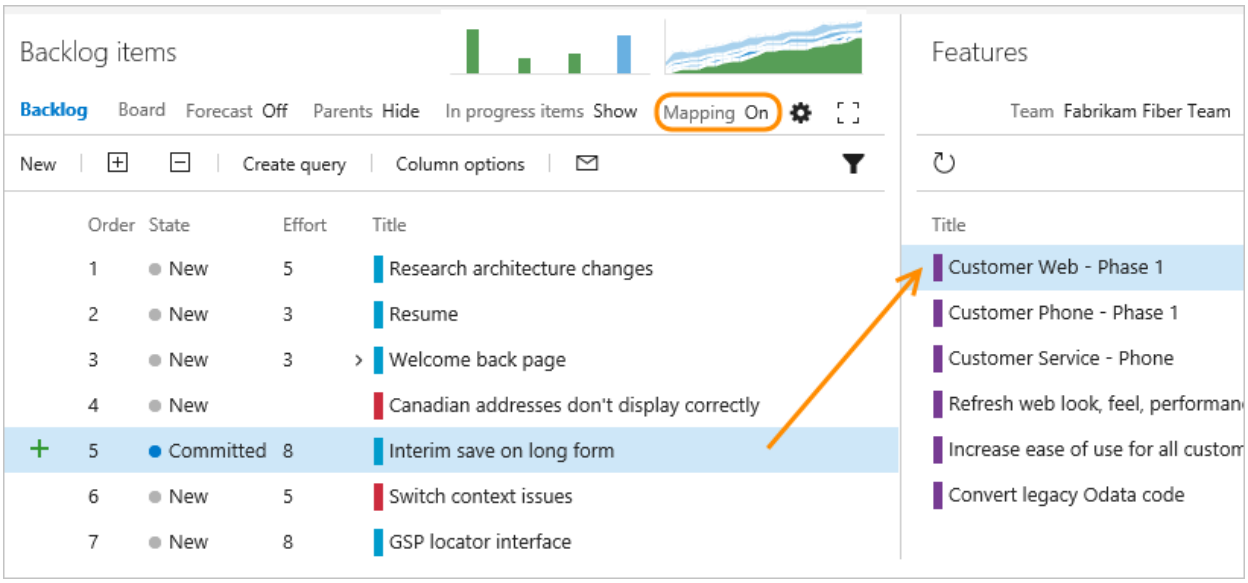
2. Find Unparented backlog items that may appear at the end of the parented set of backlog items. Parents must be turned on in view options.



3. To map features to epics, select the **Features** backlog from the backlog selector. The Epics Mapping pane will automatically display.

To map a backlog item under a feature, you first turn on mapping from your backlog (Backlog items, Stories, or Requirements). Next, find the Unparented backlog items group by turning the Parents control to Show. Unparented backlog items will appear at the end of the parented set of backlog items.

Drag items that are currently unparented to the feature under which they belong. Also, you can drag a backlog item to a different feature to change its parent. This mapping creates parent-child links from feature to user stories, which is captured in the (links) tab.



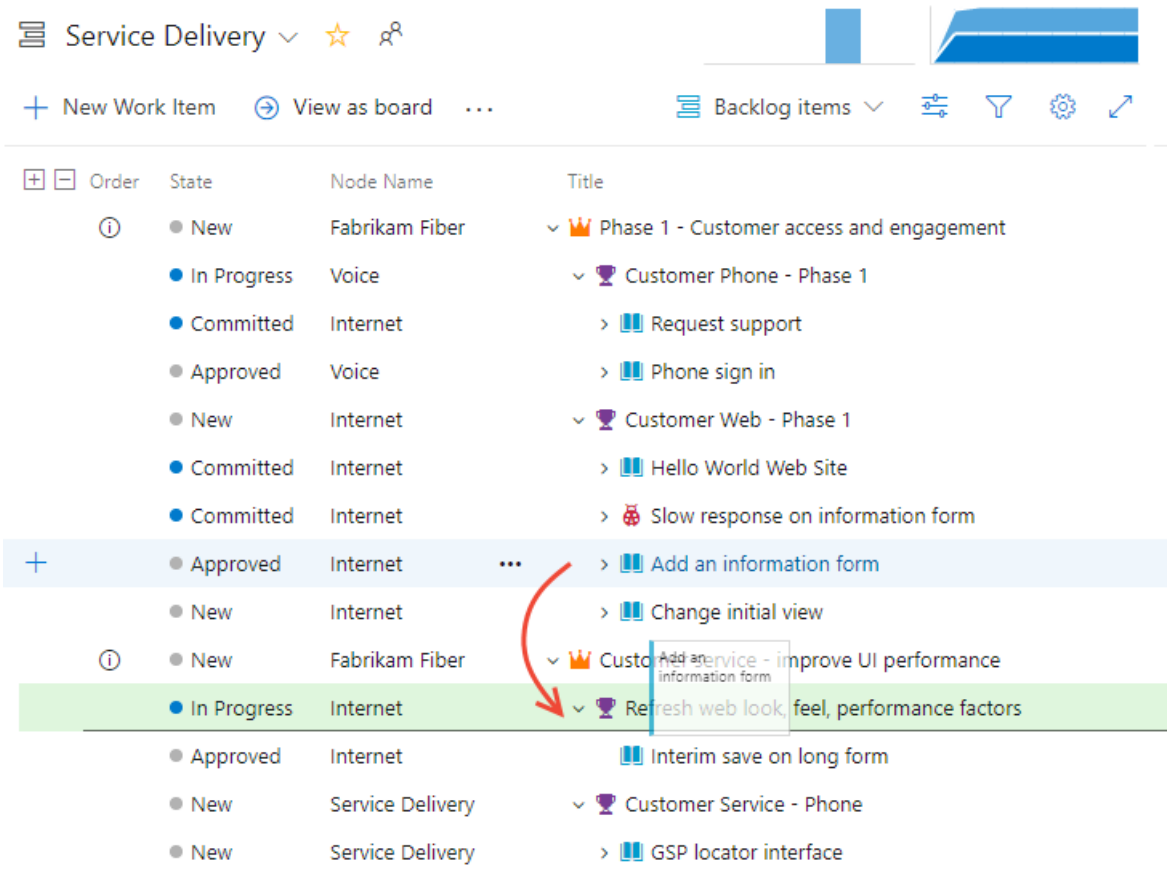
You can multi-select backlog and sprint backlog items in the same way as you multi-select items from query results.

It's the same process to map features to epics. From the Features backlog, drag features to an epic listed under the mapping pane.

Change the parent task and reorder items

When you need to change the order or grouping, drag the item to its new location.

You can reparent an item using the mapping pane, or drag it within the hierarchy to change its parent.



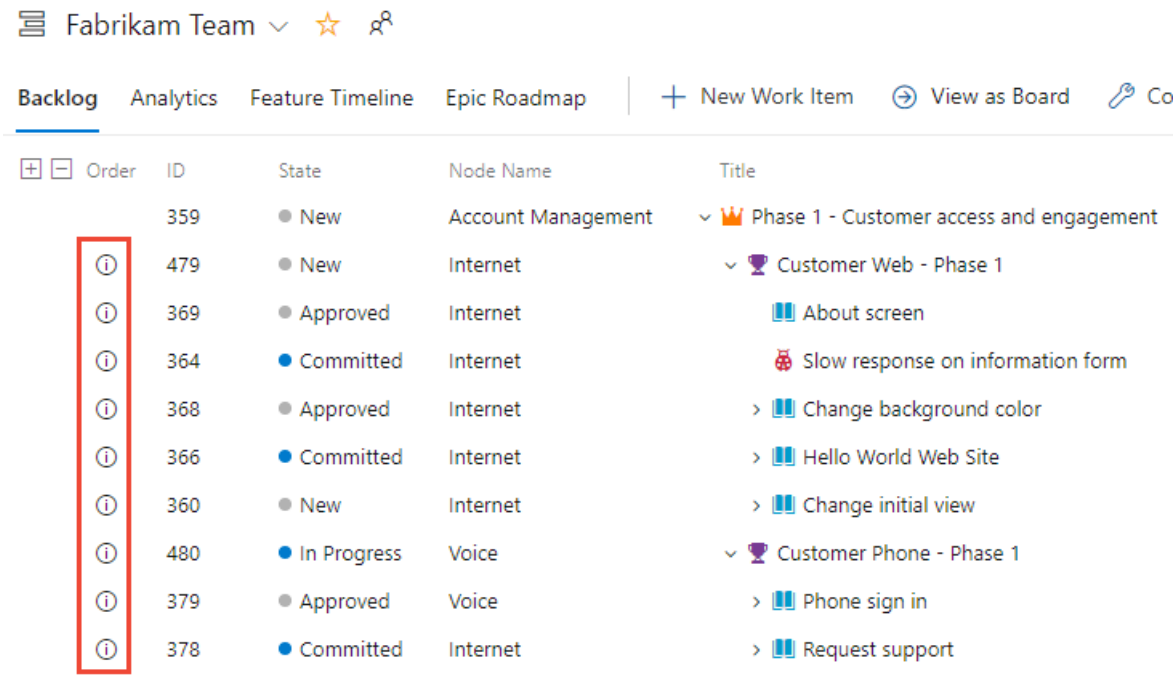
You can only reparent backlog items under other features, and features under other epics.

Also, to change an item's priority within a group, drag the item up or down within its hierarchical group.

Reordering from a portfolio backlog works the same as when you moved items into priority order on your product backlog.

Limitations on reordering backlog items owned by other teams

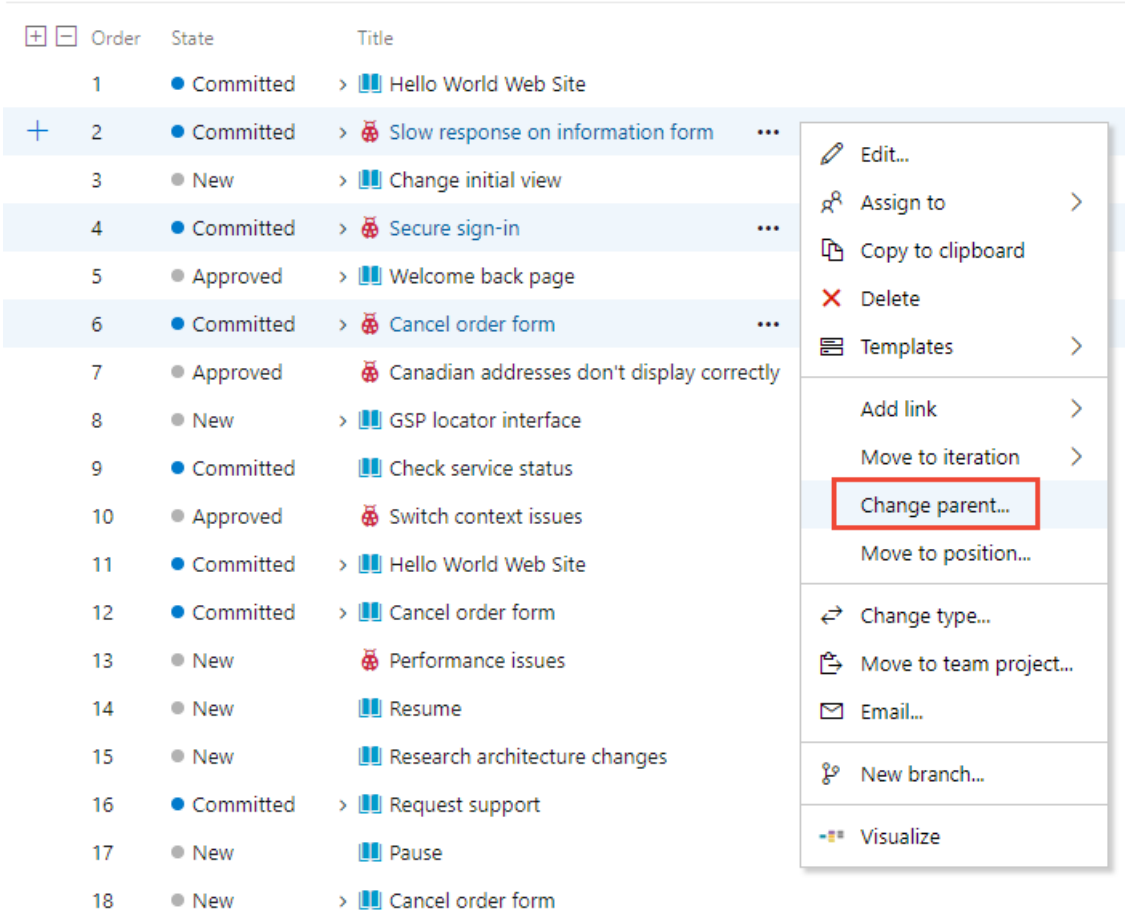
If you find you can't reorder a backlog item, check whether the ⓘ info icon appears in the first column as shown in the following image.



You can reparent items owned by other teams, but you can't reorder items owned by other teams. For more information, see Backlogs, portfolios, and Agile project management, Work with multi-team ownership of backlog items.

Change a parent of multiple backlog items

From the product backlog, you can multi-select several work items and choose **Change parent...** to link the items to a parent work item.



Add portfolio backlog levels and boards

If you need more than two portfolio backlogs, you can add up to two more for a total of five backlog levels.

You can add them by customizing your process, adding new work item types, and then configuring your backlogs and boards. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. For details, see Customize an inheritance process and Customize your backlogs or boards (Inheritance process).

You can add them by either customizing your inherited process or modifying your XML definition files. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. To learn how, see the following articles based on the process model used to update your project:

Inheritance process model:

- Customize your backlogs or boards for a process.
- Customize an inheritance process.

On-premises XML process model:

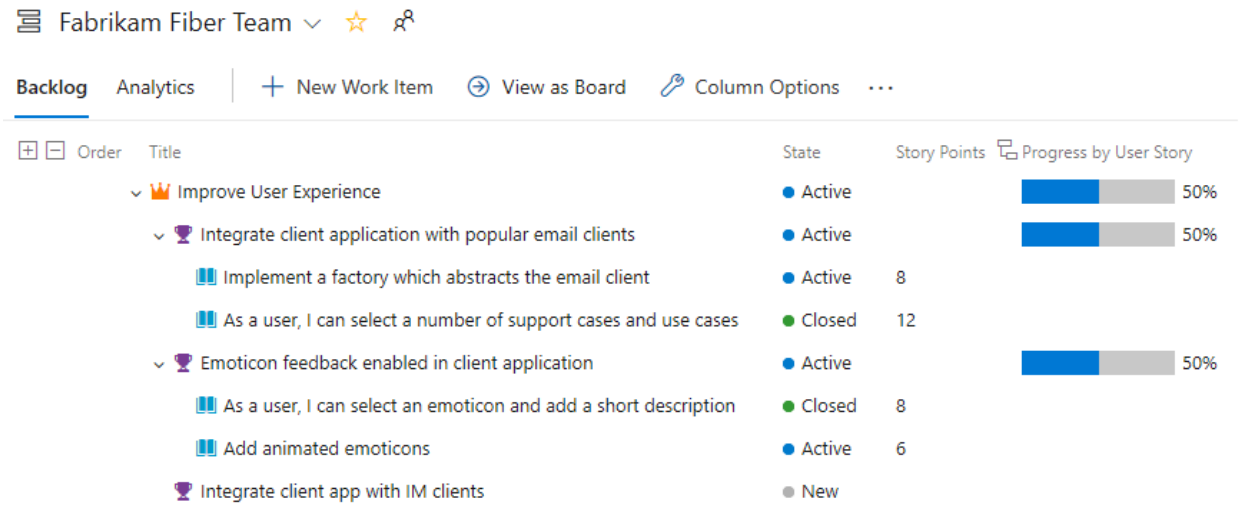
- Customize work tracking
- Add portfolio backlogs
- Customize the On-premises XML process model

You can add them by defining additional work item types and then customizing your process configuration. You can also add or modify the fields defined for a work item type (WIT) or add a custom WIT. To learn more, see Customize the On-premises XML process model and Add a portfolio backlog level.

Display rollup progress, counts, or totals

Product and portfolio backlogs support the display of rollup columns. You can add one or more rollup columns to display progress bars, counts of descendant work items, or totals of select numeric fields. Your settings persist for each page you customize and are only valid for your views.

Progress bars indicate the percentage of descendant items within a hierarchy that are closed or completed. *Counts* display the total number of descendant items. And, *Totals* provide a sum of numeric fields, such as Effort, Story Points, Completed Work, or Remaining Work of descendant items.
For example, progress bars are shown here for a portfolio backlog.



Related articles

- Define features and epics
- Configure your backlog view
- Work with multi-team ownership of backlog items
- Select backlog navigation levels for your team
- Product backlog controls
- Filter product and portfolio backlogs
- Keyboard shortcuts

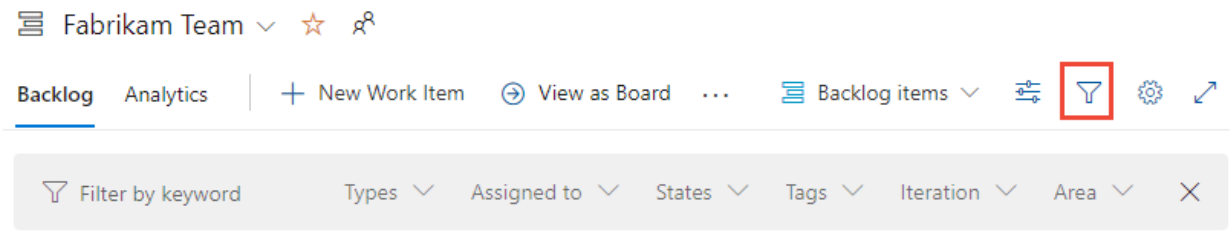
Interactively filter backlogs, boards, queries, and plans in Azure Boards

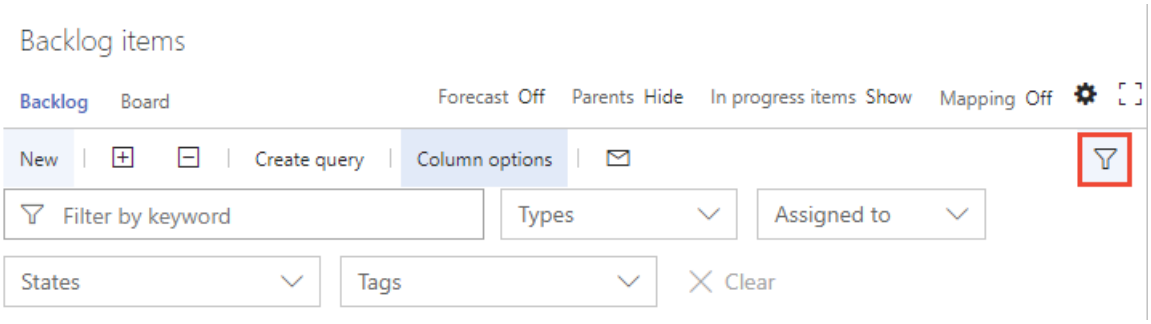
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

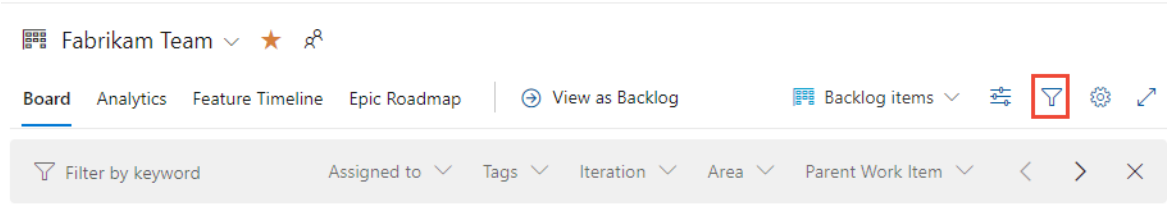
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

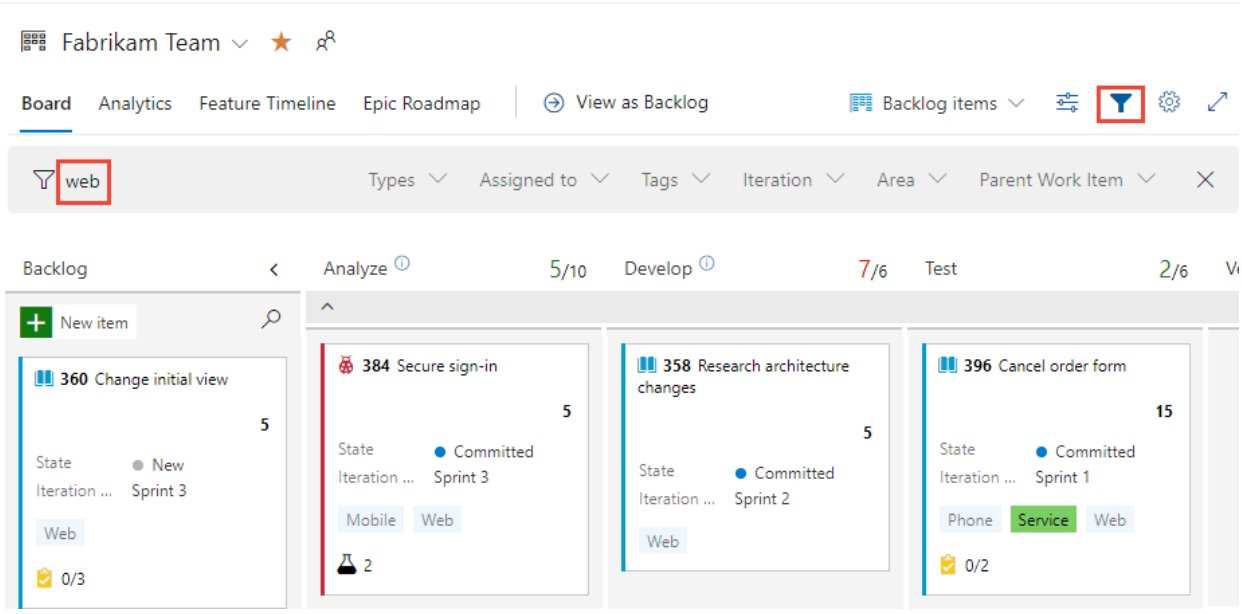
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

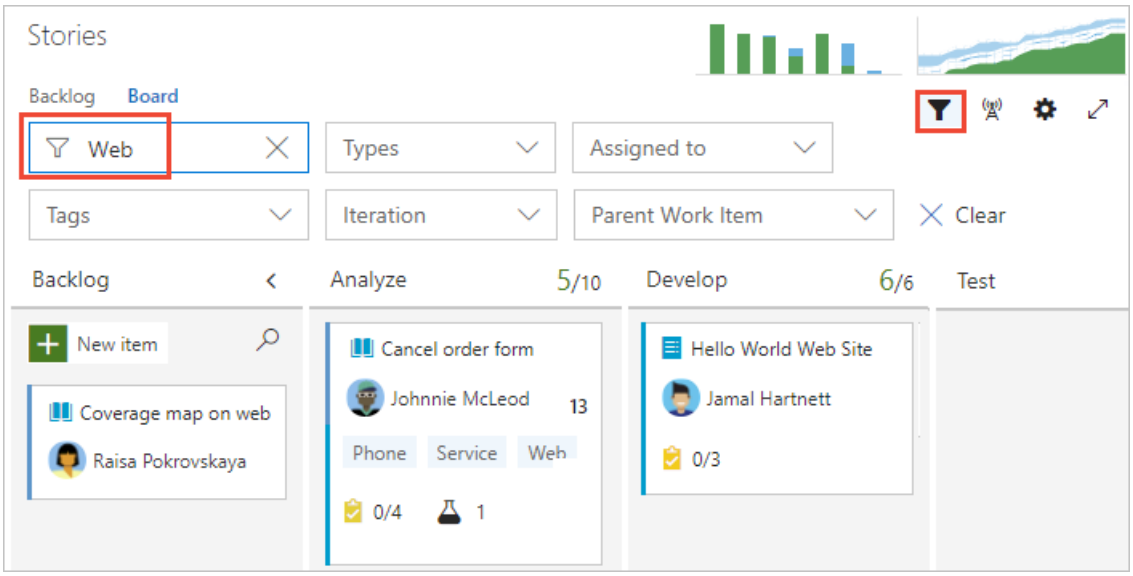
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

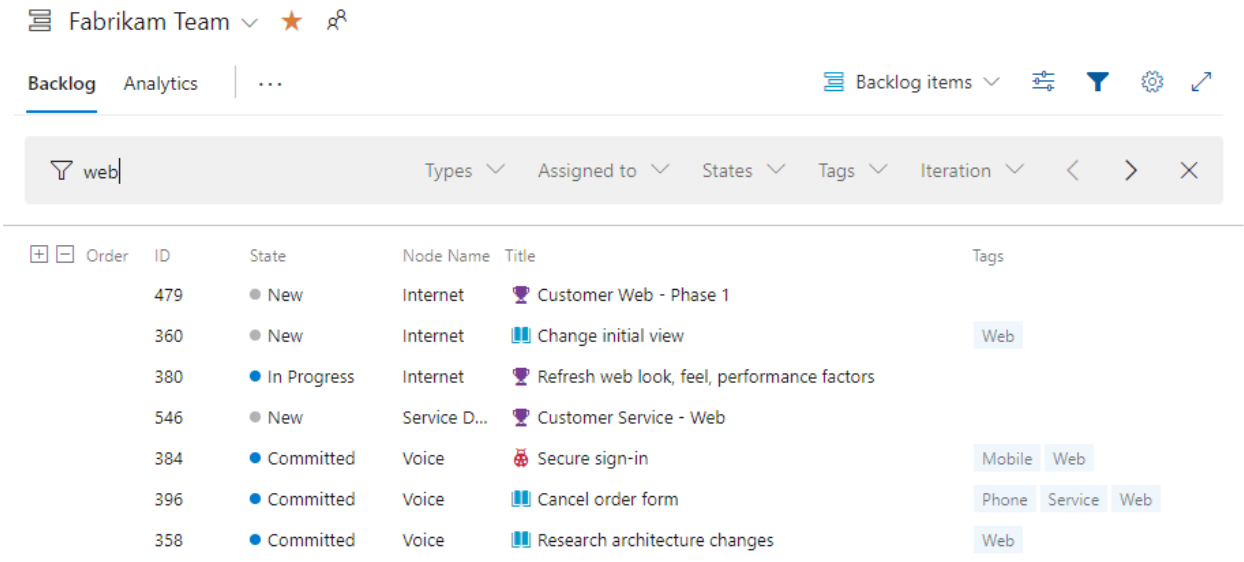


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

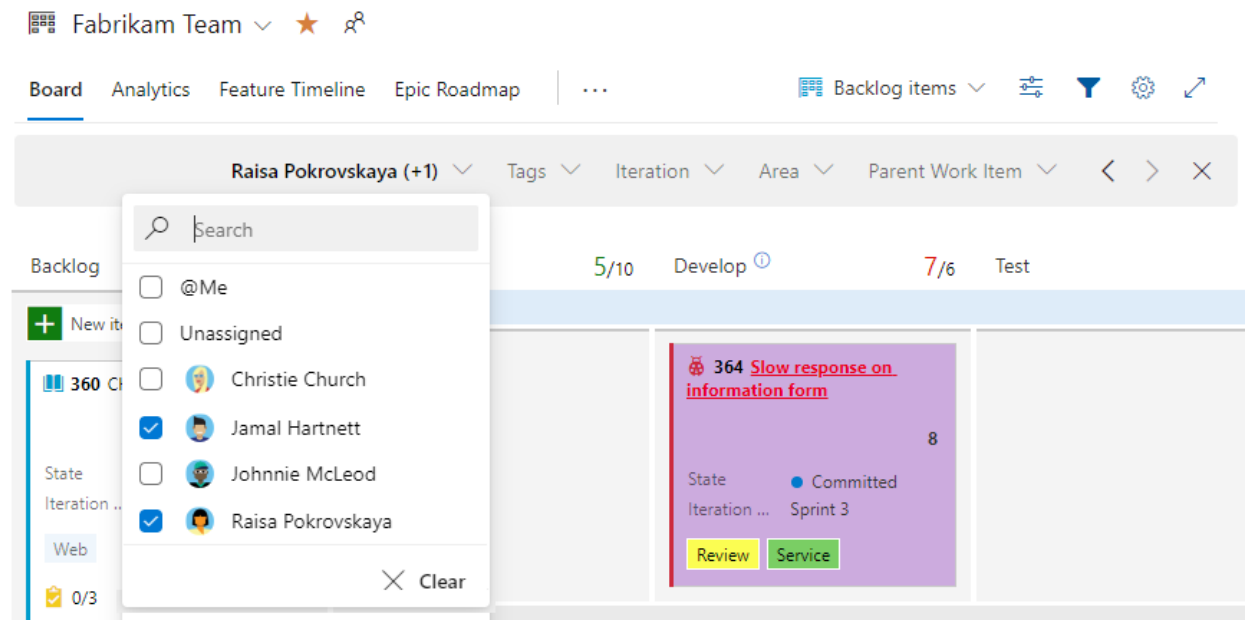
Features board

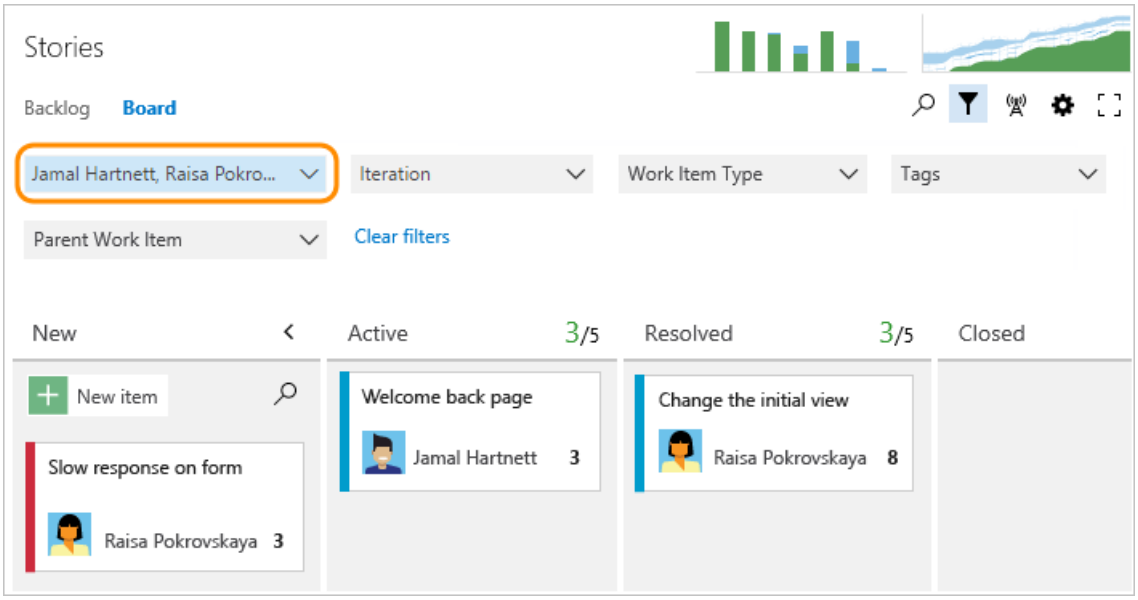
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

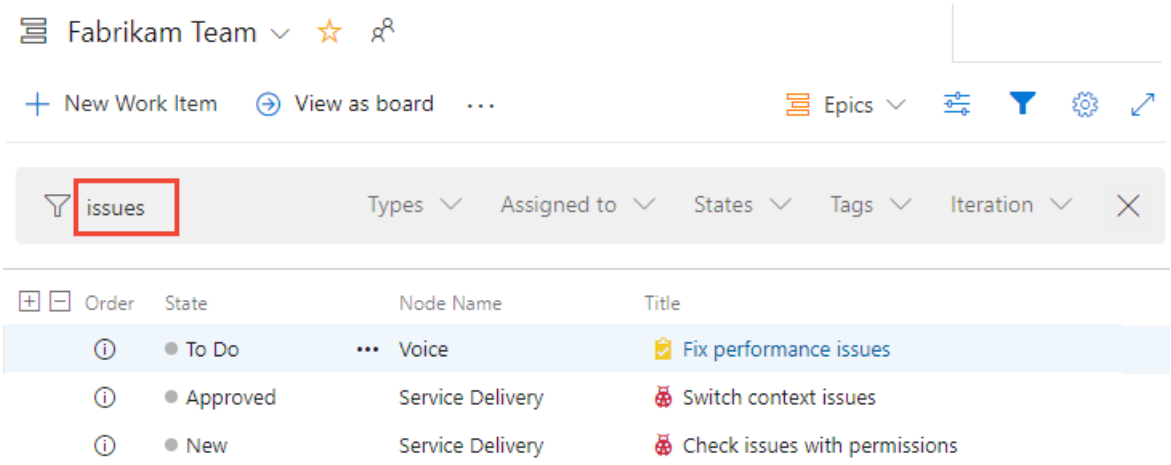
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

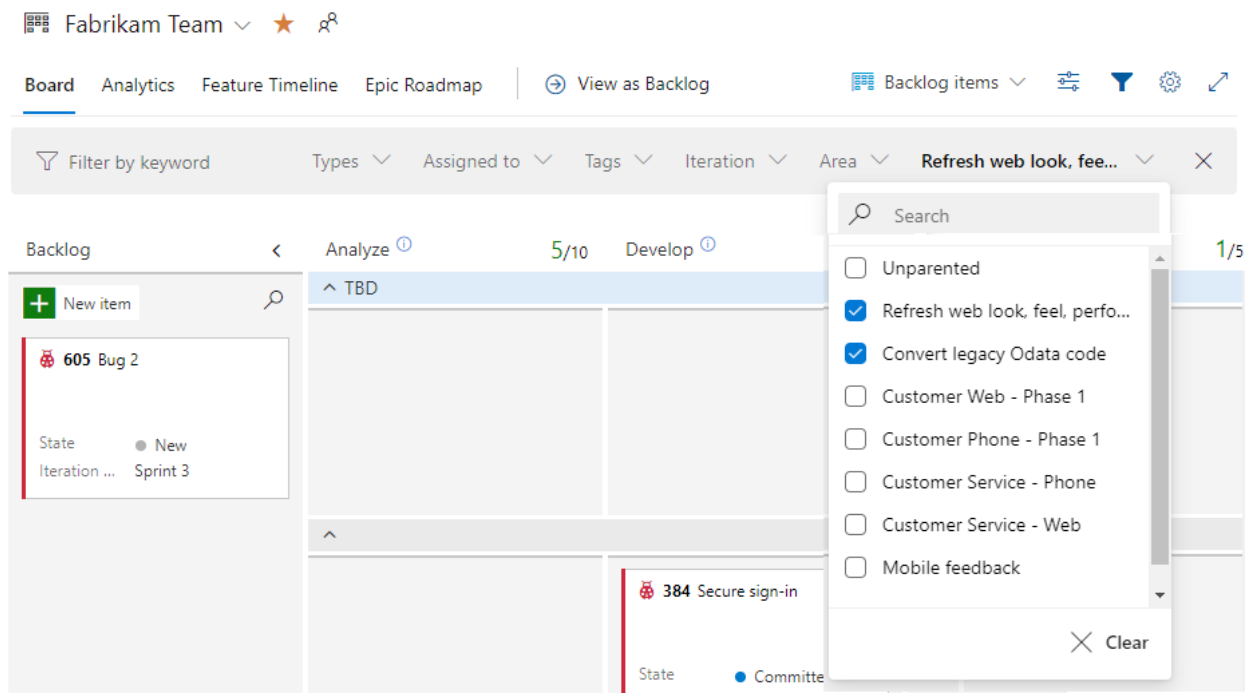
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

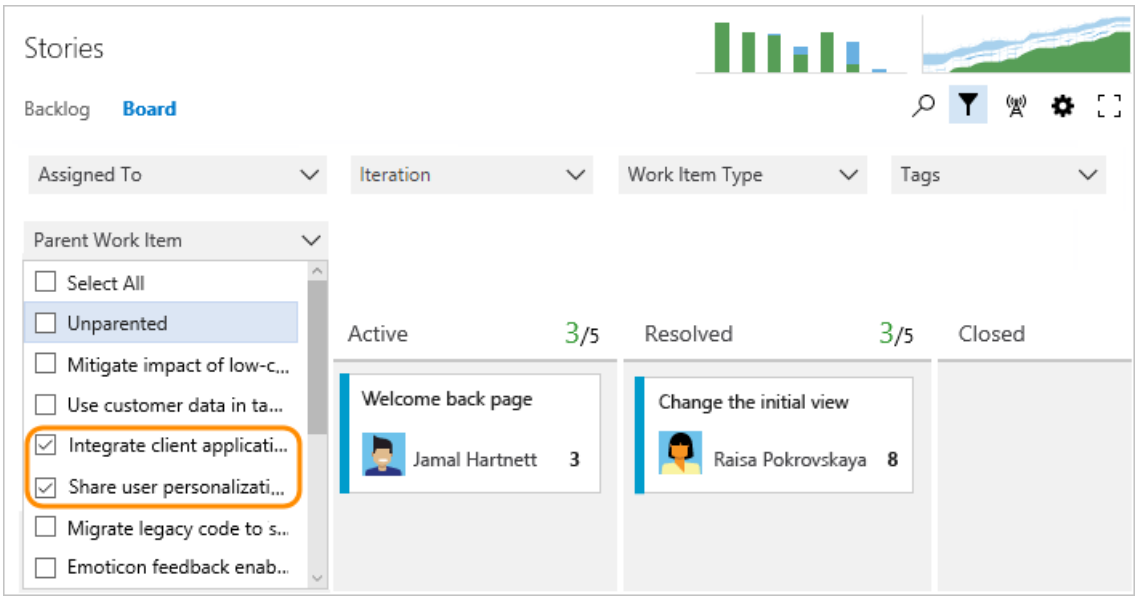
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



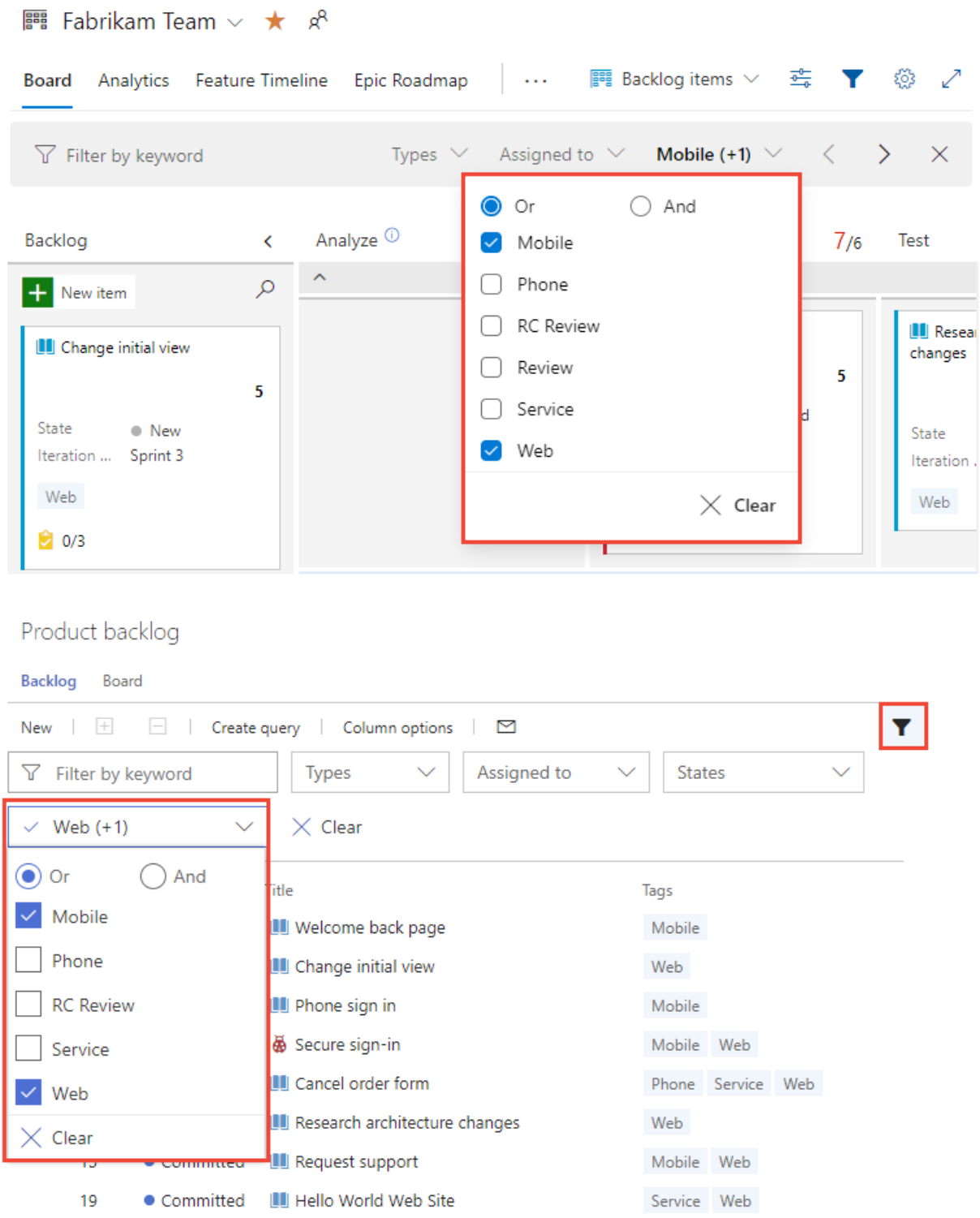


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

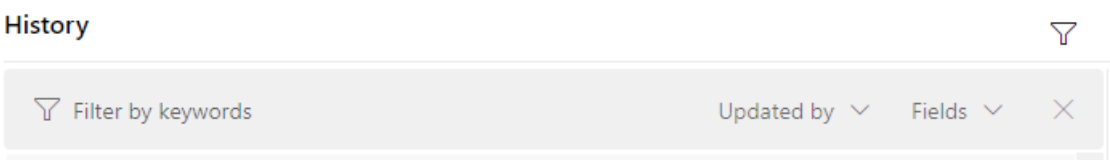


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Forecast your product backlog

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Teams use the forecast tool to help in their sprint planning efforts. By plugging in a value for the team velocity, the forecast tool will show which items in the backlog can be completed within future sprints. Both tools are team-specific tools that rely on the team's ability to estimate backlog items. Once your team has completed a sprint or two, they can use the team velocity to forecast how much of the backlog they can finish within the upcoming sprints.

Use this article to learn:

- How to forecast upcoming sprints
- Required and recommended team activities to support forecasting

> **NOTE**
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans. If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

Prerequisites

- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** security group. If you're not on a project or team, get added now.
- You must be granted **Basic** access or higher to use the forecast feature. For details, see About access levels.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** security group. If you're not on a project or team, get added now.
- You must be granted **Basic** access or higher to add or modify work items. For details, see About access levels.

Required and recommended activities

Here's what you need to have in place before you attempt to forecast your team's backlog.

Required:

- Define iteration paths (sprints) and configure team iterations
  - Sprints should be of the same duration.
  - Select enough future sprints to forecast your entire product backlog.
- Define and estimate backlog items. If you work from your team's backlog, the items you create will automatically be assigned to the current sprint (Iteration) and to your team's default Area Path.

- Update the status of backlog items once work starts and when completed. Only backlog items whose State maps to a state category of *Proposed* or *In Progress* show up on the velocity chart. (for details, see Workflow states and state categories).

Recommended:

- Define and size backlog items to minimize variability.
- Determine how your team wants to treat bugs. If your team chooses to treat bugs like requirements, bugs will show up on the backlog and be counted within the Velocity chart and forecasting.
- Set your team's area path. The forecast tool will forecast those items based on your team's default settings. These settings can specify to include items in area paths under the team's default or exclude them.
- Don't create a hierarchy of backlog items and bugs. The display of the leaf node, the last node in a same-category hierarchy, may only appear on Kanban boards, sprint backlogs, and taskboards. To learn more, see Fix reordering and nesting issues, How backlogs and boards display hierarchical (nested) items.
  Instead of nesting requirements, bugs, and tasks, maintain a flat list—only creating parent-child links one level deep between different-category items. Use Features to group requirements or user stories. You can quickly map stories to features. The map creates parent-child links in the background.
- At the end of the sprint, update the status of those backlog items that the team has fully completed. Incomplete items should be moved back to the product backlog and considered in a future sprint planning meeting.

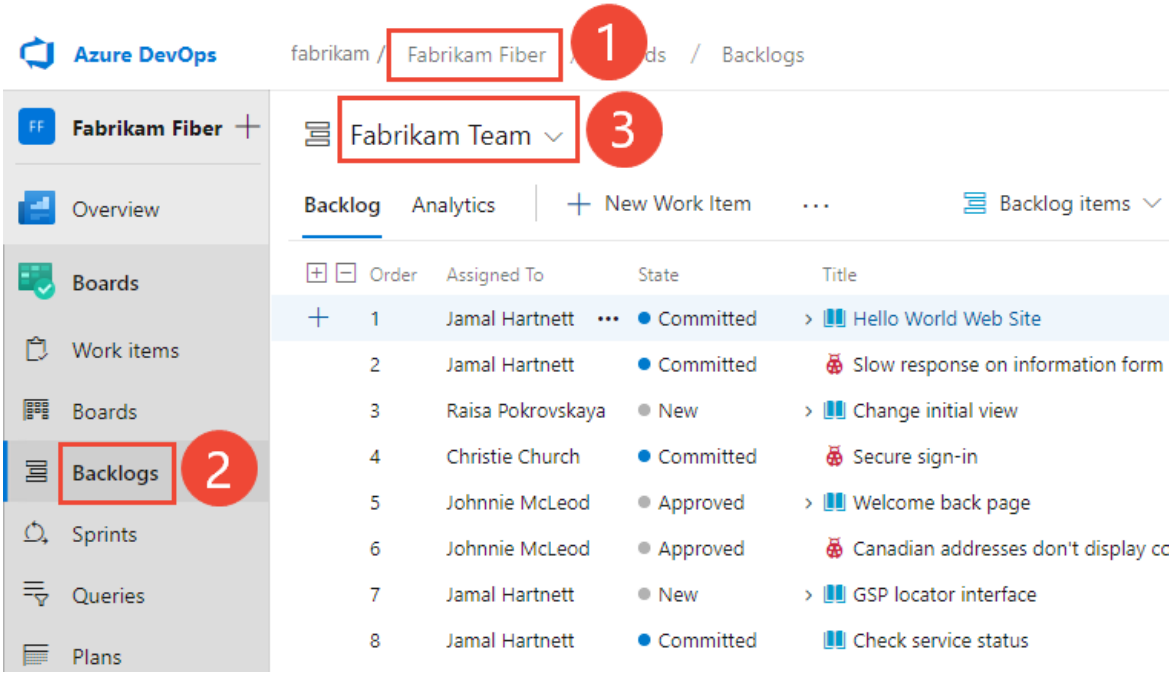
> **NOTE**
> If you work with several teams, and each team wants to work with their own backlog, velocity chart, and forecast tool, you can create additional teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items to only include those whose assigned area paths and iteration paths meet those set for the team.

Forecast upcoming sprints

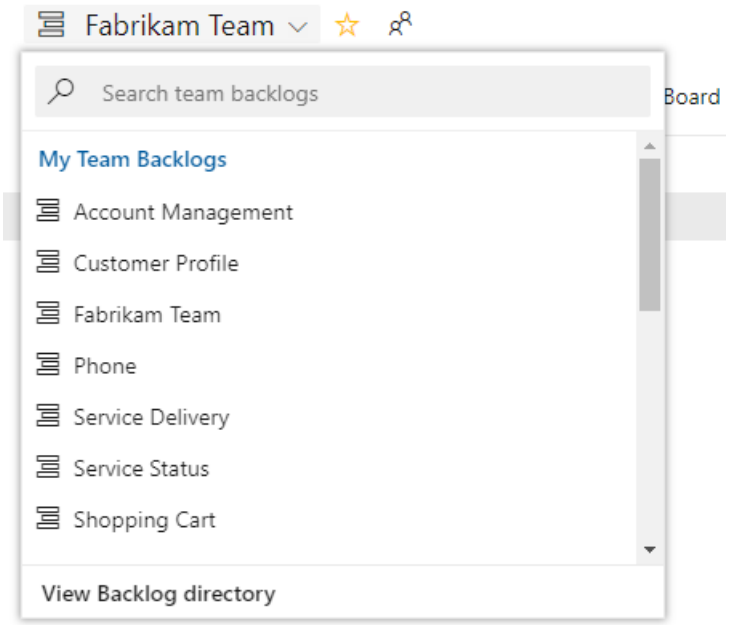
Use the forecast tool to get an idea of how many items you can complete within a sprint. By plugging in a velocity, you can see which items are within scope for the set of sprints the team has activated.

To forecast your product backlog, complete the following actions.

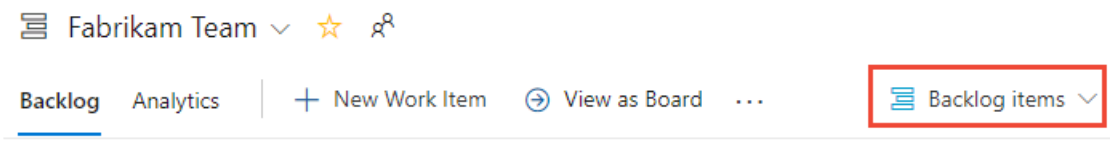
1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



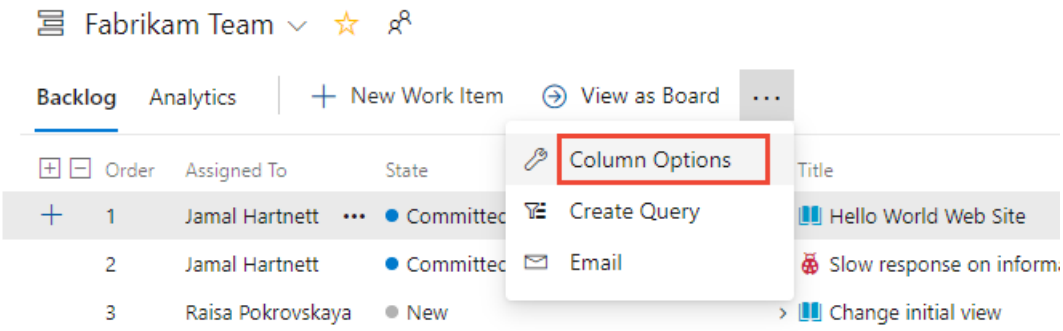
To select another backlog, open the selector and then choose a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.



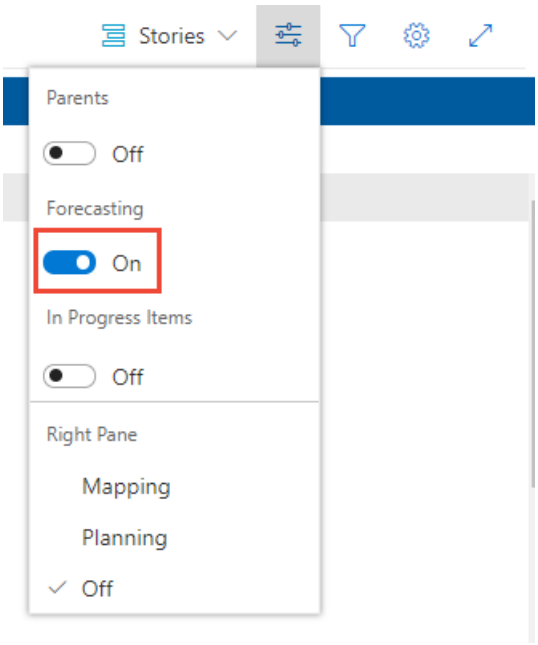
2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.

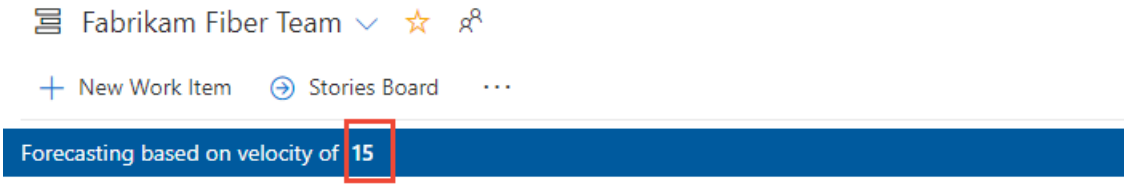


4. Choose the view options icon and slide **Forecasting** to **On**. To keep things simple, turn the **Mapping** and **Planning** panes **Off**.



Set **In Progress Items** to **Off** to hide those items that won't be counted in the forecast. The forecast tool ignores Scrum items set to *Committed* or *Done* and Agile and CMMI items set to *Active*, *Resolved*, or *Completed*.

5. Enter your team's predicted velocity.

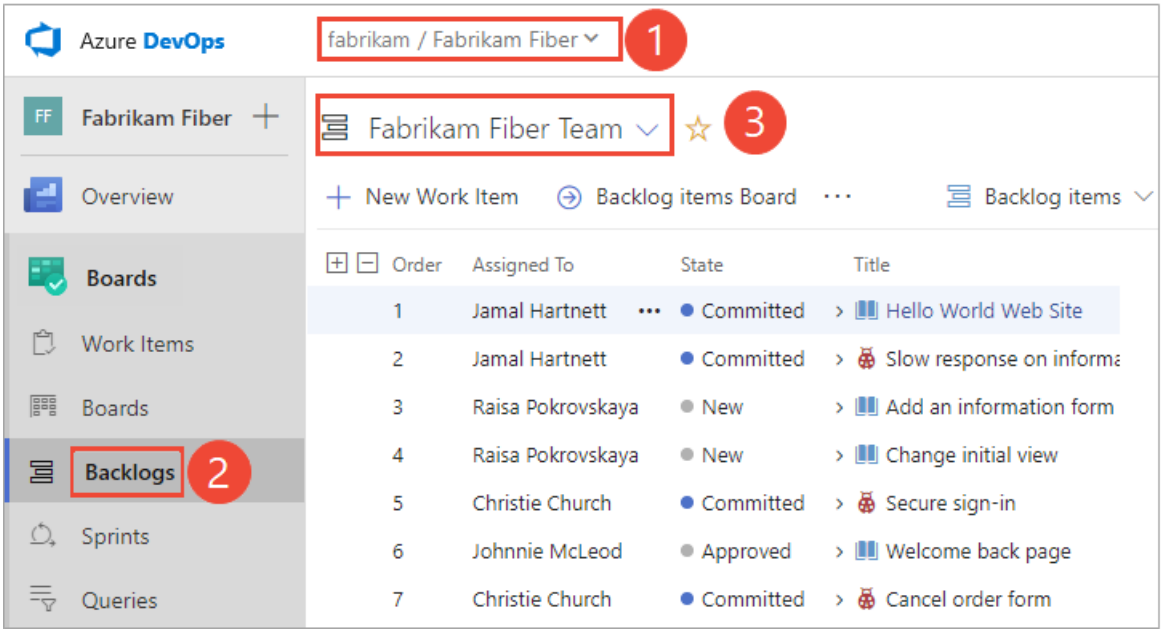


TIP

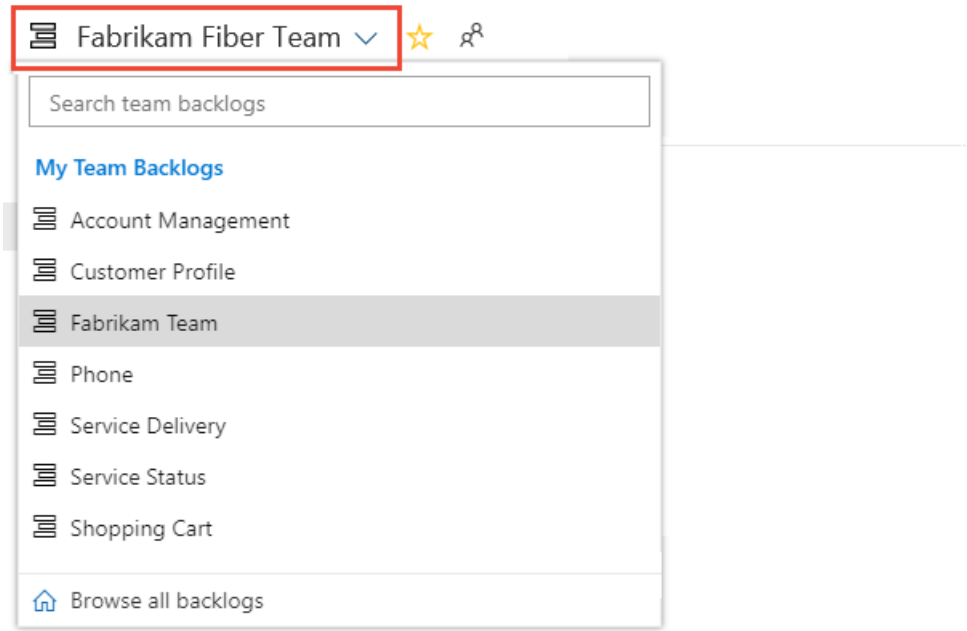
If your team has been working for several sprints, you can gain an idea of your team's velocity from the Velocity widget.

The tool draws lines for each future sprint selected by the team. The Forecast lines show how much work your team can complete in future sprints. Typically, items above the first line are already in progress for the current sprint. Items that fall between the first and second forecast lines indicate what can be completed in the named sprint.

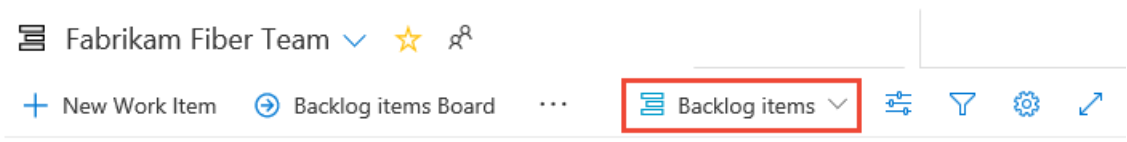
1. From your web browser, open your product backlog. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



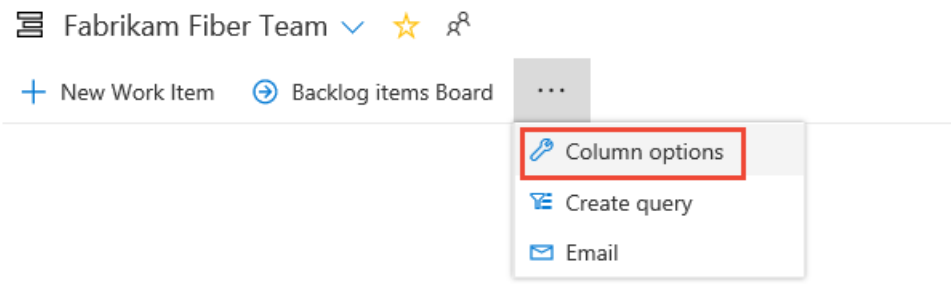
To choose another team, open the selector and select a different team or choose the **Browse all team backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



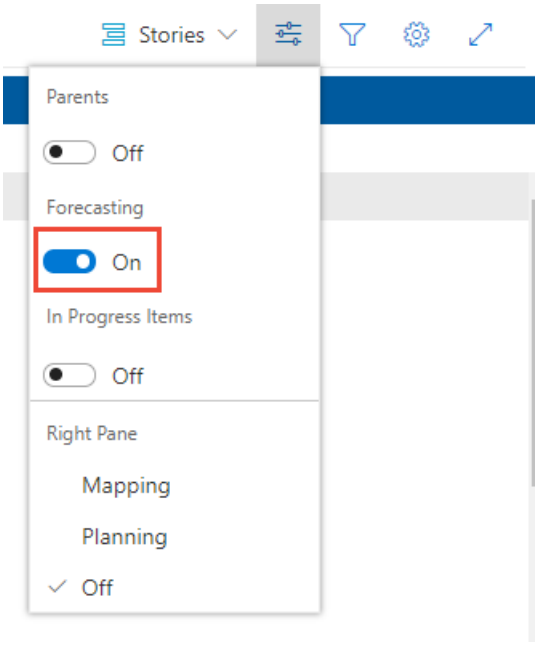
2. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level. You can only forecast a product backlog. You can't forecast a portfolio backlog such as Features or Epics.



3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.

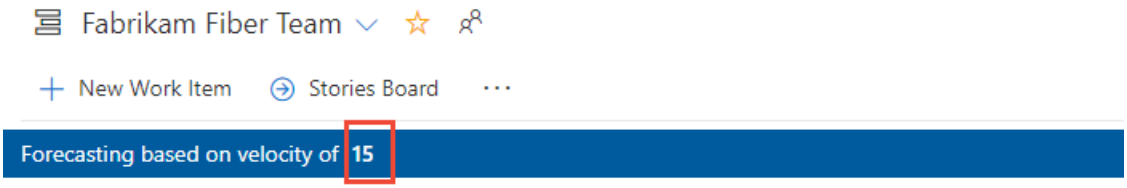


4. Choose the view options icon and slide **Forecasting** to **On**. To keep things simple, turn the **Mapping** and **Planning** panes **Off**.



Set **In Progress Items** to **Off** to hide those items that won't be counted in the forecast. The forecast tool ignores Scrum items set to *Committed* or *Done* and Agile and CMMI items set to *Active*, *Resolved*, or *Completed*.

5. Enter your team's predicted velocity. If the **Forecasting** bar doesn't appear.



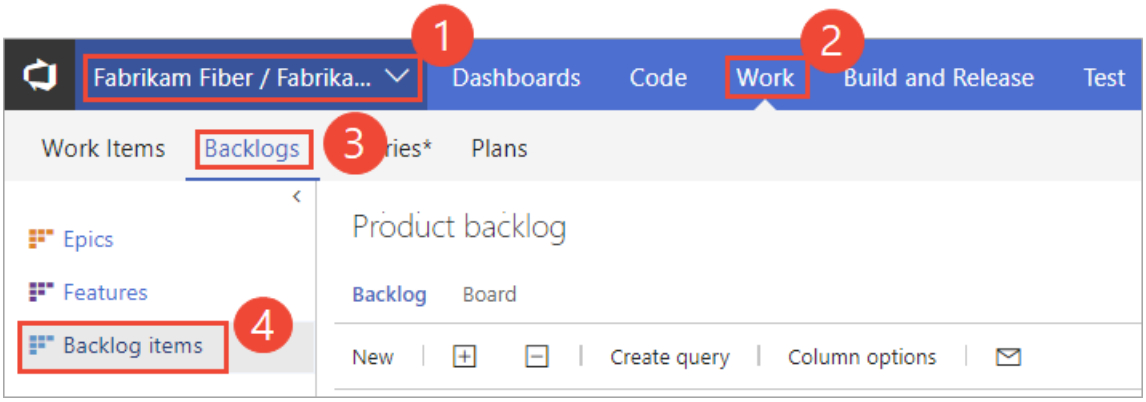
TIP

If your team has been working for several sprints, you can gain an idea of your team's velocity from the Velocity widget.

The tool draws lines for each future sprint selected by the team. The Forecast lines show how much work your team can complete in future sprints. Typically, items above the first line are already in progress for the current sprint. Items that fall between the first and second forecast lines indicate what can be completed in the named sprint.

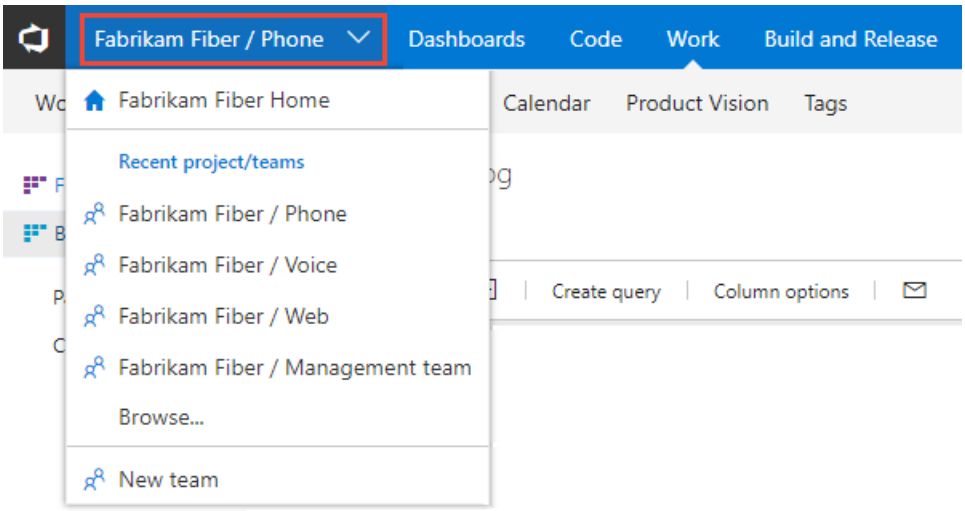
To forecast your product backlog, complete the following actions.

1. From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).



   You can only forecast the product backlog of Stories, Backlog items, or Requirements.

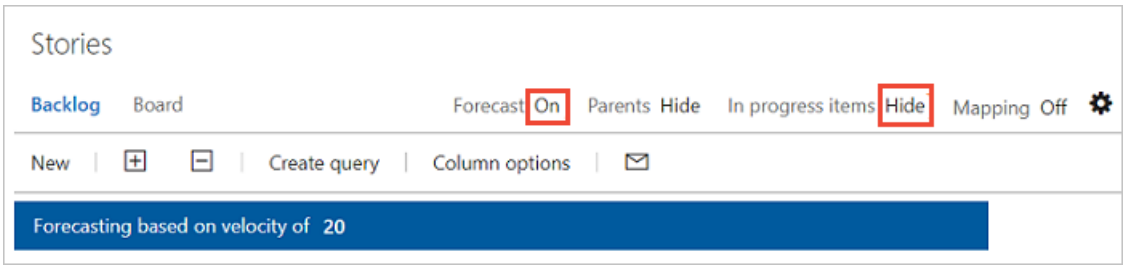
   To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



> **NOTE**
> If you work with several teams, and each team wants to work with their own backlog, velocity chart, and forecast tool, you can create additional teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items to only include those whose assigned area paths and iteration paths meet those set for the team.

2. (Optional) To choose which columns should display and in what order, choose **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.

3. Set **Forecast** to **On** and enter your team's predicted velocity. If the Forecasting bar doesn't appear, set **Parents** to **Hide**.



4. Set **In progress** items to **Hide** to hide those items that won't be counted in the forecast. The forecast tool ignores Scrum items set to Committed or Done and Agile and CMMI items set to Active, Resolved, or Completed.

   The tool draws lines for each future sprint selected by the team. The Forecast lines show how much work your team can complete in future sprints. Typically, items above the first line are already in progress for the current sprint. Items that fall between the first and second forecast lines indicate what can be completed in the named sprint.

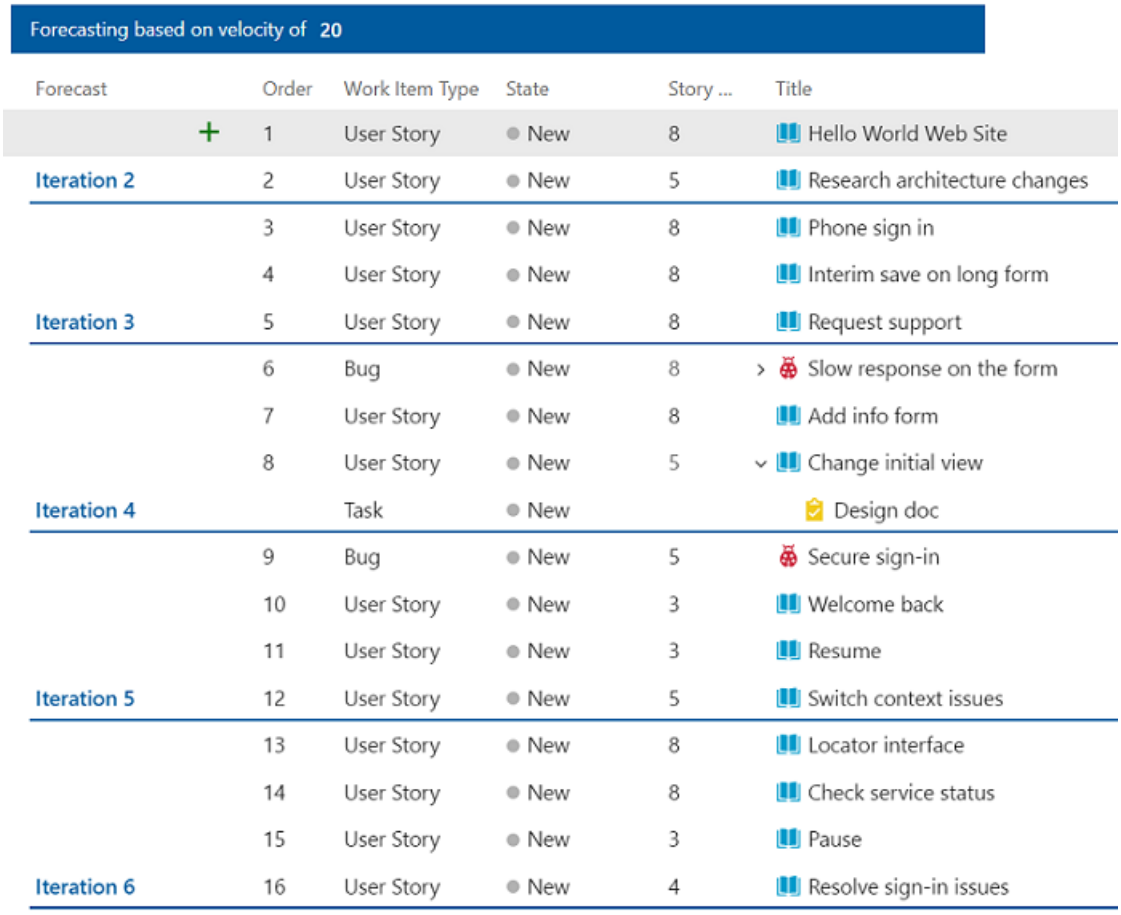
Review the forecast results

- Check the results manually to understand discrepancies in what you expect and what the forecast tool displays.
- Check the amount of effort (Effort, Story Points, or Size) forecasted per sprint.
- Question forecast results where the effort of an item is near to, or greater than, team velocity.

In this example, a Velocity of 20 is used. The forecast tool limits the number of items that are shown between the forecast lines to those items that can be completed within the sprint or using unused velocity points from the previous sprint.

The forecast tool shows between two and four items can be worked on during Iterations 2 through 6 based on the number of Story Points you assigned to each user story or bug. The forecast logic carries over velocity points from one sprint to the next.

- **Iteration 2**: 13 Story Points, items 1 and 2 can be completed; 7 velocity points are carried over to the next sprint
- **Iteration 3**: 24 Story Points, items 3 through 5 can be completed; 3 (=20+7-24) velocity points are carried over to the next sprint
- **Iteration 4**: 21 Story points, items 6 through 8 can be completed; 2 (=20+3-21) velocity points are carried over to the next sprint
- **Iteration 5**: 16 Story points, items 9 through 12 can be completed; 6 (=20+2-16) velocity points are carried over to the next sprint
- **Iteration 6**: 23 Story points, items 13 through 16 can be completed; 3 (=20+6-23) velocity points are carried over to the next sprint



Determine the velocity needed to complete all items in the backlog

Another way to use the forecast tool is to enter different velocity values until all the backlog items are completed within a given set of sprints. This forecast provides an estimate of what velocity is required to complete your backlog of items.

You can then assess the delta between the current team's velocity and the required velocity. The delta helps determine what other resources are required to meet production demands within a required time.

Next steps

Assign work to a sprint

Related articles

- Team velocity
- Define iteration paths (sprints) and configure team iterations
- Use the taskboard to track work during your sprint
- Monitor the sprint burndown chart to determine if your team is on track to complete the sprint plan

Configure your backlog view in Azure Boards

10/4/2022 • 11 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Your backlogs are designed to support many project management tasks. Chief among them are:

- Define work to be done
- Prioritize work
- Group work into a hierarchical view
- Assign work to iterations
- Forecast work

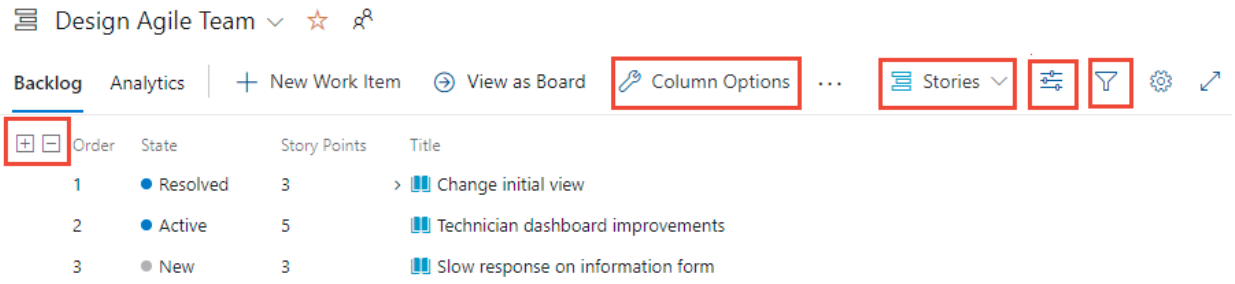
Each backlog—product or portfolio—is a tool you share with your team members. When you add backlog items, prioritize items, or link work items using parent-child links, team members see the changes when they refresh their backlog.

To effectively perform select tasks, it's good to know how to set your view options to support those tasks.

> **TIP**
> You can't sort your backlog on a column. To view a sorted listed, select **Create query** from your backlog. Save and open the query. Modify the query if needed to be a flat list query. You can then sort the query results. To learn more about queries, see Use the query editor to list and manage queries.

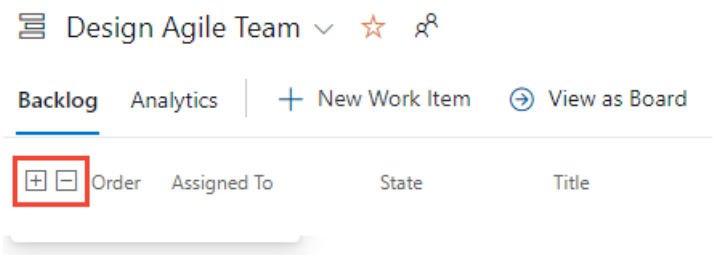
Backlog configuration options

You have the following tools to configure your backlog view: **Expand/Collapse one level**, **Column Options**, **Backlog level selector**, **View options**, and **Filter** toolbar. The options you set for each backlog level—stories, features, epics (Agile process) or product backlog items, features, epics (Scrum process)—are distinct. The options you set persist until you change them.



Expand and collapse the hierarchy

The default view when you select a backlog level is to show the collapsed view. Only those items associated with the backlog level selected are shown. You can expand and collapse the hierarchy by using the ⊞ and collapse ⊟ icons to expand or collapse one level of the hierarchy. Your selection doesn't persist once you switch to a different page or view.



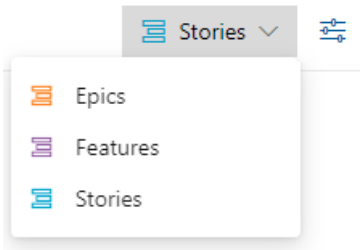
Backlog level selector

The number of backlog levels available to you are set by your team administrator and may have been customized to add custom work item types or backlog levels. Each backlog automatically applies the filters associated with the **Area Paths** and **Iteration Paths** selected for the team.

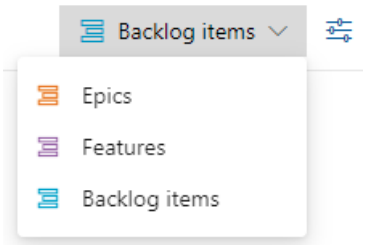
> **NOTE**
> Prior to using the tools described in this article, we recommend that you review Set up your project's backlogs and boards to ensure you have configured your backlog to support your team's needs.

From the **Backlogs** page, you can select the product backlog or a portfolio backlog. You select a backlog from the backlog level selector next to the **View options** icon. The labels within this selector vary depending on the process selected for your project, customization made to that process, and configurations made by your team administrator as shown in the following images.

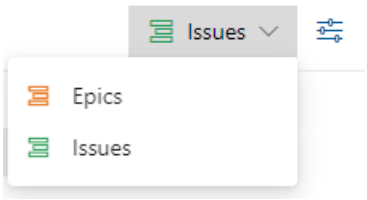
Agile process



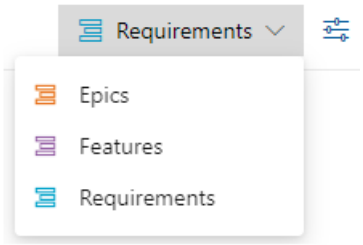
Scrum process



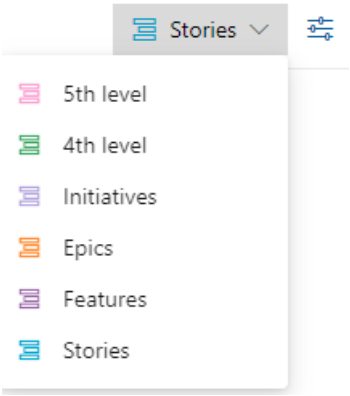
Basic process



CMMI process



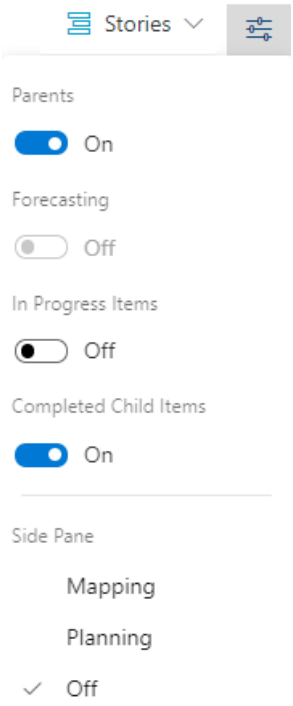
Customized process



For information on team configuration of backlog levels, see Select backlog navigation levels for your team.

View options menu

The **View options** menu controls the following options.



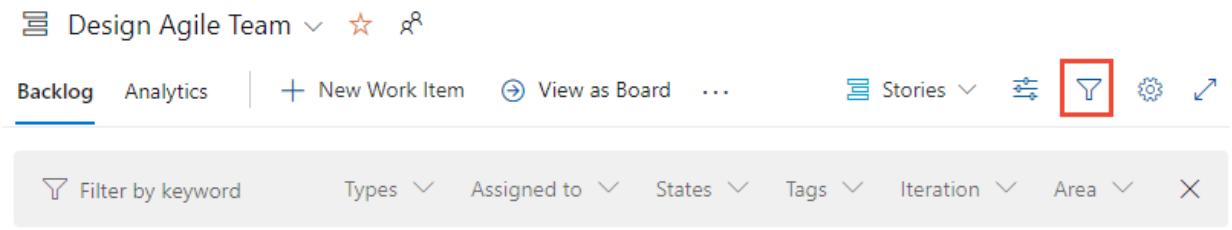
- **Parents**: Show the hierarchical grouping of parent-child work items. Useful when adding child work items, reparenting a work item, or displaying rollup columns.
- **Forecasting**: Show the **Forecast** tool and forecast lines. The **Forecast** option only appears for the first-level backlog and depends on the assignment of **Story Points**, **Effort**, or **Size**.
- **In Progress Items**: Show items whose workflow **State** corresponds to an *In Progress* workflow state category. If you turn the **In Progress** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or a custom workflow state defined in the In Progress state category won't appear in the backlog. To learn more about category workflow states, see How to use workflow states and state

categories.

- **Completed Child Items**: Show child items that have been completed. Typically you turn this **On** when reviewing reviewing a rollup column.
- **Mapping**: Shows the **Mapping** pane to support drag-and-drop linking of work items to parent items. The **Mapping** option doesn't appear when you've selected the highest backlog level configured for your team.
- **Planning**: Shows the **Planning** pane to support drag-and-drop of work items to **Iteration Paths**.

Filter bar

Turn on filtering when you want to find one or more work items based on a keyword, tag, assignment, or other field you display using **Column Options**. You enable the filter feature by choosing **Filter**.



Filtering displays a flat list of all items in the hierarchy when you have selected to show **Parents**. The hierarchical grouping is restored once you dismiss the filter toolbar. The filter toolbar persists until you dismiss it.

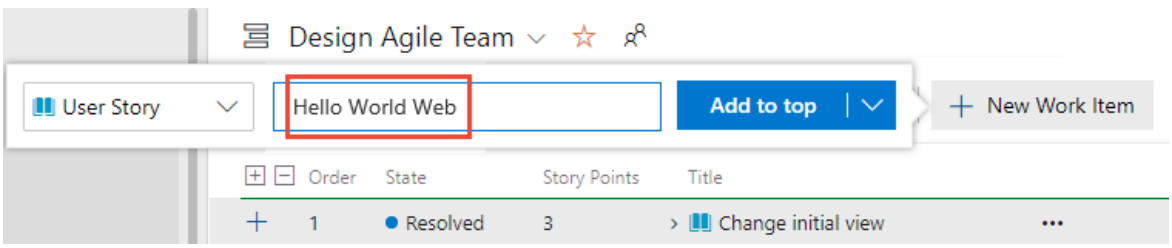
To learn more, see Filter backlogs, boards, and plans.

Define backlog items

It's useful to be able to add work items quickly and refine details later when more information becomes available. Often you use a query to help triage work added to the backlog to review, refine, and add details to work items that you entered through your backlog.

To quickly define many items to a backlog, perform the following steps.

1. Select the backlog level that you want to add items to.
2. From the **View options** menu, turn the slider for **Parents** and **Forecasting** to **Off**.
3. (Optional) turn **In Progress Items** on or off.
4. (Optional) Minimize the number of columns displayed on your backlog, or select those fields you want to view.
5. Choose **New Work Item**, enter a title, choose **Add to top** or **Add to bottom**, and then press **Enter**. We recommend you add items to the bottom of the backlog if your team has a process for prioritizing backlog items.



Work items are automatically assigned the default **Area Path** and **Iteration Path** selected for the team.

> **NOTE**
> If you have **Stakeholder** access , you can only add work items to the bottom of the backlog. For details, see Stakeholder access quick reference.

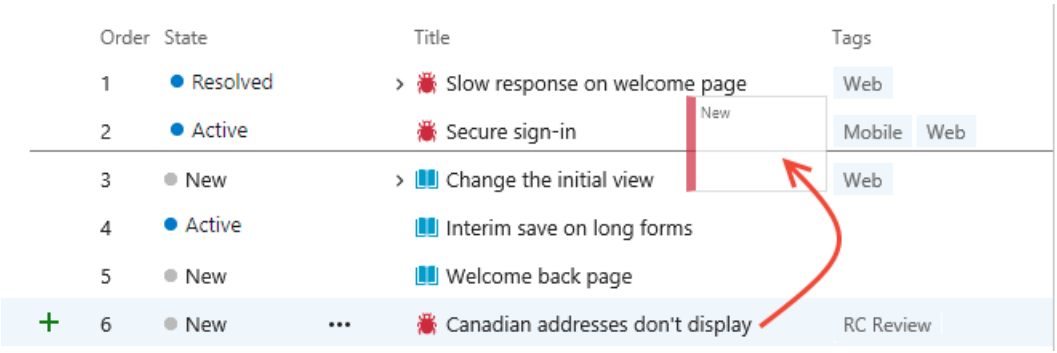
6. Continue entering a **Title** and pressing the **Enter** key to define more backlog work items.

To learn more, see Create your product backlog and Define features and epics.

Prioritize your product backlog

If your team follows Agile or Scrum methods, they'll want to prioritize the backlog to make sure that the most important work to complete is situated at the top of the backlog. To prioritize a backlog, follow these steps.

1. Select the backlog level you want to prioritize.
2. Turn the **Parents** view option **Off**.
3. Drag the work items up or down within the backlog. Or, if you prefer to use the keyboard, hold down the Alt key and use the up and down arrows.



To reorder a backlog, you must have **Basic** or higher level access. If you have Stakeholder access, you can't reorder backlog items. For more information, see Stakeholder access quick reference.

> **NOTE**
> Changes you make to the priority impact all backlog items. When other team members refresh their backlog, they'll see the new priorities. A background process updates the **Stack Rank** (Agile, Basic, and CMMI processes) or **Backlog Priority** (Scrum process) fields. These fields are used by the system to track the relative ranking of items on the product, feature, epic, or other portfolio backlog. By default, these fields don't appear on the work item form. The priority ranking is assigned separately to each backlog level, which you can check by adding the field to a backlog and viewing a hiearchical list.

Backlogs that participate in portfolio management or that contain nested same-type child items might not allow you to reorder the items. For more information, see these articles:

- Backlogs, portfolios, and Agile project management, Work with multi-team ownership of backlog items
- Fix reordering and nesting issues

Prioritize a portfolio backlog

The method for prioritizing a portfolio backlog is similar to that described for a product backlog. The main difference is that you prioritize child items within each portfolio item. Each backlog level—**Stories**, **Features**, **Epics**—supports priority ordering distinct from every other level.

Prioritize the portfolio items:

1. Select the portfolio backlog level you want to prioritize.
2. Turn the **Parents** view option **Off**.
3. Drag the work items up or down within the backlog.
4. Within each item, you can expand to see child items and drag these items into priority order.

Prioritize child items:

1. Expand each portfolio item.
2. Drag each child item up or down within the expanded item.

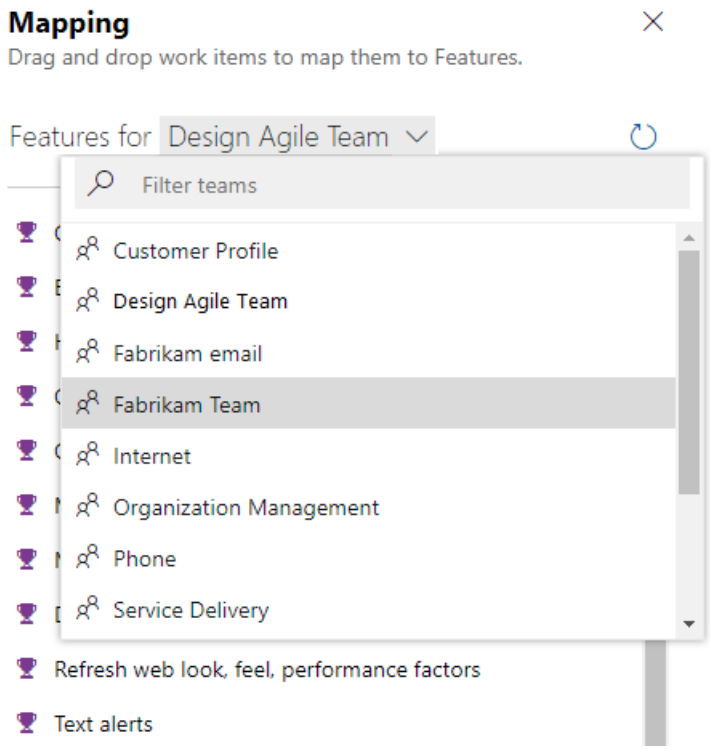
> **NOTE**
> To reorder a backlog, you must have **Basic** or higher level access. If you have Stakeholder access, you can't reorder backlog items. For more information, see Stakeholder access quick reference.

Link work items to a parent (mapping)

You can drag items to quickly link one or several work items to a parent portfolio item.

> **TIP**
> Prior to opening mapping work items, add the portfolio backlog items you want to link work items to and prioritize them. The **Mapping** pane lists the portfolio backlog items in priority order.

1. Select the backlog level you want to link to parent items. For example, choose **Stories** to link to **Features**.
2. Open **View options** and choose **Mapping**. The Mapping pane opens. By default, the pane lists the next-level up portfolio items for the current team.
3. (Optional) To map items to parent items owned by a different team, choose it from the team selector in the **Mapping** pane as shown in the following image.



4. Drag work items from the backlog to the portfolio item listed in the **Mapping** pane. The system creates a parent-child link in the background. The backlog item turns bold and then unbold as the system saves the changes.

   Note, you can select multiple backlog items and drag them to a portfolio item. To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the **Ctrl** key. Then, you can drag the selected items.

5. (Optional) You can also drag a backlog item within the expanded hierarchical view to reparent a work item.

> **TIP**
> To view the work items that are unparented, you can add the **Parent** field as a column. The **Title** of the parent item is listed for those items that have been linked to a parent.

To learn more, see Organize your backlog and map child work items to parents.

Add child items to a portfolio backlog item

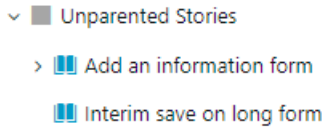
2. Select the portfolio backlog level, such as **Features**, that you want to add items to.
3. Choose + **Add User Story, Bug** for the feature you want to add the child item to as shown in the following image. Your labels may differ based on process and customizations.
4. In the work item form that appears, enter a **Title** and any other required fields or details. Save the work item to close it.

To learn more, see Define features and epics, add child items.

View or find unparented work

To view or find unparented work:

1. Select the backlog level you want to inspect for unparented items.
2. Open **View options** and choose **Parents**.
3. Scroll to the bottom of the backlog and expand **Unparented Stories**, **Unparented Features**, or similar entries. Unparented work items are listed under these entries.



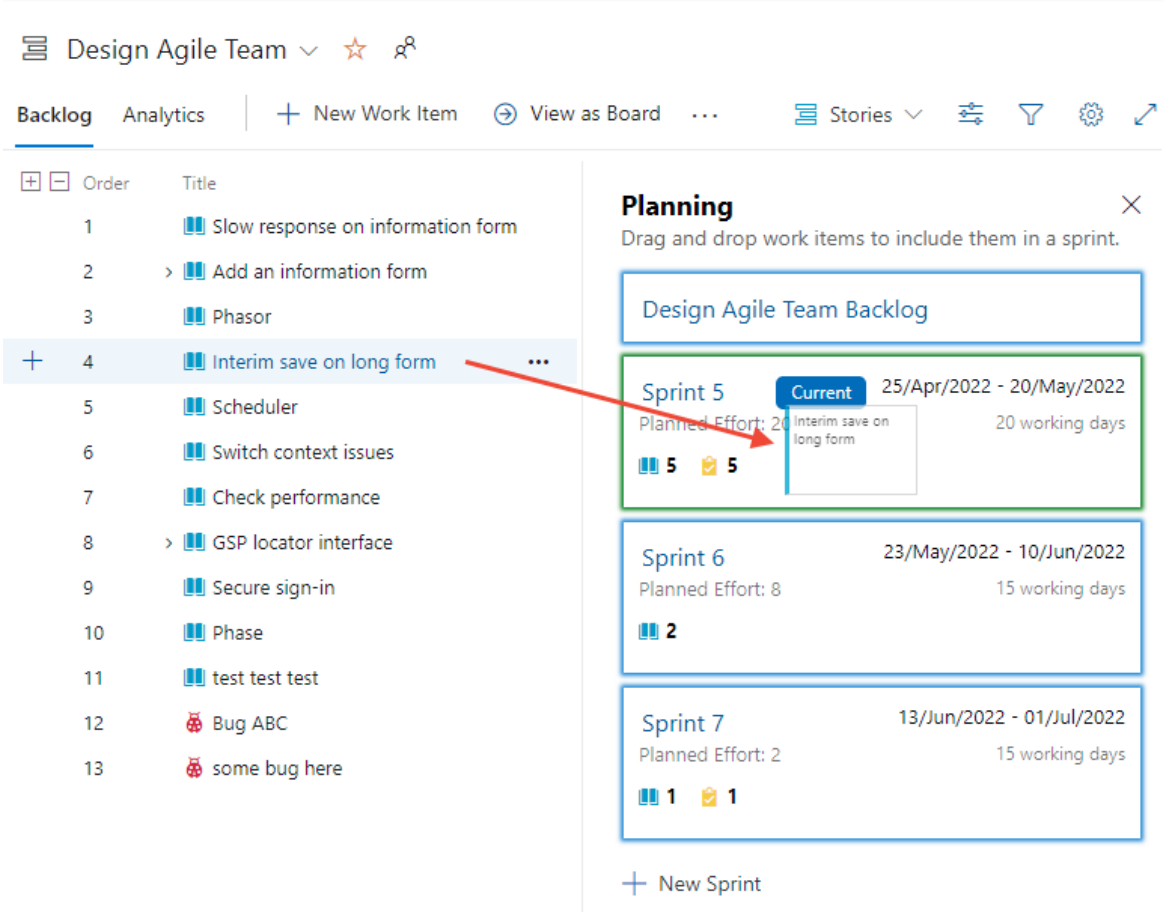
Assign work to a sprint or iteration

Similar to using the **Mapping** pane, you can use the **Planning** pane to assign one or more work items to an **Iteration Path** or sprint.

1. Make sure all **Iteration Paths** have been selected for your team that you want to show in the **Planning** pane.
2. Choose the backlog level that contains the work items you want to assign.
3. Open **View options**, turn off **Completed Child Items** and choose **Planning**.
4. Drag work items from the backlog to the portfolio item listed in the **Mapping** pane. The system creates a parent-child link in the background. The backlog item turns bold and then unbold as the system saves the

changes.

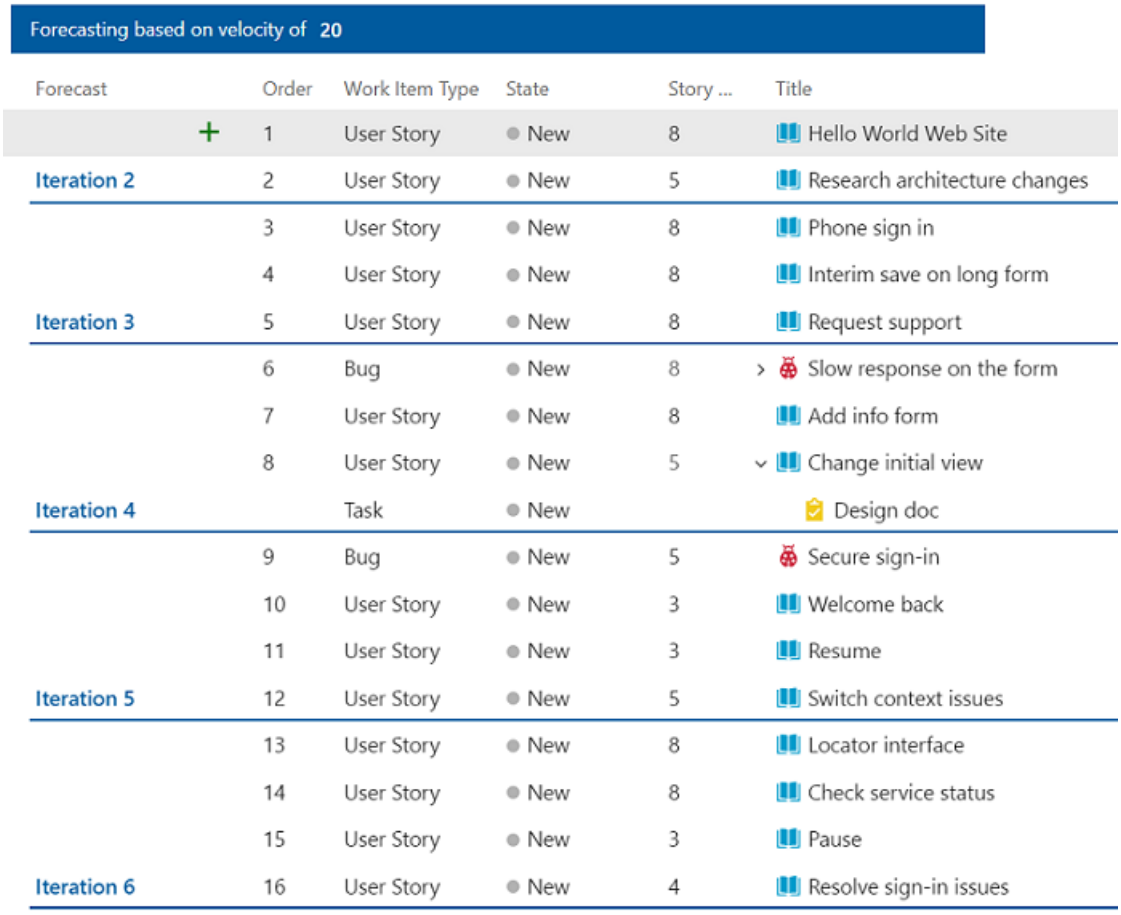
Note, you can select multiple backlog items and drag them to a portfolio item. To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the **Ctrl** key. Then, you can drag the selected items.



Forecast a backlog

The **Forecast** tool is only available for the product backlog. Use these steps to forecast your backlog. To use the **Forecast** tool, you must have **Basic** or higher level access. This feature isn't available to users granted **Stakeholder** access.

1. Make sure that future **Iteration Paths** have been selected for your team.
2. Choose the backlog level for your team.
3. (Optional) Add the **Story Points**, **Effort**, or **Size** field as a column based on the process your project uses.
4. Open **View options** and turn off **Parents** and **In Progress** items, and **Completed Child Items**. Turn on **Forecast**.
5. Enter a velocity estimate in the **Forecasting based on velocity** of box.
6. Review the forecast lines that appear, similar to the ones shown in the following image.



The forecast tool doesn't reference any iteration assignments made to the product backlog items.

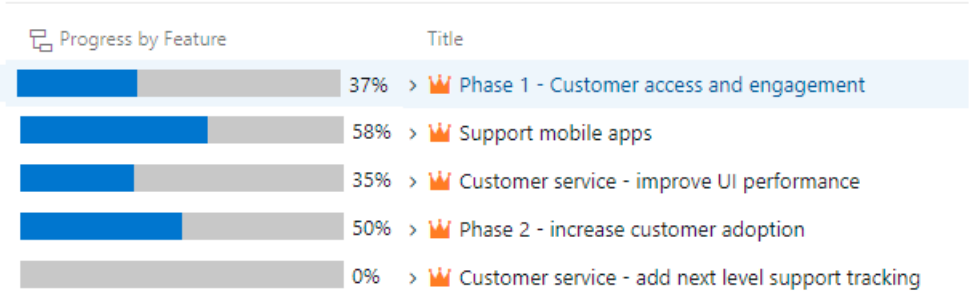
> **TIP**
> You can drag items to reprioritize them with forecast lines shown. You can also use the **Planning** pane with the **Forecast** tool turned on.

To learn more, see Forecast your product backlog.

Review progress made to your backlog

You can add a rollup progress bar, count of work items, or sum of any integer or numeric field as a column to the backlog. This option allows you to review progress made to parent work items based on the completion of their child items. These child items can be Tasks for User Stories, User Stories and Bugs for Features, or Features for Epics.

1. Select the backlog level you want to view progress on
2. Open **View options**, show **Completed Child Items**
3. Open **Column Options**, choose **Add a rollup column**, and select the progress bar or count to display.



It can take several moments for the progress bar or count to appear.

To learn more, see Display rollup progress or totals.

Related articles

- Set up your project's backlogs and boards
- Create your product backlog
- Define features and epics
- Organize your backlog and map child work items to parents
- Configure team settings

Bulk modify tools

- Bulk modify (web)
- Bulk add or modify (Excel)
- Import or update work items in bulk by using CSV files

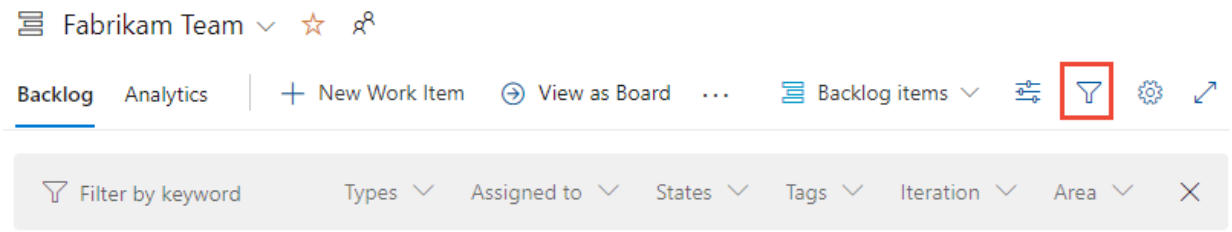
Interactively filter backlogs, boards, queries, and plans in Azure Boards

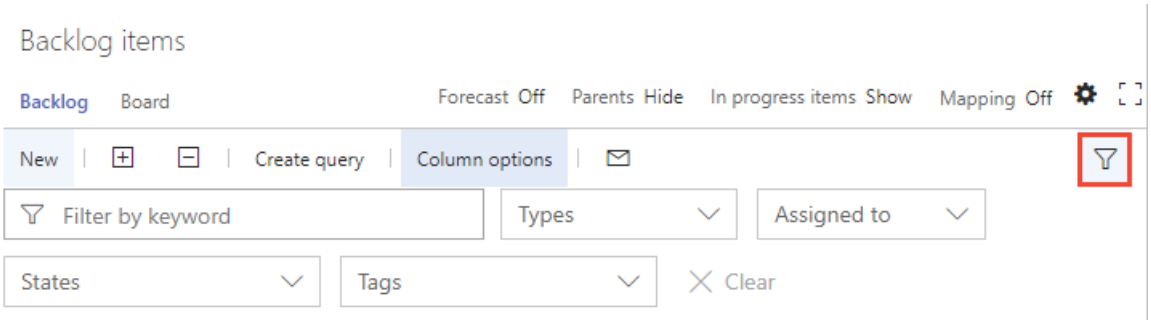
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

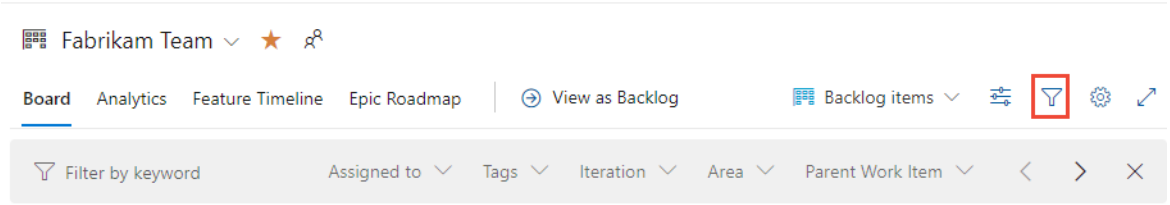
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

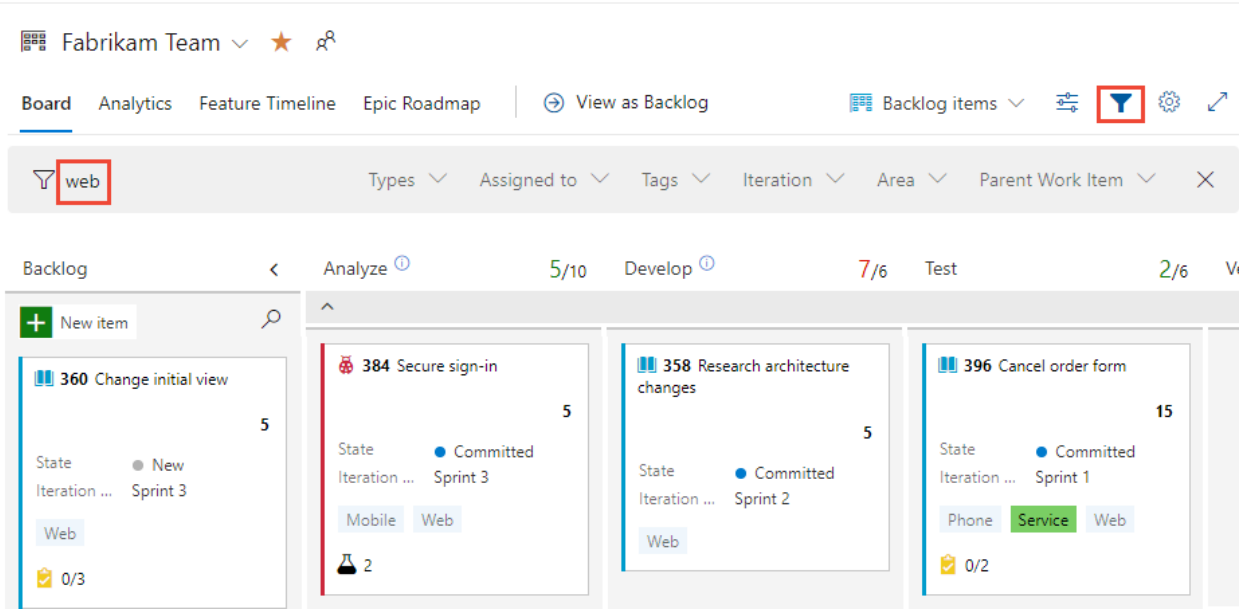
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

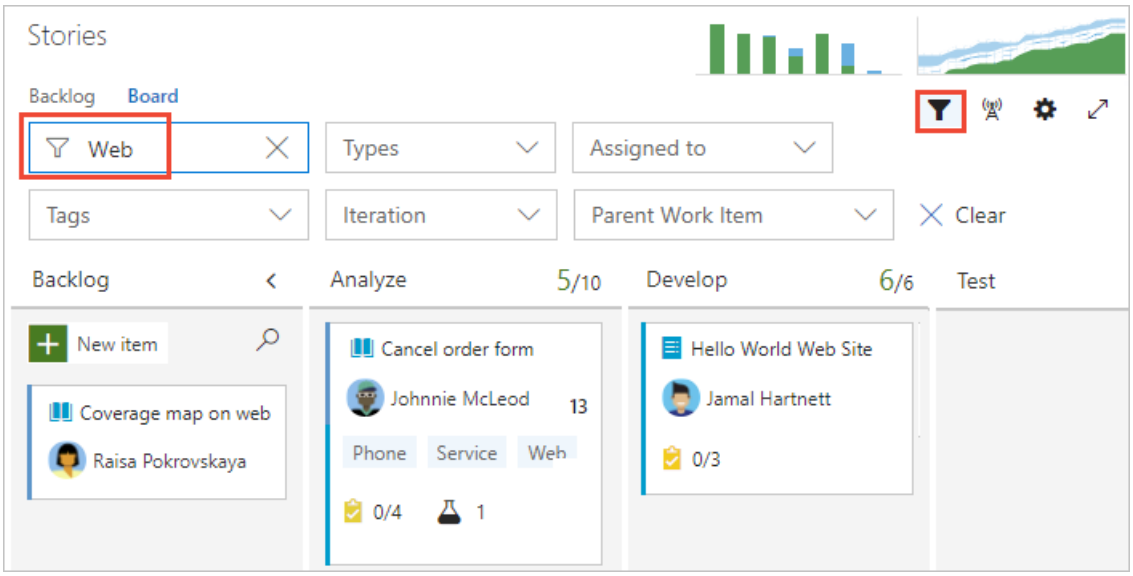
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

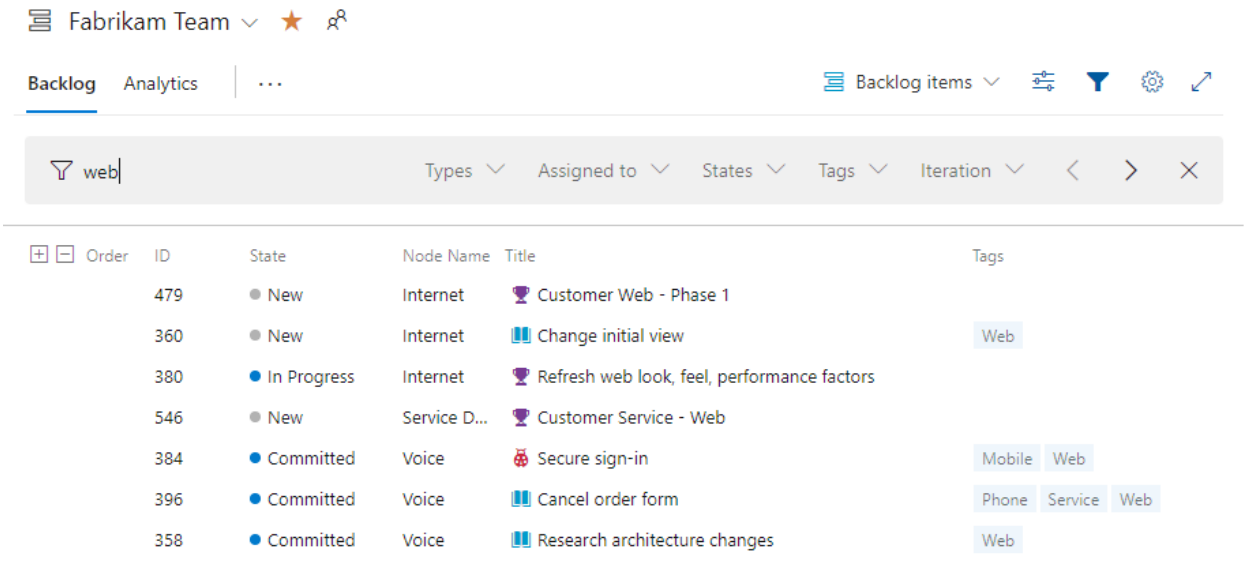


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

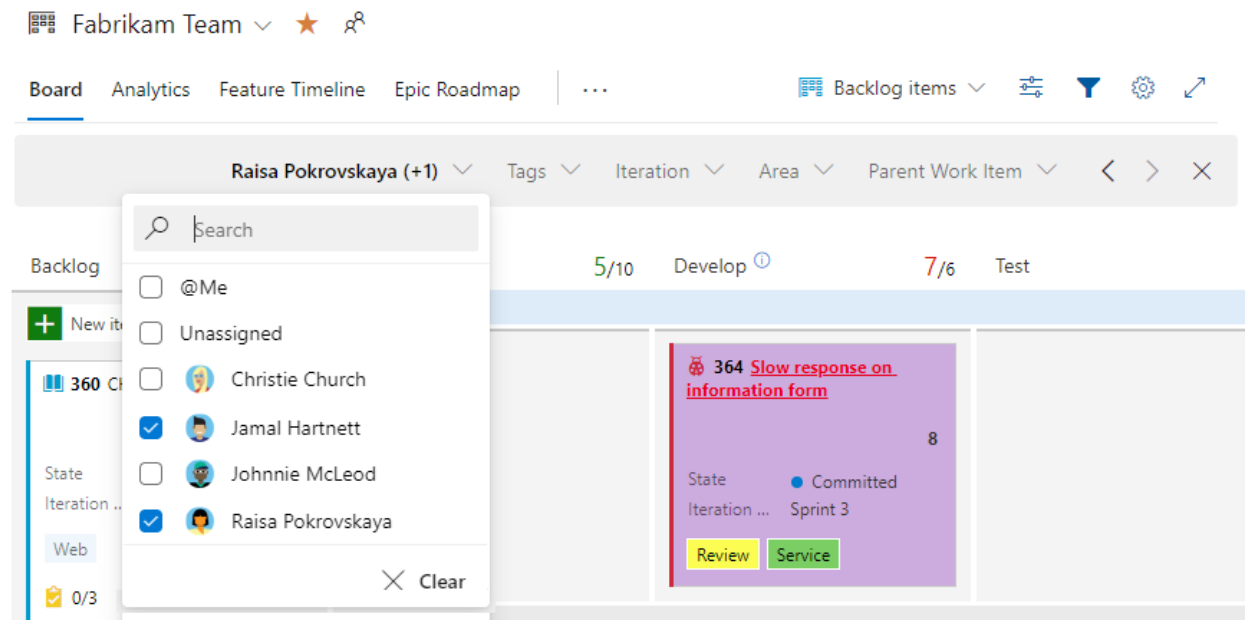
Features board

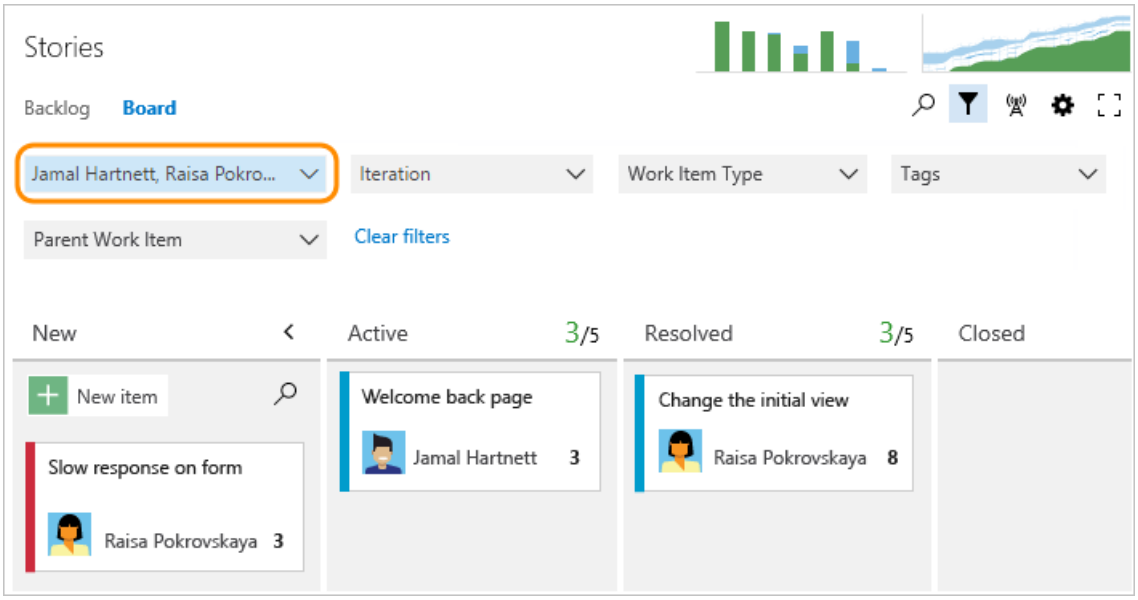
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

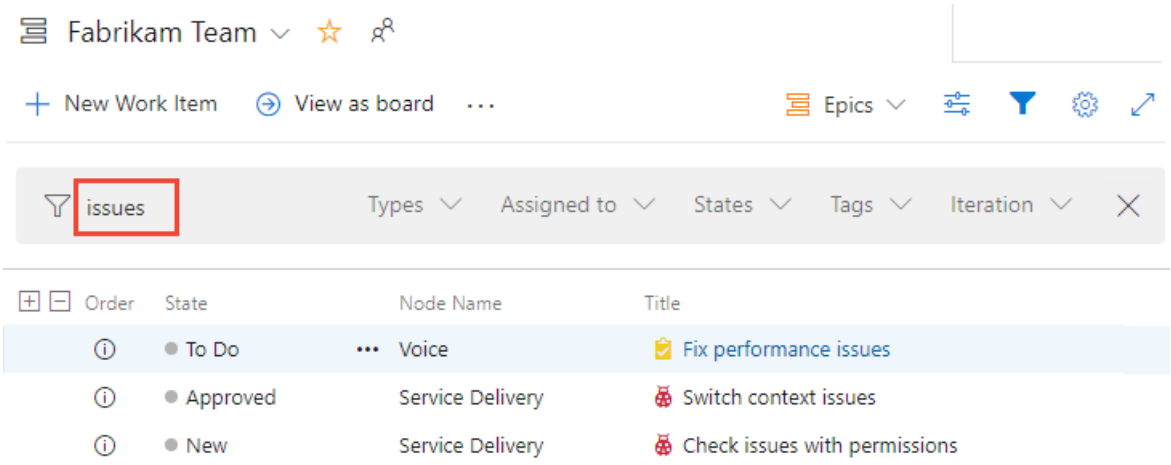
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

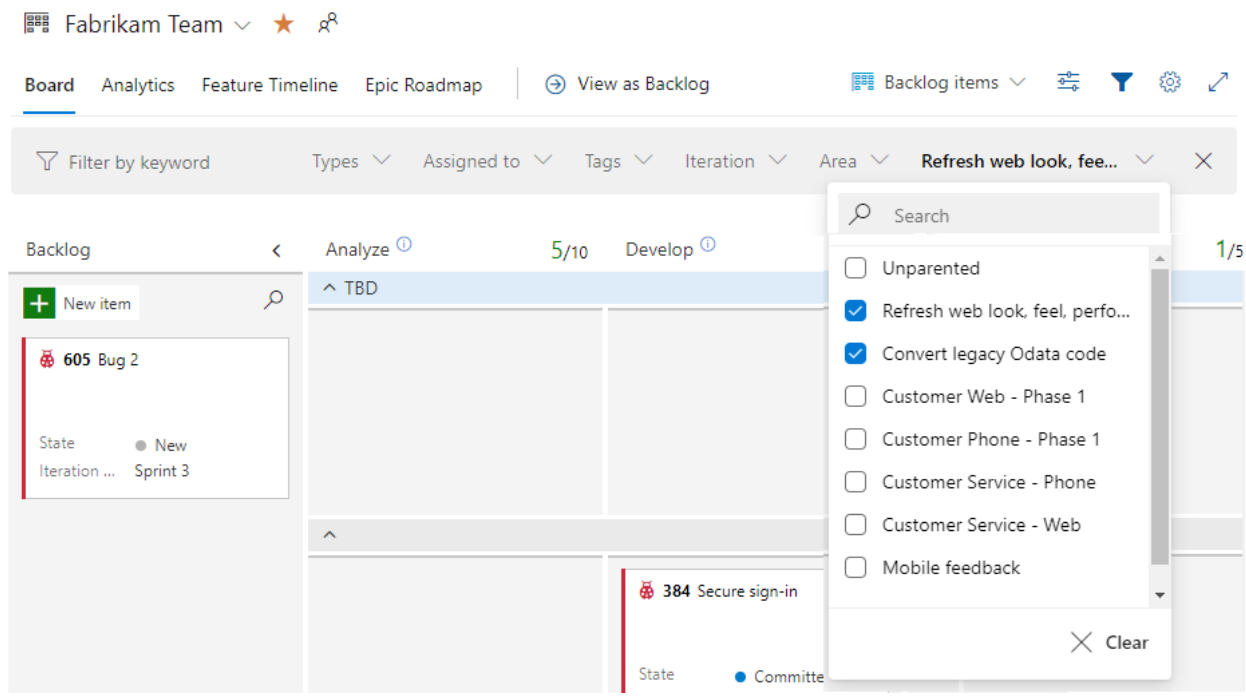
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

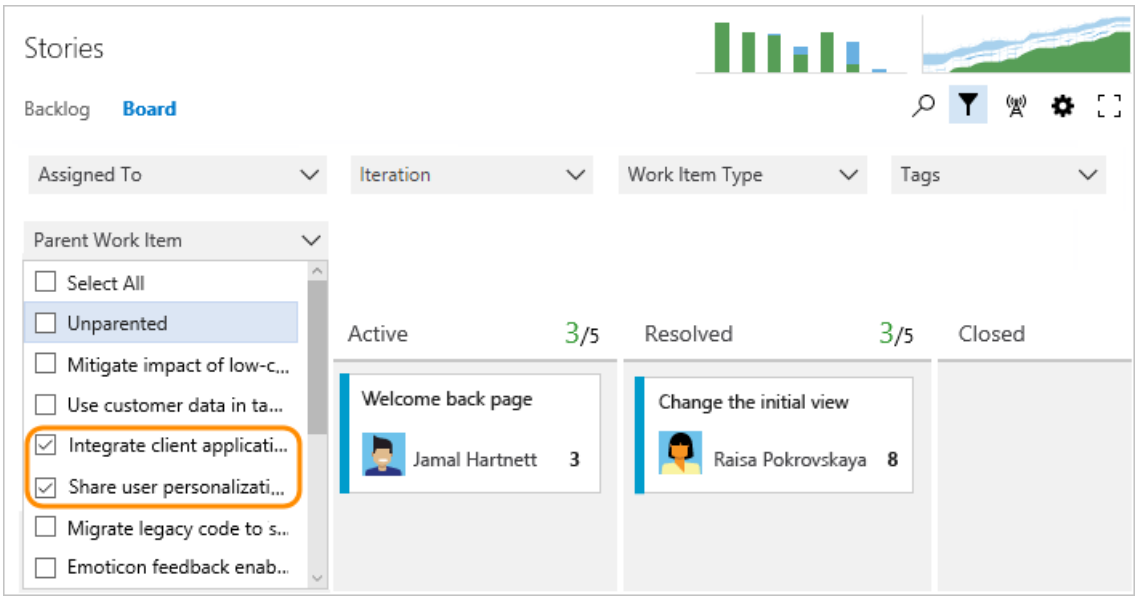
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



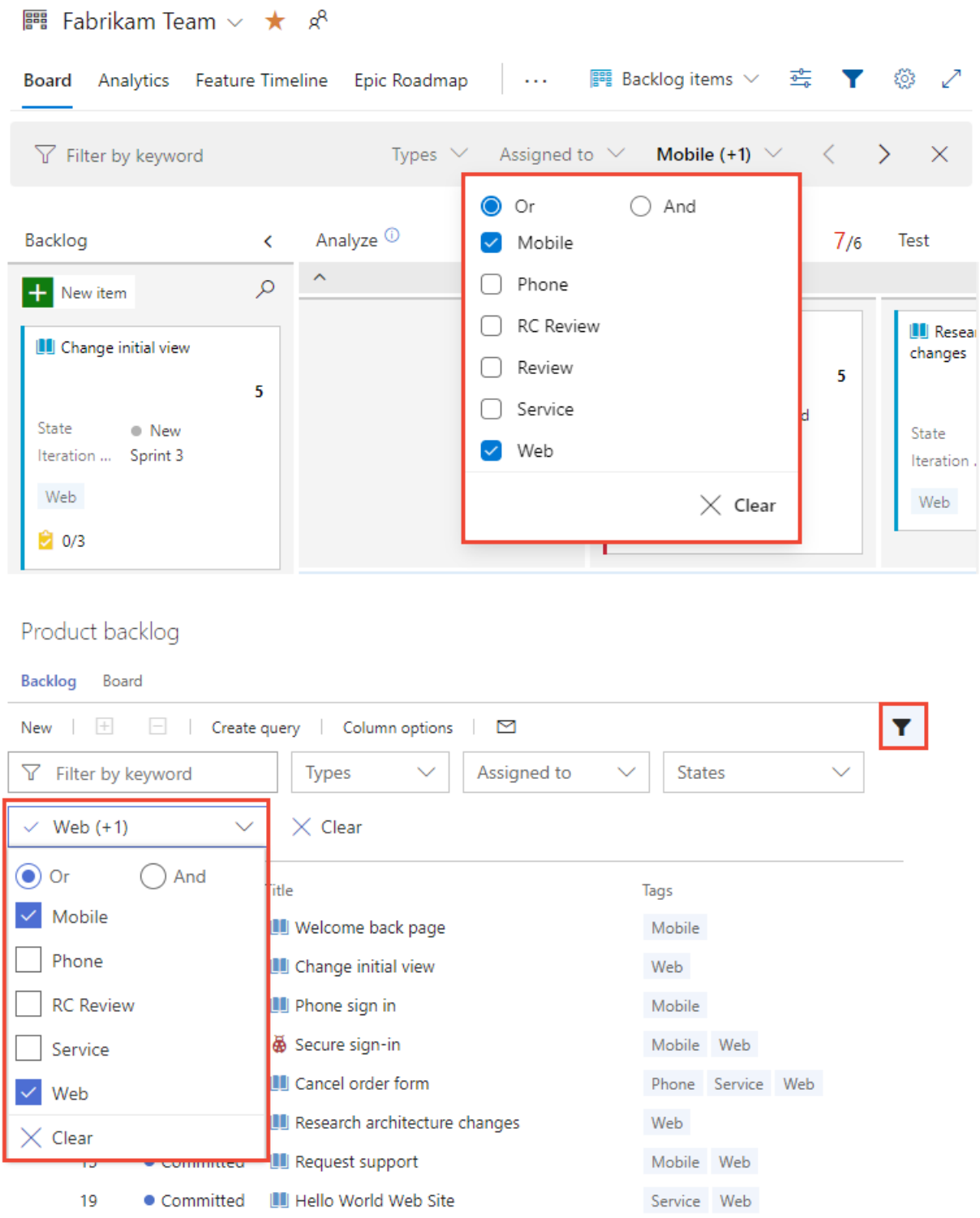


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

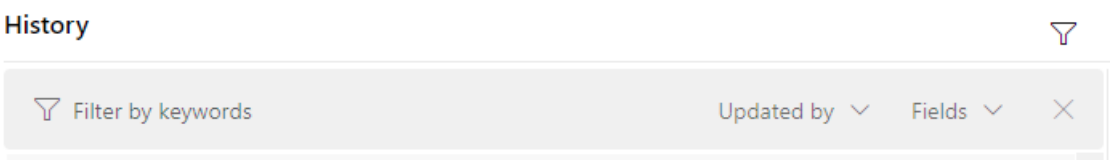


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Set up your project's backlogs and boards in Azure Boards

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

In most cases, you can start using your product and portfolio backlogs once your project is created. A default team is created along with associated backlogs and boards. You can start adding work items to your product backlog using the Backlog or Board.

However, you may need to ensure you've configured your backlogs and boards correctly. Ensure the configuration if you've added a team and want to start using the team backlogs and boards. Changes may be made to a project or team configuration over time. These changes can influence the work items that appear on your backlog and boards.

For an overview of the tools associated with your team, see Manage and configure team tools.

Default backlog and board work items

The first thing you need to understand is that your product **Backlog** and **Board** display work items that meet the following criteria:

- Work item type belongs to the Requirements category. The types differ depending on the process selected for your project:
  - Basic : Issue, Backlog name=**Issues**
  - Agile: User Story, Backlog name=**Stories**
  - Scrum: Product Backlog Item, Backlog name=**Backlog items**
  - CMMI: Requirement, Backlog name=**Requirements**
- Work item **Area Path** matches one of the selected team's Area Paths
- Work item **Iteration Path** is under the team's Default Iteration Path

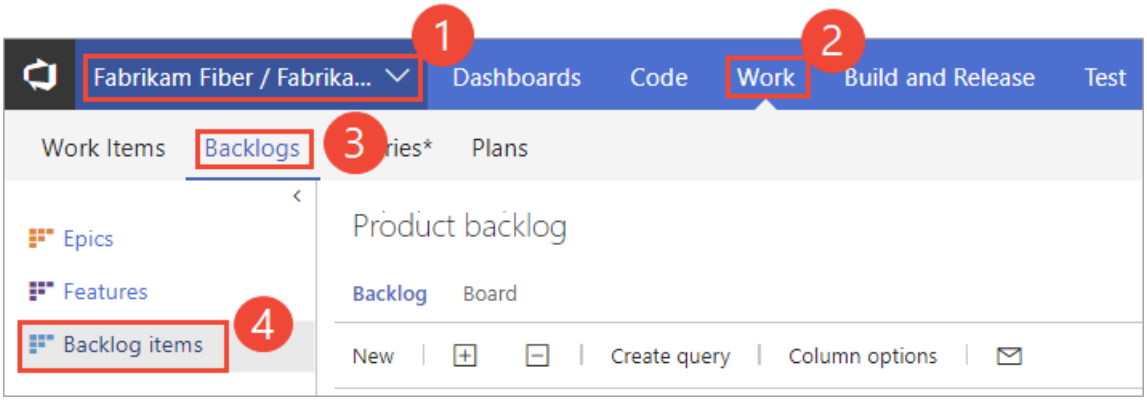
> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

- Work item type belongs to the Requirements category. The types differ depending on the process selected for your project:
  - Agile: User Story, Backlog name=**Stories**
  - Scrum: Product Backlog Item, Backlog name=**Backlog items**
  - CMMI: Requirement, Backlog name=**Requirements**
- Work item **Area Path** matches one of the selected team's Area Paths
- Work item **Iteration Path** is under the team's Default Iteration Path

You can determine the work item types that belong to your Requirements category. Determine the items by opening your product Backlog and checking the product backlog name.



As shown in the following image, (1) choose the team, (2) **Work**, (3) **Backlogs**, and then the product backlog.



Look up your team's Area Path(s) and Iteration Paths. For more information, see Define area paths and assign to a team and Define sprint paths and configure team iterations.

Default sprint backlog and Taskboard work items

Your sprint backlog and Taskboard apply the filters associated with your team's default backlog and board work items along with the Iteration Path you select.

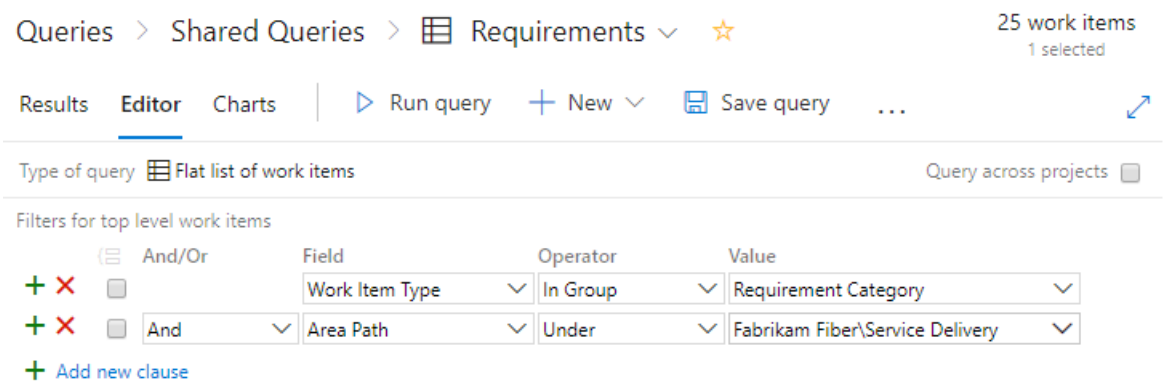
You can only select Iteration Paths that have been preselected by your team.

Your sprint backlog displays only those work items assigned to the selected sprint. Child tasks assigned to other sprints aren't displayed.

Review checklist for work items, backlogs, and boards

If you don't see the work items you expect on your product Backlog or Kanban board, complete the following checks:

1. Make sure you've selected the team backlog or board of interest. To learn how, see Use breadcrumbs and selectors to navigate and open artifacts.
2. Create a query of your backlog items, specifying the work item types that belong to your Requirements category and the Area Path associated with your team, for example:



3. Add the **State**, **Area Path**, and **Iteration Path** fields to the column options.

4. Check the query results and that the values of the work items you expect to show up on your backlog meet these criteria:

   - **Area Path** belongs to your team's area path(s)
   - **Iteration Path** belongs under your team's default iteration path
   - **State** isn't Closed, Completed, Done, or Removed.

> **NOTE**
> You can also filter your product backlog to show or hide work items that are in an **In Progress** state category, corresponding to an Active, Resolved, Committed, Doing workflow state.

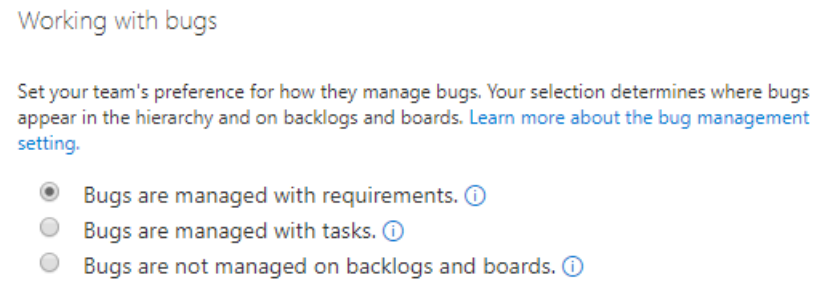
Add bugs to your backlogs and boards

For all processes except the Basic process, each team manages the way bugs are tracked. Track bugs in the Requirements category because they show up on the Backlog and Kanban board or the Tasks category. They can also show up on the Taskboard or the Bugs category where they don't appear on either backlogs or boards.

> **NOTE**
> Bug work item types aren't available with the Basic process. The Basic process tracks bugs as Issues and is available when you create a new project from Azure DevOps Services or Azure DevOps Server 2019.1 or later versions.

Each team can manage the way bugs are tracked. You can track bugs as belonging to the Requirements category. Those bugs show up on the Backlog and Kanban board or the Tasks category. They can show up on the Taskboard or the Bugs category where they don't appear on either backlogs or boards.

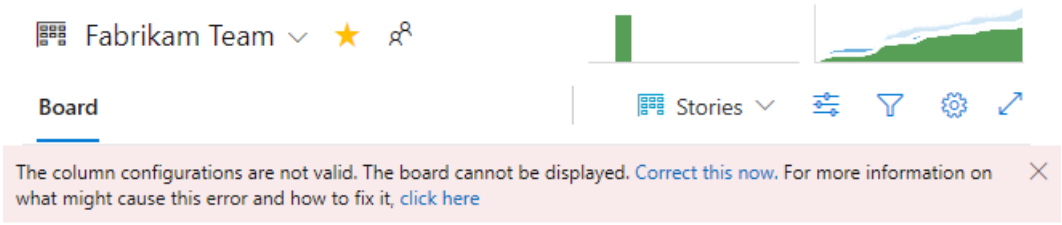
If you want bugs to show up on your Backlog and Board, choose **Bugs are managed with requirements**.



For details, see Show bugs on backlogs and boards.

Correct your Kanban board configuration

If you see the following error when you open your Kanban board, you need to correct the configuration. The main reason for this error is that the workflow states of work item types that have been added to the Requirements category aren't mapped to the column.

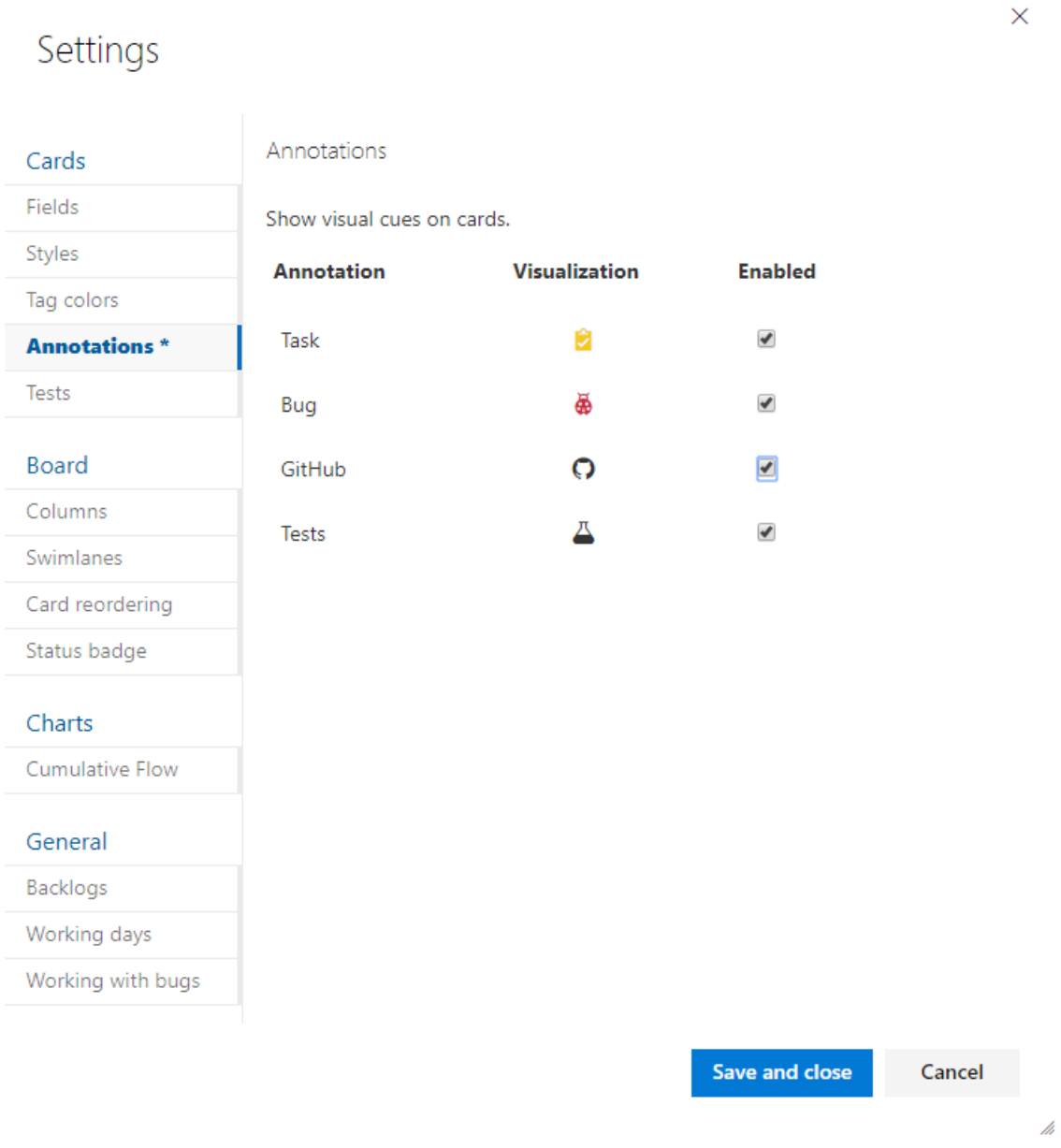


Choose **Correct this now** to open the Settings dialog. To map the workflow states, refer to Add columns to your Kanban board, Update Kanban column-to-State mappings.

Customize your Kanban board checklist items

Checklists are a great way to create work items that are automatically linked with a parent-child link to another work item on a Kanban board. You can customize the work item types that you can add as a checklist by opening the Board Settings, choose **Annotations**, and enable the work item types you want to appear on the board. For details, see Customize cards.

For example, here we've chosen to track bugs along with tasks, and enable Task, Bug, GitHub objects, and Tests to appear within checklists.



NOTE

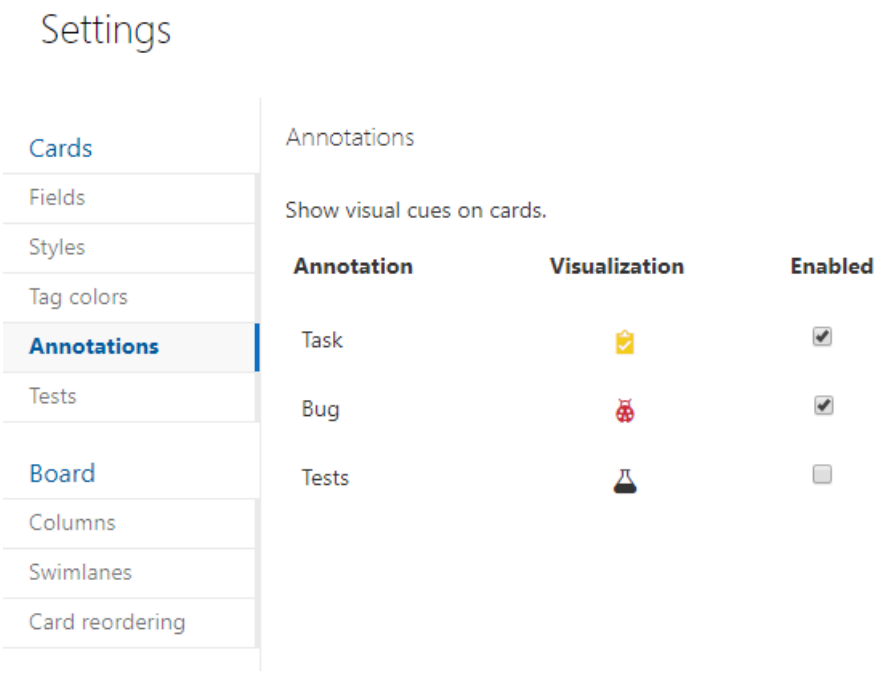
The GitHub annotations require Azure DevOps Server 2019 update 1 or later version.

To learn more about checklists, see the following articles:

- Add tasks or child items as checklists
- Add, run, and update inline tests

- Link GitHub commits, pull requests, and issues to work items

For example, here we've chosen to track bugs along with tasks, and enable Task and Bug to appear within checklists.



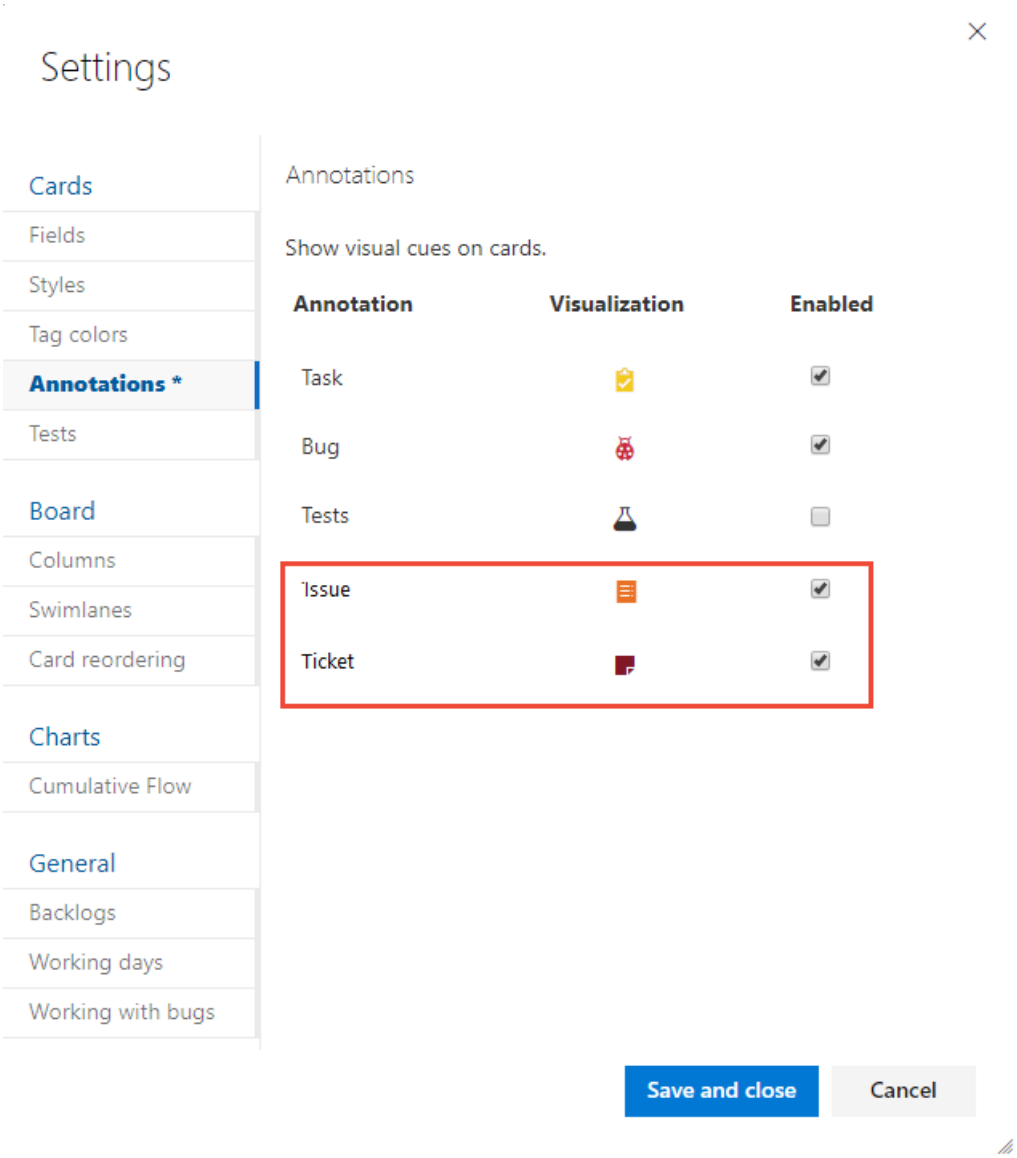
To learn more about checklists, see the following articles:

- Add tasks or child items as checklists
- Add, run, and update inline tests

Add other work item types to your Kanban board checklist

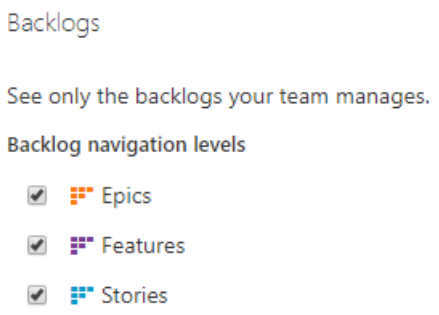
If you added work item types to the Task Category as described in Add custom work item types to your Taskboard later in this article, you can choose if these types appear within a checklist on your product Kanban board. You make this choice by opening **Board Settings**, choose **Annotations**, and enable the work item types you want to appear on the board. You can enable up to five annotations. For details see Customize cards.

For example, here we've chosen to track bugs along with tasks, and we enable Issue and Ticket and Task and Bug. To learn more about checklists, see Add tasks or child items as checklists and Add, run, and update inline tests.



Hide or show backlog levels

Your team can also choose to hide or show one or more backlog levels. Feature teams often manage backlog items, while management teams manage features and epics. In this situation, you can enable or disable a backlog level.



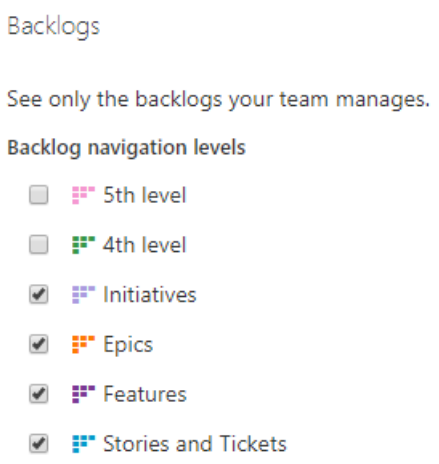
For details, see Select backlog navigation levels for your team.

Add custom work item types to your backlogs and portfolio backlog levels

If you want to track different work item types on your product backlog, you can do that by adding custom work item types and adding them to a specific backlog level.

You can also add custom work item types and add them to portfolio backlogs. You can add up to five portfolio backlogs.

For example, here we've added Initiatives, fourth level, and fifth level work item types to support five levels of portfolio backlogs. We've also added a custom work item type named Ticket and added that to the product backlog.



For details, see the following resources:

- Add and manage work item types (Inherited process)
- Customize your backlogs or boards (Inherited process)
- Customize an inheritance process

Inheritance process model:

- Add and manage work item types (Inherited process)
- Customize your backlogs or boards (Inherited process)
- Customize an inheritance process

On-premises XML process model:

- Add a work item type to a backlog and board
- Add portfolio backlogs
- Customize the On-premises XML process model

- Add a work item type to a backlog and board
- Add portfolio backlogs
- Customize the On-premises XML process model
- Add portfolio backlogs

Add custom work item types to your Taskboard

To add custom work item types to appear on your sprint Taskboard, follow the steps outlined next based on the process model your project uses.

> **NOTE**
> You can enable work item types that you add to your Iteration backlog to appear as a checklist on your product Kanban board. To learn how, see Customize your Kanban board checklist items provided earlier in this article.

Track custom work items with the Inherited process model

For example, if you want to track a custom work item type, Tickets, along with Tasks and Bugs, you do the following tasks:

1. Define the Ticket custom work item type. See Add and manage work item types.
2. Add the Ticket work item types to the Iteration backlog. For details, see Customize your backlogs or boards for a process.

Add existing and custom work item types with the On-premises XML process model

For on-premises deployments that use the On-premises XML process model to customize work tracking, you can add existing and custom work item types to the sprint Taskboards. For example, if you want to track Issues (or Impediments for the Scrum process) and a custom work item type, Tickets, along with Tasks and Bugs, you would do the following tasks:

1. Define the Ticket custom work item type. See Add or modify a work item type.
2. Add Issue and Ticket work item types to the Task Category by modifying the Categories XML file. For details, see Categories XML element reference.

   For example, here we add Issue and Ticket to the Task Category.

   ```
   <CATEGORY name="Task Category" refname="Microsoft.TaskCategory">
     <DEFAULTWORKITEMTYPE name="Task" />
   <WORKITEMTYPE name="Issue" /
   <WORKITEMTYPE name="Ticket" /
   </CATEGORY>
   ```

3. Make sure that the Issue and Ticket workflow states are mapped to category states. As needed, modify the ProcessConfiguration XML file to add Issues and Tickets to the `TaskBacklog` section.

   For example, here the New, Active, and Closed states are mapped for the Task Category.

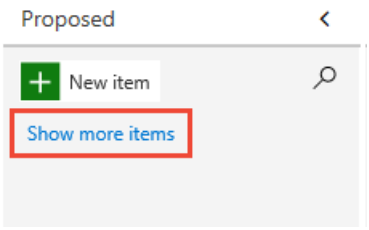
   ```
     <TaskBacklog category="Microsoft.TaskCategory" pluralName="Tasks" singularName="Task"
   workItemCountLimit="1000">
       <States>
         <State value="New" type="Proposed" />
         <State value="Active" type="InProgress" />
         <State value="Closed" type="Complete" />
       </States>
   . . .
     </TaskBacklog>
   ```

4. To verify your changes, open a Sprint backlog and make sure you can add an Issue or Ticket in the same way you add a Task. See Add tasks.

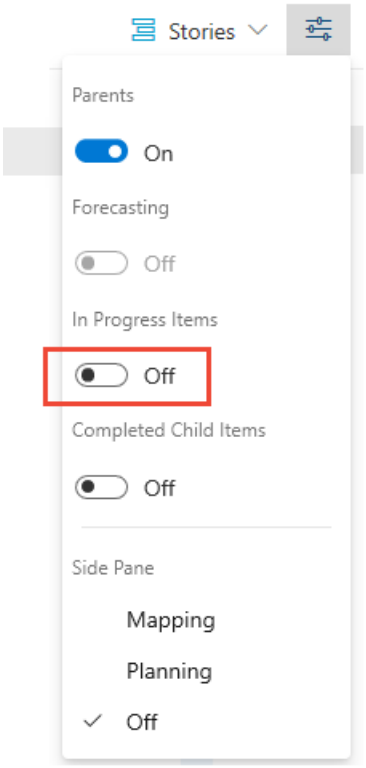
Other factors that can affect work items in your backlogs and boards

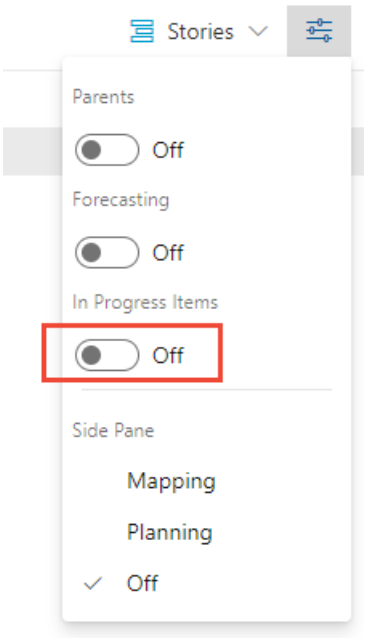
The following settings can influence the type and number of work items that will appear in your backlogs and boards.

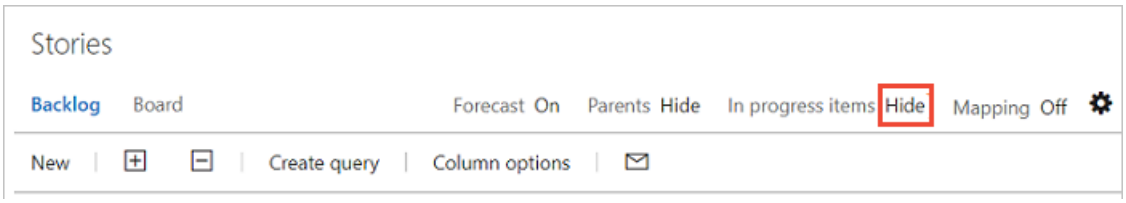
- In your Kanban board, newly added work items may not appear if they're stack ranked lower within the product backlog. By choosing **Show more items**, you can cause the board to refresh and display more work items.



- If you have nested work items that belong to the same category, only leaf nodes may appear on the Kanban board (for TFS 2018.1 and earlier versions). For this reason, we recommend that you don't nest work items of the same work item type or belonging to the same category. To learn more, see Fix reordering and nesting issues, How backlogs and boards display hierarchical (nested) items.
- If you've turned off the **In Progress** view, then those work items where work has started won't appear in the backlog list.







- Work items appear in the priority order in which they're added or moved to. This order or sequence is managed by the **Stack Rank** (Basic, Agile, and CMMI processes) or **Backlog Priority** (Scrum) field. To learn more, see the Stack rank section in Backlogs, portfolios, and Agile project management.
- Each backlog can display up to 999 work items. If your backlog exceeds this limit, then you may want to consider adding a team and moving some of the work items to the other team's backlog.
- Sprint backlogs show only those work items that meet the team's area path and the **Iteration Path** defined for the sprint.
- Inheritance process model: If an administrator disables or deletes a work item type, it will no longer appear on backlogs and boards.
- On-premises XML process model: If an administrator deletes or destroys a work item type, it will no longer appear on backlogs and boards.

Related articles

- Add a team, move from one default team to several teams
- Refine your backlog
- Create your backlog
- Backlog priority or stack rank order
- Use categories to group work item types
- Workflow states & state categories

Use product backlog controls in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

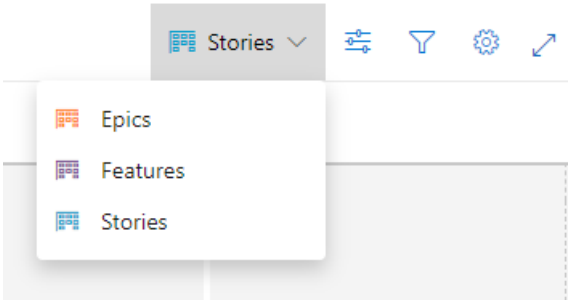
Once you've defined your product backlog, you can use the following controls to change or filter the view.

> **IMPORTANT**
> If you turn the **In Progress** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or in the In Progress category workflow state won't appear in the backlog. To learn more about category workflow states, see How to use workflow states and state categories.

For details on using each of these controls, see Configure your backlog view.

Icon or Link	Control	Function
Backlog	Switch to backlog view	
Analytics	Switch to Analytics in-context reports	
Board	Switch to Kanban board	
Forecast	Turn forecasting Off/On	
Mapping	Turn mapping Off/On	
Parents	Show/Hide parents	
In progress items	Show/Hide in progress items	
	Email a copy of your backlog	
	Backlog selector	

Switch backlog view

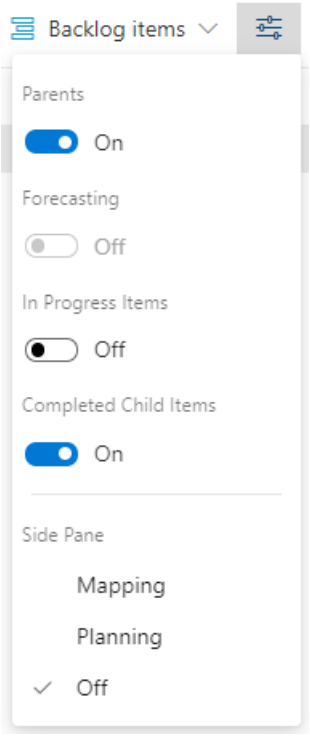


NOTE

Your backlog levels may differ from the level shown in the previous image based on the process selected or customizations you made to your process. Other common labels are Backlog items (Scrum), Requirements (CMMI), and Issues (Basic). To learn more, see Choose a process.

View options

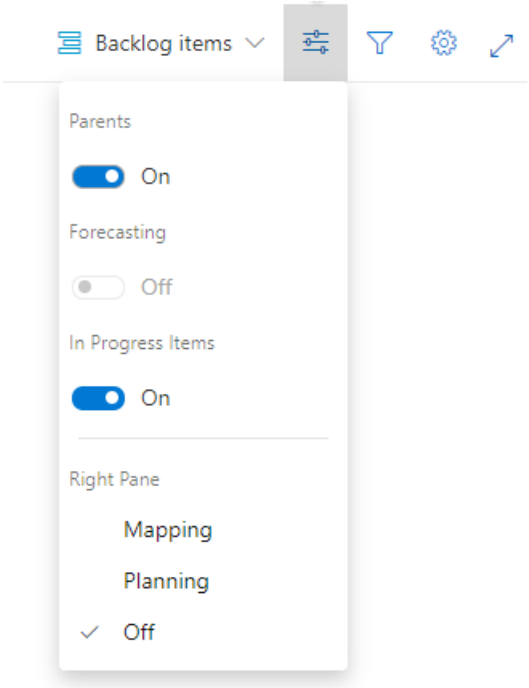
- Turn Parents on/off (not available for top-level portfolio backlog)
- Turn Forecasting on/off (Only available on product backlog)
- Turn In Progress items on/off
- Turn Completed child items on/off
- Show Mapping (not available for top-level portfolio backlog)
- Show Planning



View options

- Turn Parents on/off (not available for top-level portfolio backlog)
- Turn Forecasting on/off (Only available on product backlog)
- Turn In Progress items on/off

- Show Mapping (not available for top-level portfolio backlog)
- Show Planning



Filter

Turn filtering On/Off

Settings

Manage teams and configure team tools

/

Full screen

Enter or exit full screen mode

Filter

Turn filtering On/Off

Settings

Manage teams and configure team tools

/

Full screen mode

Enter or exit full screen mode

/

Expand/Collapse

Expand or collapse one level of the tree hierarchy

More commands

- Set column options
- Create Query
- Email

NOTE

Even if you have show parents turned on, the **Create Query** and **Email** controls will only list items at the currently selected level.

Related articles

- Backlogs, portfolios, and Agile project management
- Workflow states and state categories

Manage columns in a work item list in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Each column corresponds to a work item field. You can add and remove columns within work item lists to show the fields of interest to you. Or, you can drag a column to a new position. Your settings persist for each page you customize and are only valid for your views.

Specifically, you can do the following actions from the following list views:

Action	Backlogs	Sprint backlogs	Queries	Work items
Add or remove a column field	Yes	Yes	Yes	Yes
Add or remove the Parent field	Yes	Yes	Yes	Yes
Add or remove a rollup column	Yes	No	No	No
Sort on a column	No	No		

Yes

Yes

> **TIP**
> Unlike a query result, you can't sort a backlog by a column. However, you can use the **Create Query** link on each backlog to create a query that you can sort on any field column you choose from the **Sorting** tab of the Column options dialog. While you may be able to add a field to sort on, not all fields are supported. For example, selection of the **Parent**, **History**, **Description**, or other rich-text field will result in the display of an error message as you can't sort on these fields.

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. You can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. If your project uses the Inherited process model, you can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project.

About persistence of column options

Once you set the column options for a specific view, your settings persist until you change them. The following notes apply to specific views.

- Column options you set for a backlog apply only to the active team and backlog. Options you set for the product backlog differ from the options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query.
- Column options you set for one of the supported Work items views persists for a specific view only, such as **Assigned to me**, **Following**, **Mentioned**, and so on.

- Column options you set for a backlog apply only to the active team and backlog. Options set for the product backlog differ from those options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query. For Shared queries, your settings can get overwritten when someone else saves different column option settings.

> **NOTE**
> You can't set column options for other members of your team, nor can you set default column options.

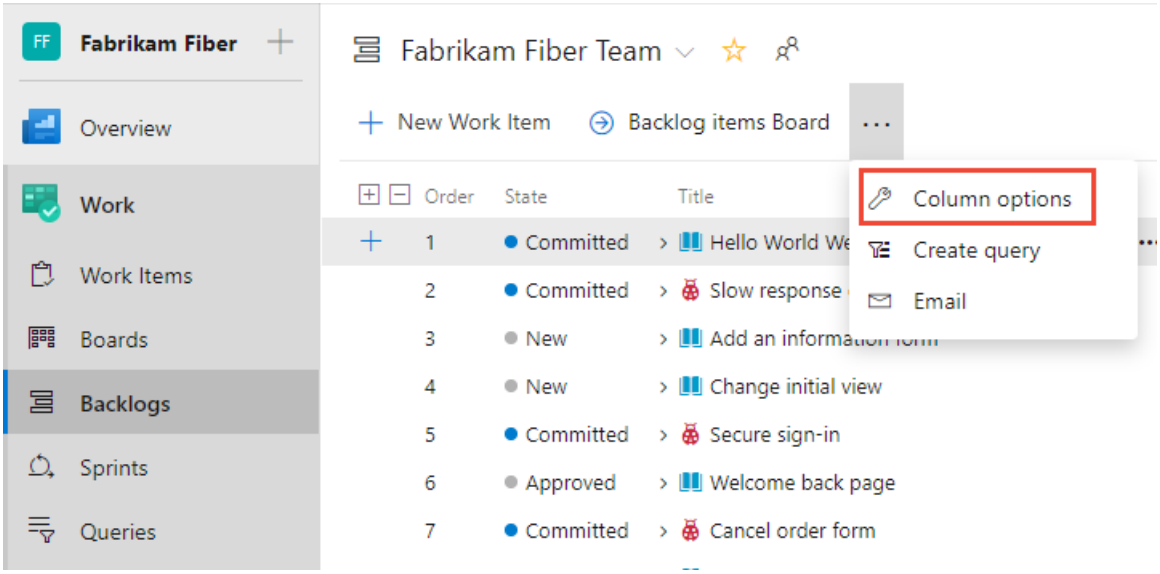
> **NOTE**
> You can't set column options for other members of your team. Also, for projects that use the Inheritance process model, you can't set default column options. For projects that use the On-premises XML process model, you can set the default column options for product, portfolio, and sprint backlogs. To learn how, see Process configuration XML element reference.

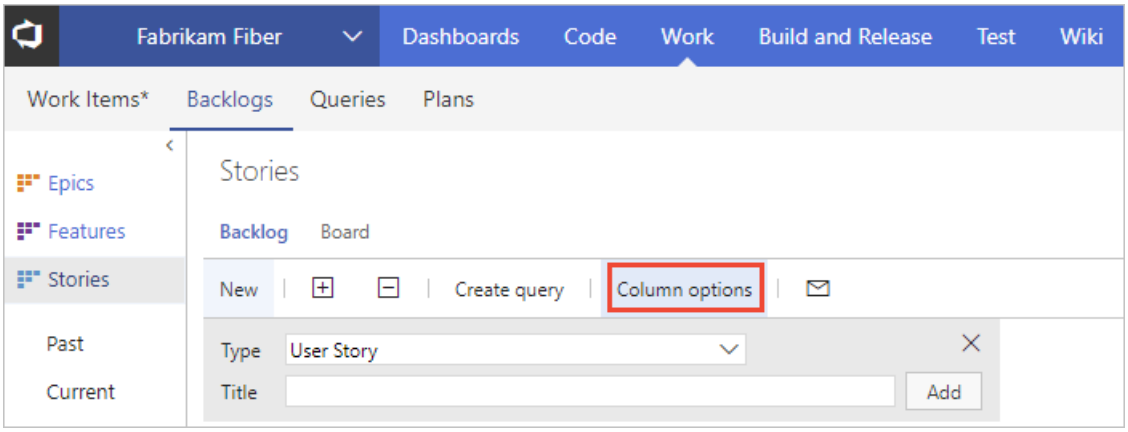
NOTE

You can't set column options for other members of your team. For projects that use the On-premises XML process model, you can set the default column options for product and portfolio backlogs. To learn how, see Process configuration XML element reference.

Open the Column options dialog

Start by opening the **Column Options** dialog. If you don't see the option, choose the ... and choose from the options provided.





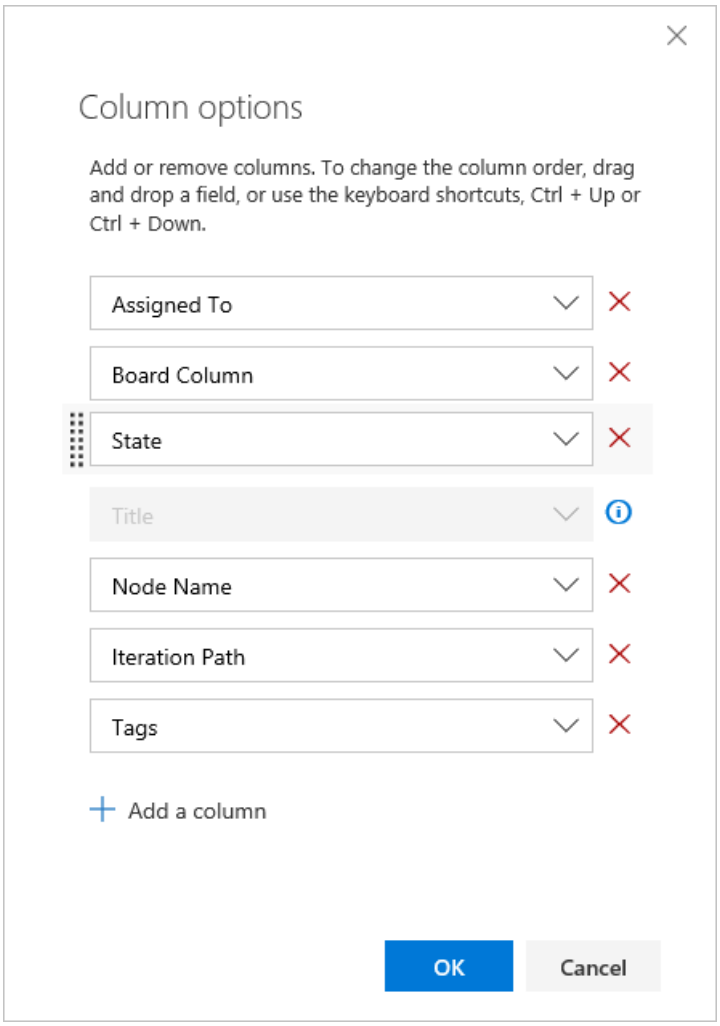
Add or remove columns

- Browser
- Visual Studio

In the Column options dialog, choose **Add a column** to add a field that isn't shown. To change the order of the fields, drag-and-drop the field where you want it within the set of selected fields. And, to remove a field, choose the ✕.

NOTE

The following dialog is available from TFS 2018.2 and later versions.

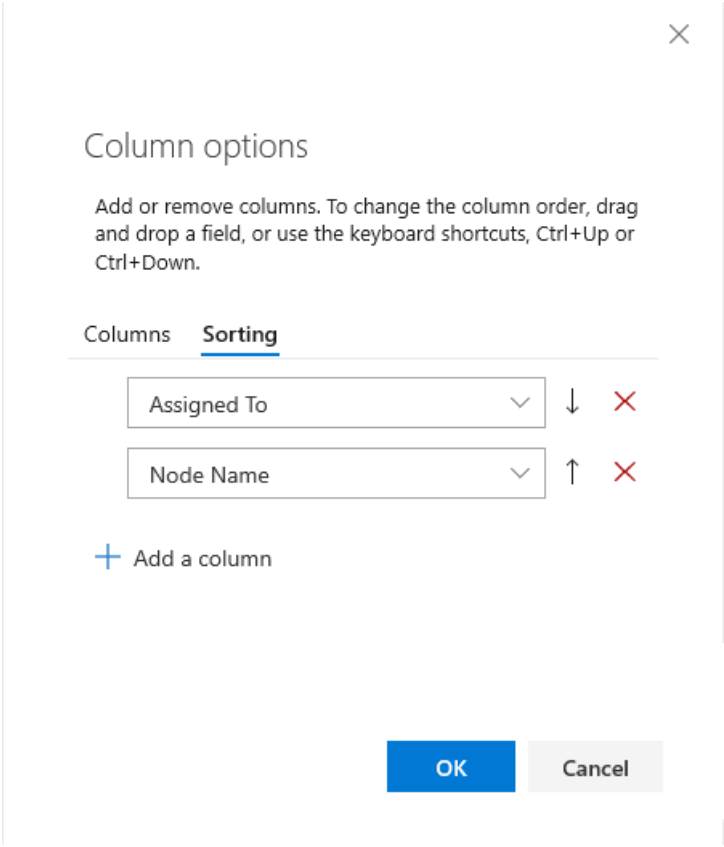


Add or remove rollup columns

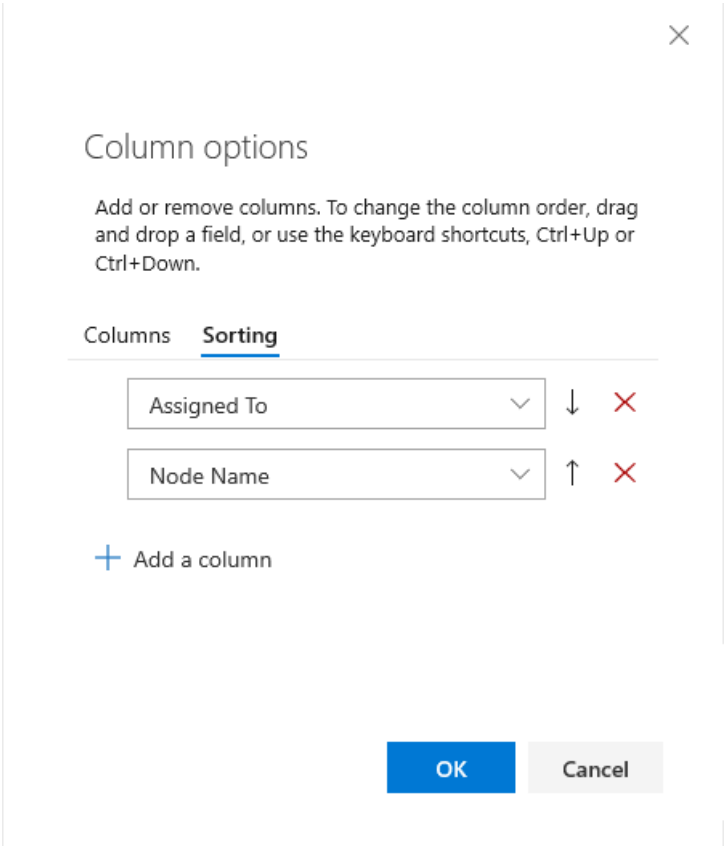
Rollup columns can display progress bars or the sum of numeric fields of child items. You can add them to any product or portfolio backlog. To learn more, see Display rollup progress or totals.

Sort on a column

You can sort query results and Work items views. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



You can sort query results. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



Use keyboard shortcuts to change the column order, column width, or sort options

You can change the column order, column size, or sort options by using the following keyboard commands:

- To change the column order, choose the field and drag it to a new location
- To resize a column, choose the column divider to the right of the field and drag to a new location
- For query results:
  - Add the field as a column in order to sort by that field
  - To sort by a column, hold down the SHIFT key and select on the field
  - To reverse the sort order, SHIFT+click on the field
  - To sort by multiple columns, SHIFT+click on each column in the order you want to sort

For other keyboard shortcuts, enter **?** to display available shortcuts based on the page you're on.

Related articles

- Display rollup progress or totals
- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- View, run, or email a work item query
- Create managed queries
- Customize a sprint Taskboard

- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- Create managed queries

Display rollup progress or totals in Azure Boards

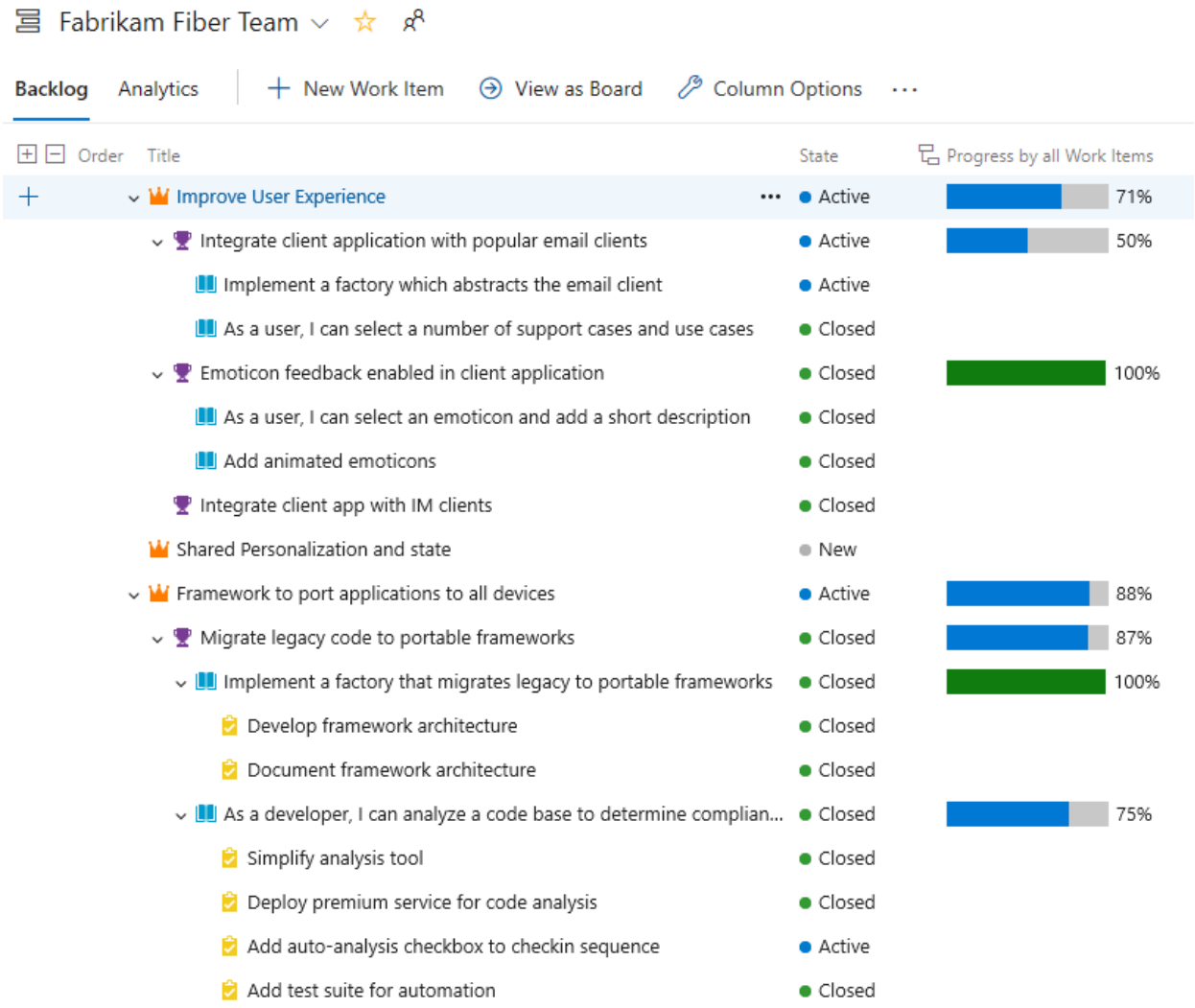
10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 | Azure DevOps Server 2020

Rollup columns allow you to view progress bars or totals of numeric fields for descendant items within a hierarchy. Descendant items correspond to all child items within a hierarchy. You can add one or more rollup columns to a product or portfolio backlog. Support for sprint backlogs isn't supported. For information on linking work items in a hierarchy, see Linking, traceability, and managing dependencies, Parent-child work item links.

For example, here we show **Progress by Work Items** which displays progress bars for ascendant work items based on the percentage of descendant items that have been closed. Descendant items for Epics include all child Features and their child or grand-child work items. Descendant items for Features include all child User Stories and their child work items.

> **IMPORTANT**
> Rollup data supports progress bars, counts of work items, or sums of numeric fields within a project. Child items that link to a different project aren't counted within the parent rollup calculations. Also, links to test cases or test artifacts are also not included in rollup calculation. These items are linked using a test-specific link types.



> **NOTE**
> You can also view rollup progress from the new version of Delivery Plans that is available in public preview for Azure Boards. This feature is now part of Azure Boards and not an extension. To enable it, see Manage or enable features and turn on **New Delivery Plans Experience**. To learn more, see Review team Delivery Plans.

Prerequisites

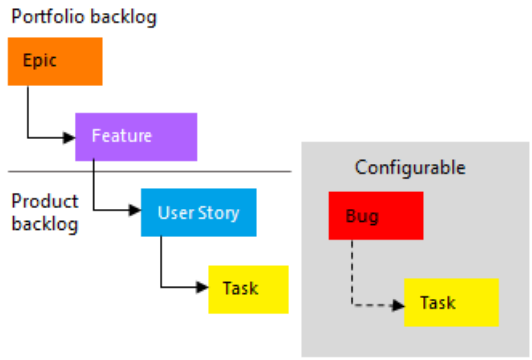
- Rollup column data is calculated from the Analytics service.
- To add a rollup column, the Analytics service must be enabled on your on-premises Azure DevOps Server. To learn more, see Install/uninstall or enable/disable the Analytics service.

Rollup and hierarchical work items

The default set of backlog work items that supports a natural hierarchical grouping varies depending on the process chosen for your project. The easiest way to group work items into a hierarchy is by mapping them, or adding them to the parent item on a Kanban board. To learn more, see Organize your backlog, map child work items to parents and Kanban board features and epics.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. Each team can configure how they manage bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting.



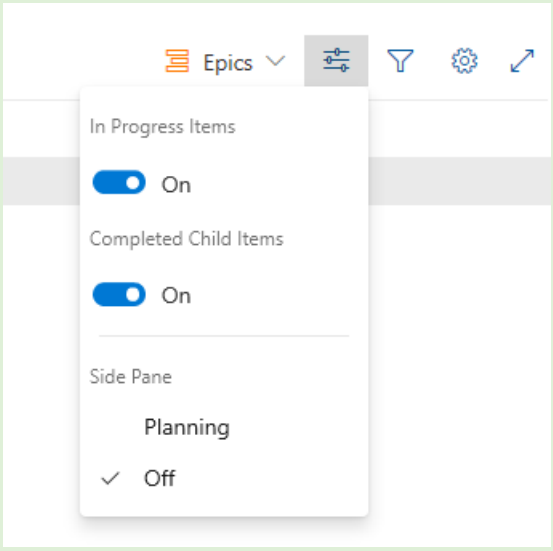
Open a product or portfolio backlog

Each user can set their own column options that persist for each backlog across the user's sessions.

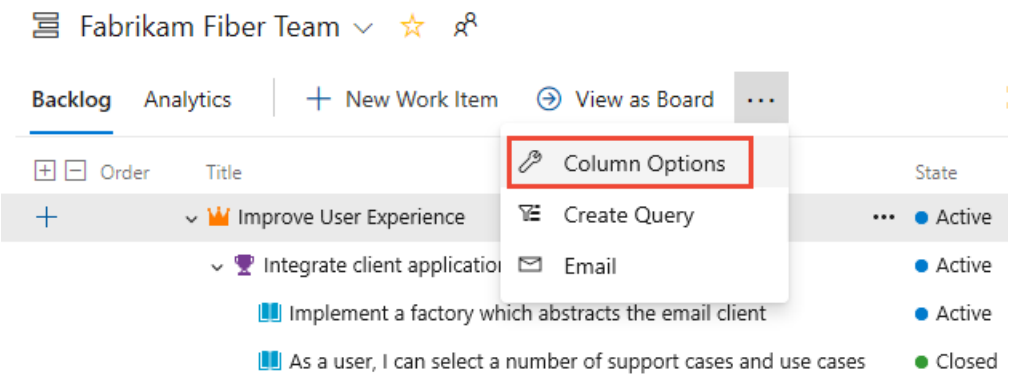
1. Open a product or portfolio backlog. Optionally, enable **Show parents** from your view options. Even if child items aren't listed, rollup for them displays.

TIP

Consider opening a portfolio backlog and choose to view **In Progress Items** and **Completed Child Items**. That way, you can check the State value of items against the rollup value.



2. Choose **Column options**, or choose the ··· actions icon and then select **Column options**.

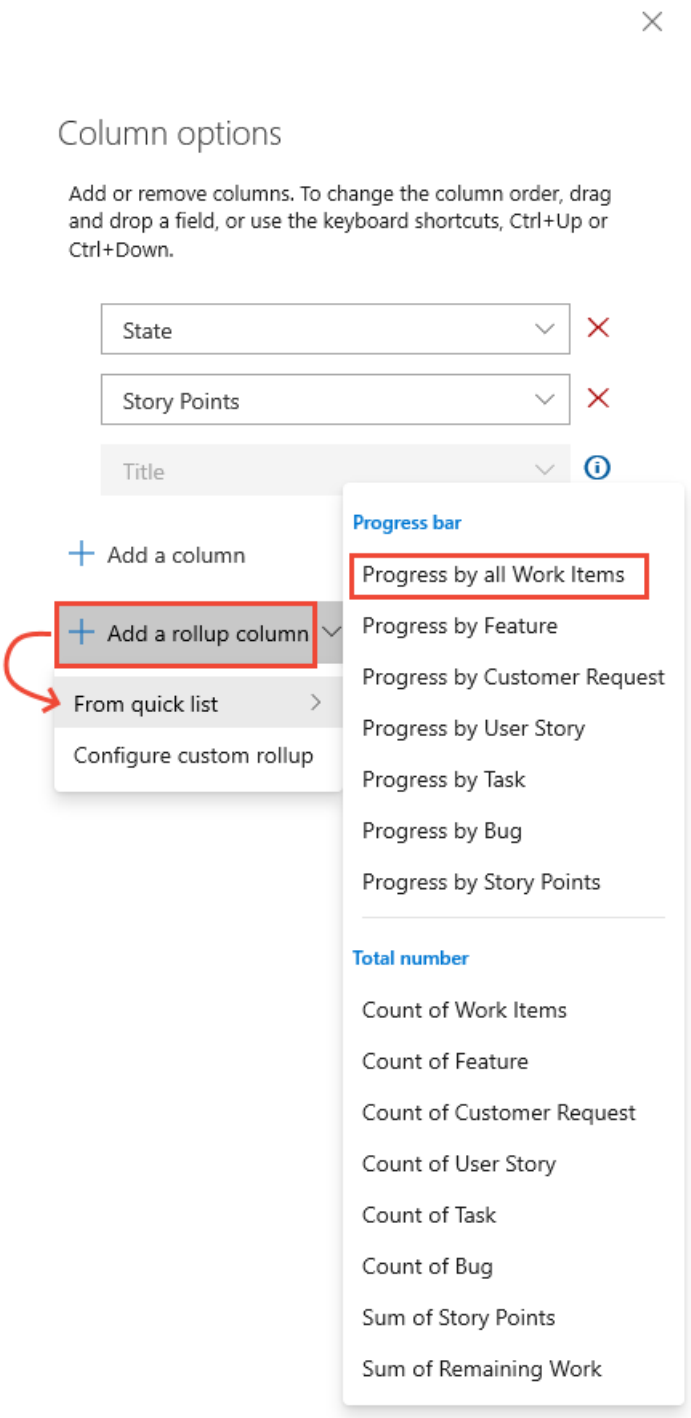


TIP

Remember that the Column options you choose are for the selected backlog level. They persist across your sessions until you change them.

Add a rollup column

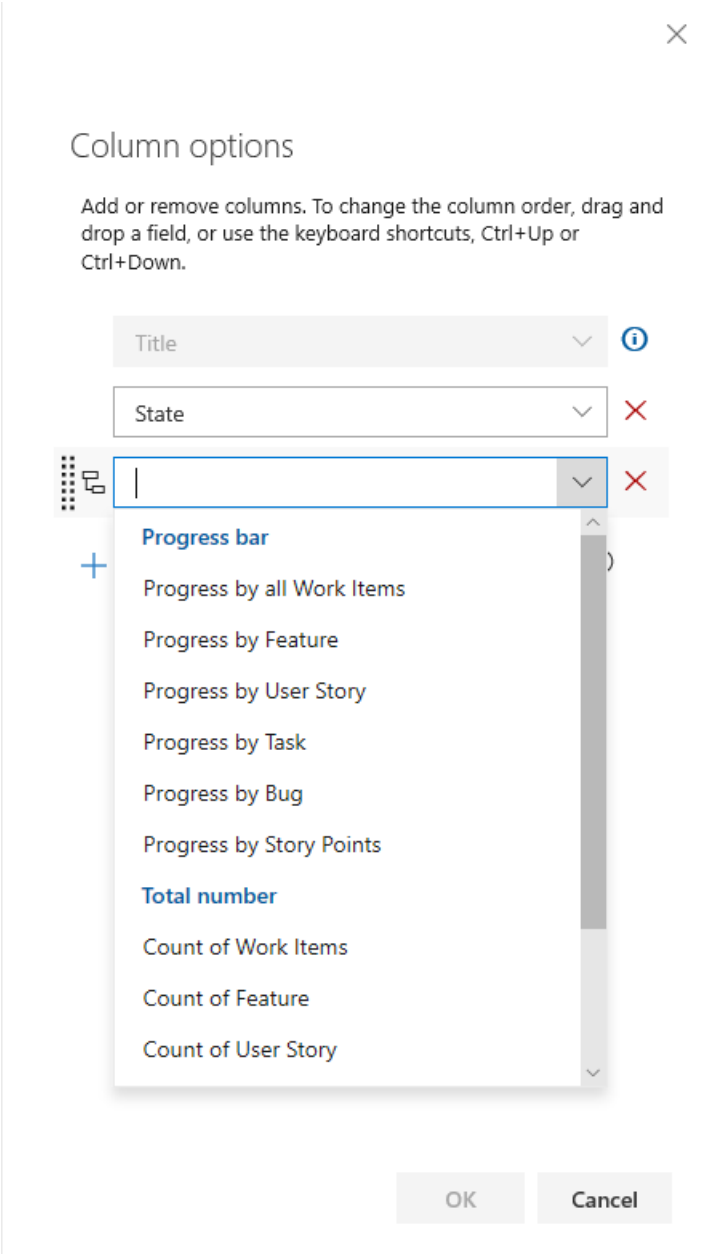
1. In the Column options dialog, choose **Add a rollup column**, select **From quick list**, and then choose from one of the options listed.



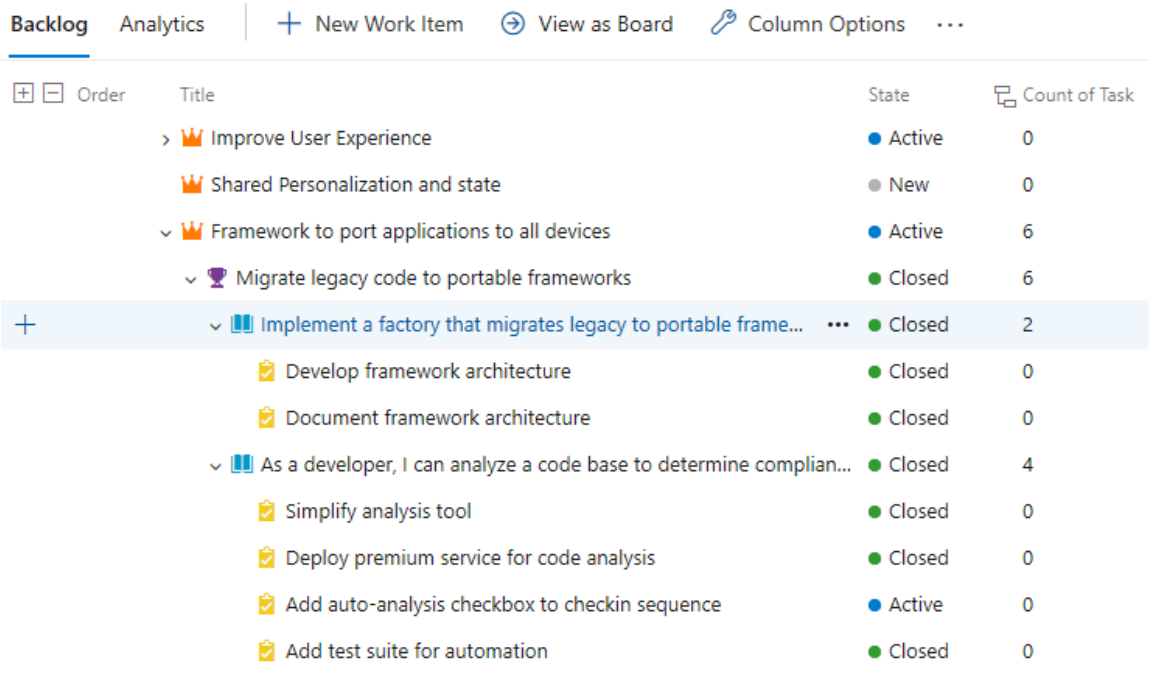
> **NOTE**
>
> The menu options vary depending on the process chosen for your project, the backlog level you've selected, and whether or not you have the **Show parents** view option enabled.

2. Choose from the menu provided.

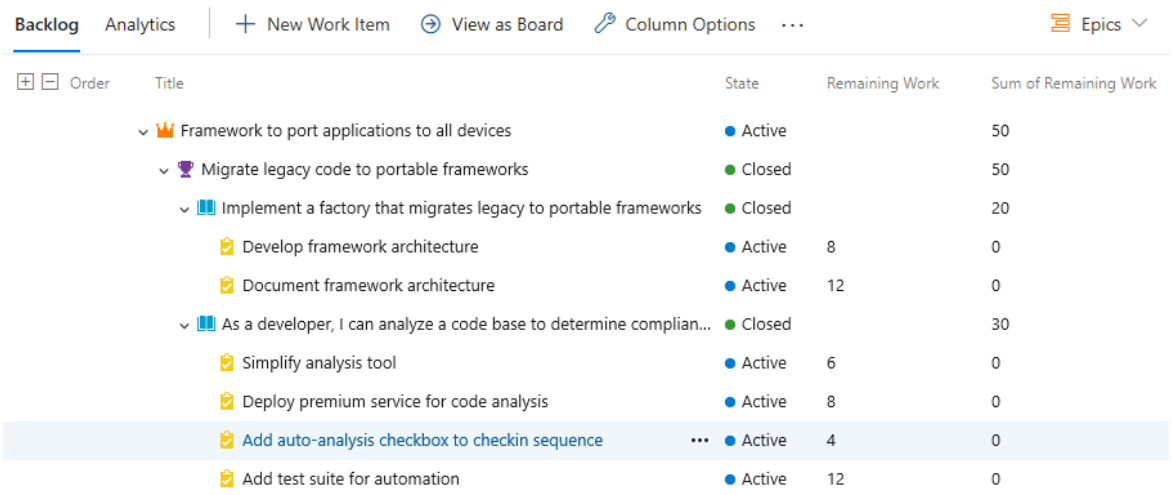
- **Progress bar** displays progress bars based on the percentage of associated descendant work items that have been completed or closed.
- **Total number** displays the sum of descendant items or the associated fields of descendant items. Totals provide a measure of the size of a Feature or Epic based on the number of its child items. For example, **Count of Tasks** shows the sum of all tasks that are linked to parent items. The active or closed state is ignored.



For example, the following image shows the Count of Tasks for the parent user stories is 2 and 4, respectively. The Count of Tasks is 6 for the parent Feature and Epic.



3. **Remaining Work of Tasks** shows the sum of Remaining Work of tasks that are linked to the parent item.

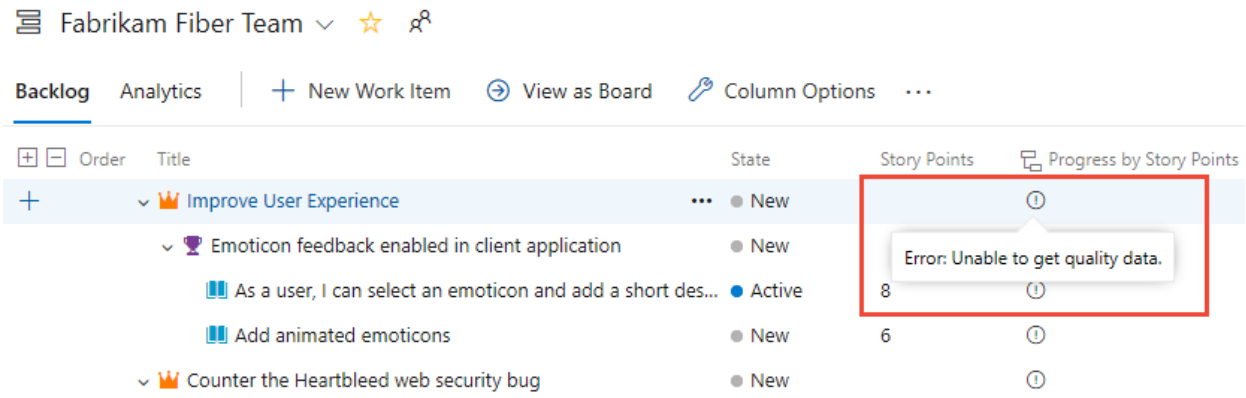


> **TIP**
> Reminder that when a task is closed, the Remaining Work field is set to zero.

Analytics, latency, and error states

Rollup data is calculated from the Analytics service. When there's a large amount of data, it's possible to experience some latency in displaying rollup. If you hover over the rollup icon, you can determine the state of the data.

If an error occurs in retrieving rollup data, you'll see an info icon and empty rows. Errors indicate when the Analytics data was last updated. This means that the Analytics services are still processing changes made which may affect rollup calculations. Once the Analytics data is up to date, the rollup columns will refresh with the latest data.



To learn more about the service, see What is Analytics?.

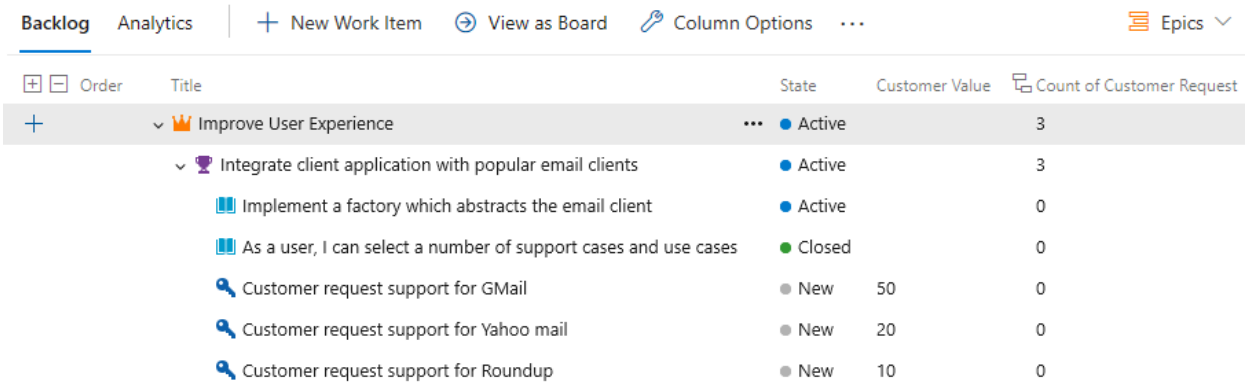
Change the column order or remove a rollup column

To change the order of the fields, drag-and-drop the field where you want it within the set of selected fields. And, to remove a field, choose the ×.

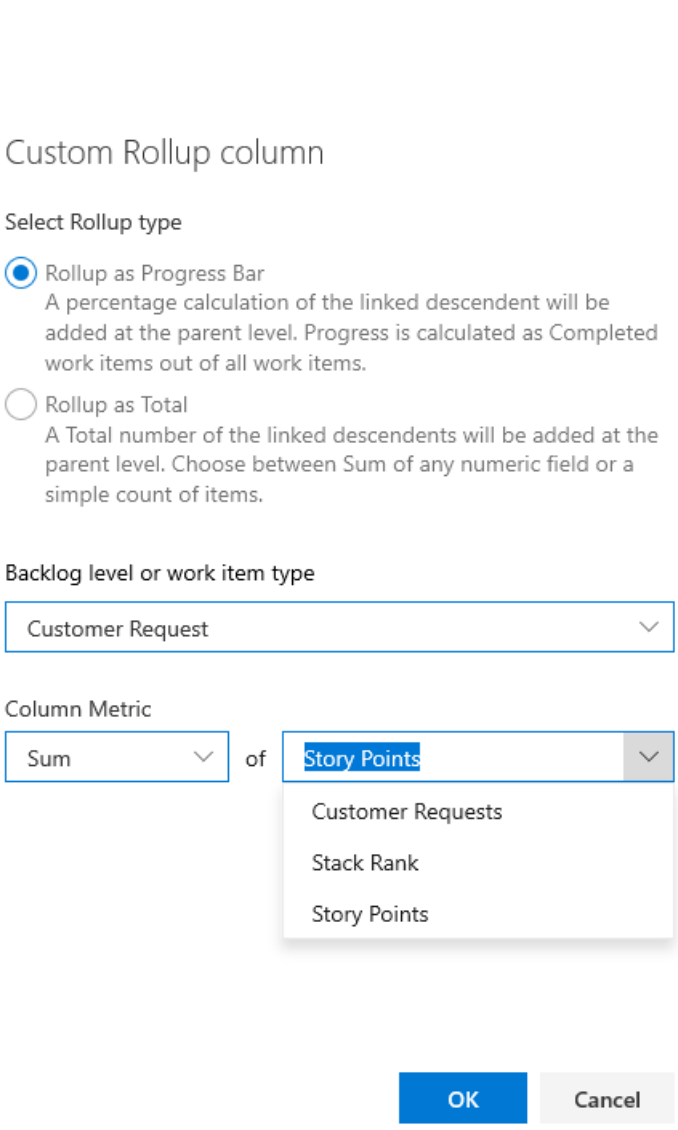
Rollup of custom work item types or custom fields

If you add a custom work item type or field to a backlog level, you can view rollup based on those options. For

example, the Customer Request type is added to the Requirements category, and a Count of Customer Requests is shown in the following image.



1. From the Column options dialog, choose **Add a rollup column**, select **Configure custom rollup** option.
2. Choose the options you want from the Custom Rollup column dialog.



3. Choose **OK**. and then **OK** to complete your operations.

> **TIP**
> If you add custom fields or custom work item types, you must refresh the backlog page to reflect your changes.

Use keyboard shortcuts to change the column order, column width, or sort options

You can change the column order, column size, or sort options by using the following keyboard commands:

- To change the column order, select the field and drag it to a new location
- To resize a column, choose the column divider to the right of the field and drag to a new location

Related articles

- Change column options
- Work item field index
- Product backlog controls
- Backlogs, boards, and plans
- Create managed queries

Select backlog navigation levels for your team

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Each team can determine the backlog levels that they use. For example, feature teams may wish to only focus on their product backlog, while a management team may choose to only show feature and epics (the two default portfolio backlogs). You configure which backlog levels appear from your team settings dialog.

If you need additional portfolio backlogs, see the following articles based on the process model you use:

- **Inheritance**: Customize your backlogs or boards for a process
- **On-premises XML**: Add portfolio backlogs.

For an overview of process models, see Customize your work tracking experience.

If you need additional portfolio backlogs, see Add portfolio backlogs.

Prerequisites

- To configure team settings, you must be added as a team administrator or be a member of the **Project Administrators** group. See Change project-level permissions.

Set your team's preferences for backlog levels

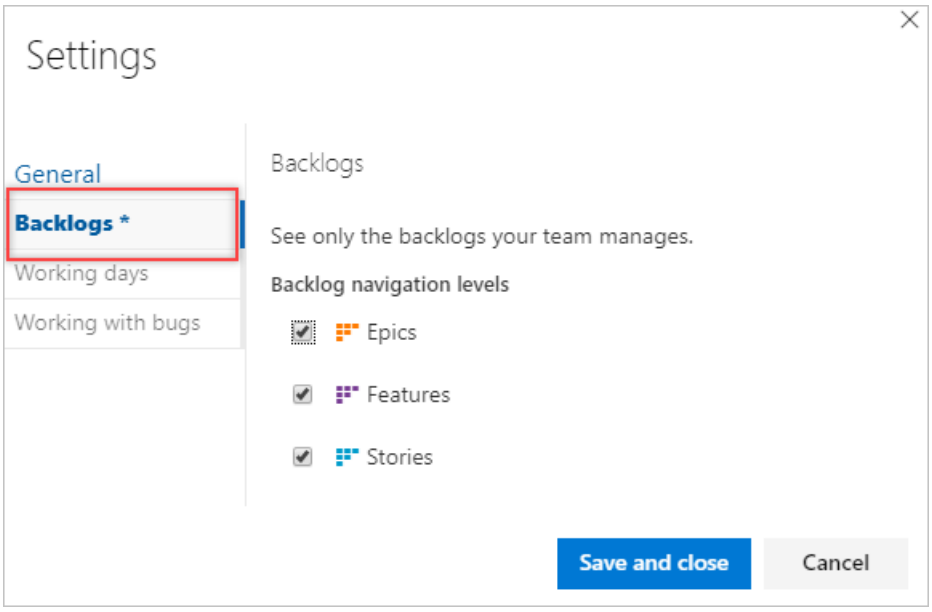
Because this setting affects all team members' view of the team backlogs and boards, you must be a team administrator to change the setting. Changing the setting is disabled if you're not a team administrator. To get added as a team administrator, see Add team administrator.

You can change the setting from a backlog or board view. Here we show how to change it from the board view.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Select **Configure team settings** to configure the board and set general team settings.



3. Choose **Backlogs** and check the boxes of those backlog levels you want your team to manage.

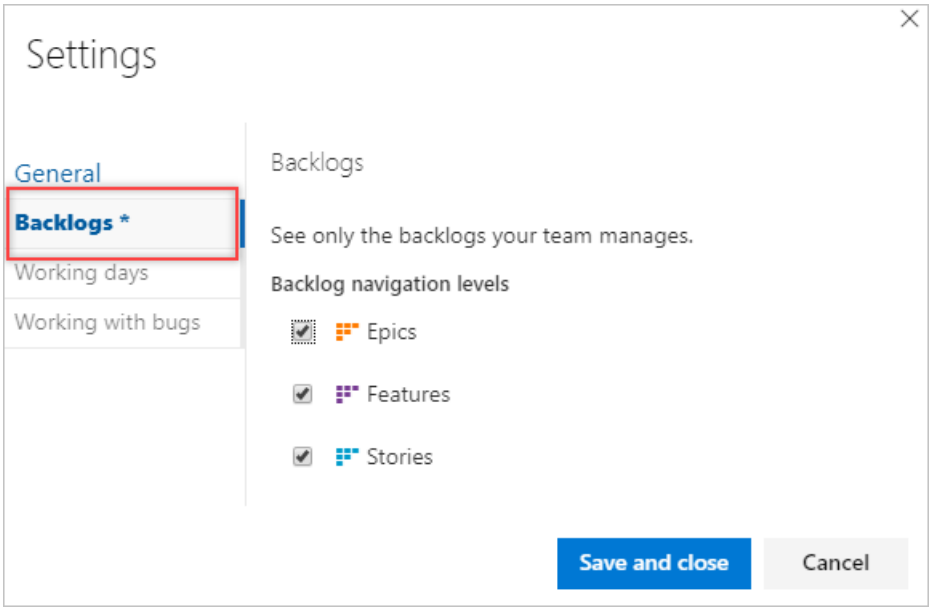


4. When done with your changes, choose **Save and close**.
5. To see the changes, open or refresh your team's backlog.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Select **Configure team settings** to open the settings dialog.



3. Choose **Backlogs** and check the boxes of those backlog levels you want your team to manage.



4. When done with your changes, choose **Save**.
5. To see the changes, open or refresh your team's backlog.

Related articles

- Get started with Agile tools to plan and track work
- Backlogs, boards, and plans
- Create your backlog
- Define features and epics
- Organize your backlog

Show bugs on backlogs and boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As your team identifies code defects or bugs, they can add them to the backlog and track them similar to tracking requirements. Or, they can schedule bugs to be fixed within a sprint along with other tasks. When you *track bugs as requirements*, they appear on the product Backlogs and Kanban Boards. When you *track bugs as tasks*, the bugs appear on the Sprint Backlogs and Taskboards. For more information about other work item types, see Add other work item types to backlogs or boards.

You can define this team setting for the Agile, Scrum, and CMMI processes. The Bug work item type isn't defined for the Basic process, so there isn't a team setting for Basic. Instead, you should track bugs and code defects using the Issue work item type.

> **NOTE**
> *Requirements* specify expectations of users for a software product. In Azure Boards, requirements are defined by work items that appear on your product backlog. They correspond to User Stories (Agile), Product Backlog Items (Scrum), Issues (Basic), or Requirements (CMMI) based on the process selected for your project. They also belong to the Requirements Category which manages which work item types appear on the product backlog.

Prerequisites

- To configure team settings, you must be added as a team administrator or be a member of the **Project Administrators** group. See Change project-level permissions.

Options your team has to track bugs

The following table summarizes the three options teams have to track bugs. Before making your choice, we recommend you review the information provided in Define, capture, triage, and manage bugs. That article provides an overview of the Bug work item type and supported tools for managing bugs.

Option	Choose when you want to...
Track bugs as Requirements	• Prioritize (stack rank) bugs along with requirements • Estimate Bug effort for forecasting • Update bug status on Kanban board • Include Bugs in Velocity charts and Cumulative Flow Diagrams • Can use the **Forecast** tool to support sprint planning • Can drag and drop bugs onto **Planning** pane to assign bugs to a sprint • Can view Bugs on Delivery Plans

NOTE

- Bugs are assigned to the Requirements Category

Track bugs as Tasks

- Estimate work for bugs similar to tasks
- Update bug status on sprint Taskboards
- Link bugs to requirements as child items
- Can drag and drop bugs onto Planning pane to assign bugs to a sprint

NOTE

- Bugs are assigned to the Task Category
- User Stories (Agile), Product Backlog Items (Scrum), or Requirements (CMMI) are the natural parent work item type for Bugs
- Bugs won't be visible on Delivery Plans

Bugs don't appear on backlogs or boards

- Manage bugs using queries

NOTE

- Bugs are associated with the Bugs Category and won't appear on either backlogs or boards
- Bugs won't be visible on Backlogs, Boards, Sprint Backlogs, Taskboards, or Delivery Plans
- Can't drag and drop bugs onto Planning pane to assign bugs to a sprint

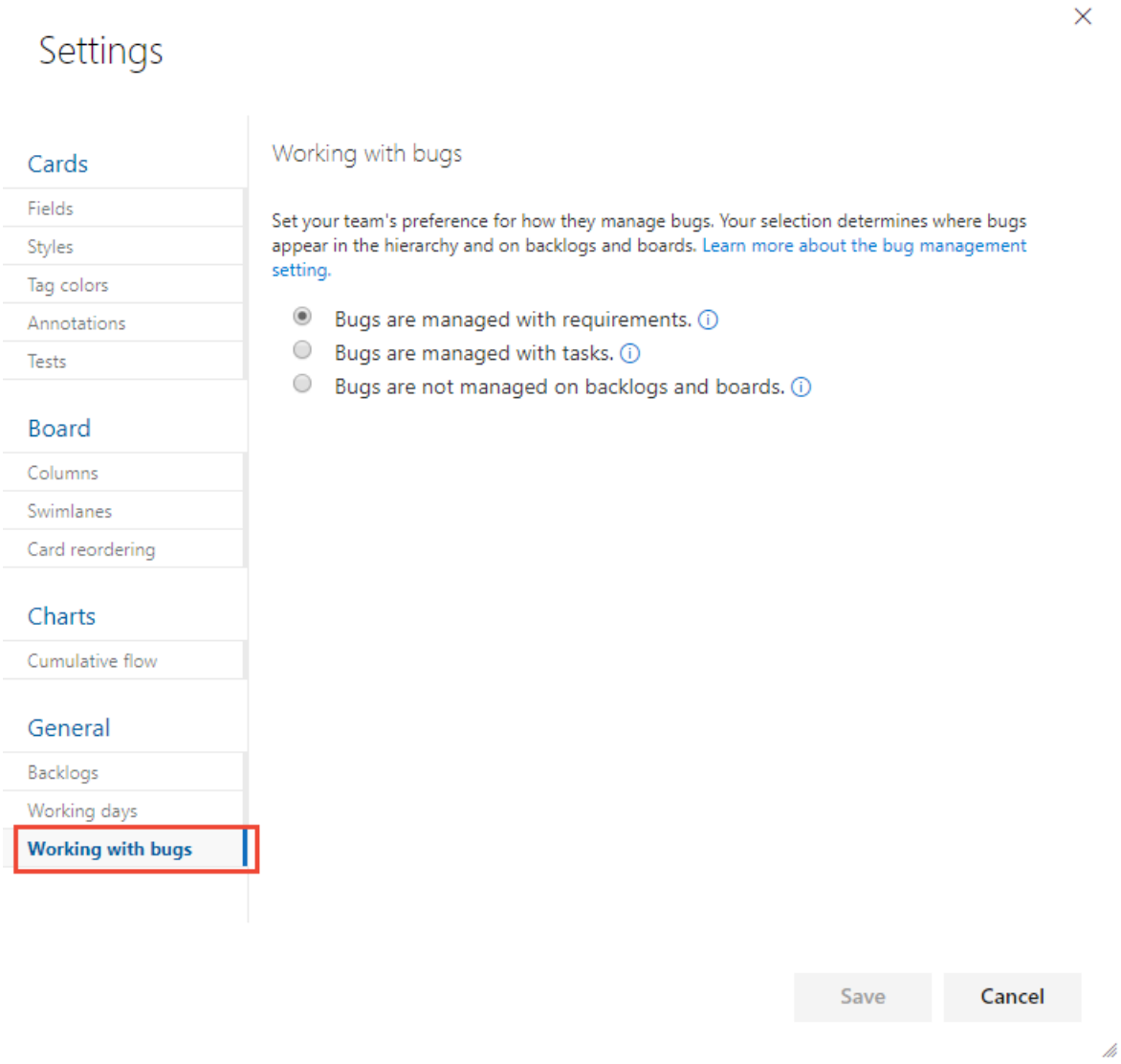
Set your team's preference for bug tracking

You can change the setting from a backlog or board view, or from **Project Settings>Team Settings**. Here we show how to change it from the board view.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose **Board settings** to configure the board and set general team settings.



3. Choose **Working with bugs** and then choose the option that best meets your team's way of working.

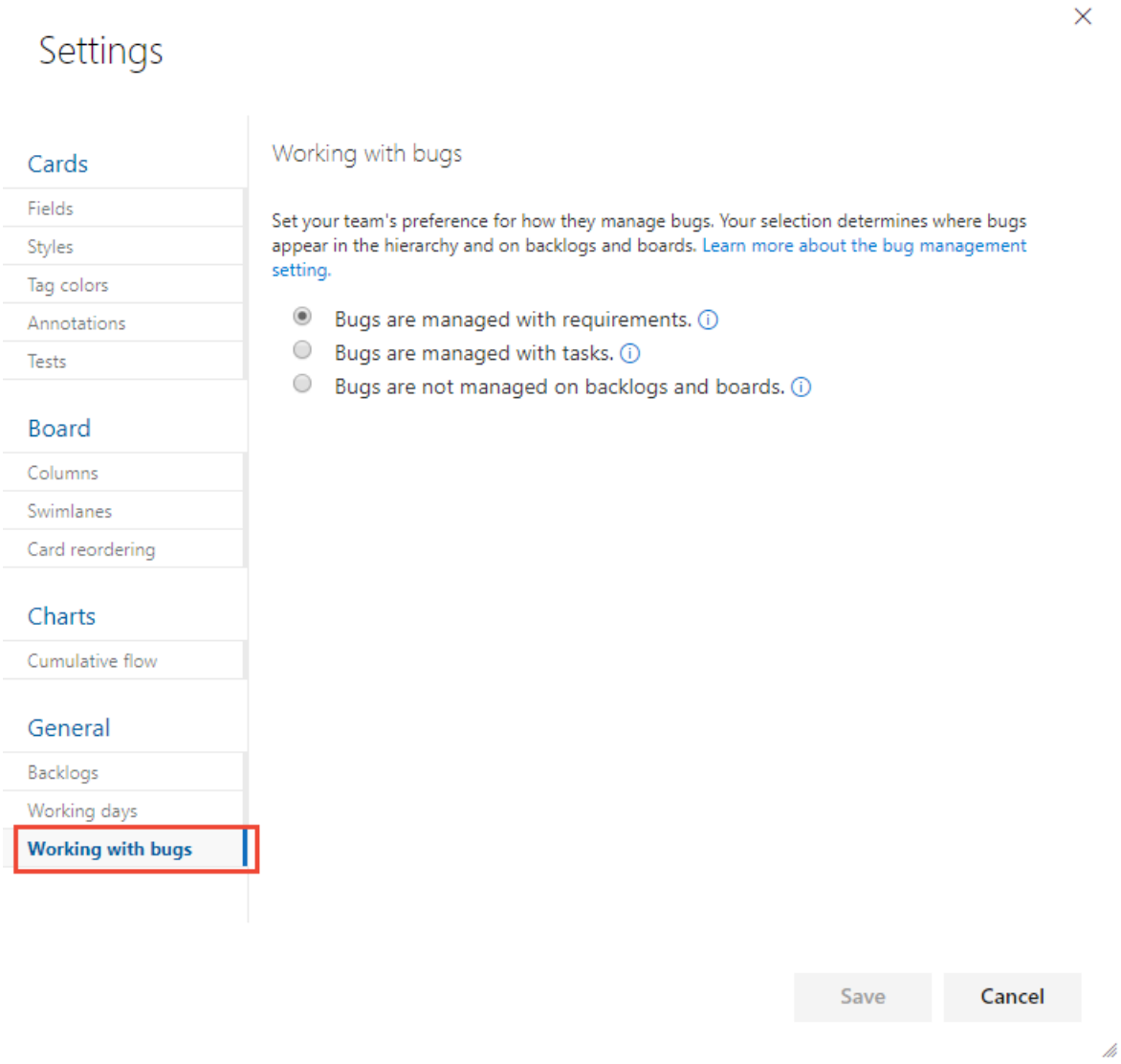


4. When you're done with your changes, choose **Save**.
5. To see the changes, open or refresh the team's backlog or Kanban board.

1. Open your Kanban board. If you're not a team admin, get added as one. Only team and project admins can customize the Kanban board.
2. Choose **Board settings** to open the settings dialog.



3. Choose **Working with bugs** and then choose the option that best meets your team's way of working.



4. When done with your changes, choose **Save**.

5. To see the changes, open or refresh your team's backlog or Kanban board.

Nested items

> **TIP**
> If, after refreshing a backlog or board, you don't see bugs where you expect to see them, review How backlogs and boards display hierarchical (nested) items. Only leaf nodes of nested items appear on the Kanban or task boards.

When you manage bugs with requirements or tasks, they appear on one or more of your Agile tool backlogs and boards. However, if you nest items—create parent-child links of items that belong in either the Requirements or Task categories—not all items may appear on your backlogs and boards. To learn more about how nested items are treated, see How backlogs and boards display hierarchical (nested) items.

Add other work item types to your backlogs or boards

Bugs are a common item that teams want to track, and choose how they track them. For more information, see Manage bugs.

However, what if you want to track other work item types on your backlogs and boards?

You can add other work item types—such as change requests, issues, or impediments—by customizing your

process or project, based on the process model you use. For details,

- For the Inheritance process model, see Customize your backlogs or boards for a process.
- For Hosted XML and On-premises XML process models, see Add a work item type to a backlog and board.

You can add other work item types—such as change requests, issues, or impediments—by customizing your process or project, based on the process model you use. For details, see Add a work item type to a backlog and board.

For an overview of process models, see Customize your work tracking experience.

Create, list, and manage bugs

Bugs that are managed with requirements can be added through the product backlog or Kanban board. When bugs are managed along with tasks, you can add them to a sprint backlog or task board. Or, capture them using other tools. For more information, see Define, triage, and manage bugs.

> **TIP**
> Effort should automatically be part of a bug, but if you don't see it, customize the bug work item type for it to appear.

You can review bugs defined for your project by creating a query and specifying the **Work Item Type=Bug**. Or, open a predefined query, **Active Bugs** (Agile and CMMI), or **Work in Progress** (Scrum).

Related articles

- Define, capture, triage, and manage bugs
- Enable backlog levels of interest to your team
- Manage teams and configure team tools
- View, run, or email a work item query
- Triage work items
- Query by assignment or workflow changes

Backlog management in Azure Boards

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

A great backlog conveys customer needs and value. Over the course of the project, your team will add detailed information to each backlog item, break them down into smaller items, prioritize, and estimate them, and finally, implement them and deliver the results to your customers.

To get started, see Create your backlog.

Role of the product owner

Product owners play an important role in Scrum, primarily as the interface between customers and the team. To enable product owners to fulfill the following responsibilities, they need to be added to the Contributors group.

- Analyzing customer requirements and express them as user stories, features, or requirements
- Building, prioritizing, and refining the product backlog
- Representing customer and stakeholder requirements to the team and responding to questions your team has about them
- Meeting regularly with stakeholders to address their needs and keep them informed
- Helping stakeholders understand the decisions underlying the priority order of your backlog
- Responding to all requests from your team for more information concerning backlog priorities and requirements

If they'll also be responsible for configuring team settings, add them as a team administrator.

A product owner can reduce the need for detailed specifications by being more responsive to the team's questions about implementation details and clearly expressing acceptance criteria within each requirement.

Acceptance criteria

Acceptance criteria define what "Done" means by describing the conditions that the team should use to verify whether a requirement or bug fix has been fully implemented. You can capture these criteria in the work item. Clear acceptance criteria help with estimating and developing requirements and with testing.

Product owners are the ultimate deciders of the criteria that create customer value.

> **Tips from the trenches: Start to love and embrace acceptance criteria.**
>
> Ask 10 mature agile teams "How do you know when you're "done done"? and you'll get the same answer from each one. . . get serious about writing acceptance criteria.
>
> Acceptance criteria are the handshake between the product owner and the team on what "done done" really means.
> Until the acceptance criteria are met, the team isn't done with the story. Period. However, the value of acceptance criteria only starts here.
>
> Acceptance criteria provide the stage for some of most meaningful conversations and interactions that can happen on an agile team. On my own team we routinely have some of our best interactions as we start digging into the acceptance criteria for each story on our backlog. Inevitably we all start with our own ideas

about what "done" means for a given story.

However, as we begin to discuss the acceptance criteria presented by the product owner what ensues is a series of "ah-ha moments." A shared understanding of the story begins to emerge. A comment one team member might elicit the following response from someone else. . . "Ah-ha, great point. . . I never thought of that."

Regardless of who is being enlightened, the power is in the fact that the product owner and the team are building together a shared understanding of what "done" means for each backlog item. And, this is happening before the team has written a single line of code... before any work has been done... before commitments have been made... and before the sprint has begun.

By collaborating on acceptance criteria the team is minimizing risk and greatly increasing the chance of delivering successfully. I don't think it's a coincidence that the first bullet in the Agile Manifesto states ". . . we have come to value **individual and interactions** over processes and tools". Agile teams work together. And by working together, they create better software.

Start learning to love acceptance criteria and see if your team isn't more successful delivering software.

—Aaron Bjork, Principal Product Manager, Visual Studio Cloud Services

Refine your backlog

Backlog refinement supports your sprint planning efforts and helps minimize these often seen challenges:

- Long, unfocused, and ineffective sprint planning meetings
- Insufficient thought given to design requirements
- Poor sprint planning and execution
- Defocus on the business value team wants to achieve
- Inability to forecast

A meeting to refine the backlog should occur separate from the sprint planning meeting. Use this meeting to complete these activities:

- Right-size backlog items by splitting larger items into smaller items. No backlog item should be larger than it will take to complete in a single sprint.
- Identify and fill in gaps in the product backlog. Capture new ideas and stories, architecture and design requirements, and other spikes.
- Reorder the backlog to represent today's priorities and business value focus.
- Ensure well-defined acceptance criteria has been added to each item.
- Revisit estimates made to backlog items and adjust upwards or downwards based on recent understanding about scope and acceptance criteria.
- Review all potential backlog items to consider for the upcoming sprint to make sure they're well understood and that any other work required to support their development is well understood by both product owner and the team.

You'll know that you've done a good job refining your backlog when your sprint planning meetings run smoothly and efficiently. Such meetings shouldn't contain many surprises, and your team should feel they can contribute fully.

Capture and manage spikes

Also, to new features and requirements to build, you can capture non-feature work that still needs to be done

for a healthy ecosystem of delivery. This work can include necessary research, design, exploration, or prototyping. Any work done that doesn't directly lead to shippable software can be considered and captured as a spike.

As the need to do this work arises, capture it along with other items on your backlog. To track that it's a spike, you can either preface the title with the word "[Spike]" or add the tag "Spike" to the work item.

Other resources

- What is Agile?
- Building productive, customer focused teams

Rollup of work and other fields

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Rollup provides summed values of select fields for all child work items of a parent. Because Azure DevOps Services and Team Foundation Server (TFS) support multiple levels of nesting, when you perform rollup, you want to make sure you don't double-count values. Most project managers are interested in getting rollup of estimated or completed work, effort, size, or story points.

> **NOTE**
> The system doesn't support rollup of the Effort, Story Points, or Size fields across product backlogs and portfolio backlogs.

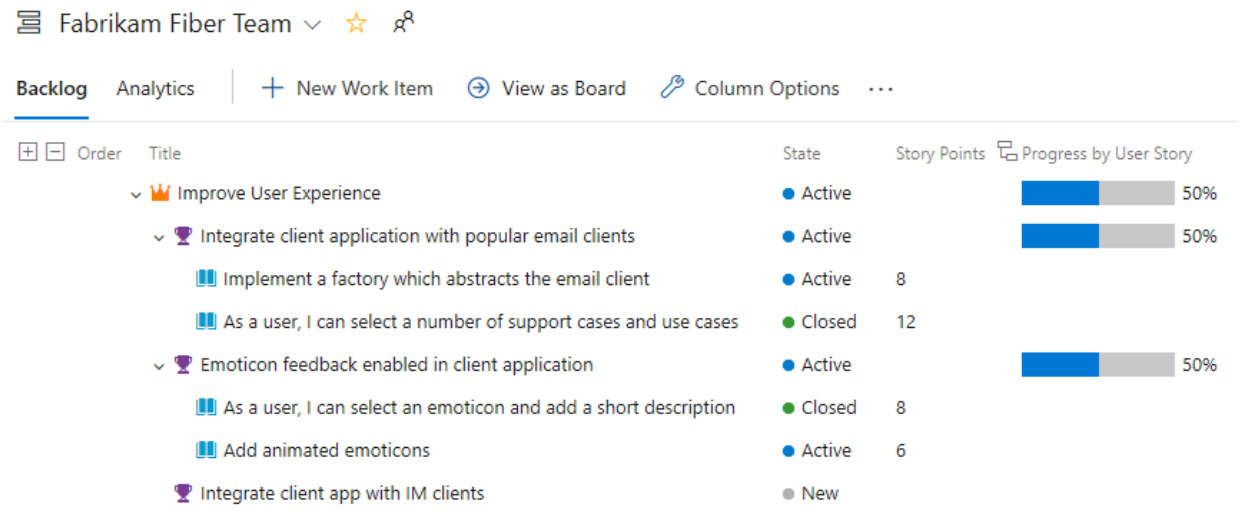
Native support of rollup within the web portal

Azure Boards supports rollup columns on product and portfolio backlogs, rollup within the sprint Planning pane, as well as rollup on sprint backlogs and taskboards.

Display rollup progress bars, counts, or totals

From any product or portfolio backlog, you can add rollup progress bars, counts of descendant work items within a hierarchy, or sum of select fields, such as Story Points or Remaining Work.

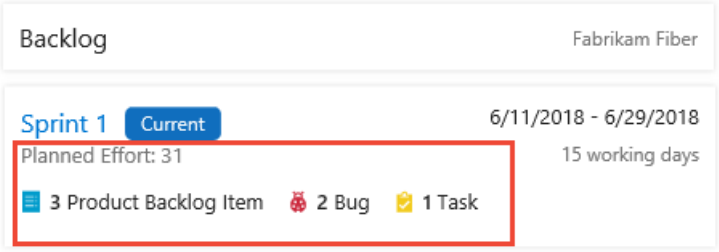
For example, progress bars are shown here for a portfolio backlog.



To learn more, see Display rollup progress or totals.

View rollup within sprint Planning pane

As you assign backlog items to a sprint using the Planning pane, the sprint window updates with a running tally of the number of backlog items and tasks, as well as the **Planned Effort**.



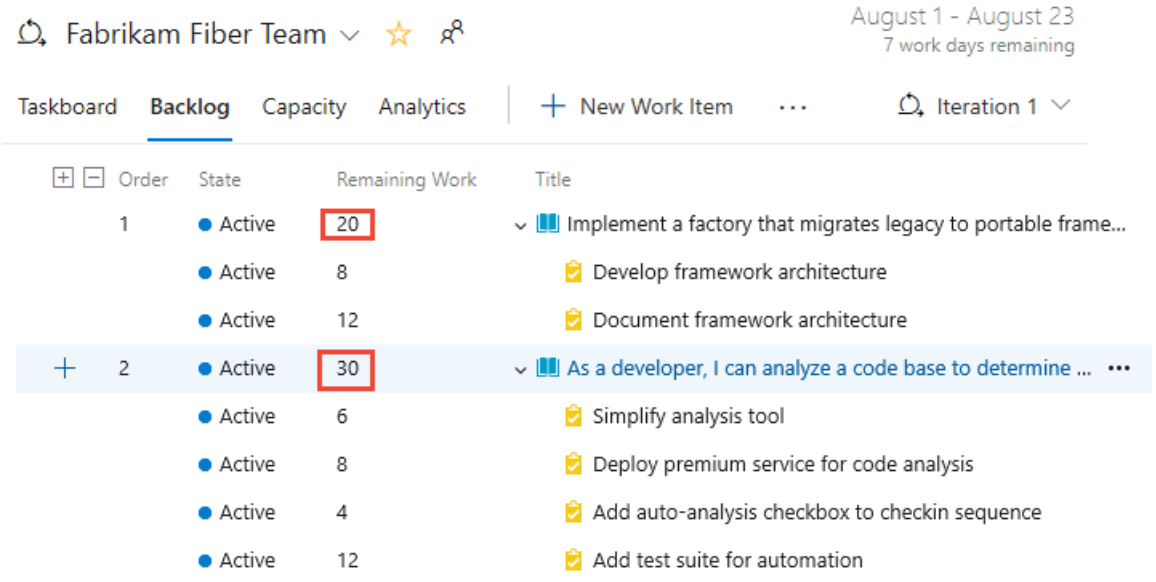
Planned Effort provides a sum of all *Story Points* or *Effort* defined for backlog items assigned to the sprint. To learn more, see Assign backlog items to a sprint.

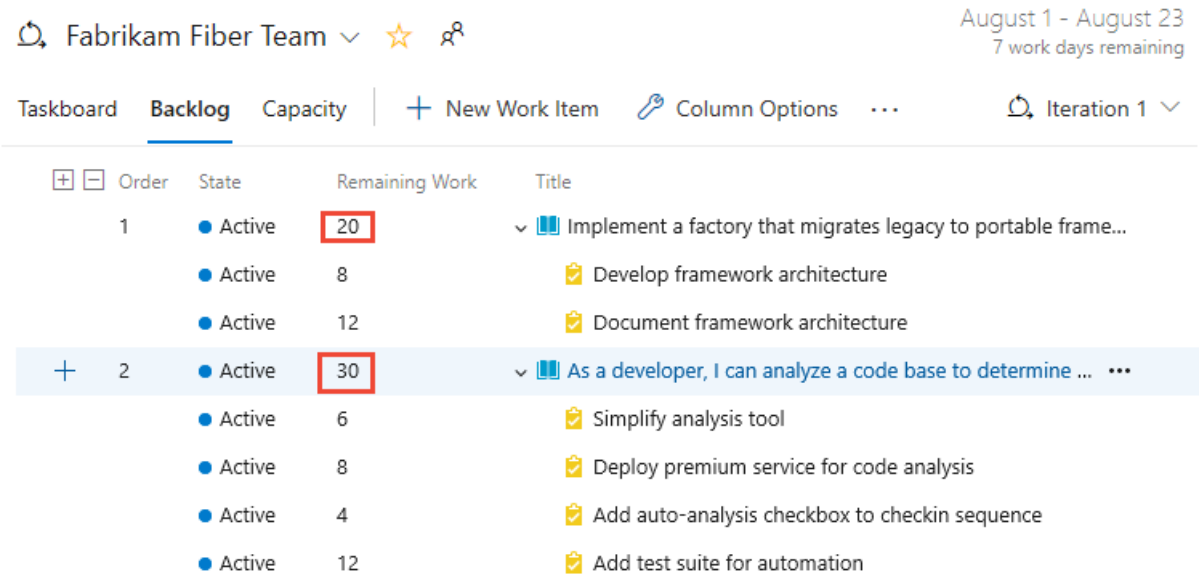
View rollup of sprint backlogs and taskboards

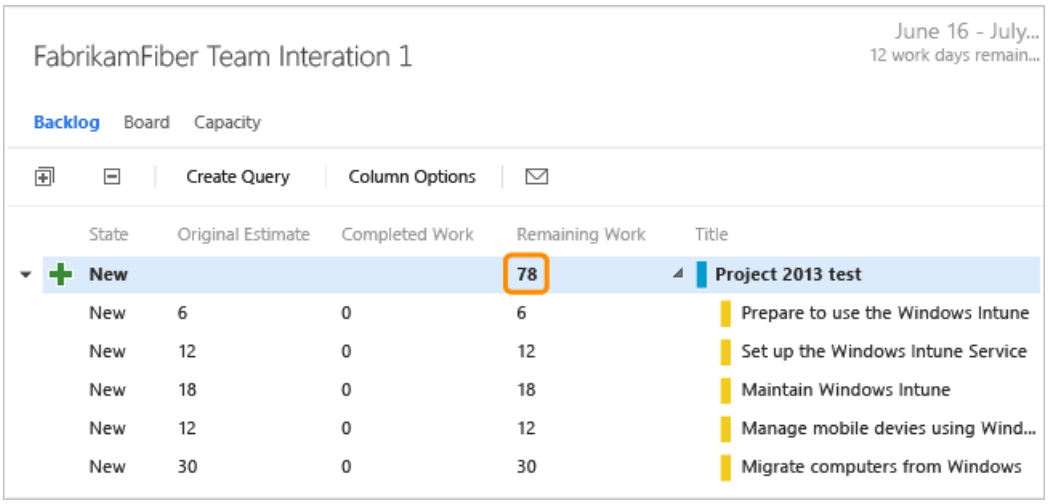
In addition, you can view rollup of Remaining Work from a sprint backlog or taskboard.

You can view rollup of Remaining Work from a sprint backlog or task board.

From the sprint backlog, the sum of all **Remaining Work** defined for all tasks is displayed for the parent work item. This value also displays on the parent work item card when you view the task board.

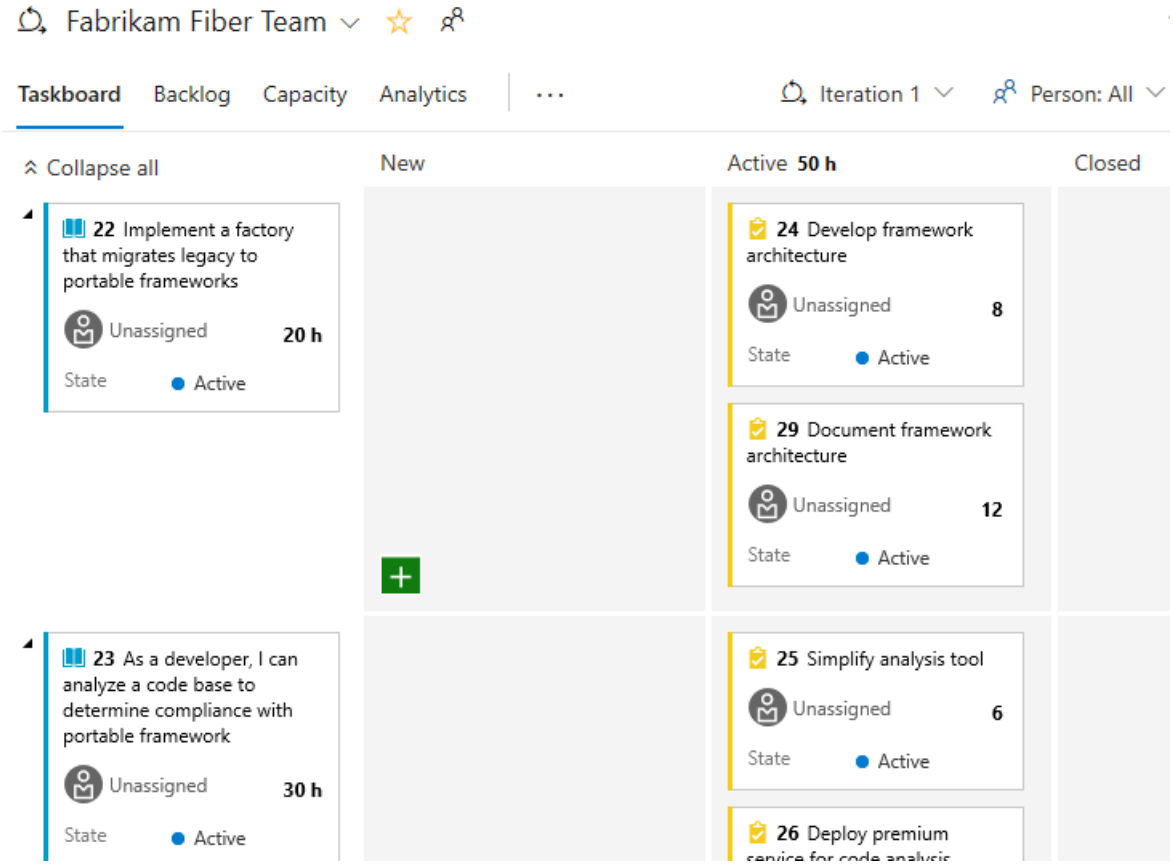


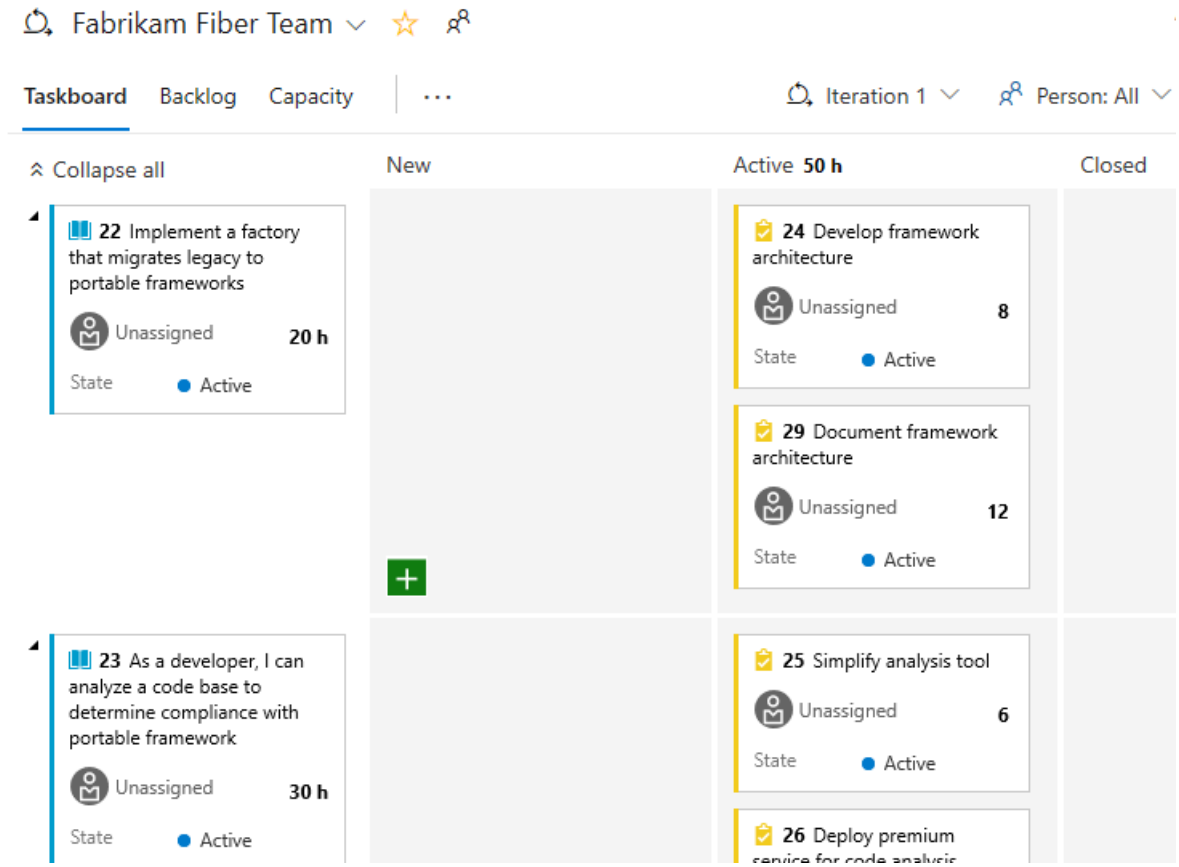


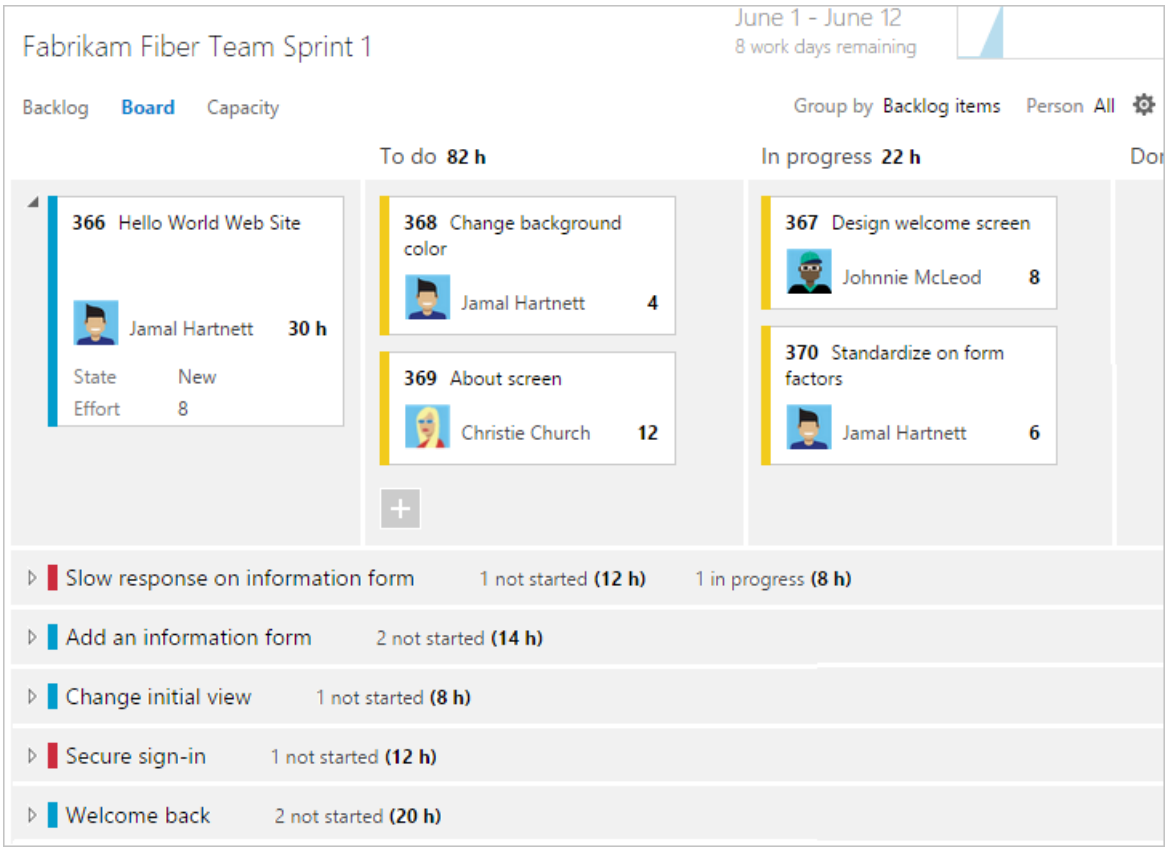


From a sprint task board, there are three types of rollup:

- The rollup of Remaining Work displays on the card for the parent work item
- The sum of all Remaining Work defined for all tasks within a column displays at the top of each column
- The sum of all Remaining Work defined for all tasks for a backlog item displays within each row, grouped by column.







When you update the status of a task as Completed, the system automatically zeros out the Remaining Work for that task. To learn more, see Task board.

Other tools that support rollup

You can obtain rollup of additional data fields in Azure DevOps work tracking data by using one of the following methods:

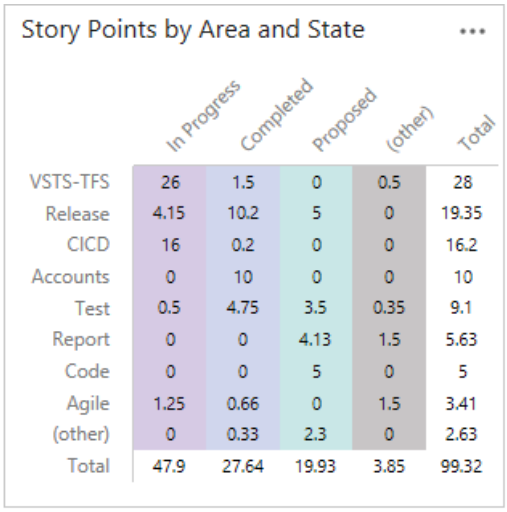
METHOD	AZURE DEVOPS SERVICES	ON-PREMISES AZURE DEVOPS
Work item query charts	✓	✓
Microsoft Excel	✓	✓
Marketplace extensions	✓	✓
Analytics	✓	✓
SQL Server Reporting Services report	✗	✓

- Work item query charts
- Microsoft Excel
- Microsoft Project
- Marketplace extensions
- SQL Server Reporting Services report

Work item query charts and rollup

You can create a flat list query that sums the values of a field you specify. To learn more, see Track progress by creating status and trend query-based charts. Charts support a count of work items or a sum of a field.

For example, here we show a pivot table that sums the story points for user stories by area path and work item state.



Microsoft Excel and rollup of work tracking data

You can export a query to Excel that contains the work items you want to provide rollup. You can then write an Excel macro to get the sums and publish data back to TFS. To learn more about Excel integration, see Bulk add or modify work items with Excel.

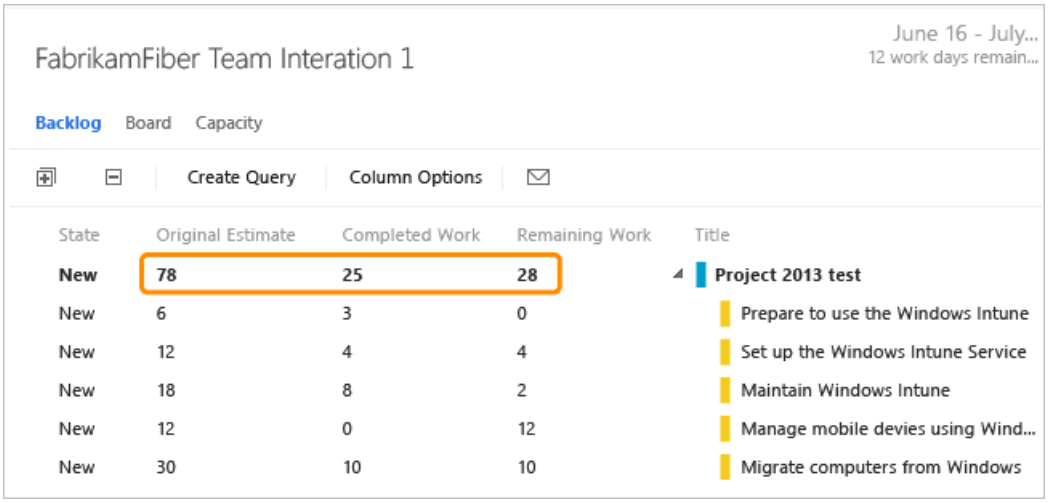
To learn more about Excel macros, see Automate tasks with the Macro Recorder.

TIP

To provide support for opening work items and query results in Excel from the web portal, add the VSTS Open in Excel Marketplace extension to your organization or collection.

Microsoft Project and rollup of work tracking data

Project natively supports rollup of summary tasks. With Project, you can round trip work tracking data to obtain rollup values.



To learn how, see Rollup estimated and actual work.

Marketplace extensions and custom controls

There are several extensions available from the Marketplace that provide rollup. Here are a few that may support your needs:

- VSTS Rollup, supports Azure DevOps Services only
- TFS Aggregator, supports Azure DevOps Services and TFS

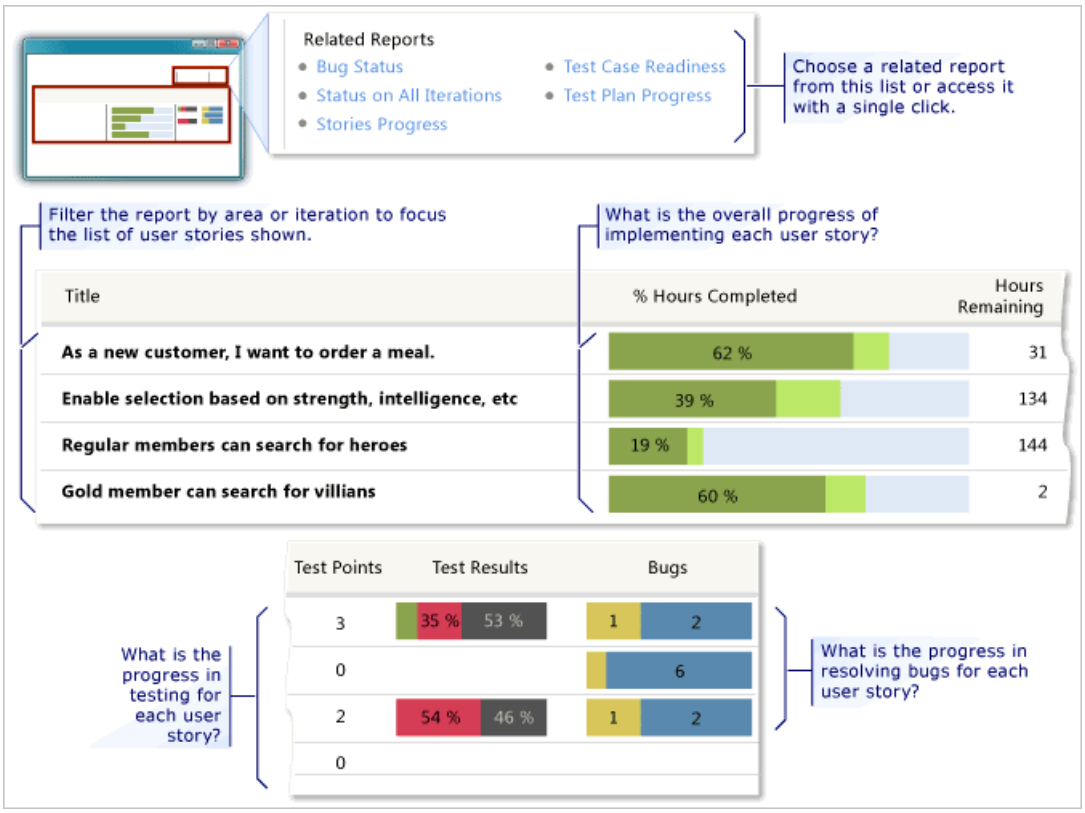
Or, you can write an extension using the REST API for work tracking to get rollup. A code sample available on github that can get you started is TFS Aggregator.

Analytics service

You can use the Analytics Service to answer quantitative questions about your projects. With this service, you can add Analytics widgets to your dashboard. Or, you can create additional reports using Power BI.

SQL Server Reporting Services report

Several OOB reports provide rollup. Here's an example of rollup of completed and remaining work that the Stories Overview report provides. This report is part of the default TFS Agile process template.



If you have SQL Server Analysis Services deployed, you can get rollup for backlog items from these reports. The refresh frequency for these reports is 2 hours.

- Backlog Overview
- Stories Overview
- Requirements Overview

If you need to add reports to your on-premises TFS deployment, see Add reports to a project.

Rollup requirements

To support rollup, structure your work items according to the following recommendations:

- Use parent-child links to link work items that contain values that you want to rollup.
- Add required fields to the WITs that will capture the rollup values. Default fields used to schedule work are only present on the task work item. These fields are:
  - Original Estimate (Microsoft.VSTS.Scheduling.OriginalEstimate): The amount of work required to complete a task. (Agile and CMMI)
  - Completed Work (Microsoft.VSTS.Scheduling.CompletedWork): The amount of work that has been spent implementing a task. (Agile and CMMI)
  - Remaining Work (Microsoft.VSTS.Scheduling.RemainingWork): This field is used to support burndown charts.

    If your project was created using the Visual Studio Scrum process template, only Remaining Work is defined in the task.

    To learn more about adding fields, see Modify a field or add a custom field.
- Determine the unit of time used to track work and make sure it is used consistently across your team or organization. For example, you can track tasks using hours or days.

- Determine if you want to make rollup values read-only on the work item form. By making them read-only you prevent users from entering inaccurate data. You make fields read-only using the `Control` field `Readonly` attribute.

Q & A

Q: Can I get rollup of team capacity?

A: No. The data entered for team capacity isn't stored in the regular data stores.

View or configure team velocity

10/4/2022 • 13 minutes to read • Edit Online

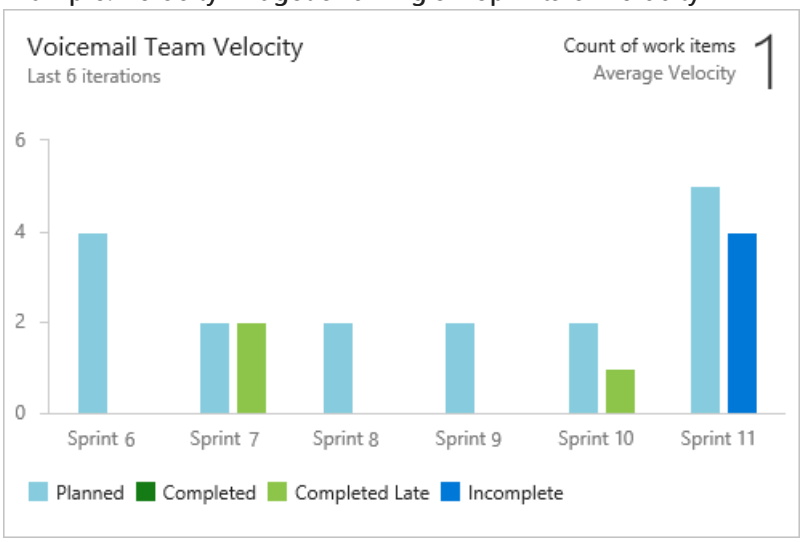
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Teams track their velocity to help them determine how much work they can do sprint-over-sprint. Velocity provides an indication of how much work a team can complete during a sprint based on either:

- A count of work items completed.
- The sum of estimates made to:
  - Effort (product backlog items).
  - Story Points (user stories).
  - Size (requirements).

There are two velocity charts: the in-context report you can view from a team backlog or Kanban board and the Velocity widget you can add to a dashboard.

Example: Velocity widget showing six sprints of velocity



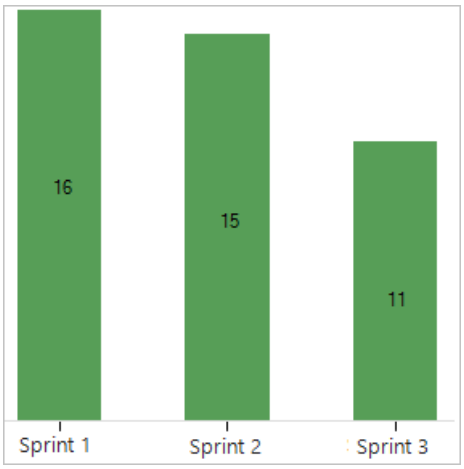
> **NOTE**
> The Velocity widget is based on Analytics data. Analytics is generally available for Azure DevOps Services and Azure DevOps Server 2020 and later versions. Analytics is in preview as an extension for Azure DevOps Server 2019. For TFS 2018 and earlier versions, you have access to the velocity chart provided by the work tracking data store.

Velocity provides a useful metric for gaining insight into how much work your team can complete during a sprint cycle. Each team is associated with one and only one velocity chart.

> **NOTE**
> For TFS 2018 and earlier versions, you can only view team velocity. There is no configuration of this report.

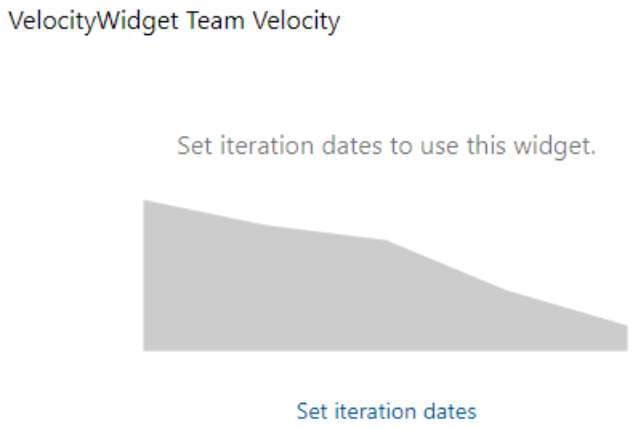
Velocity will vary depending on team capacity, sprint over sprint. However, over time, the velocity should indicate a reliable average that can be used to forecast the full backlog.

Example Velocity chart from the work tracking data store



Once your team has completed a few sprints, they can use their velocity to forecast how much of the backlog they can finish within upcoming sprints. If your team hasn't completed a sprint or if you're working on items before a sprint start date, Velocity would have no data to analyze and forecast. You might see this message: *Set iteration dates to use this widget*. To resolve this situation, set an iteration date range to include present date or wait for the sprint to start. For usage guidance, see Velocity metrics and usage guidance.

Example: Velocity showing Set iteration dates to use this widget



> **NOTE**
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your web app, options that you or your admin have enabled, and which process was chosen when creating your project—Agile, Basic, Scrum, or CMMI. The Basic process is available with Azure DevOps Server 2019 Update 1 and later versions.

Use this article to learn:

- How to configure the Velocity widget (Analytics)
- How to view the Velocity in-context report (Analytics)
- Required and recommended team activities to support velocity tracking

- How to configure the Velocity widget (Analytics)
- How to view the Velocity in-context report (work tracking data store)
- Required and recommended team activities to support velocity tracking

- How to view the Velocity in-context report (work tracking data store)
- Required and recommended team activities to support velocity tracking

Prerequisites

- You must be a member of a project. If you don't have a team project yet, create one.
- If you haven't been added as a project member, get added now.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Have enabled or installed Analytics. You must be an account owner or a member of the **Project Collection Administrators** group to add extensions or enable the service.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

Open your backlog from the web portal

1. Check that you selected the right project, and select **Boards** > **Backlogs**. Then select the correct team from the team selector menu.

   To select another backlog, open the selector and then select a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.

2. To view the in-context reports for the product backlog, check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level. Or

1. Check that you selected the right project, and select **Boards** > **Backlogs**. Then select the correct team from the team selector menu.

   To select another backlog, open the selector and then select a different team or select the **Browse all backlogs** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.

2. To view the in-context reports for the product backlog, check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level.

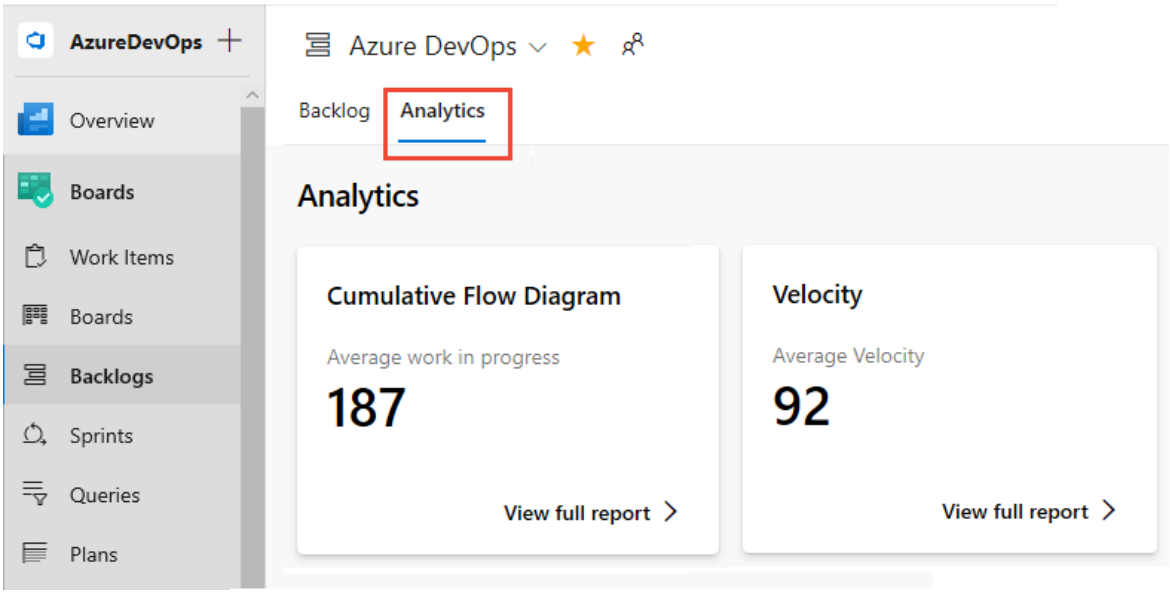
From your web portal, open your team's product backlog and select the team from the project and team selector. Then select **Work** > **Backlogs**. Select the product backlog, which is **Backlog items** for Scrum, **Stories** for Agile, or **Requirements** for CMMI.

To select another team, open the project and team selector. Select a different team, or select the **Browse** option.

View the Velocity in-context report

Velocity reports are available for each backlog level, both product and portfolio backlogs. Each report provides interactive controls to provide each user the view of interest to them.

1. You open the Velocity report for your product or portfolio backlog by choosing **Analytics**.



2. To change to a different backlog, choose from the backlog selector.
3. Next, select **View full report** for Velocity.
4. Use the interactive controls to select the count or sum field and number of iterations. Select **Custom iterations** to specify any number of iterations between 1 and 15.

   Hover over a column area to show a summary of planned and completed work items.

   For example, for the 07_2019 sprint, 131 items are planned.



With the following assignments made to the color bars. All work items counted are assigned to the sprint or iteration.

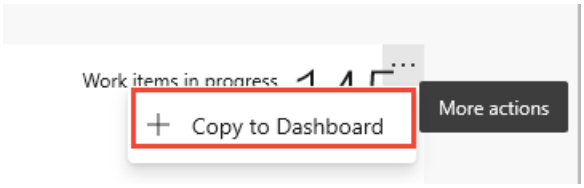
- **Planned**: calculated based on the amount of work assigned to the sprint before the start of the sprint. This count includes work that was moved to a different sprint after the start of the sprint, but doesn't include work that was added later after the sprint started.

  > **TIP**
  > To list the work items included in the count, click the velocity bar. A query results page will open with the list of work items included.

- **Completed**: calculated based on the amount of work assigned to the sprint and whose workflow **State** corresponds to the *Completed* category state, and completed on or before the sprint end date.
- **Completed Late**: calculated based on the amount of work assigned to the sprint that is completed after the sprint end date. If you assign work items to a sprint, even one in the past, they will show up in this calculation once the workflow **State** corresponds to a *Completed* category state.
- **Incomplete**: Amount of work not completed, calculated based on the amount of work assigned to the sprint, and the workflow **State** corresponds to the *In Progress* category state.

  > **NOTE**
  > Items assigned to a *Proposed* or *Resolved* workflow state category aren't included in any of the calculations for **Completed**, **Completed Late**, or **Incomplete**. To learn more about category states, see How workflow category states are used in Azure Boards backlogs and boards. The selections you make are only set for you, and persist across sessions until you change them.

5. To add the report to a dashboard, select the ··· actions icon and select **Copy to Dashboard**.

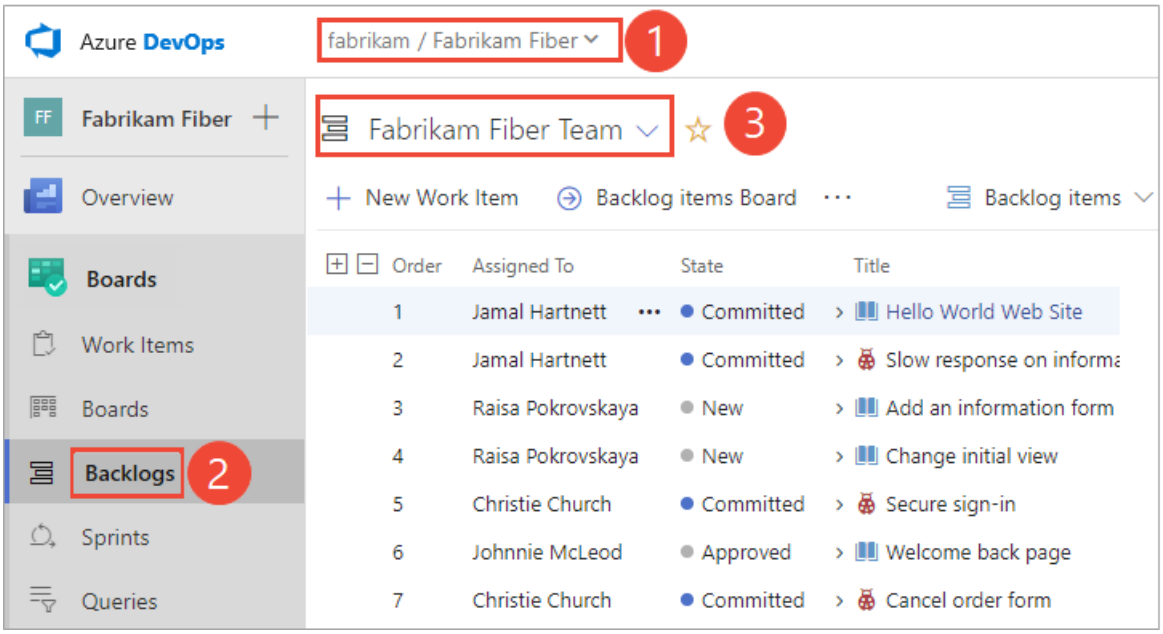


Select the dashboard and select **OK**.

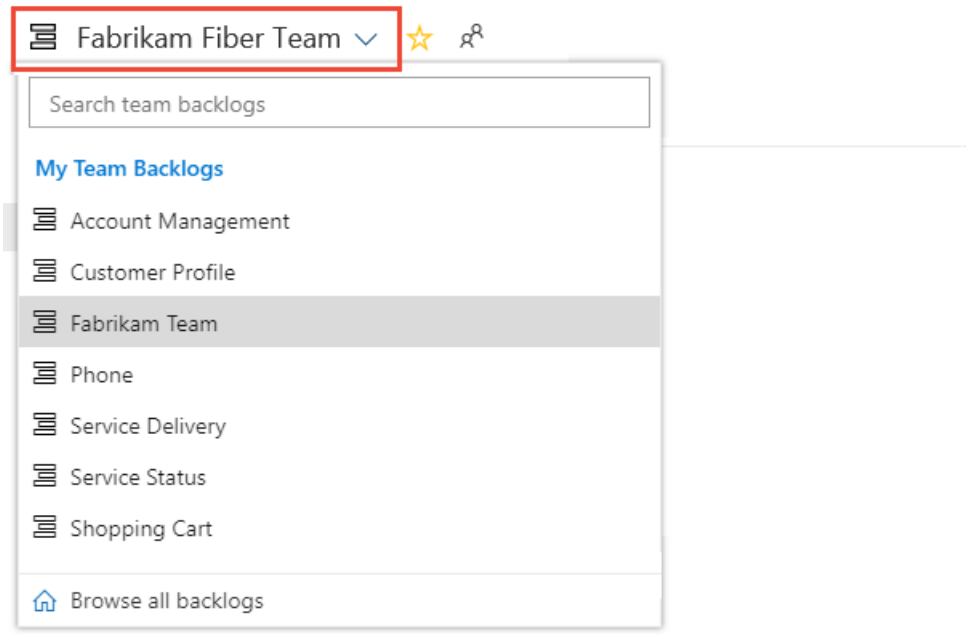
6. To return to the Analytics summary, select the ← back arrow.

1. From the web portal, open your product backlog.

2. (1) Check that you've selected the right project, (2) select **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



To select another team, open the selector and select a different team or select the **Browse all backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



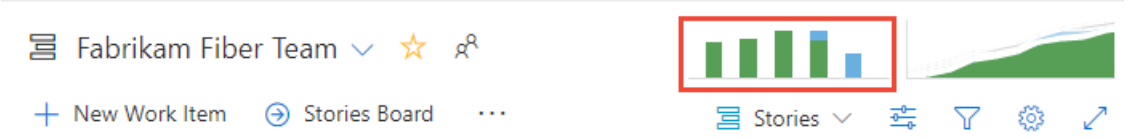
TIP

Select the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

3. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level.



4. Open the velocity chart.

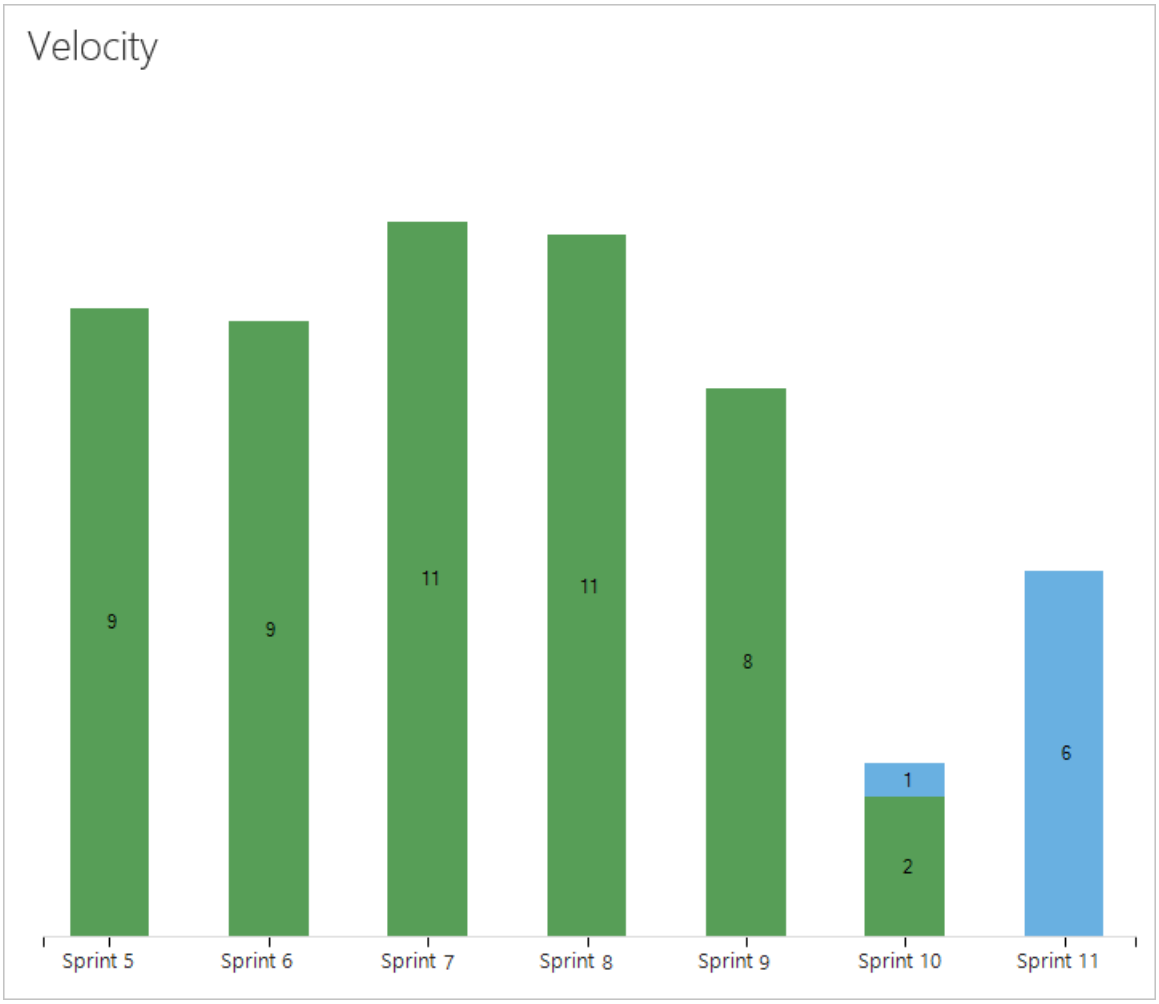


   For charts to appear, your team must carry out these activities:

   - Select sprints for your team.
   - Assign backlog items to sprints.
   - Estimate backlog items by defining the Effort, Story Points, or Size.

5. The chart tracks your estimated backlog work (sum of Effort, Story Points, or Size) that your team has completed (green) in the previous sprints, or that are still in progress (blue).

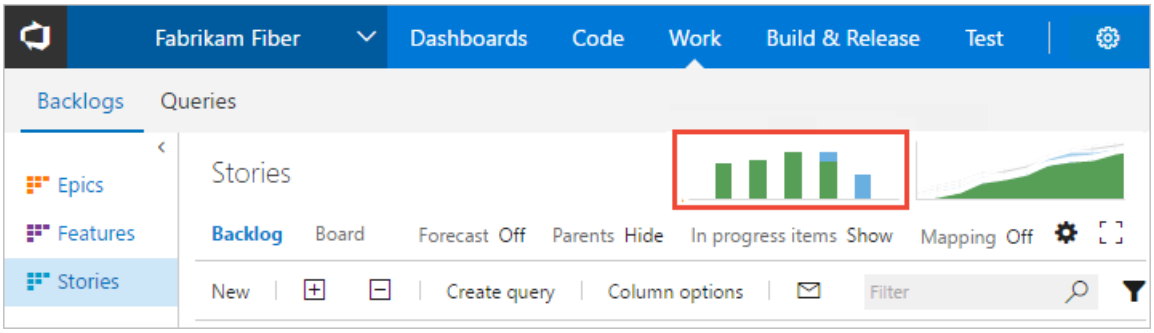
   As this chart shows, velocity tends to fluctuate from sprint-to-sprint for different kinds of reasons. However, you can quickly determine the average velocity by averaging the values shown in green for each sprint. You can then plug the average into the Forecast tool.



NOTE

Work items based on the Scrum process get counted in the chart once their State is set to Committed, whereas items based on the Agile and CMMI processes get counted once their State is set to Active. This behavior is set through the workflow states to category state mappings.

1. From the web portal, open the product backlog and then select the velocity chart.



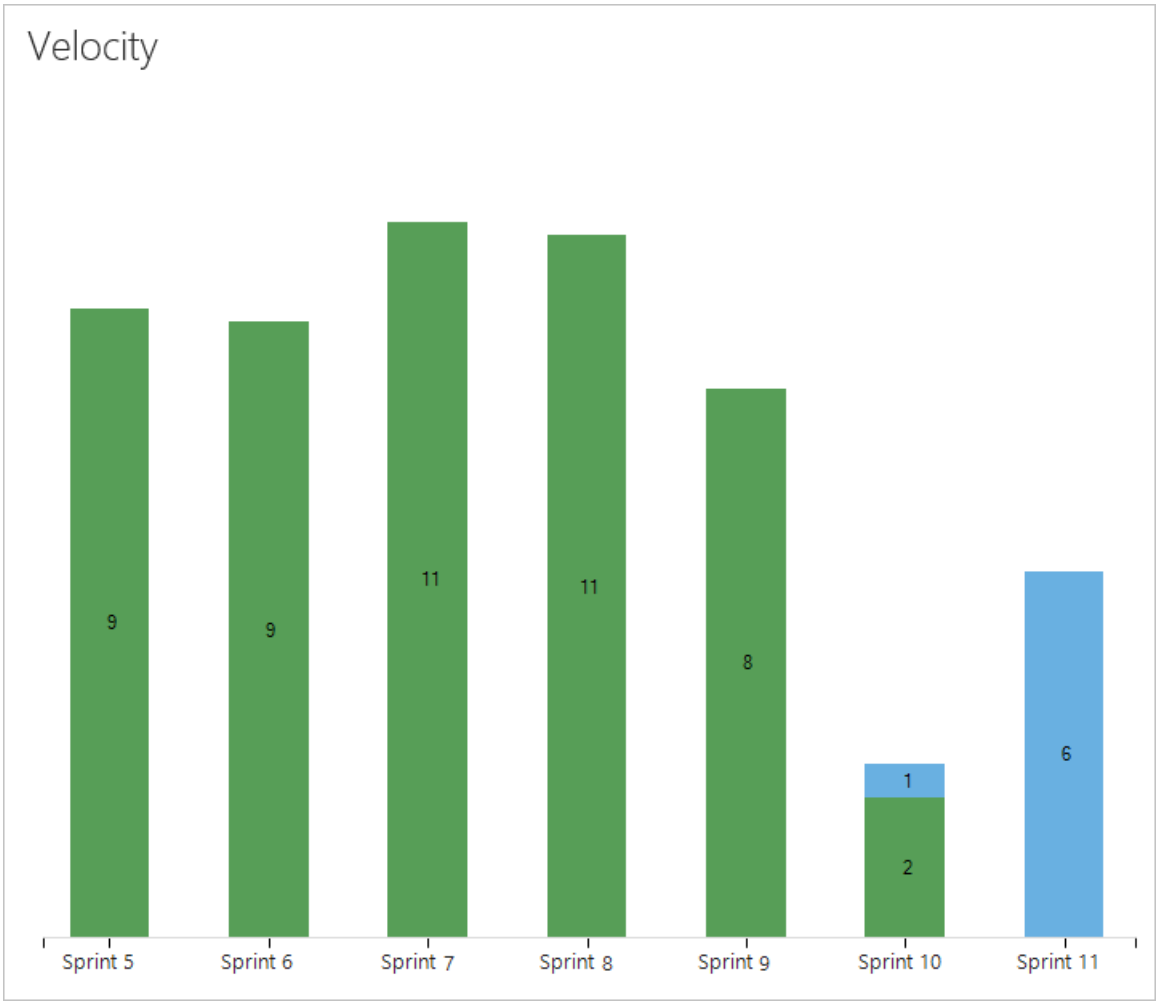
   For charts to appear, your team must carry out these activities:

   - Select sprints for your team.
   - Assign backlog items to sprints.
   - Estimate backlog items by defining the Effort, Story Points, or Size.

2. The report tracks your estimated backlog work (sum of Effort, Story Points, or Size) that your team has completed (green) in the previous sprints, or that are still in progress (blue).

   As this chart shows, velocity will fluctuate from sprint-to-sprint for different kinds of reasons. However, you can quickly determine the average velocity by averaging the values shown in green for each sprint.

You can then plug the average into the Forecast tool.



NOTE

Work items based on the Scrum process get counted in the chart once their State is set to Committed, whereas items based on the Agile and CMMI processes get counted once their State is set to Active. This behavior is set through the workflow states to category state mappings.

Add the Velocity widget to your dashboard

- If you haven't yet added the Velocity widget to your dashboard, do that now.

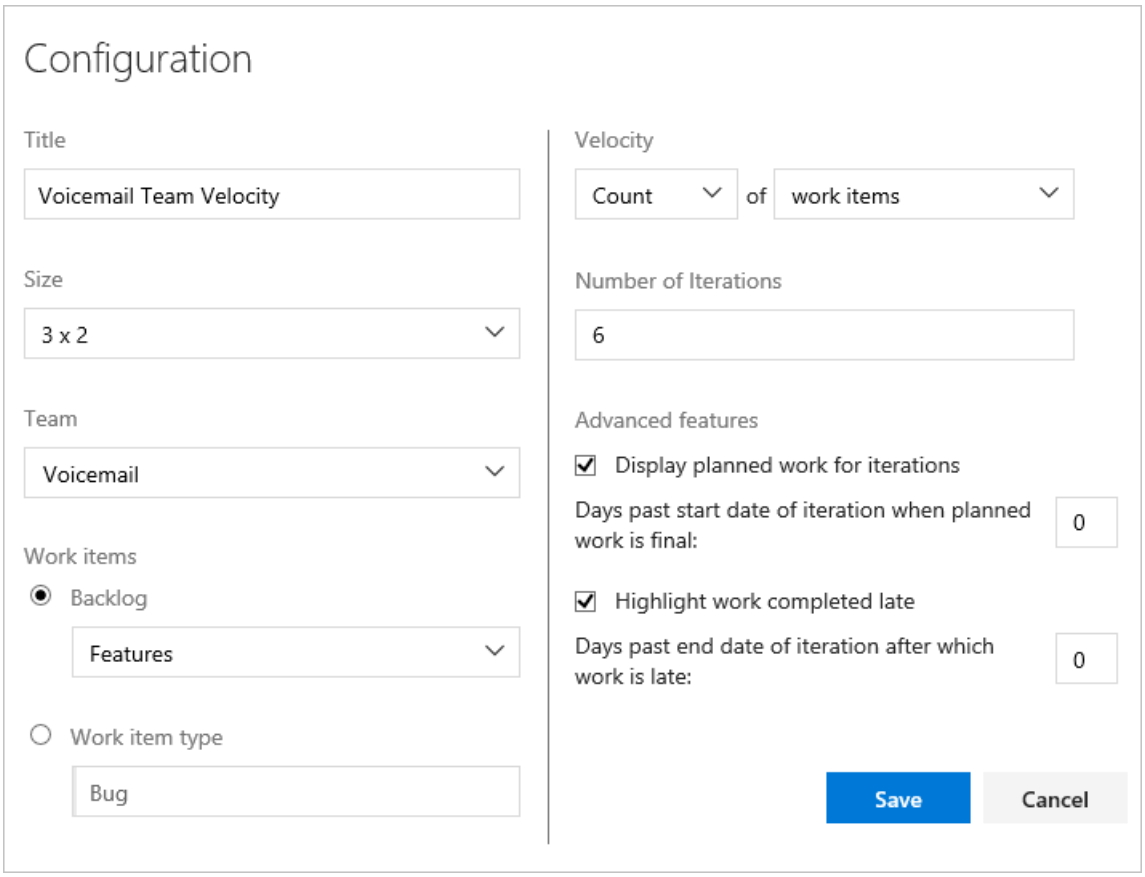
1. If you haven't yet enabled or installed Analytics], do that now.
2. If you haven't yet added the Velocity widget to your dashboard, do that now.

Configure the Velocity widget

You configure your velocity widget for a single team. If you want to view the velocity for several teams, then you must configure a portfolio management team that rolls up from several teams. To learn more about teams, see Add teams.

1. Select the ··· actions icon and select the Configure option to open the configuration dialog.

   Modify the title, select the team, and then select either the backlog level or work item type to track. Select whether you want to track a count of work items or a sum of a numeric field. The most common summed field is that of Effort, Story Points, or Size.



2. Specify the number of sprints you want to view. The default is 6 and the maximum is 15.

3. (Optional) Select the check boxes to show additional information for work completed later than planned for each sprint.

   Displayed planned work for iterations: Check this box to display the amount of work planned for an iteration at the start of the iteration. This is useful for comparing your planned work to actual deliverables. By default, the count of planned work begins on the start date of the iteration.

   - ***Days past start date of iteration when planned work is final:*** Specify the number of days past the start date to count planned work. For example, if the first two days of an iteration are for planning, then you can enter "3", and planned work will be counted on the third day.

     For example, if the Iteration starts on 01/01/2018, and three backlog items are assigned to the iteration on 01/01/2018 end-of-day, then those three backlog items will be considered as Planned. If your team doesn't complete planning until a few days into the iteration, then you can update the Days past start date of iteration when planned work is final.

     > **NOTE**
     > Work is considered Planned if it is assigned to the iteration as-of the Iteration Start Date.

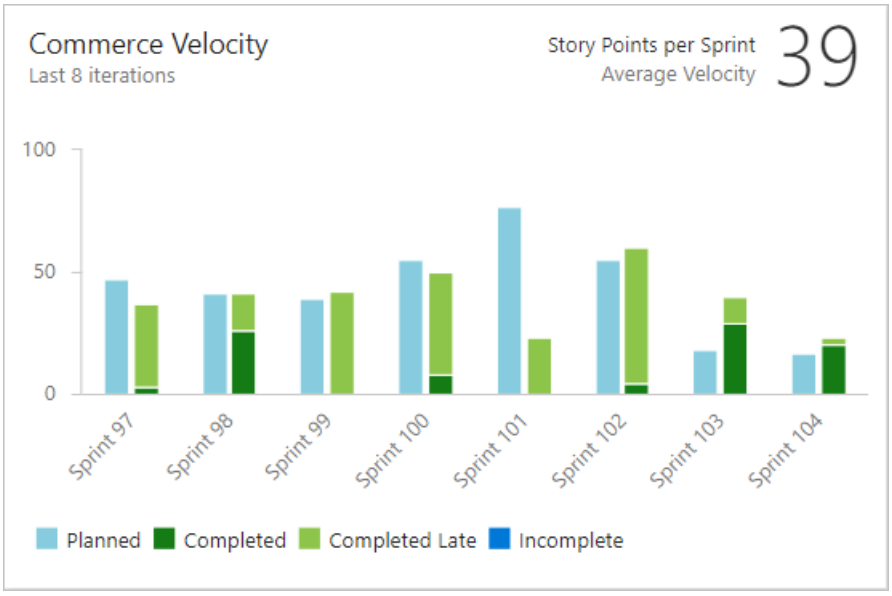
     Highlight work completed late: Work items marked complete after the iteration end date are considered to be completed late and will show as light green. This is useful for spotting a trend where work items are marked complete after the iteration is complete.

   - ***Days past end date of iteration after which work is late:*** Specify the number of days past which you consider a work item late if its status is still new or in progress.

     For example, entering three days will give the team 3 days after the end of an iteration to mark work items complete or done, before they're considered late.

> **NOTE**
>
> A work item is considered late when the work item's Completed Date is later than End Date of the Iteration the work item is *currently* assigned to.
>
> It will take into account the value you enter for *Days past end date of iteration after which work is late*.

4. Select **Save** when done. The following image shows Velocity based on Story Points and eight sprints of data.



For information on **Planned**, **Completed**, **Completed Late**, and **Incomplete**, see the velocity legend earlier in the article.

Required and recommended activities

For your team to gain the greatest utility from the velocity charts, follow these required and recommended tasks.

Required:

- Define iteration paths (aka sprints) and configure team iterations Sprints should be of the same duration.
- Define and estimate backlog items. If you work from your team's backlog, the items you create will automatically be assigned to the current sprint (Iteration) and to your team's default Area Path.
- Update the status of backlog items once work starts and when completed. Only backlog items whose State maps to a metastate of In Progress or Done will show up on the velocity chart or velocity widget.

Recommended:

- Define and size backlog items to minimize variability.
- Determine how your team wants to treat bugs. If your team chooses to treat bugs like requirements, bugs will show up on the backlog and be counted within the Velocity chart and forecasting.
- Set your team's area path. The forecast tool will forecast those items based on your team's default settings. These settings can specify to include items in area paths under the team's default or exclude them.
- Don't create a hierarchy of backlog items and bugs. The Kanban board, sprint backlog, and task board only show the last node in a hierarchy, called the leaf node. For example, if you link items within a hierarchy that is four levels deep, only the items at the fourth level appear on the Kanban board, sprint backlog, and task board.
  Instead of nesting requirements, bugs, and tasks, we recommend that you maintain a flat list-only creating

parent-child links one level deep between items. Use Features to group requirements or user stories. You can quickly map stories to features, which creates parent-child links in the background.

- At the end of the sprint, update the status of those backlog items that the team has fully completed. Incomplete items should be moved back to the product backlog and considered in a future sprint planning meeting.

Add other teams

If you work with several teams, and each team wants to work with their own backlog view, velocity chart, and forecast tool, you can add teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items to only include those whose assigned area paths and iteration paths meet those set for the team.

Next steps

Velocity guidance

Implement Scrum practices for your team in Azure Boards

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Your Sprints tools include a filtered backlog based on an Iteration Path and a similarly filtered taskboard. These tools are useful for implementing Scrum practices. With Scrum, you can schedule and plan sprints, update your taskboard, and monitor your sprint burndown.

Scrum methods use Iteration Paths, also referred to as sprints, to plan work to be performed by a team within a specific time period and cadence. To get started, several sprints are predefined for your team. If you're new to Scrum, get an overview from What is Scrum?.

> **NOTE**
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans.
> If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

Use Azure Boards to implement Scrum

The general sequence of steps for implementing Scrum using Azure Boards is as follows:

Configure teams and sprints

1. Define project-level Iteration Paths and set dates
2. (Optional) Add project-level Area Paths (Or, add an area path when you configure each team)
3. Add teams
4. Select team-level Iteration Paths.

Create team backlog

1. Create and organize your team backlog.
2. (Optional) Forecast your team backlog.

Implement a sprint

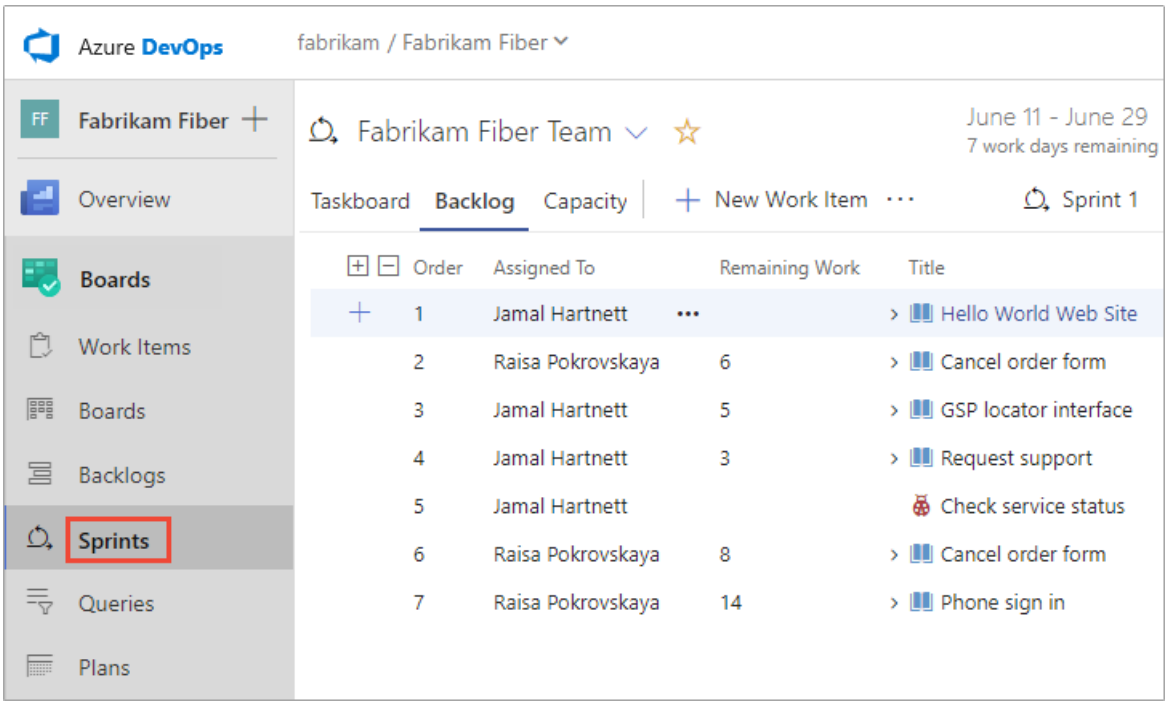
You can quickly assign work items to a sprint by dragging and dropping them from the product backlog to the sprint.

1. Assign backlog items to a sprint
2. Add tasks to backlog items
3. Set sprint capacity
4. Adjust work to fit sprint capacity
5. (Optional) Share your sprint plan
6. Update the Taskboard
7. Monitor your sprint burndown

Sprint backlogs and taskboards overview

Sprint backlogs and taskboards provide a filtered view of work items a team has assigned to a specific iteration

path, or sprint. Sprints are defined for a project and then selected by teams. From your backlog, you can map work to an iteration path using drag-and-drop, and then view that work in a separate **sprint backlog**.

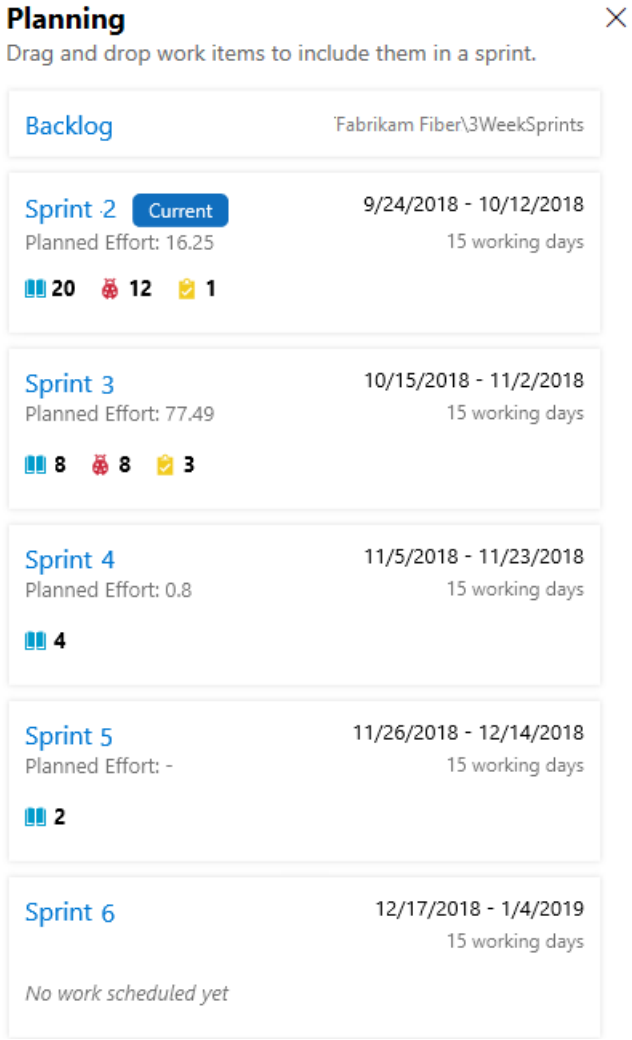


"Web portal, choose Boards>Backlogs, Sprint

How selected sprints show up on the backlog

Each sprint that you select for your team provides access to a sprint backlog, taskboard, and other Agile tools for planning and tracking work.

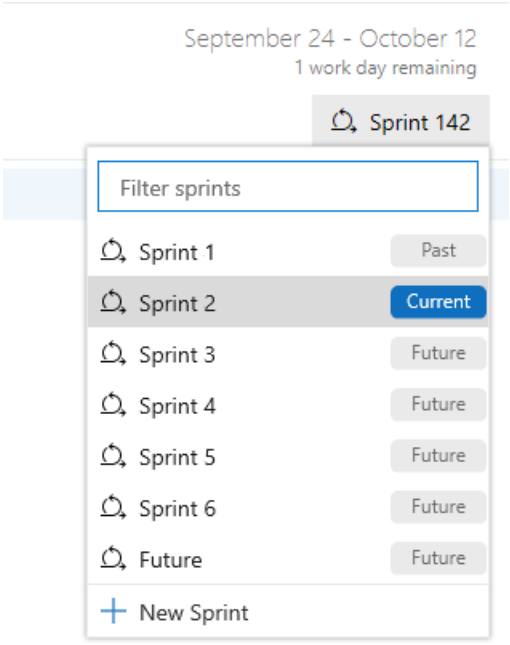
1. You can gain an overview of your sprint planning by turning on the **Planning** view option. From the product backlog or any sprint backlog, choose the view options icon and select **Planning**.



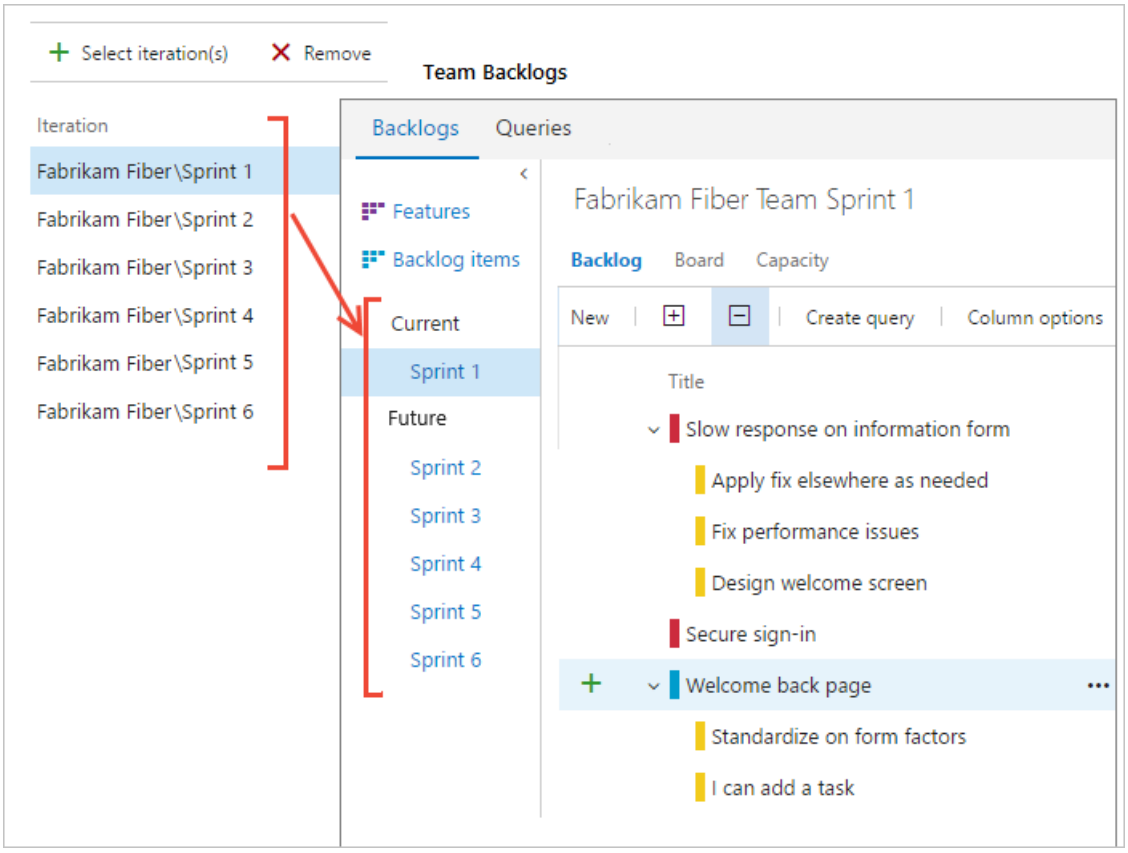
> **NOTE**
> The Planning pane will only show the current sprint and the next 10 future sprints in the list, even if more have been selected for the team.

The set of sprints selected for your team appears. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. To select a sprint backlog, you can choose one of the sprint links from the **Planning** pane, or from a Sprint backlog, choose a sprint from the sprint selector.



For example, by selecting Sprints 1 through 6, the Fabrikam Fiber team gets access to six sprint backlogs. They also get access to capacity planning tools and a taskboard for each sprint.



Track team capacity

Once you've defined iteration (sprint) paths and configured team iterations, you can start using the following tools to plan your sprint.

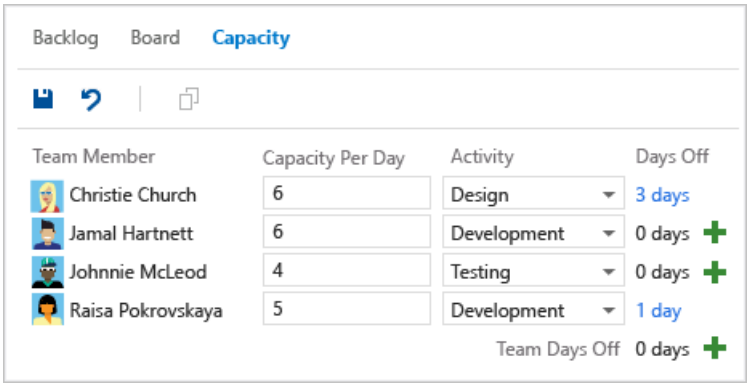
At the start of each sprint, you'll want to plan the work that your team can commit to. The three Agile tools that support this work include the sprint backlog, capacity planning, and capacity bars. The sprint backlog contains a filtered subset of backlog items whose iteration path corresponds to the current sprint.

Team capacity planning tool

By setting team capacity, the team knows exactly the total number of work hours or days the team has for each sprint. With this tool, you set individual team member capacity and days off. And, conveniently, you can set holidays or shared days off taken by the entire team.

Setting capacity for each team member working during a sprint causes the capacity bar for that individual to appear.

You set recurring days off, such as weekends, through team settings.

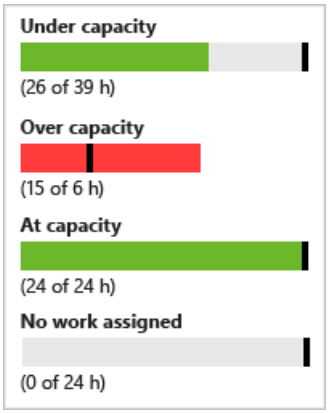


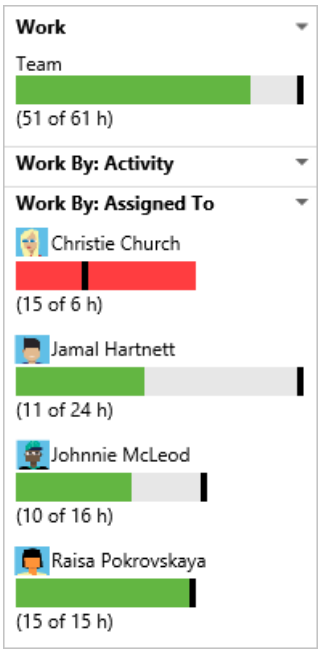
Individual and team capacity bars

With capacity bars, you can quickly see who is over, at, or under capacity. Capacity bars update with each of these activities:

- Tasks are assigned with non-zero remaining work
- Change in remaining work
- Date change within the sprint cycle. Individual and team capacity always reflects their capacity from the current day until the end of the sprint.

Here's how to interpret the capacity colors:





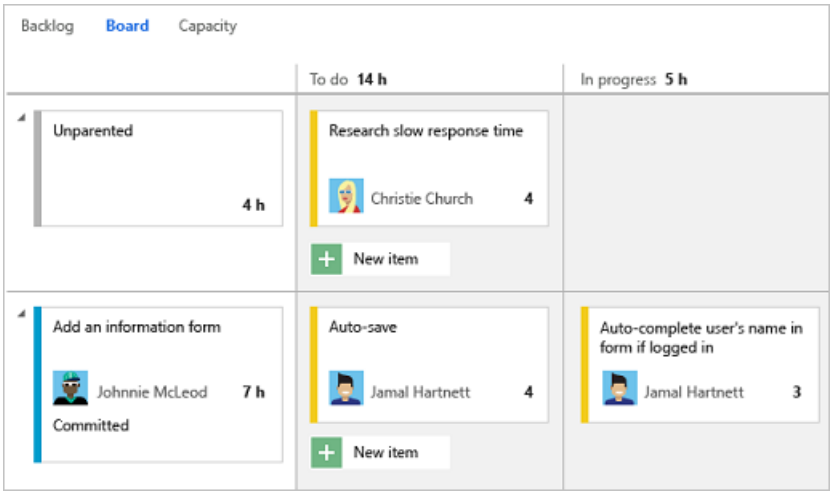
Update tasks and monitor burndown

During a sprint, use the taskboard and sprint burndown chart to track their progress. Your sprint burndown chart provides you with an at-a-glance visual to determine if your team is on track to meet their sprint plan.

Taskboard

Your Taskboard provides an interactive progress board for work required to complete the sprint backlog. During your sprint, you'll want to update the status of tasks and the remaining work for each task.

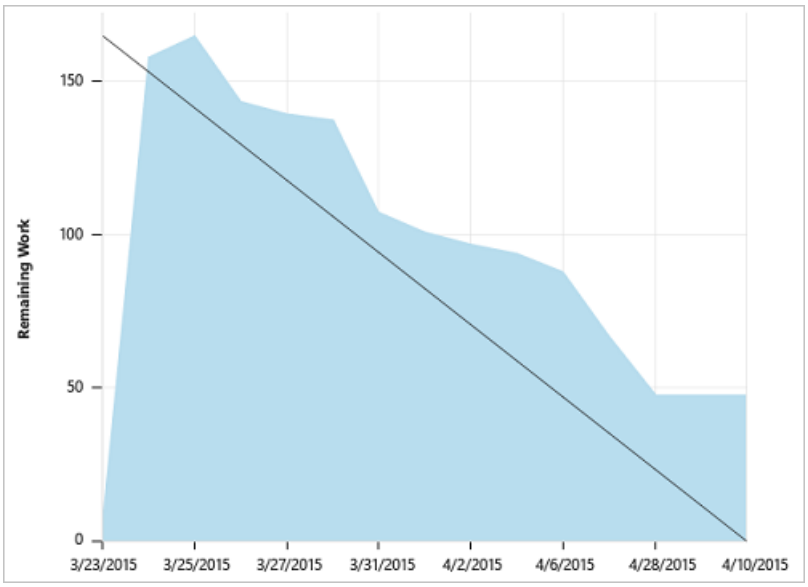
Updating tasks daily or several times a week yields a smoother burndown chart.



Sprint burndown chart

You use the sprint burndown chart to mitigate risk and check for scope creep throughout your sprint cycle. The burndown chart reflects the progress made by your team in completing all the work they estimated during their sprint planning meeting.

The ideal trend line always indicates a steady burndown. The blue area, however, represents what's actually going on. It shows the buildup of work as team members add tasks and the reduction of work as team members complete those tasks.



Use velocity and forecast tools to predict work effort

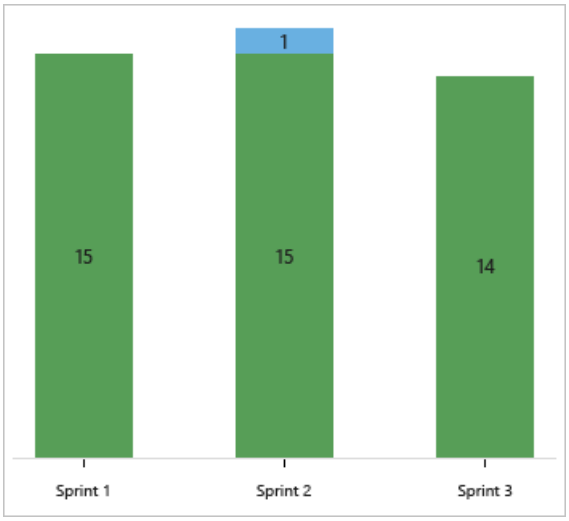
Use sprint planning and tracking tools for each sprint. Also, use the velocity and forecast tools to estimate work that can be completed in future sprints.

Velocity provides a useful metric for gaining insight into how much work your team can complete during a sprint cycle. And, the forecast tool provides a means for determining how much work your team can complete within a sprint. This amount is based on a specified team velocity.

After several sprints, use the Velocity chart and Forecast tool tool to estimate work that can be accomplished in future sprints.

Velocity chart

Each team is associated with one and only one velocity chart. The green bar within the chart indicates the total estimated effort (story points or size) of backlog items (user stories or requirements) completed within the sprint. (Blue corresponds to the estimated effort of items not yet completed.)
Velocity will vary depending on team capacity, sprint over sprint. However, over time, the velocity should indicate a reliable average that can be used to forecast the full backlog.
By minimizing the variability of backlog item size—effort or story points—you gain more reliable velocity metrics.



Forecast tool

You can use the forecast tool to get an idea of how many and which items you can complete within a sprint.

By plugging in a velocity, you can see which items are within scope for the set of sprints the team has selected. As shown here, a velocity of 15 indicates that it will take three sprints to complete the work shown.*

Forecasting based on velocity of **15**

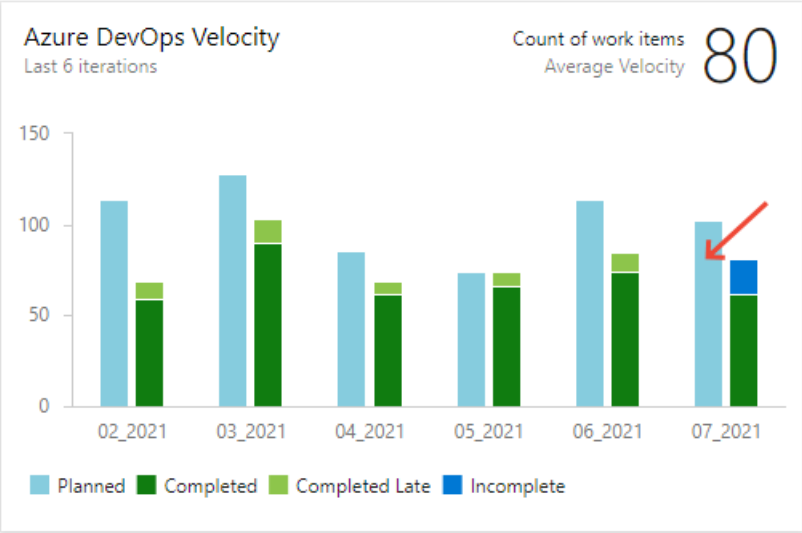
Forecast	Order	Effort	Title
	1	5	Add an information form
	2	5	Secure sign-in
Sprint 2	3	3	Welcome back page
	4	8	Interim save on long forms
Sprint 3	5	5	Canadian addresses don't display correctly
	6	5	Log on
	7	2	Lookup service outages on portal
	8	3	Coverage map on website
	9	2	Resume
Sprint 4	10	1	Pause

Query sprint scope changes

There isn't a sprint scope change chart or widget. However, you can query for work items added to a sprint or moved out of a sprint after the start of the sprint. Use the steps provided next.

List work items added after the start of the sprint

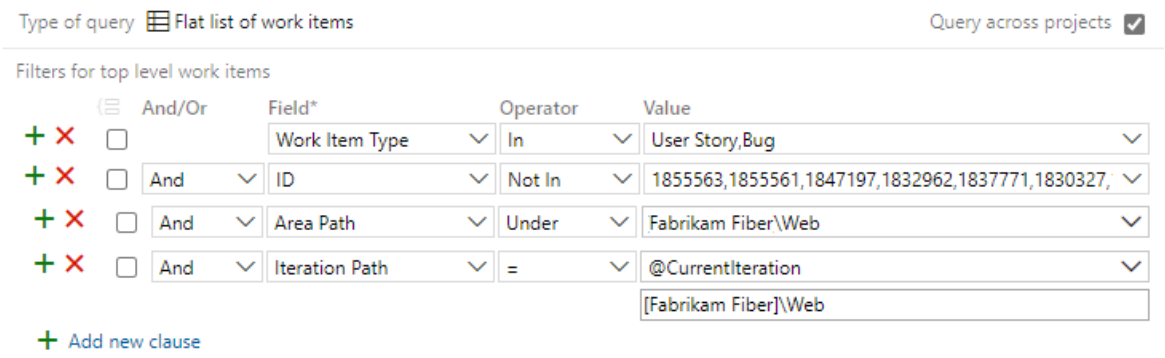
1. Open the velocity chart for the team and choose the **Planned** bar for the sprint of interest. You can use the Planned bar for a velocity chart widget or the team backlog velocity chart.



2. The Query Results page opens with a list of work items defined for the sprint at the start of the sprint, the first day of the sprint. This list is an itemized list of work item IDs.
3. Choose the Editor page to edit the query.
4. List the items that were added to the sprint after the sprint's start. To do so, change the query to add and change the following clauses:
   - Add a clause at the top to specify the Work Item Types of interest
   - Change the Operator for the ID Field to Not In.
   - Add the Iteration Path for the sprint of interest.

- Add the Area Path for the team.

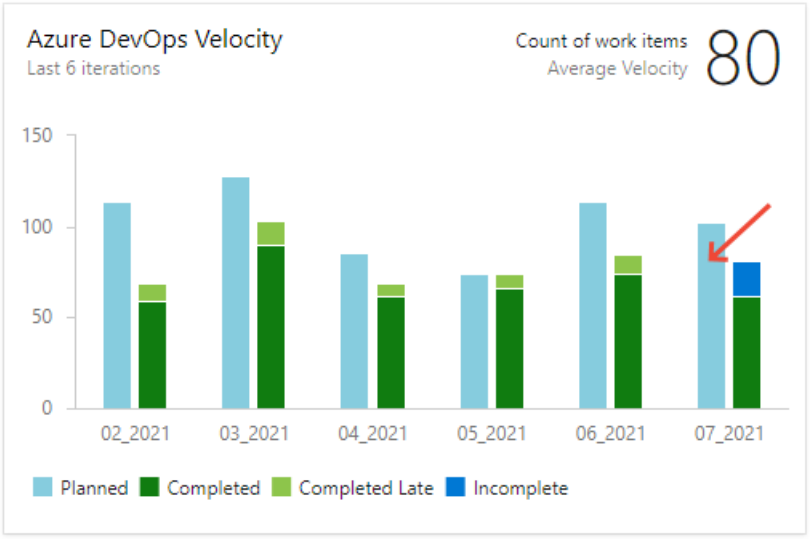
The updated query should look similar to the following image.



5. Add **Created Date** as a column option, and sort by that field. You can then view the existing work items that were added to the sprint and what newly created work items were added.

List work items moved out of the sprint

1. Open the velocity chart for the team and choose the **Planned** bar for the sprint of interest. You can use the Planned bar for a velocity chart widget or the team backlog velocity chart.



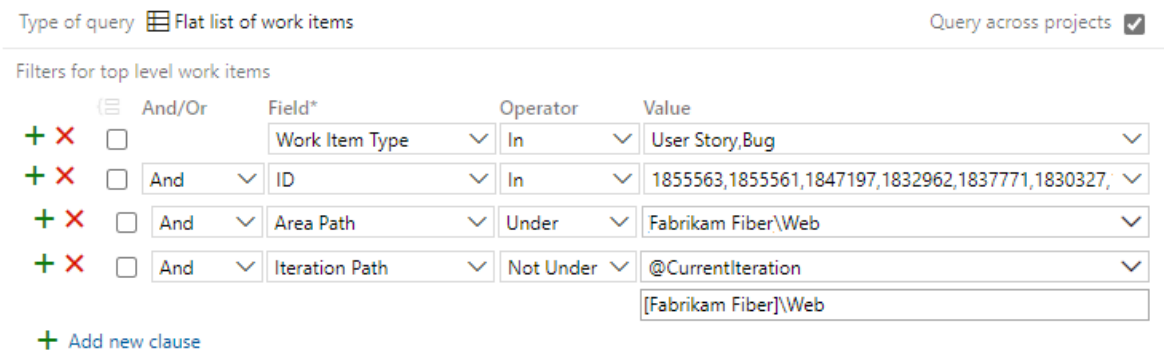
2. The Query Results page opens with a work item list defined for the sprint at the sprint's start, the first day of the sprint. This list is an itemized list of work item IDs.

3. Choose the Editor page to edit the query.

4. List the items that were moved out of the sprint after the sprint's start. To do so, change the query to add and change the following clauses:

   - Add a clause at the top to specify the Work Item Types of interest
   - Add the Iteration Path for the sprint of interest, specify **Not Under** operator
   - Add the Area Path for the team.

   The updated query should look similar to the following image.



For other options to determine changes to the sprint scope, see Query by date or current iteration, List work items moved out of a sprint.

Next step

Schedule sprints

Related articles

If you work with several teams, and each team wants their own backlog view, you can create more teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items to only include those assigned values under the team's default area path and iteration path.

- Scrum concepts
- Web portal navigation
- Backlogs, portfolios, and Agile project management
- About work items
- What is Scrum?
- What is Agile development?

Manage sprint timelines

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With Scrum, teams plan and track work at regular time intervals, referred to as a sprint cadence. You define Iteration Paths, also referred to as *sprints*, to support the assignment of work items to time-box intervals. Iteration paths are a shared resource used by all teams that select them. Many teams choose a two or three week cadence. You can, however, specify shorter or longer sprint cycles. Or, you can create a release schedule that encompasses several sprints.

> **TIP**
> If all you need is to change the iteration dates, you can do that by following the guidance provided in this article. However, if you need to define the iteration paths and tree structure, or assign team sprints, then follow the guidance provided in Define iteration paths (sprints) and configure team iterations.

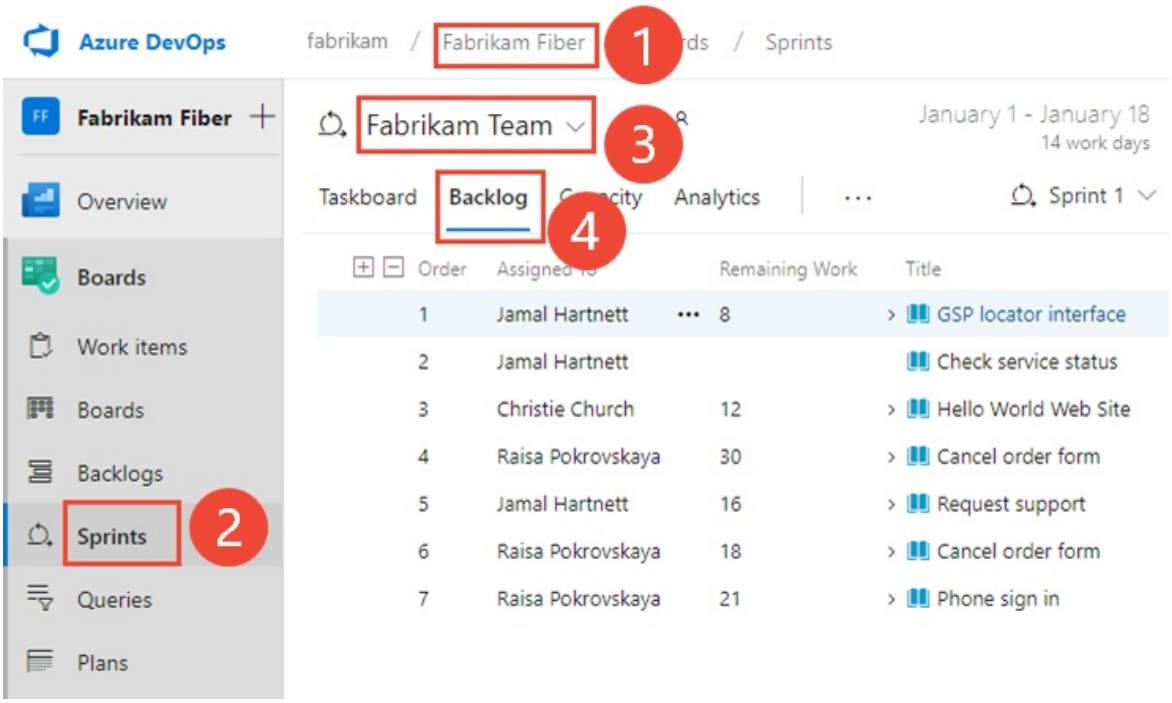
Prerequisites

- To change sprint dates, you must be a member of the **Project Administrators** security group, or have the **Edit this node** permission for the iteration child node you want to change. By default, the user who created the project has these permissions set. To learn more, see Change project-level permissions or Set permissions and access for work tracking.

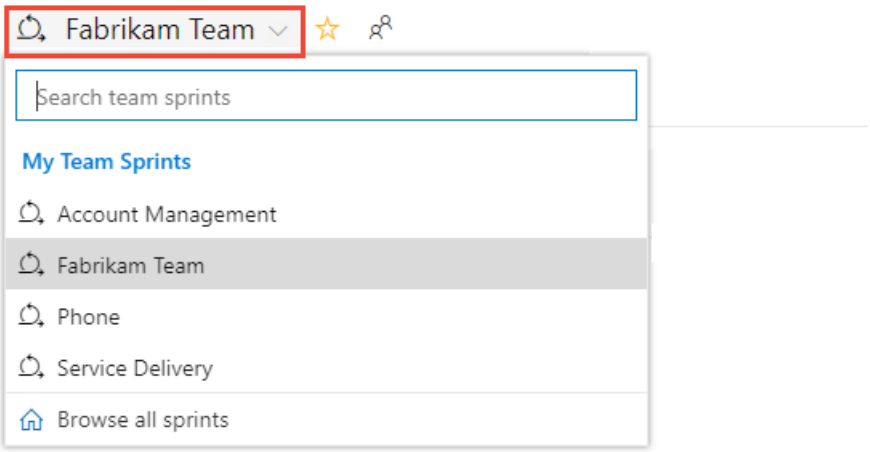
Start scheduling sprints

To quickly get started, you can use the default sprints. Default sprints are also referred to as iterations and were added when your project was created. Note, you must be a member of the **Project Administrators** group to add sprints and schedule sprint dates. If you created the project, you're a member.

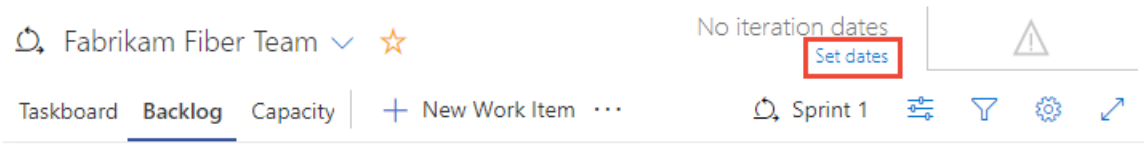
1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.



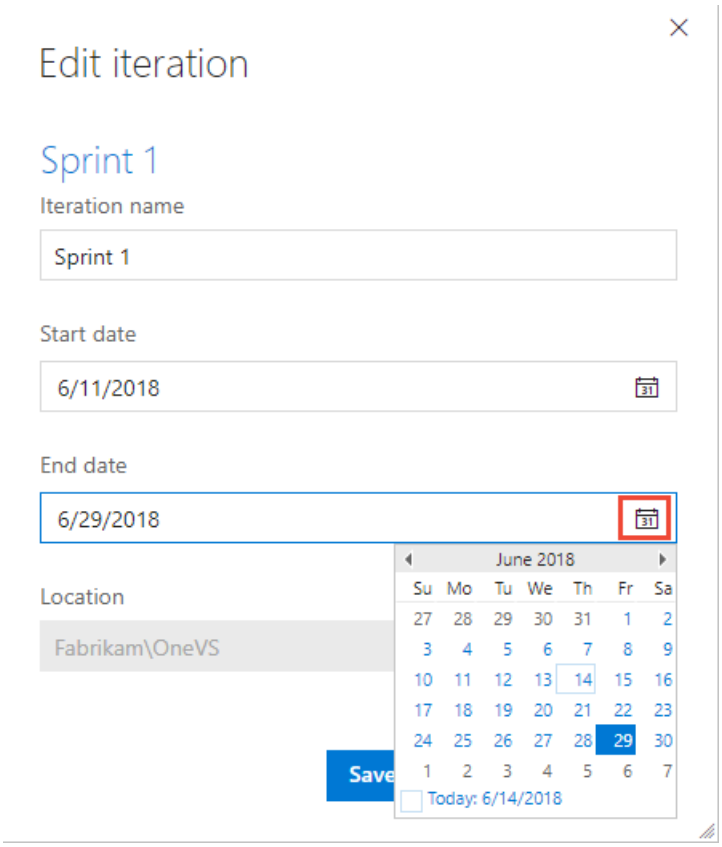
2. To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



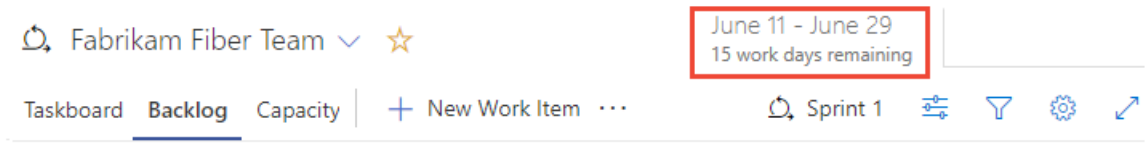
3. Choose **Set sprint dates**.



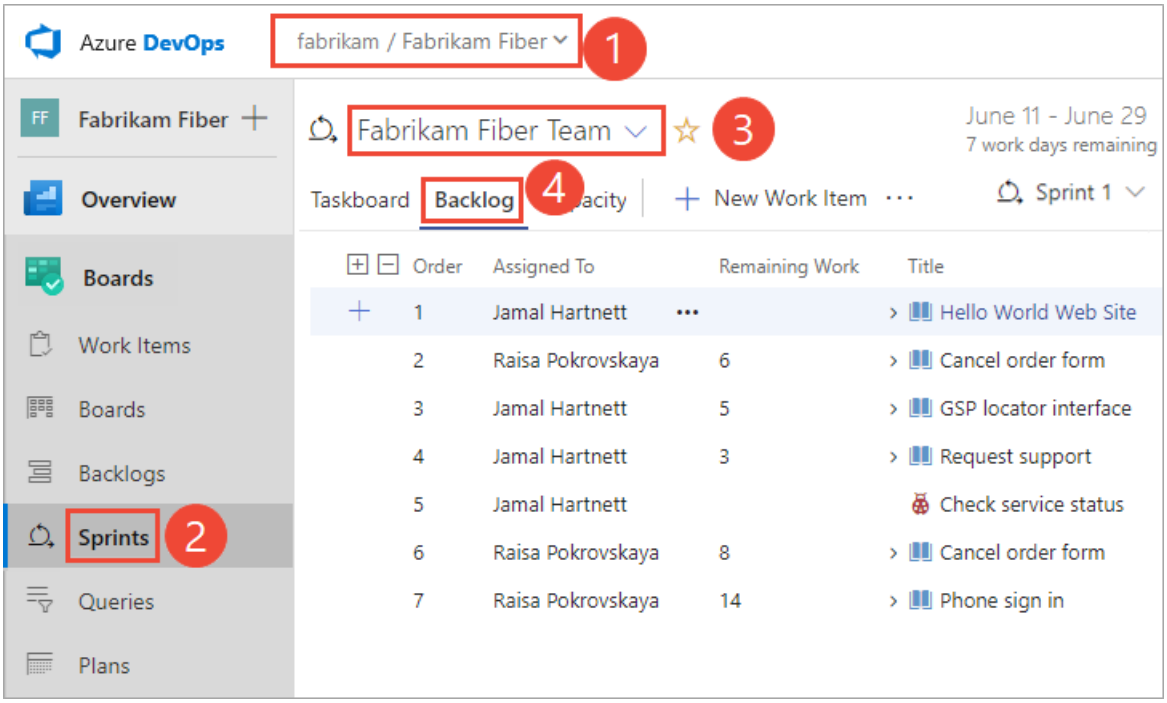
4. Choose the calendar icon to select the start date, and then the end date of the sprint.



5. Choose **Save and close**. You'll see the date listed.

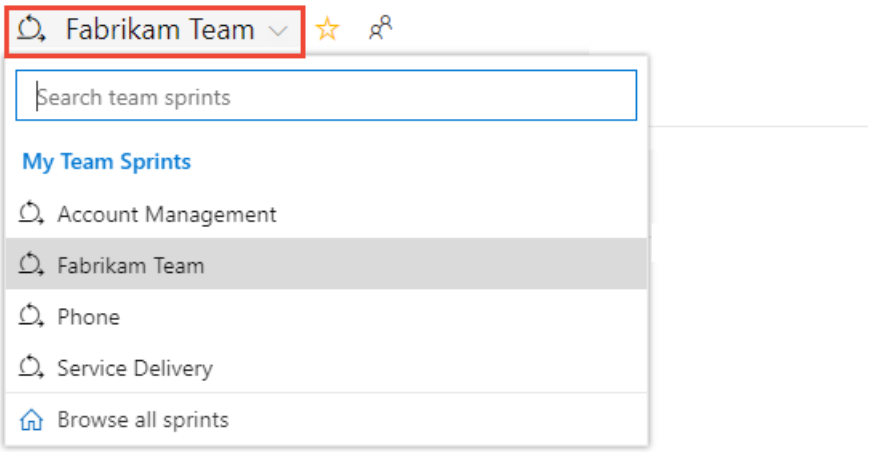


1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.

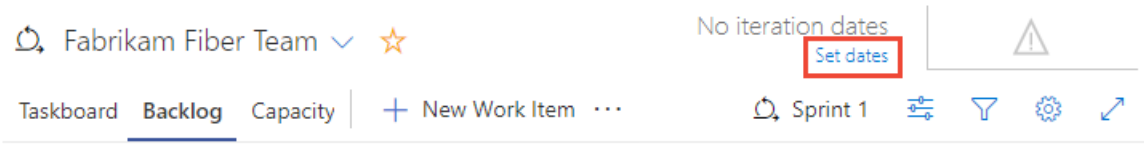


2. To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the

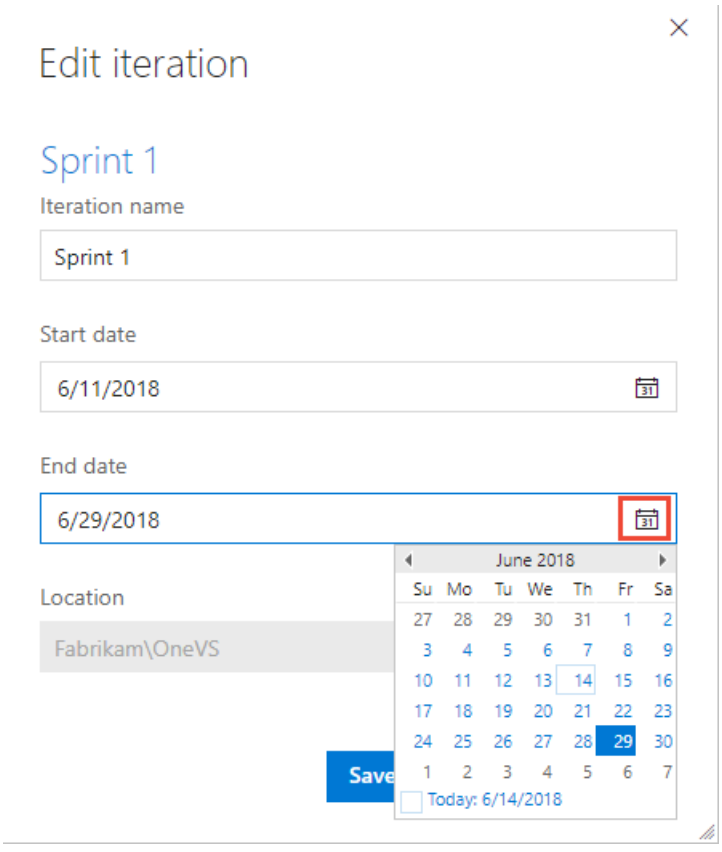
project.



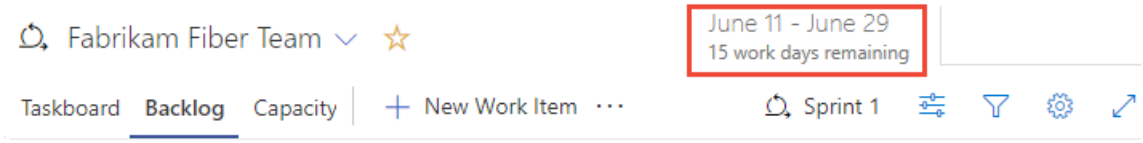
3. Choose **Set sprint dates**.



4. Choose the calendar icon to select the start date, and then the end date of the sprint.

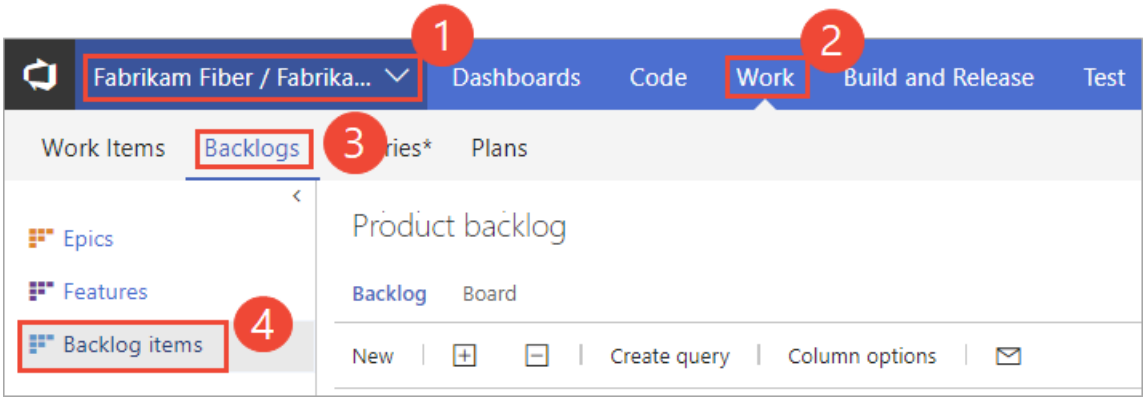


5. Choose **Save and close**. You'll see the date listed.

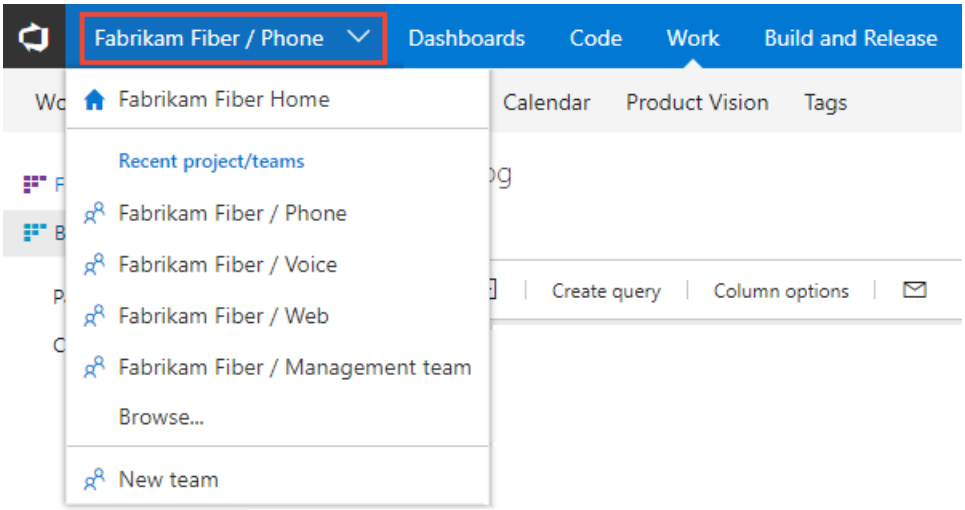


1. From your web browser, open your team's sprint backlog. (1) Select the team from the project/team

selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

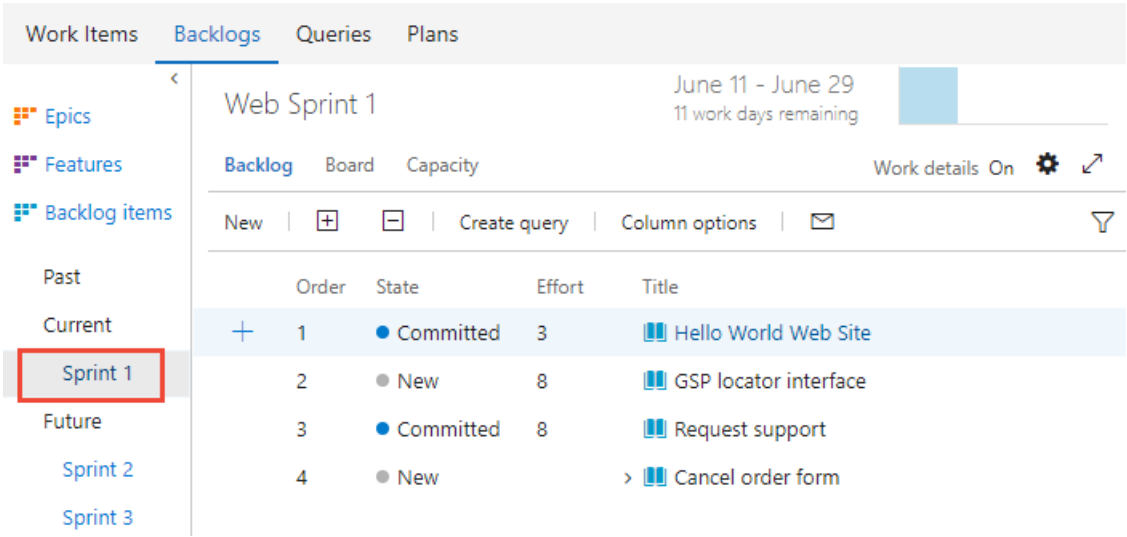


To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



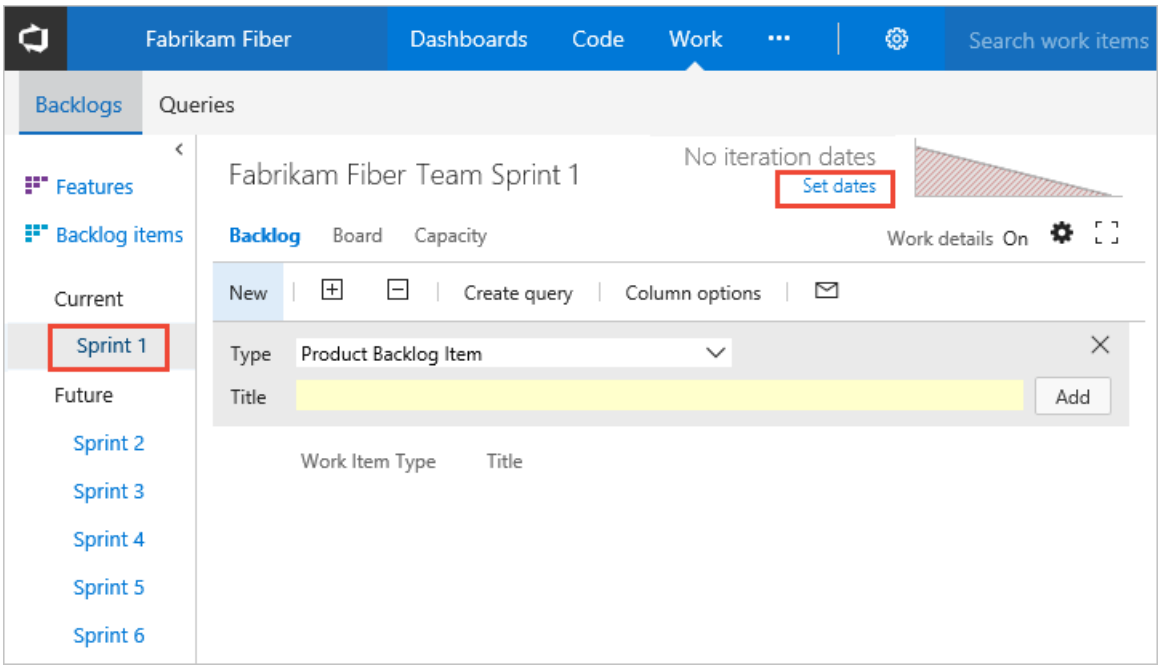
The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. Choose the sprint you want to plan.



The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then see Define iteration (sprint) paths.

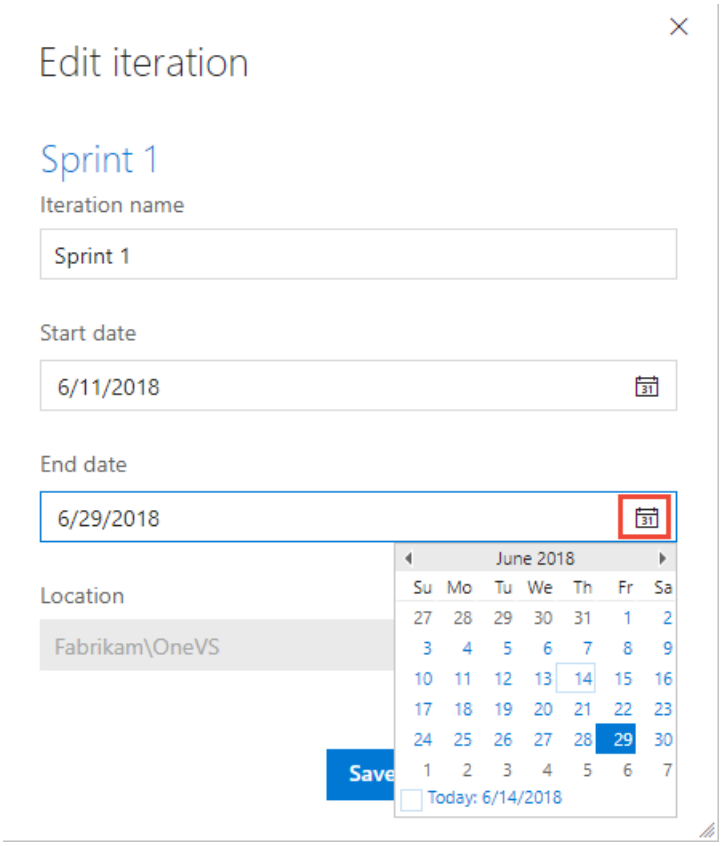
3. Choose the sprint listed under **Current** and then choose **Set dates**.



NOTE

If you don't see any sprints listed or the **Set dates** link, then no sprints have been selected for the team context you've selected. To select sprints for the team context, see Define iteration (sprint) paths and configure team iterations. To switch team context, see Switch project or team focus.

4. Choose the calendar icon to select the start date, and then the end date of the sprint.



That's it! You can now start planning your first sprint.

If you have several teams, more complex release and sprint cadences to schedule, or want to create child iterations, then read further. You define these items through the settings context for the project. See Define iteration (sprint) paths and configure team iterations.

NOTE

Terminology note: Your set of Agile tools uses the Iteration Path field to track sprints and releases. When you define sprints, you define the picklist of values available for the Iteration Path field. You use iterations to group work into sprints, milestones, or releases in which they'll be worked on or shipped.

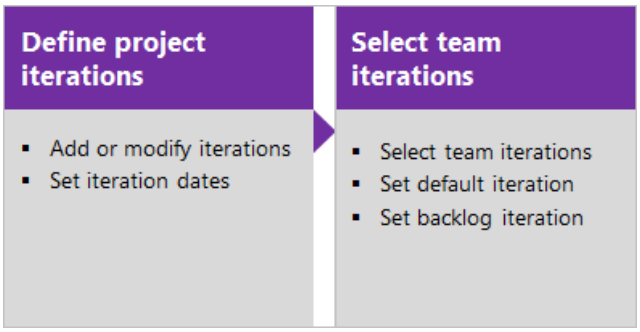
Schedule new sprints for different teams and release cadences

NOTE

Your sprint backlog and taskboard are designed to support your Scrum processes. In addition, you have access to product and portfolio backlogs and Kanban boards. For an overview of the features supported on each backlog and board, see Backlogs, boards, and plans.

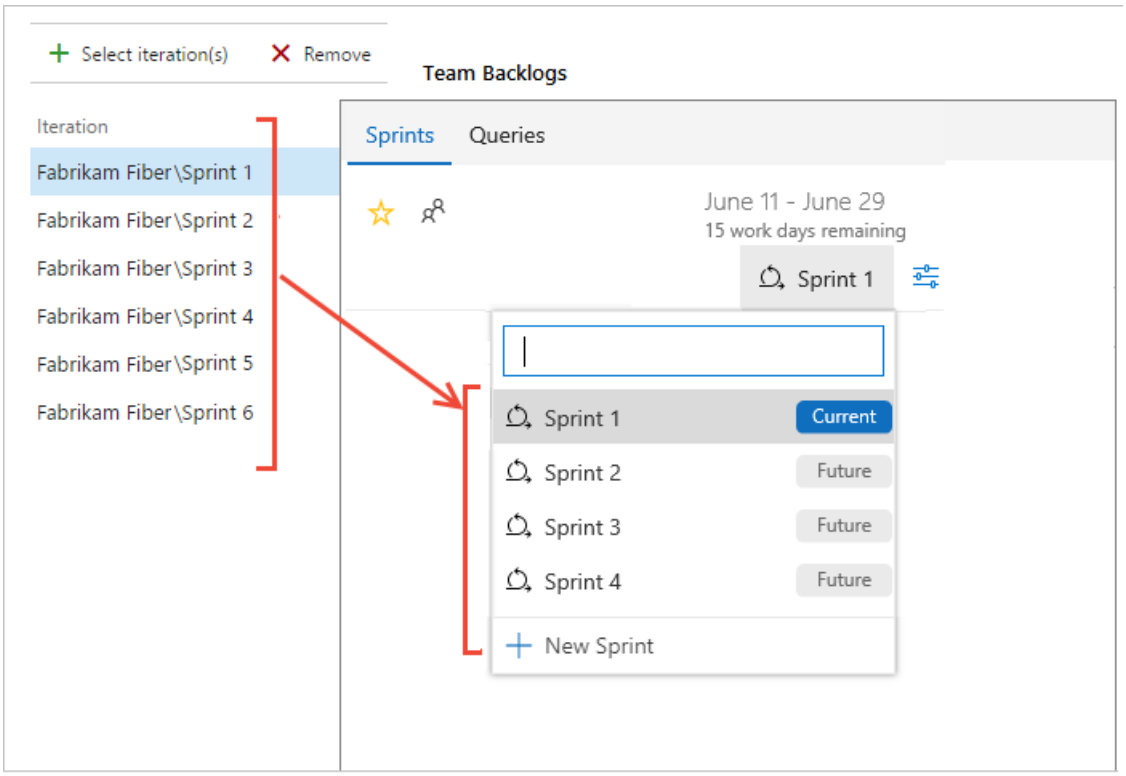
Your project comes with several sprints predefined. However, they aren't associated with any dates. For Scrum and sprint planning, you'll want to assign start and end dates for the sprints your team will use.

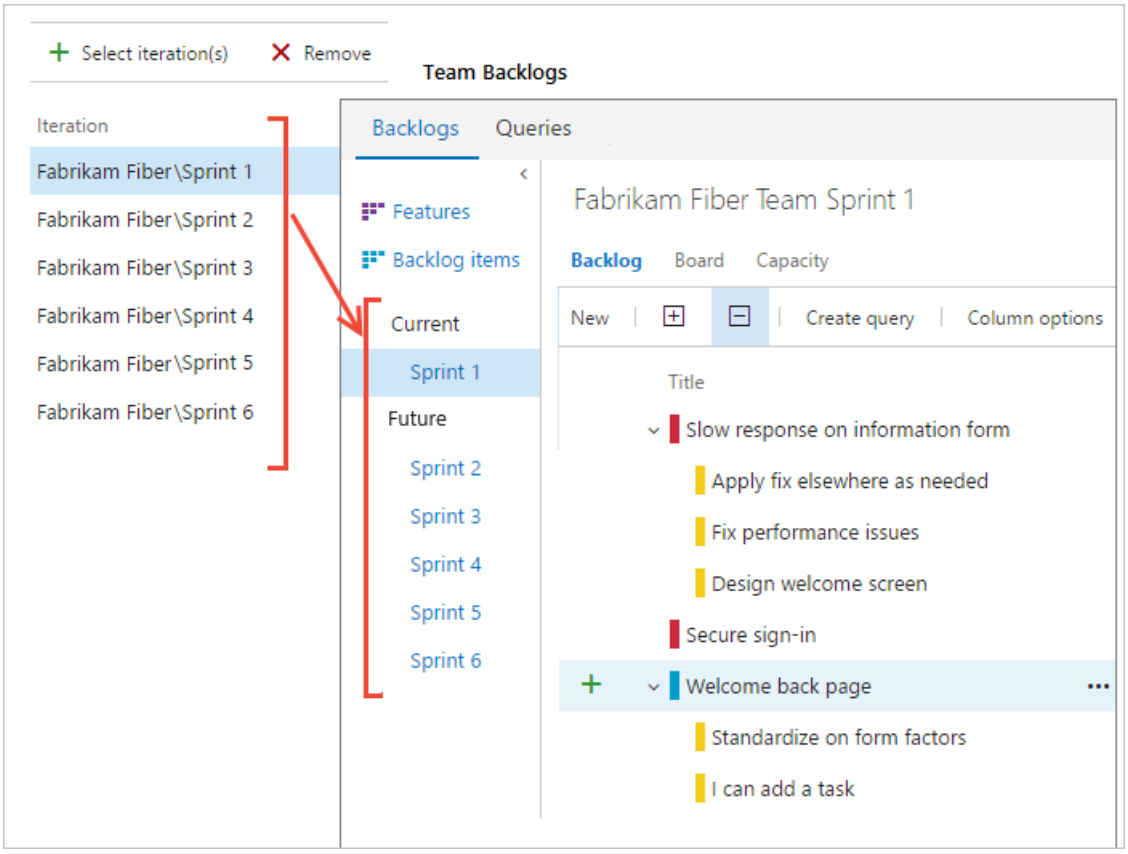
Defining other sprints is a two-step process. You first define the sprints for your project and then you select the sprints that each team will use—Define iteration (sprint) paths and configure team iterations. In this way, the system supports teams that work on different sprint cadences.



Each sprint that you select for your team provides access to a sprint backlog, taskboard, and other sprint planning tools for planning and tracking work.

For example, by selecting Sprints 1 through 6, the Fabrikam Fiber team gets access to six sprint backlogs. They also get access to capacity planning tools and a taskboard for each sprint.





Next step

Assign work to a sprint or Define iteration (sprint) paths and configure team iterations

Related articles

If you work with several teams, and each team wants their own backlog view, you can create more teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items. These items only include those assigned values under the team's default area path and iteration path.

Forecast your product backlog

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Teams use the forecast tool to help in their sprint planning efforts. By plugging in a value for the team velocity, the forecast tool will show which items in the backlog can be completed within future sprints. Both tools are team-specific tools that rely on the team's ability to estimate backlog items. Once your team has completed a sprint or two, they can use the team velocity to forecast how much of the backlog they can finish within the upcoming sprints.

Use this article to learn:

- How to forecast upcoming sprints
- Required and recommended team activities to support forecasting

> **NOTE**
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans. If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

Prerequisites

- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** security group. If you're not on a project or team, get added now.
- You must be granted **Basic** access or higher to use the forecast feature. For details, see About access levels.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** security group. If you're not on a project or team, get added now.
- You must be granted **Basic** access or higher to add or modify work items. For details, see About access levels.

Required and recommended activities

Here's what you need to have in place before you attempt to forecast your team's backlog.

Required:

- Define iteration paths (sprints) and configure team iterations
  - Sprints should be of the same duration.
  - Select enough future sprints to forecast your entire product backlog.
- Define and estimate backlog items. If you work from your team's backlog, the items you create will automatically be assigned to the current sprint (Iteration) and to your team's default Area Path.

- Update the status of backlog items once work starts and when completed. Only backlog items whose State maps to a state category of *Proposed* or *In Progress* show up on the velocity chart. (for details, see Workflow states and state categories).

Recommended:

- Define and size backlog items to minimize variability.
- Determine how your team wants to treat bugs. If your team chooses to treat bugs like requirements, bugs will show up on the backlog and be counted within the Velocity chart and forecasting.
- Set your team's area path. The forecast tool will forecast those items based on your team's default settings. These settings can specify to include items in area paths under the team's default or exclude them.
- Don't create a hierarchy of backlog items and bugs. The display of the leaf node, the last node in a same-category hierarchy, may only appear on Kanban boards, sprint backlogs, and taskboards. To learn more, see Fix reordering and nesting issues, How backlogs and boards display hierarchical (nested) items.
  Instead of nesting requirements, bugs, and tasks, maintain a flat list—only creating parent-child links one level deep between different-category items. Use Features to group requirements or user stories. You can quickly map stories to features. The map creates parent-child links in the background.
- At the end of the sprint, update the status of those backlog items that the team has fully completed. Incomplete items should be moved back to the product backlog and considered in a future sprint planning meeting.

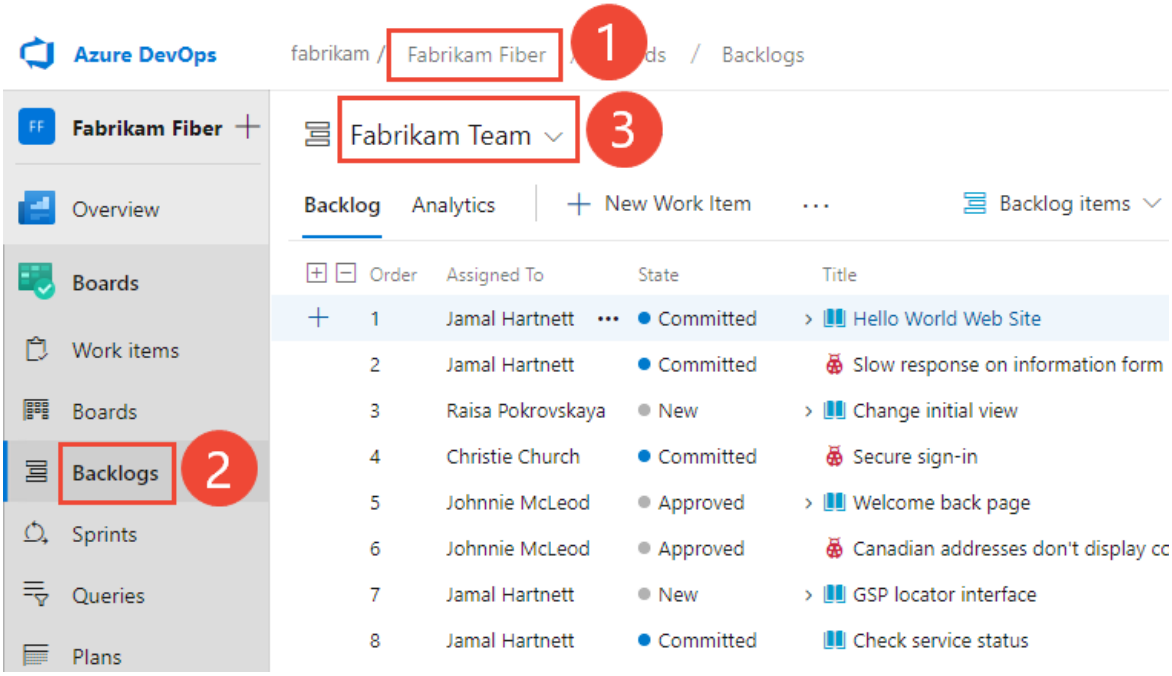
> **NOTE**
> If you work with several teams, and each team wants to work with their own backlog, velocity chart, and forecast tool, you can create additional teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items to only include those whose assigned area paths and iteration paths meet those set for the team.

Forecast upcoming sprints

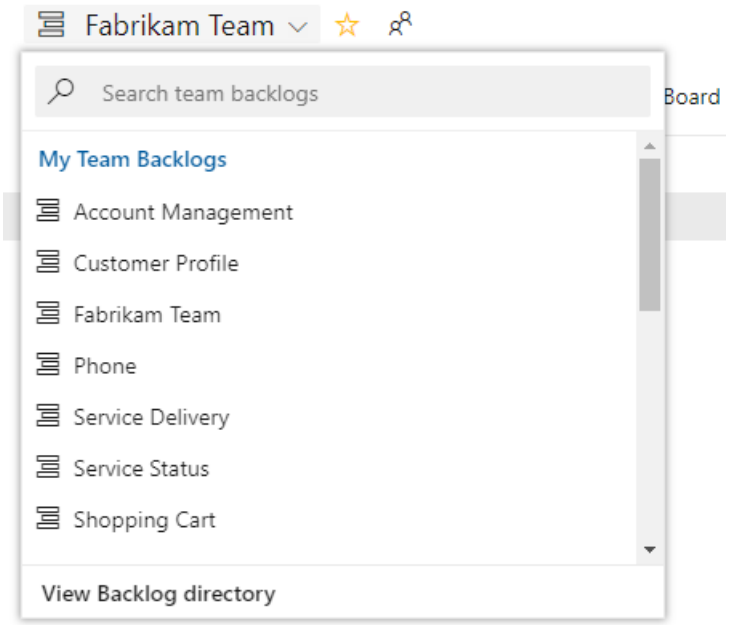
Use the forecast tool to get an idea of how many items you can complete within a sprint. By plugging in a velocity, you can see which items are within scope for the set of sprints the team has activated.

To forecast your product backlog, complete the following actions.

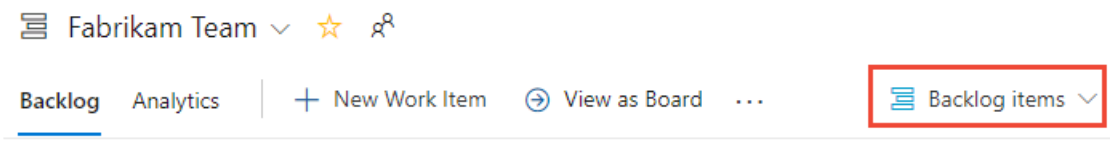
1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



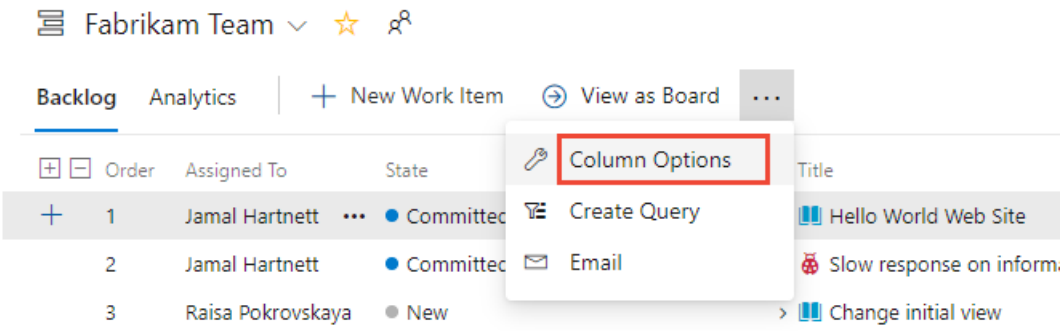
To select another backlog, open the selector and then choose a different team or select the **View Backlog directory** option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.



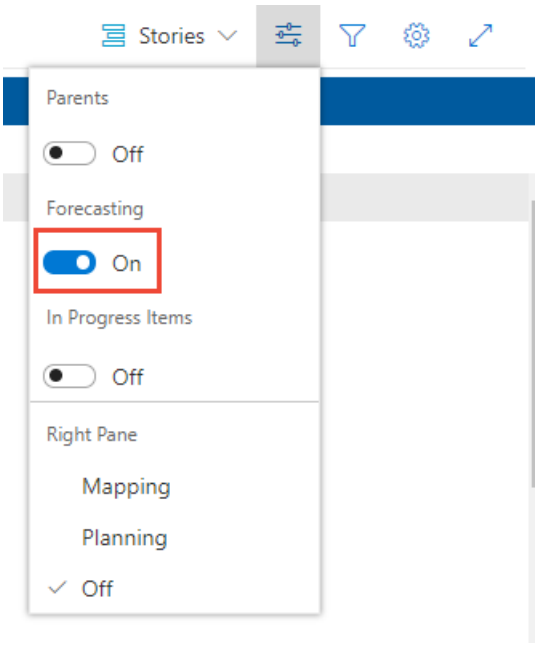
2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.

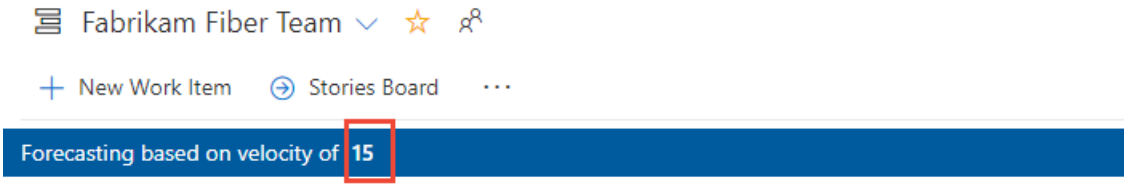


4. Choose the view options icon and slide **Forecasting** to **On**. To keep things simple, turn the **Mapping** and **Planning** panes **Off**.



Set **In Progress Items** to **Off** to hide those items that won't be counted in the forecast. The forecast tool ignores Scrum items set to *Committed* or *Done* and Agile and CMMI items set to *Active*, *Resolved*, or *Completed*.

5. Enter your team's predicted velocity.

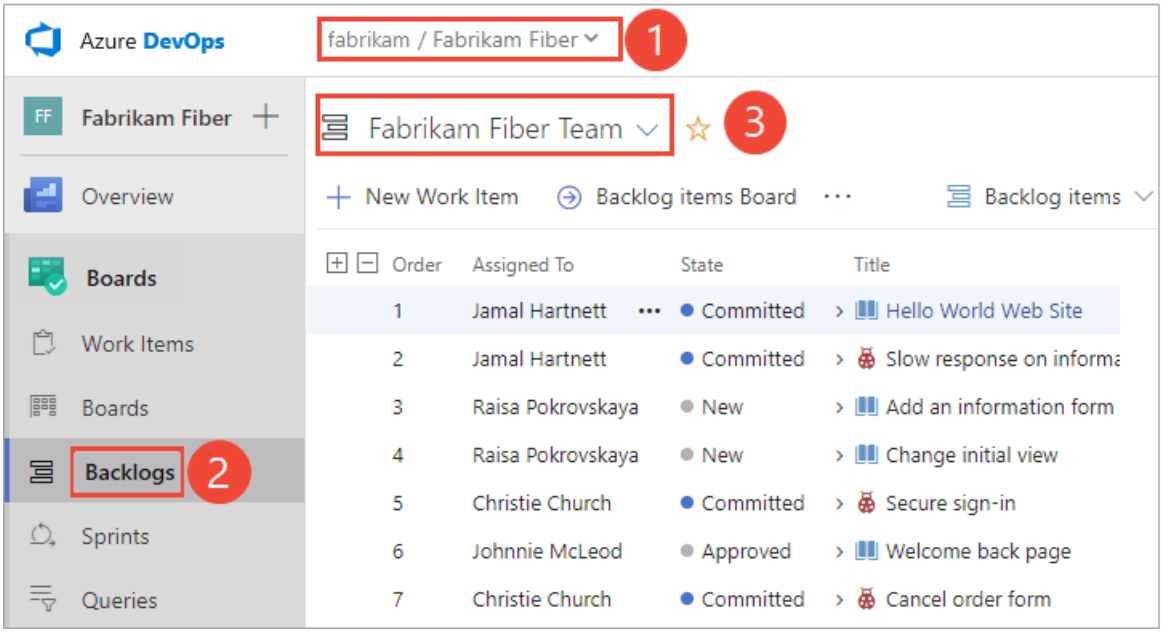


TIP

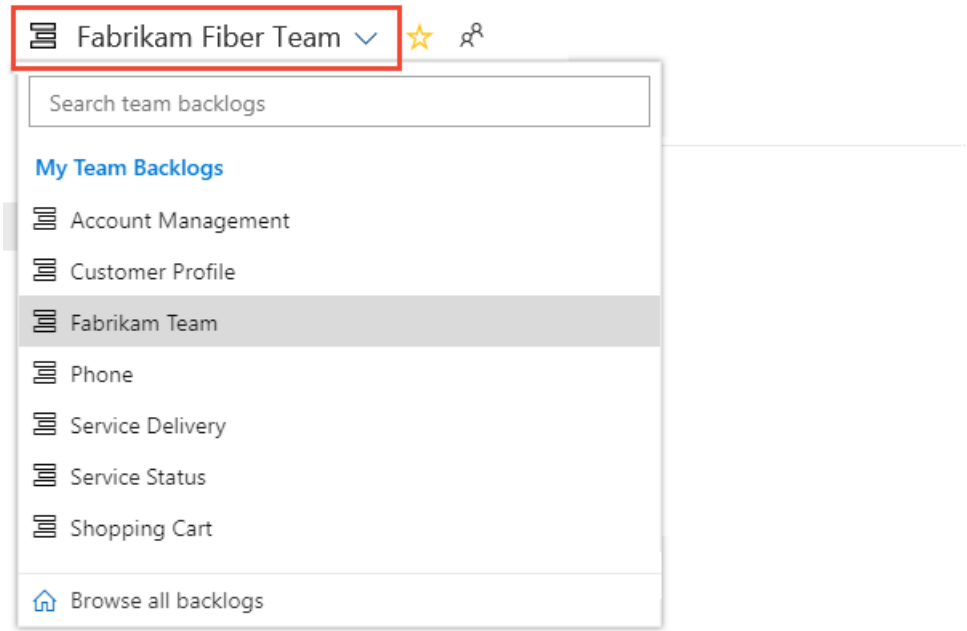
If your team has been working for several sprints, you can gain an idea of your team's velocity from the Velocity widget.

The tool draws lines for each future sprint selected by the team. The Forecast lines show how much work your team can complete in future sprints. Typically, items above the first line are already in progress for the current sprint. Items that fall between the first and second forecast lines indicate what can be completed in the named sprint.

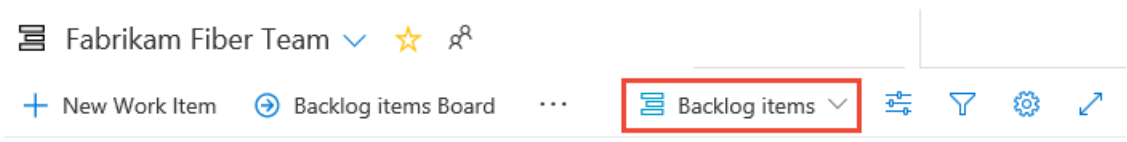
1. From your web browser, open your product backlog. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



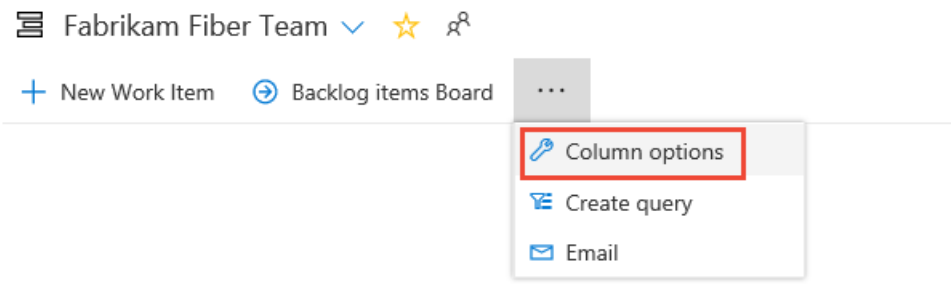
To choose another team, open the selector and select a different team or choose the **Browse all team backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



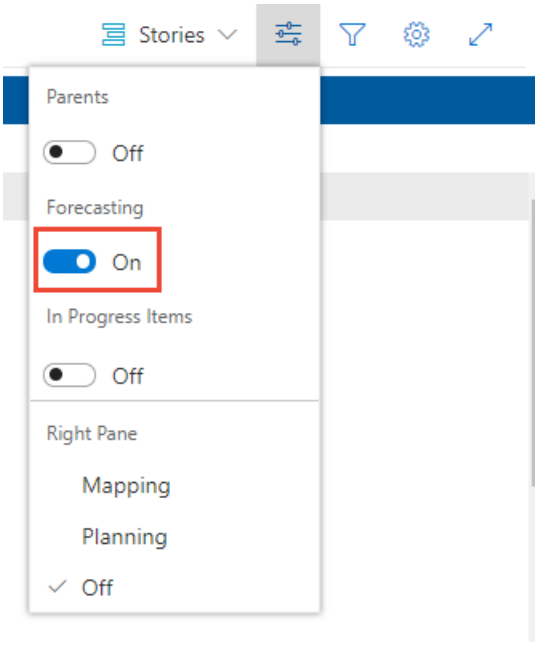
2. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level. You can only forecast a product backlog. You can't forecast a portfolio backlog such as Features or Epics.



3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.

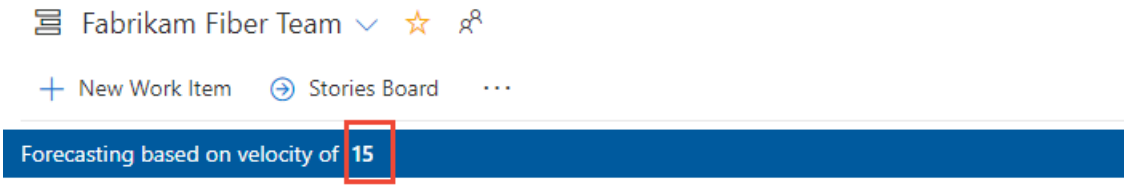


4. Choose the view options icon and slide **Forecasting** to **On**. To keep things simple, turn the **Mapping** and **Planning** panes **Off**.



Set **In Progress Items** to **Off** to hide those items that won't be counted in the forecast. The forecast tool ignores Scrum items set to *Committed* or *Done* and Agile and CMMI items set to *Active*, *Resolved*, or *Completed*.

5. Enter your team's predicted velocity. If the **Forecasting** bar doesn't appear.



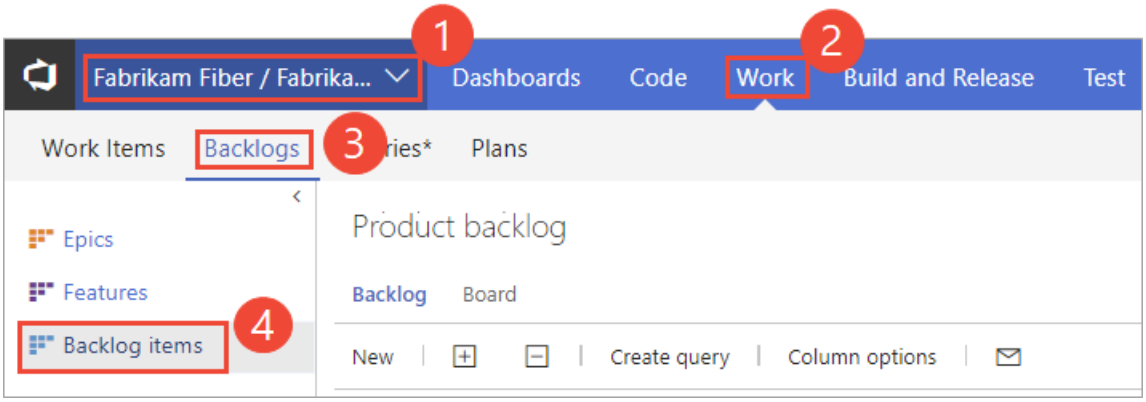
TIP

If your team has been working for several sprints, you can gain an idea of your team's velocity from the Velocity widget.

The tool draws lines for each future sprint selected by the team. The Forecast lines show how much work your team can complete in future sprints. Typically, items above the first line are already in progress for the current sprint. Items that fall between the first and second forecast lines indicate what can be completed in the named sprint.

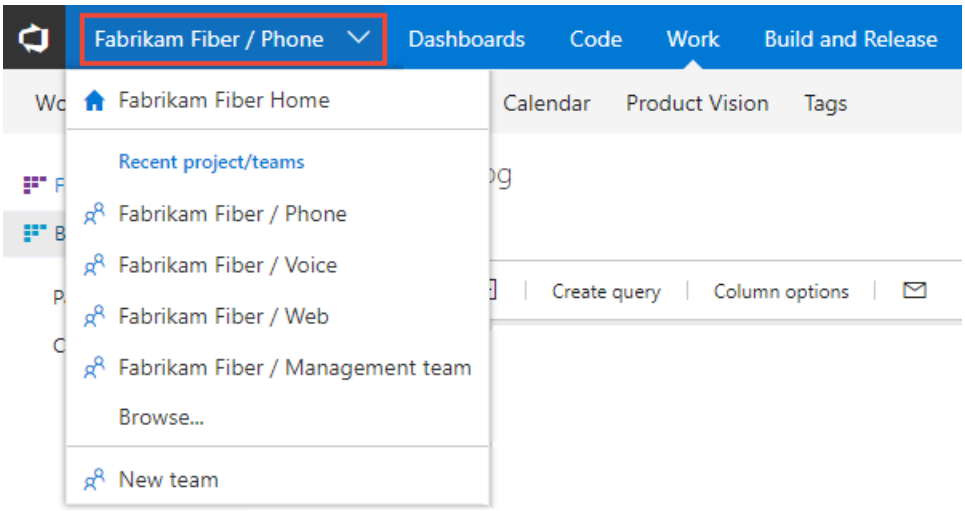
To forecast your product backlog, complete the following actions.

1. From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).



   You can only forecast the product backlog of Stories, Backlog items, or Requirements.

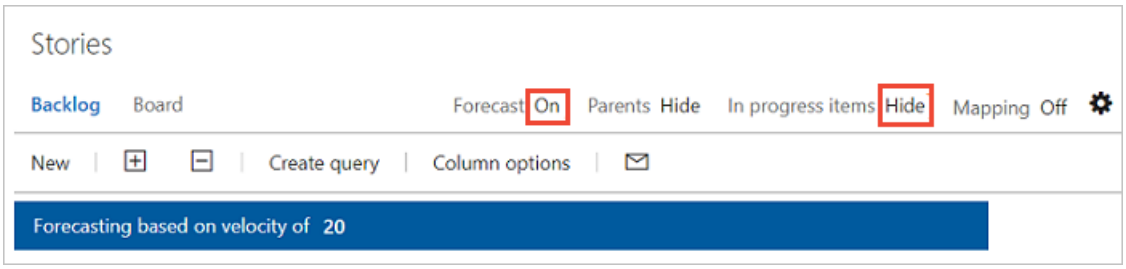
   To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



> **NOTE**
> If you work with several teams, and each team wants to work with their own backlog, velocity chart, and forecast tool, you can create additional teams. Each team then gets access to their own set of Agile tools. Each Agile tool filters work items to only include those whose assigned area paths and iteration paths meet those set for the team.

2. (Optional) To choose which columns should display and in what order, choose **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.

3. Set **Forecast** to **On** and enter your team's predicted velocity. If the Forecasting bar doesn't appear, set **Parents** to **Hide**.



4. Set **In progress** items to **Hide** to hide those items that won't be counted in the forecast. The forecast tool ignores Scrum items set to Committed or Done and Agile and CMMI items set to Active, Resolved, or Completed.

   The tool draws lines for each future sprint selected by the team. The Forecast lines show how much work your team can complete in future sprints. Typically, items above the first line are already in progress for the current sprint. Items that fall between the first and second forecast lines indicate what can be completed in the named sprint.

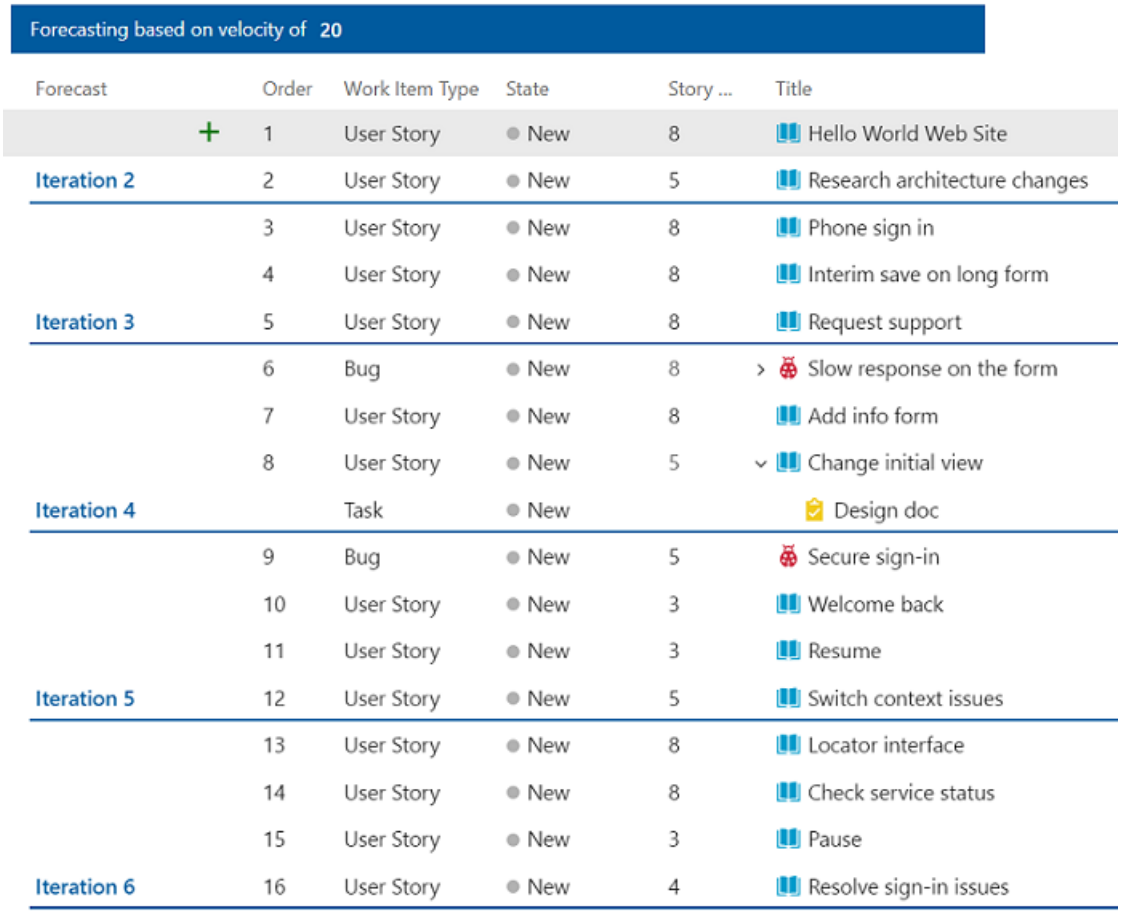
Review the forecast results

- Check the results manually to understand discrepancies in what you expect and what the forecast tool displays.
- Check the amount of effort (Effort, Story Points, or Size) forecasted per sprint.
- Question forecast results where the effort of an item is near to, or greater than, team velocity.

In this example, a Velocity of 20 is used. The forecast tool limits the number of items that are shown between the forecast lines to those items that can be completed within the sprint or using unused velocity points from the previous sprint.

The forecast tool shows between two and four items can be worked on during Iterations 2 through 6 based on the number of Story Points you assigned to each user story or bug. The forecast logic carries over velocity points from one sprint to the next.

- **Iteration 2**: 13 Story Points, items 1 and 2 can be completed; 7 velocity points are carried over to the next sprint
- **Iteration 3**: 24 Story Points, items 3 through 5 can be completed; 3 (=20+7-24) velocity points are carried over to the next sprint
- **Iteration 4**: 21 Story points, items 6 through 8 can be completed; 2 (=20+3-21) velocity points are carried over to the next sprint
- **Iteration 5**: 16 Story points, items 9 through 12 can be completed; 6 (=20+2-16) velocity points are carried over to the next sprint
- **Iteration 6**: 23 Story points, items 13 through 16 can be completed; 3 (=20+6-23) velocity points are carried over to the next sprint



Determine the velocity needed to complete all items in the backlog

Another way to use the forecast tool is to enter different velocity values until all the backlog items are completed within a given set of sprints. This forecast provides an estimate of what velocity is required to complete your backlog of items.

You can then assess the delta between the current team's velocity and the required velocity. The delta helps determine what other resources are required to meet production demands within a required time.

Next steps

Assign work to a sprint

Related articles

- Team velocity
- Define iteration paths (sprints) and configure team iterations
- Use the taskboard to track work during your sprint
- Monitor the sprint burndown chart to determine if your team is on track to complete the sprint plan

1. Assign backlog items to a sprint in Azure Boards

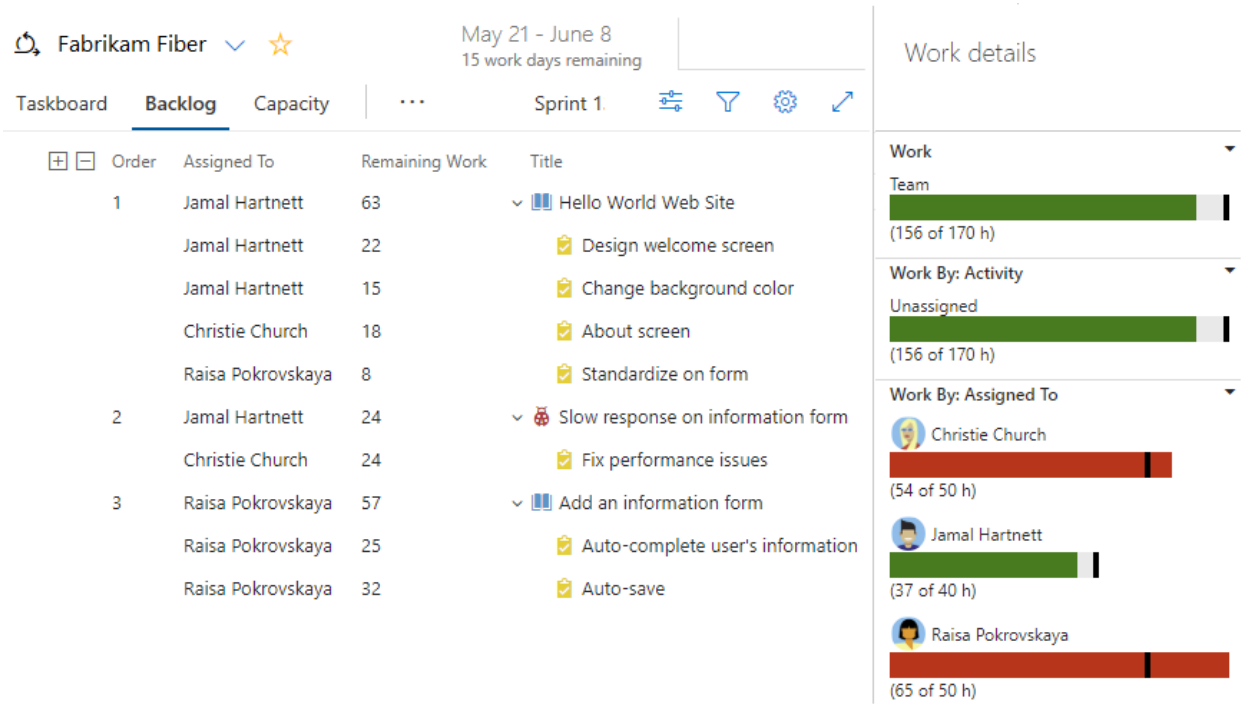
10/4/2022 • 9 minutes to read • Edit Online

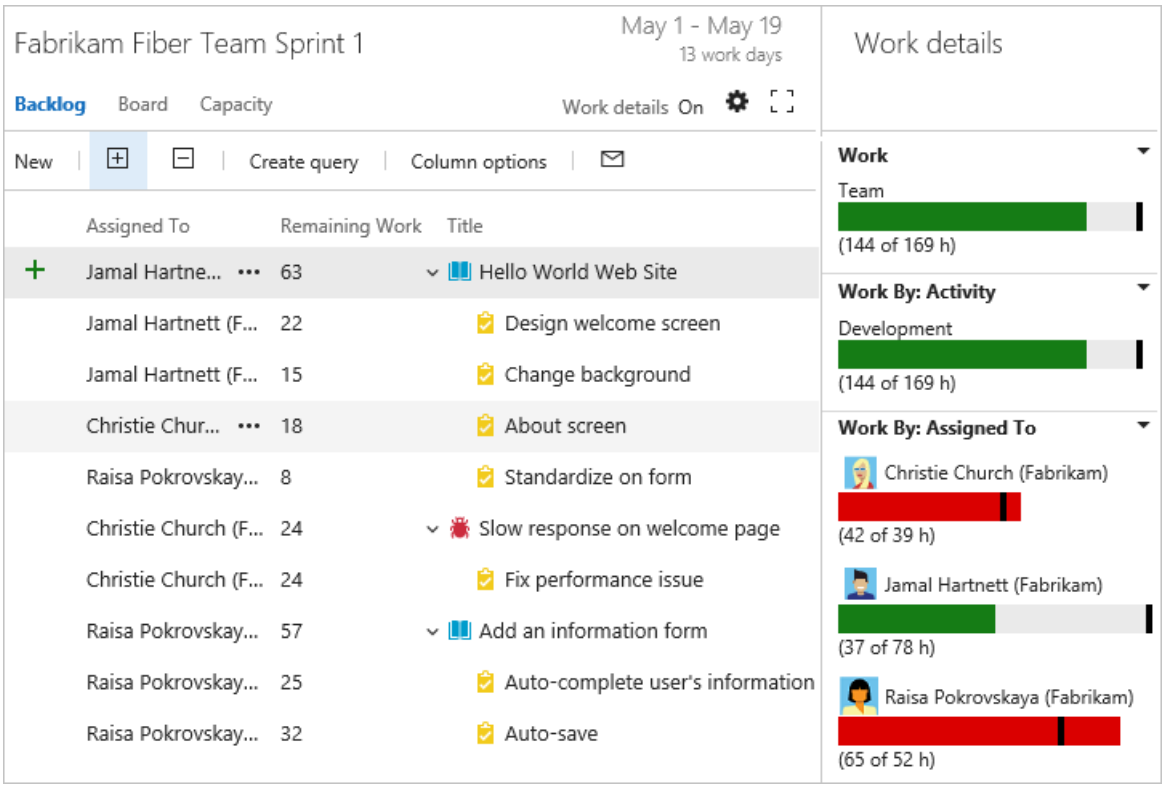
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The first step in planning your sprint is to assign work from your backlog to a sprint. Your team builds the sprint backlog during the sprint planning meeting, typically held on the first day of the sprint. Each sprint corresponds to a time-boxed interval that supports your team's ability to work using Agile processes and tools. During the planning meeting, your product owner works with your team to identify those stories or backlog items to complete in the sprint.

> **NOTE**
> Your project comes with several predefined sprints. You can quickly add more sprints from your backlog as needed. Or, change the dates of the predefined sprints. To learn more about sprints, also referred to as iterations, see About areas and iterations.

Here's an example of a sprint plan that consists of backlog items and the tasks required to complete each item. By setting team capacity and estimating tasks, the team can see when the team or a team member is at, under, or over capacity.





In this article you'll learn how to:

- Open your product backlog
- Assign backlog items to a sprint
- Use multi-select to bulk update work items

Planning meetings typically consist of two parts. In the first part, the team and product owner identify the backlog items that the team feels it can commit to completing in the sprint, based on experience with previous sprints. These items get added to the sprint backlog. In the second part, your team determines how it will develop and test each item. They then define and estimate the tasks required to complete each item. Finally, your team commits to implementing some or all of the items based on these estimates.

> **NOTE**
> Sprint planning doesn't need to be challenging. It can be fun and a time for the entire Scrum team to build camaraderie by working together to answer the question of "What can we commit to?" For examples and strategies to keep your sprint planning focused and effective, check out the What is Scrum?.
>
> When you've completed your sprint plan, your sprint backlog should contain all the information your team needs to successfully complete work within the time allotted without having to rush at the end.

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn

more, see Set permissions and access for work tracking.

- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

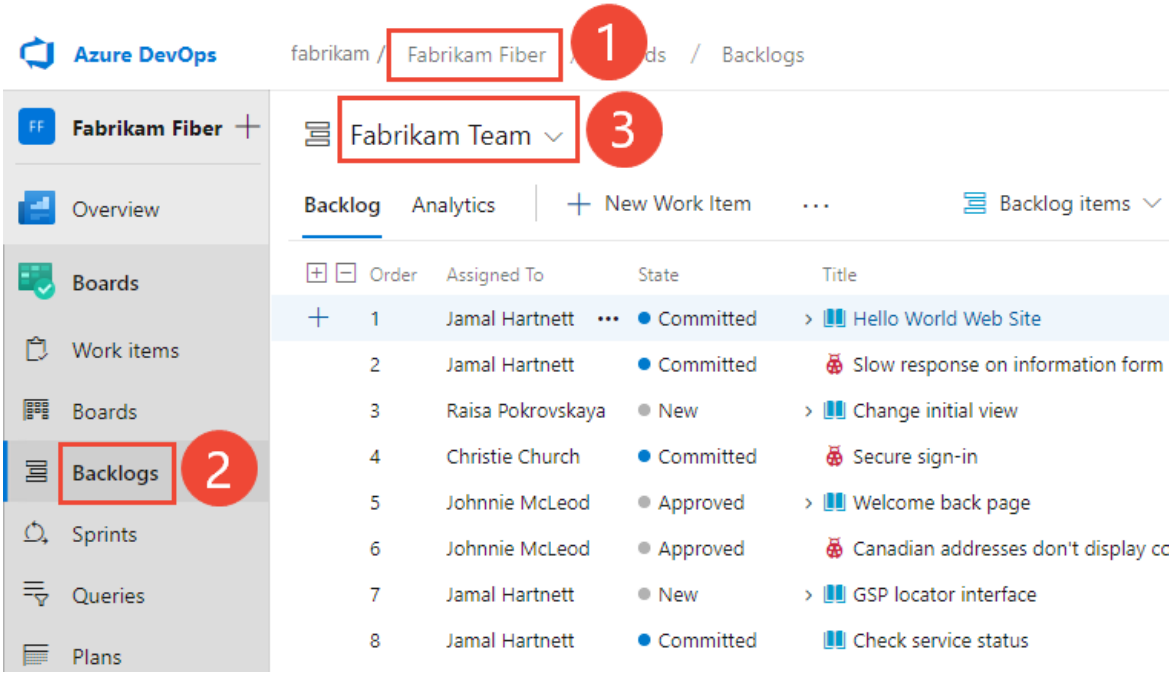
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

Open your team's product backlog

> **NOTE**
> Your sprint backlogs are one of three classes of backlogs available to you. For an overview of the features supported on each backlog and the two types of boards, see Backlogs, boards, and plans.
>
> For a beginner's guide to planning and tracking work, see Get started with Agile tools.

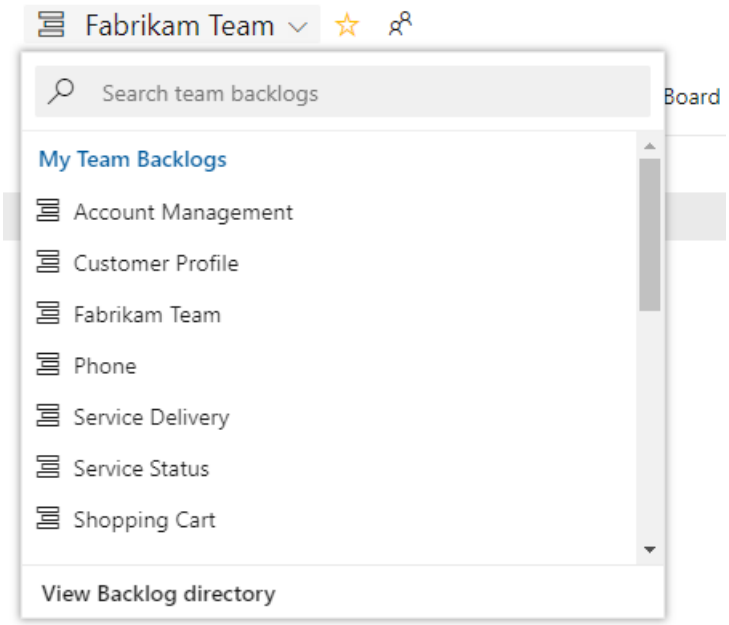
From your web browser, open your product backlog.

1. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



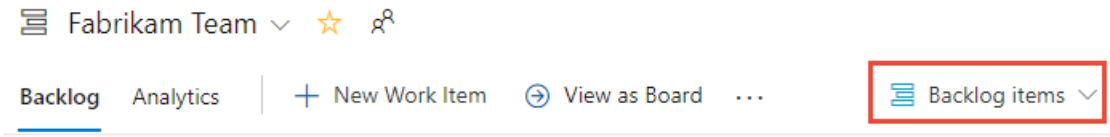
To select another backlog, open the selector and then choose a different team or select the **View Backlog**

directory option. Or, enter a keyword in the search box to filter the list of team backlogs for the project.

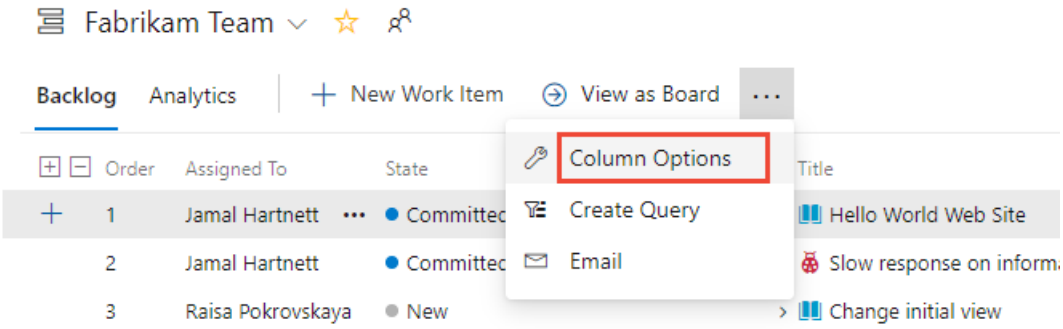


> **TIP**
> Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

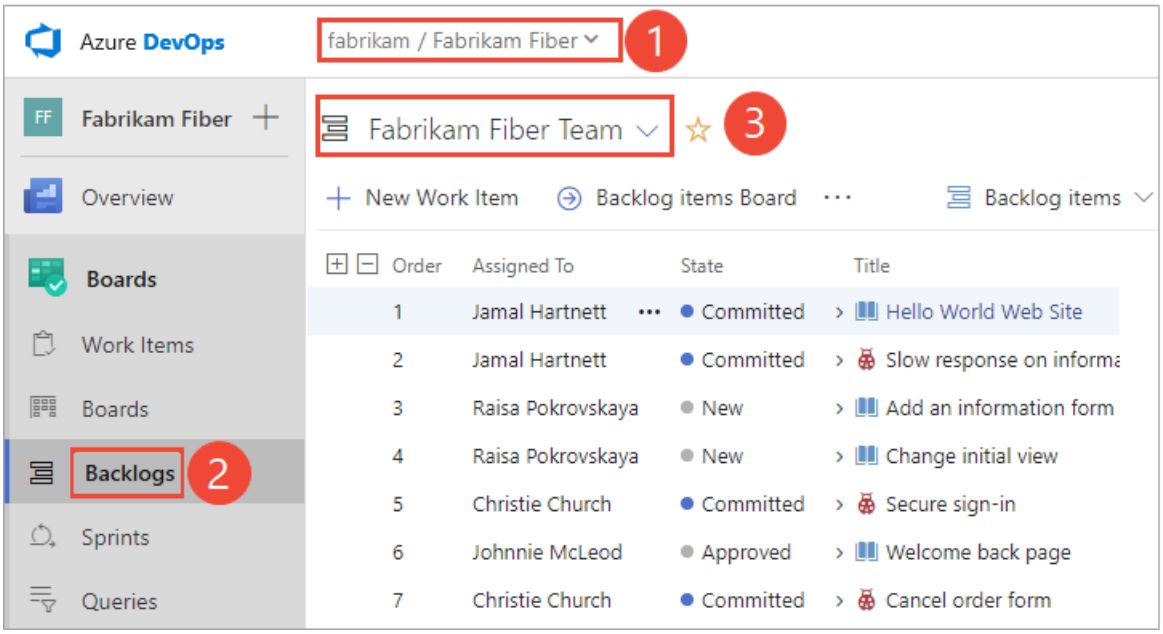
2. Check that you have selected **Stories** (for Agile), **Issues** (for Basic), **Backlog items** (for Scrum), or **Requirements** (for CMMI) as the backlog level.



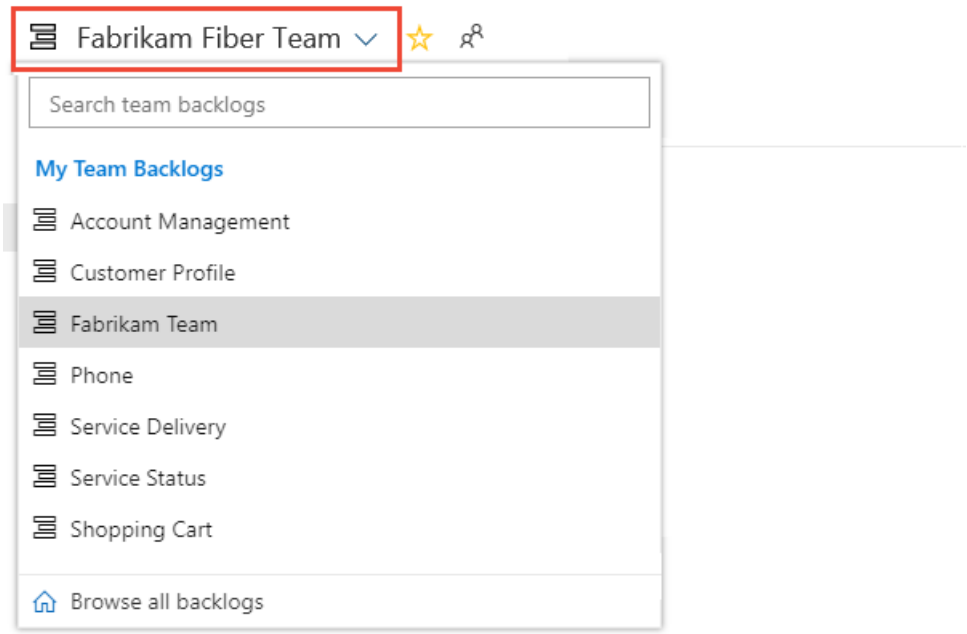
3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.



1. From your web browser, open your product backlog. (1) Check that you've selected the right project, (2) choose **Boards > Backlogs**, and then (3) select the correct team from the team selector menu.



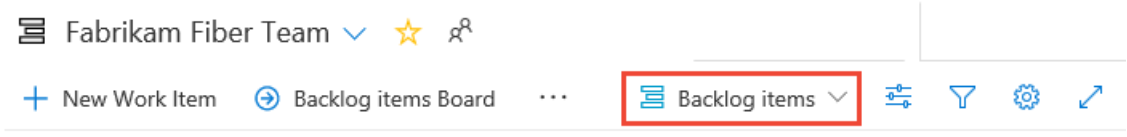
To choose another team, open the selector and select a different team or choose the **Browse all team backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



TIP

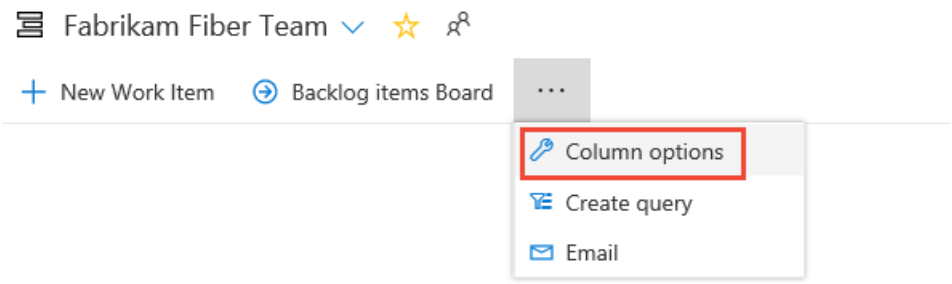
Choose the ☆ star icon to favorite a team backlog. Favorited artifacts (★ favorited icon) appear at the top of the team selector list.

2. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level.

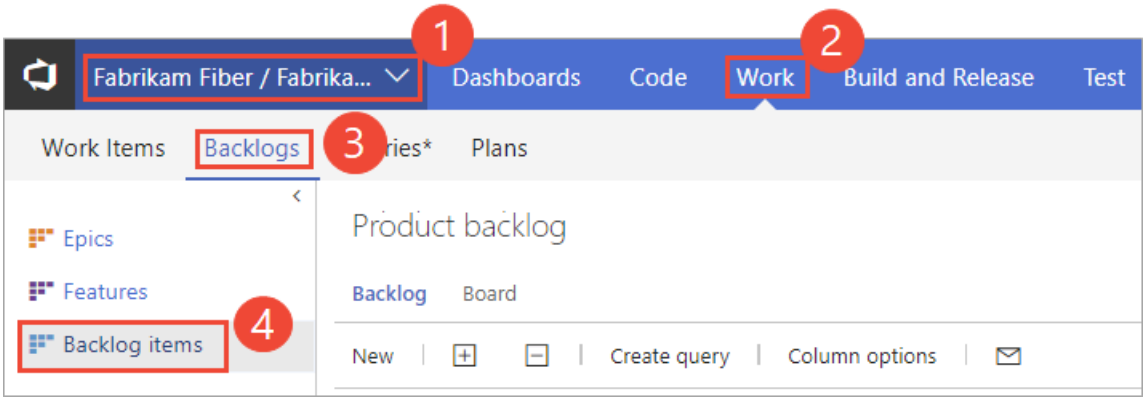


3. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and

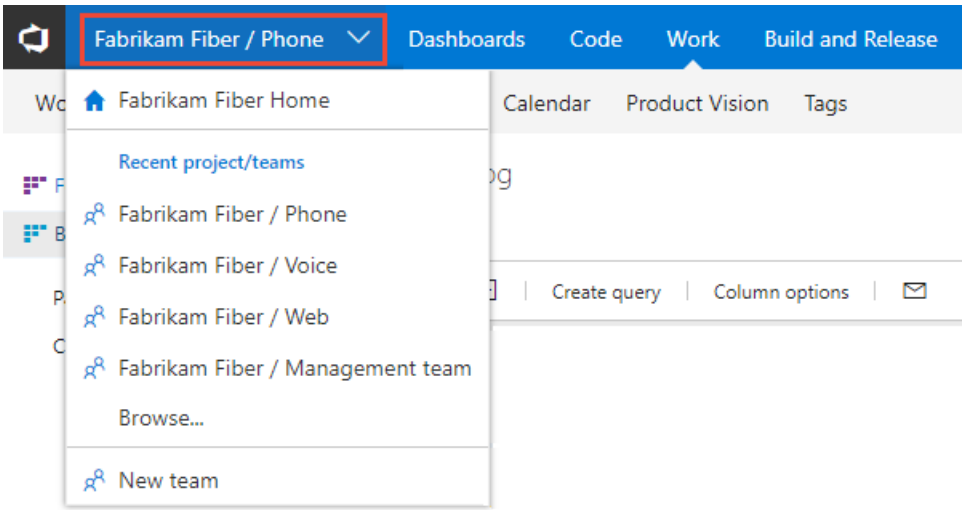
select **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.



1. From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).



To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. (Optional) To choose which columns should display and in what order, choose **Column options**. You may want to add the Iteration Path to the set of columns that appear on your backlog. To learn more, see Change column options.

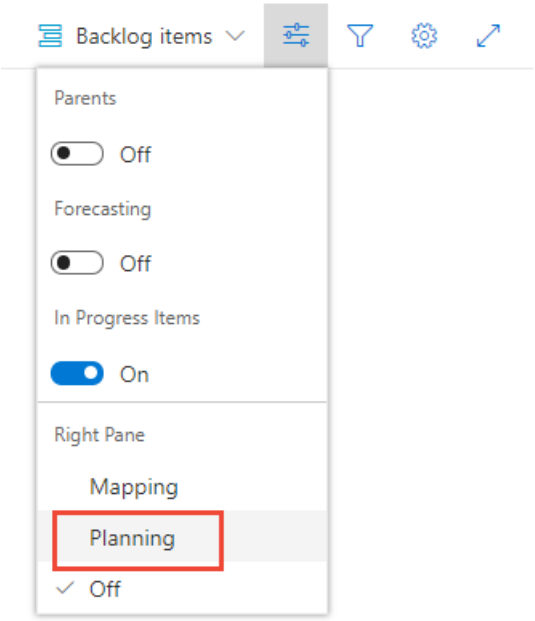
Assign work from your backlog to a sprint

Before you start planning your sprint, you'll want to have created, organized, and estimated your backlog.

Also, you'll want to have set the start and end dates for your sprint.

You can quickly assign work items to a sprint through drag-and-drop from the product backlog to the sprint.

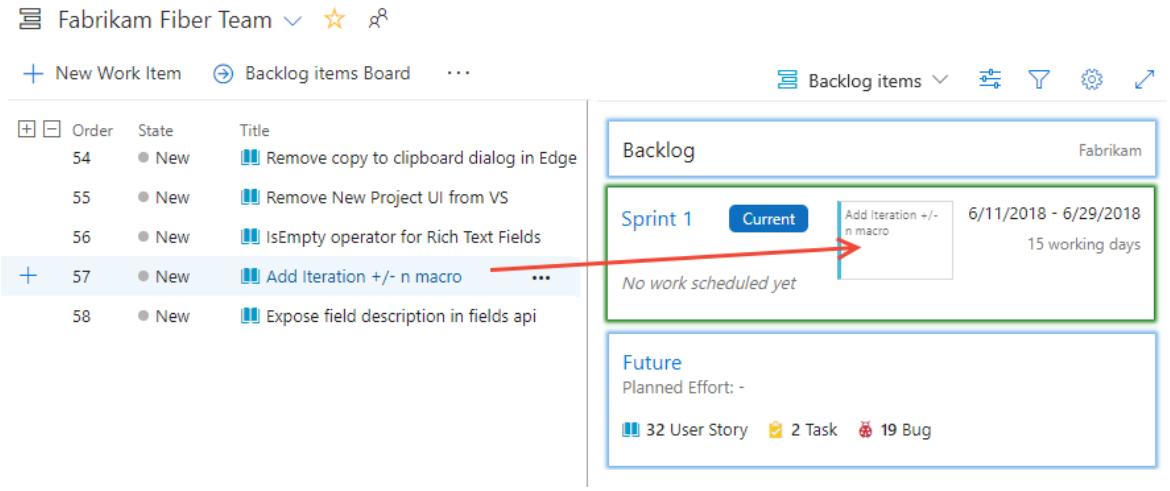
1. The next step is to open the Planning pane. Choose the view options icon and select **Planning**. While you're at it, make sure **Parents** and **Forecasting** are Off. You can choose to set **In Progress items** to On or Off.



   The set of sprints selected for your team appears. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

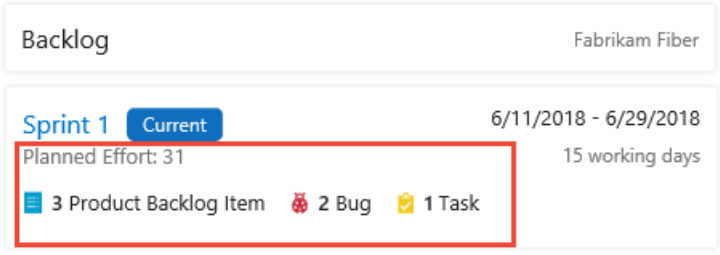
2. You can drag and drop items from the **Backlog** onto a sprint.

> **NOTE**
>
> The **Planning** pane only shows the current sprint and the next 10 future sprints in the list, even if more have been selected for the team. Only a team administrator or member of the **Project Administrators** group can select iterations for a team.



3. Select one or more items from the backlog and drag them to the sprint you're planning. This action will update the Iteration Path of the backlog items and any of its child tasks to the sprint you selected.

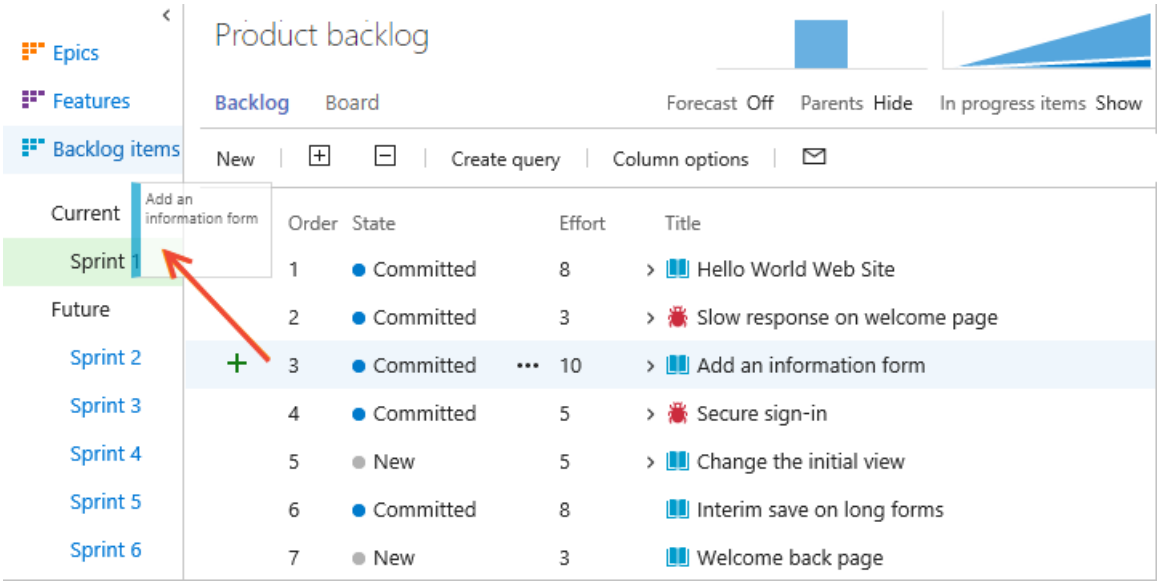
4. Check the level of effort displayed in the sprint window. As you assign backlog items to a sprint, the sprint window updates with a running tally of the number of backlog items and tasks and the **Planned Effort**.



Planned Effort provides a sum of all *Story Points* or *Effort* defined for backlog items assigned to the sprint. This sum represents your initial guess at the amount of work your team will complete in the sprint. Next, you'll define tasks, estimate that work, and use your team's capacity to make sure it fits in the sprint.

Select one or more items and drag them to one of the listed sprints.

> **NOTE**
> The **Iteration Paths** listed only show the current sprint and the next 10 future sprints in the list, even if more have been selected for the team. Only a team administrator or member of the **Project Administrators** group can select iterations for a team.



Use the multi-select feature to modify items in bulk

Multi-select of work items on the product and sprint backlogs works in the same way as multi-select works within query results.

With multi-select, you can complete several actions on several work items at once, such as:

- Move to a sprint
- Change the backlog priority
- Assign to a team member
- Change one or more field values
- Add links
- Map items or change the parent an item is linked to

To select several items in a sequence, hold down the shift key. To select several non-sequential items, use the Ctrl key. Then, you can either drag the selected items to a new position within the backlog, to a different sprint, or select an option from the context (▾) or action (•••) menu of one of the items.

To learn more, see Bulk modify work items.

Next step

Now that you've defined your sprint plan, your team's ready to begin work on the sprint tasks.

2. Add tasks

Related articles

To add or rename the sprints your team uses, see Define iteration (sprint) paths and configure team iterations.

If your backlog doesn't show the work items you expect, see Setup your Backlogs & Boards.

Add tasks to backlog items to support sprint planning

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Add tasks to backlog items when you want to track the work required to implement those tasks. You can also use tasks to estimate work that is assigned to individual team members and the team. The capacity tool tells you how much work your team can commit to. However, to compare capacity with planned work, you need to define and estimate tasks for each backlog item.

In this article you'll learn how to:

- Select a sprint backlog for a team
- Add tasks to backlog items from the sprint backlog or taskboard
- Estimate work, set *Remaining Work*

Add as many tasks as needed to capture the work required to complete each item. Tasks can represent different work to be completed, such as design, code, test, content, or sign out. Usually, each team member adds their own tasks and sets estimates for the work. However, a development lead could define the initial tasks for a story or requirement.

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

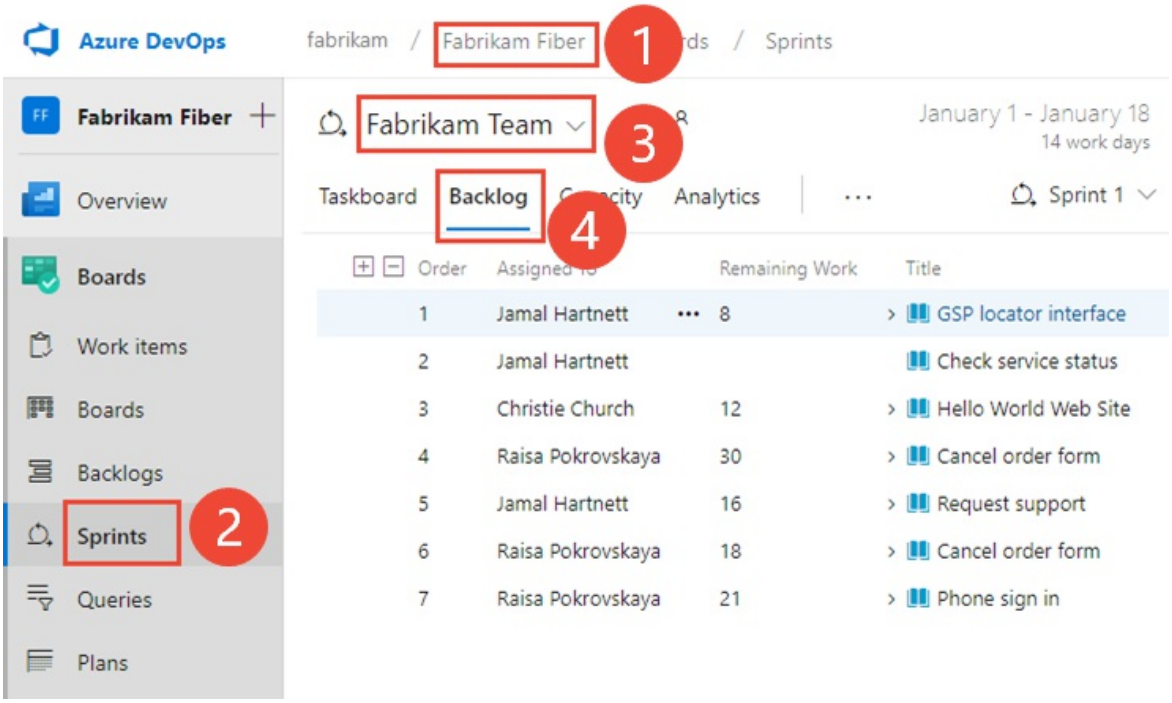
> **NOTE**
> Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.

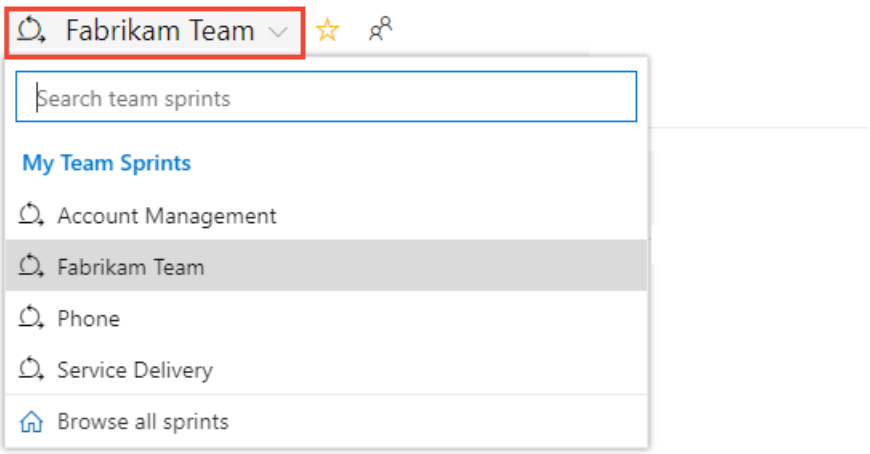
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

Open a sprint backlog for a team

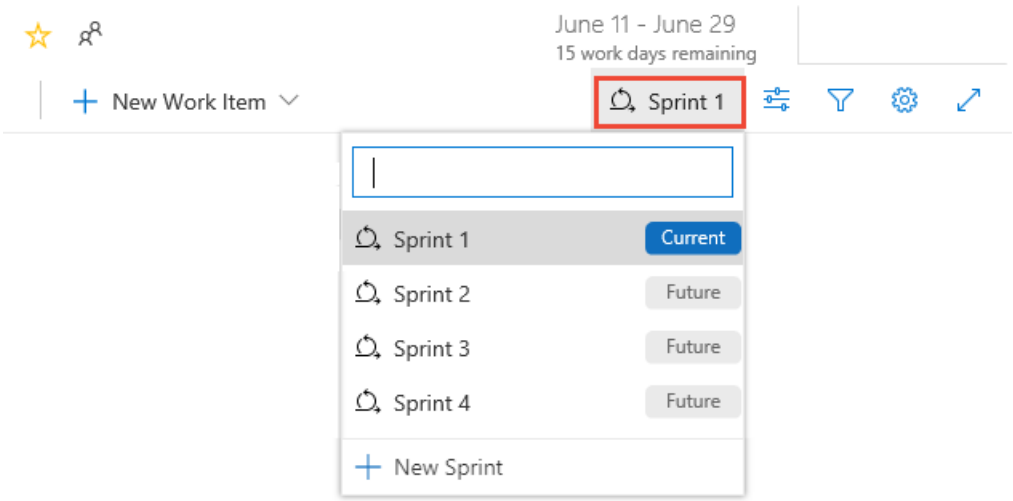
1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

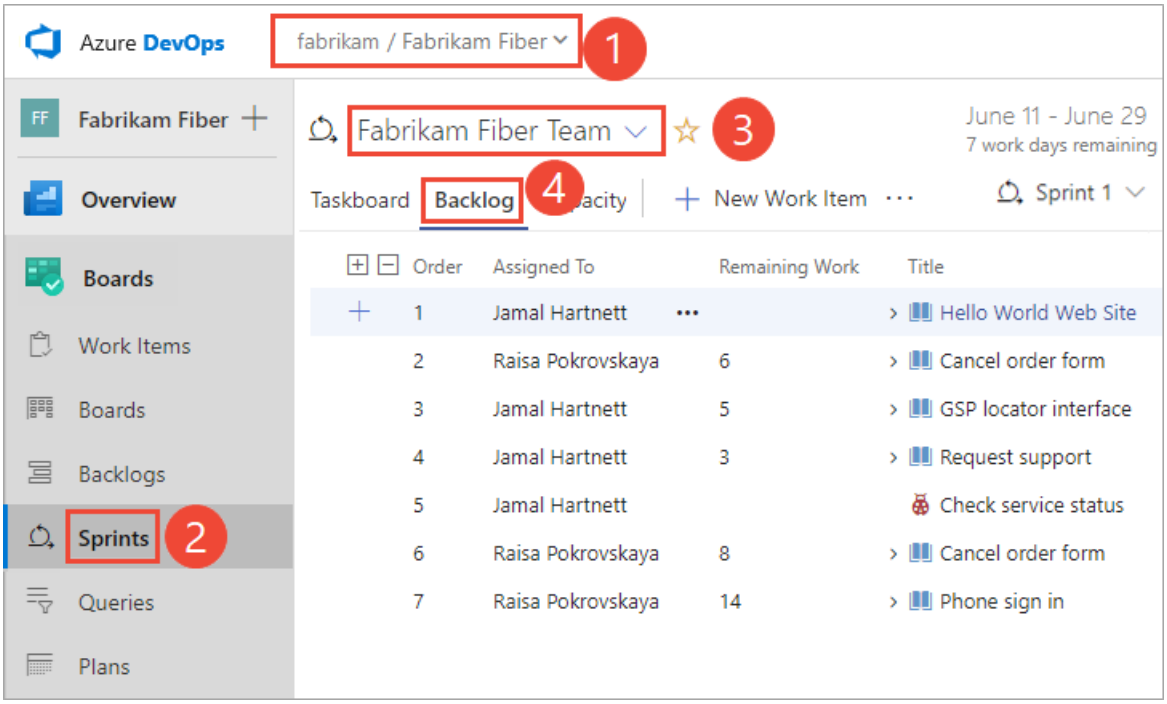


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

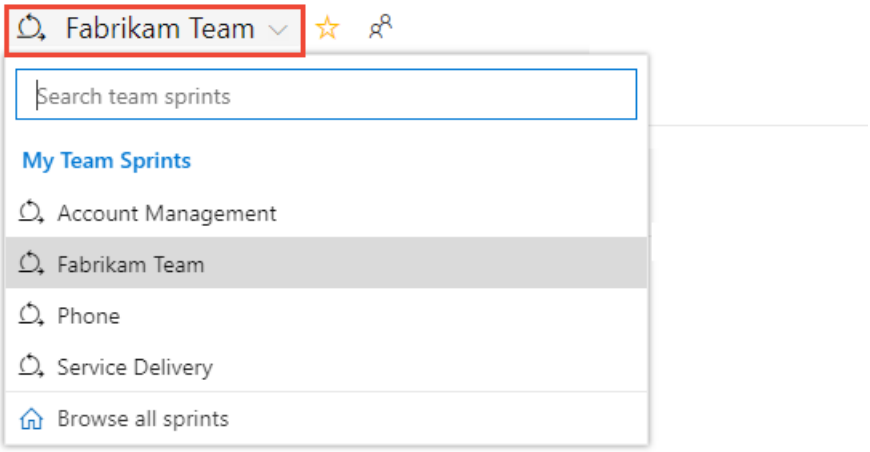


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

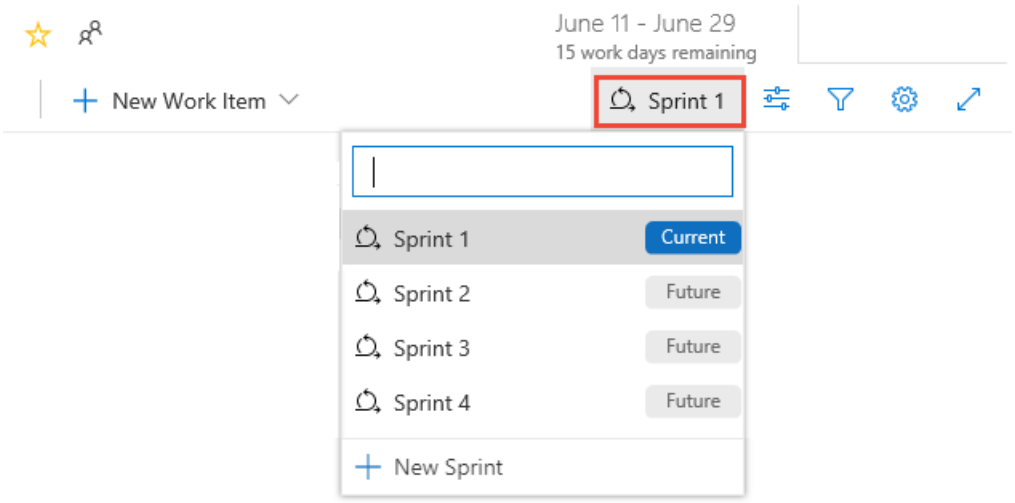
1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

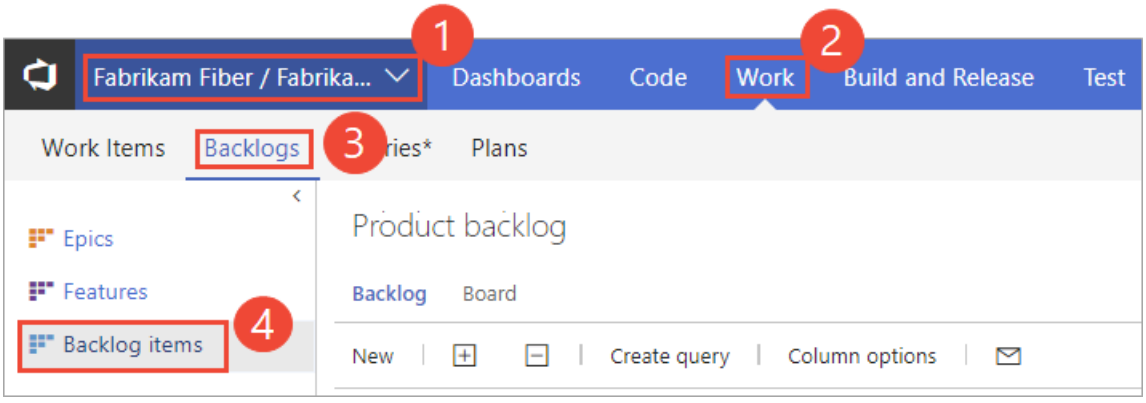


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

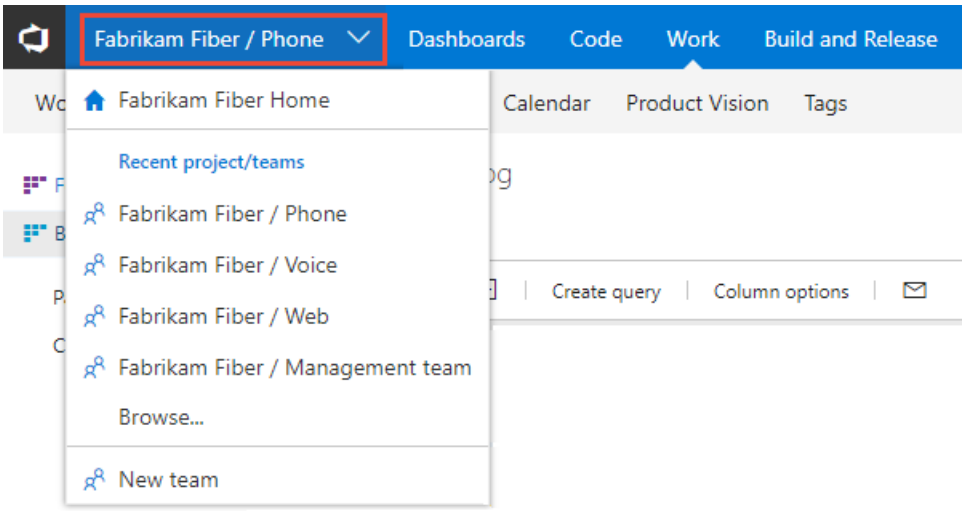


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

1. From your web browser, open your team's sprint backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

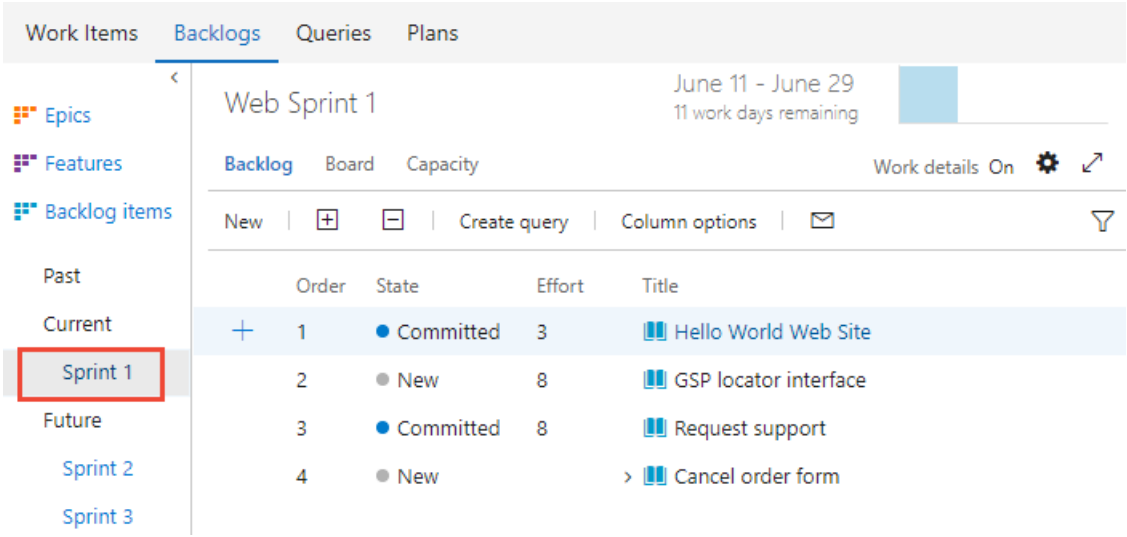


To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. Choose the sprint you want to plan.



The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then see Define iteration (sprint) paths.

Add existing work items to a sprint

If you haven't yet assigned work items to a sprint, see 1. Assign backlog items to a sprint.

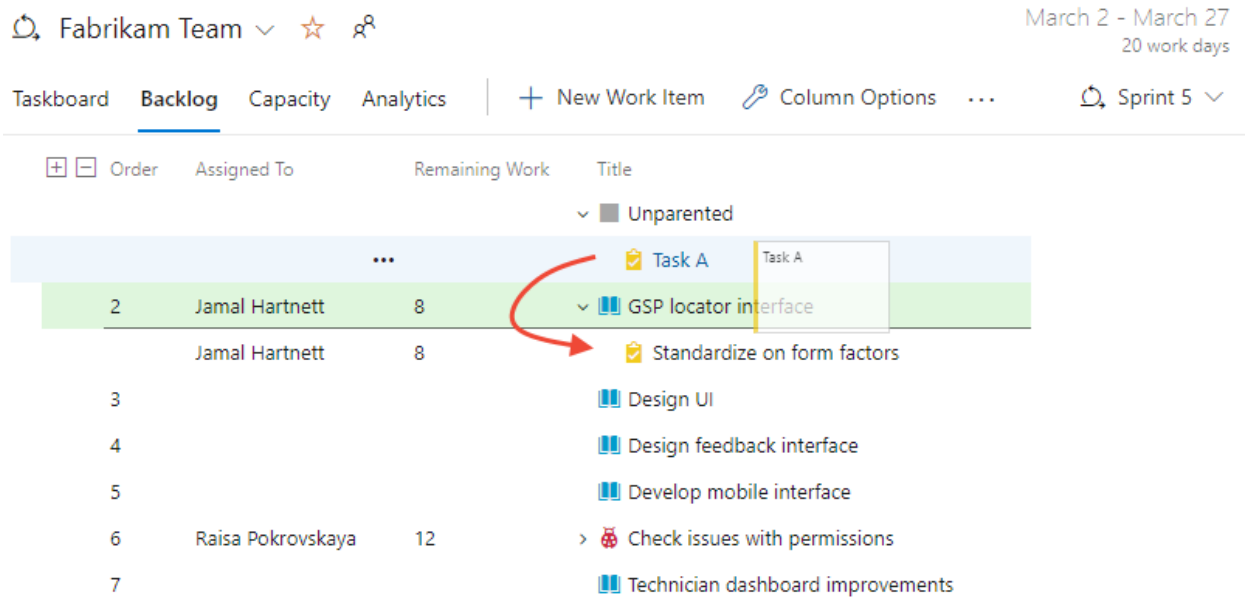
If you have added work items and they don't appear in your sprint backlog, check their area and iteration paths.

- The Area Path must match one of the areas assigned to the team
- The Iteration Path must match the iteration of the sprint that you've selected.

For details on assigning iteration paths to teams, see Define Iteration Paths and configure team iterations.

If you have added task work items, but they don't appear as a child of a product backlog item, you can parent them from the sprint backlog.

Unparented tasks assigned to the selected sprint appear at the top of the backlog under the Unparented group. Drag and drop the task on top of the work item you want to parent it to.



Add tasks to backlog items

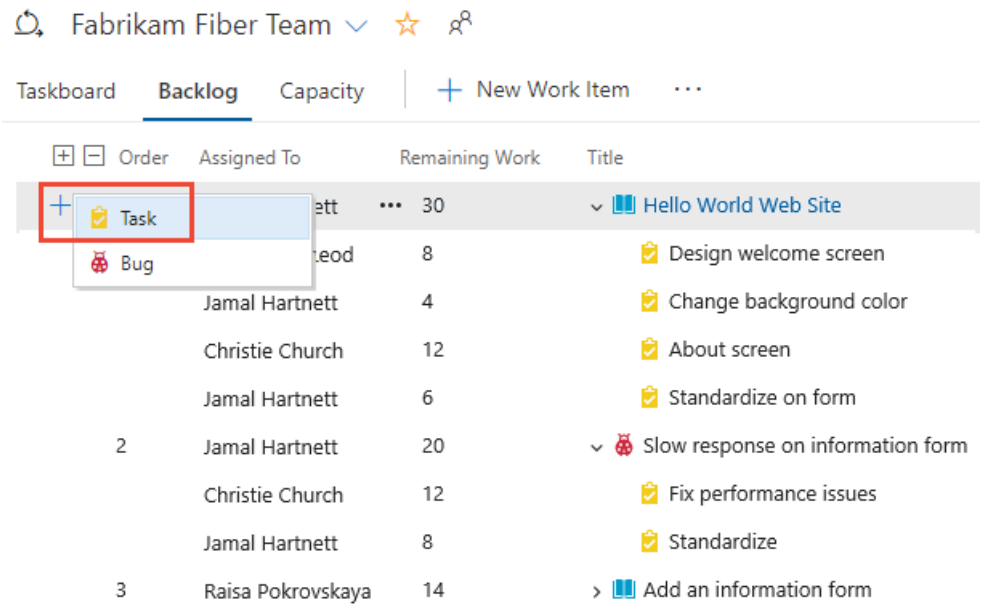
If you haven't yet assigned backlog items to your sprint, do that now. Also, you'll want to have set the start and end dates for your sprint.

For each sprint backlog item, add a task. Adding tasks from the sprint backlog or board automatically links the task to its parent backlog item.

> **TIP**
> You can quickly add several tasks on the taskboard by simply entering a title. You can then later bulk edit items to assign them or add additional details. You can also enter *Remaining Work* onto the card by making sure you add that field to display on the taskboard.

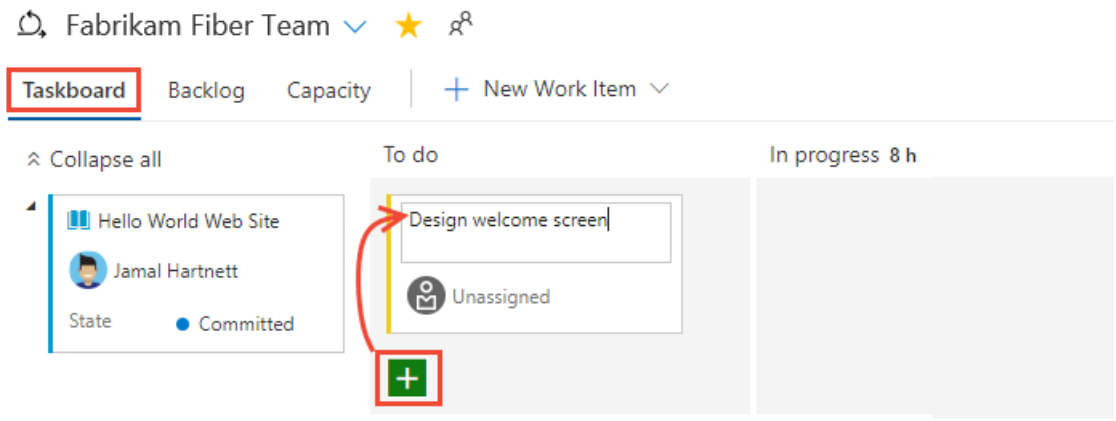
You can add tasks from the sprint **Backlog** or **Taskboard**. ALl items you add are automatically assigned the **Iteration Path** of the selected sprint.

From the **Backlog** view, choose the plus sign to open the work item form for a task.



Fill out the form as described in the next section.

Another option, is to open the **Taskboard**, and add tasks as cards. Select the plus icon, enter a title for the item, and then press Enter on your keyboard.

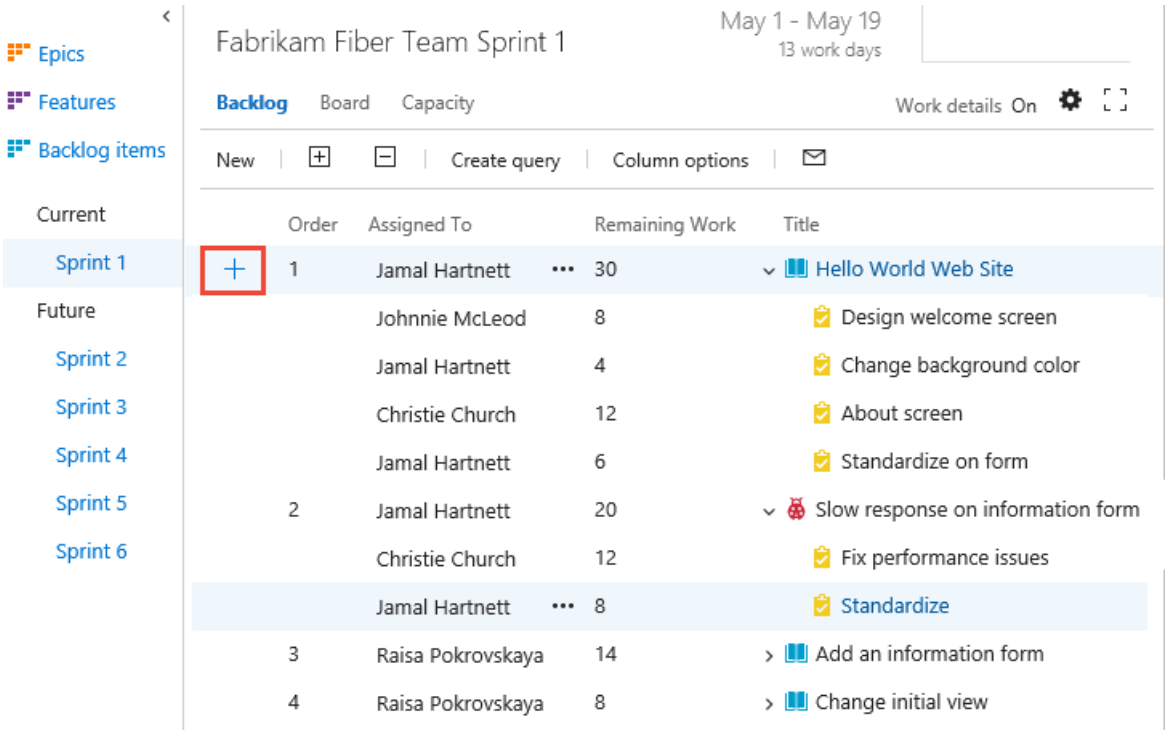


TIP

You can quickly add tasks through the **Taskboard** by just specifying the title of the work item. To show fields on the card, see Customize a sprint Taskboard.

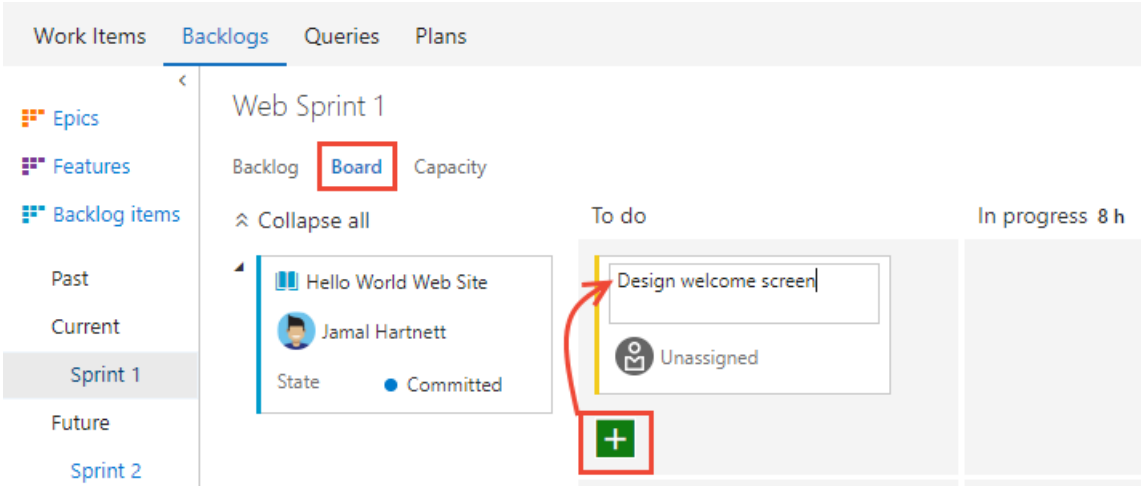
You can add tasks from the sprint **Backlog** or **Taskboard**.

From the **Backlog** view, choose the plus icon to add a new task.



Fill out the form as described in the next section.

Another option, is to open **Taskboard**, and add tasks as cards. Select the plus icon, enter a title for the item, and then press Enter on your keyboard.



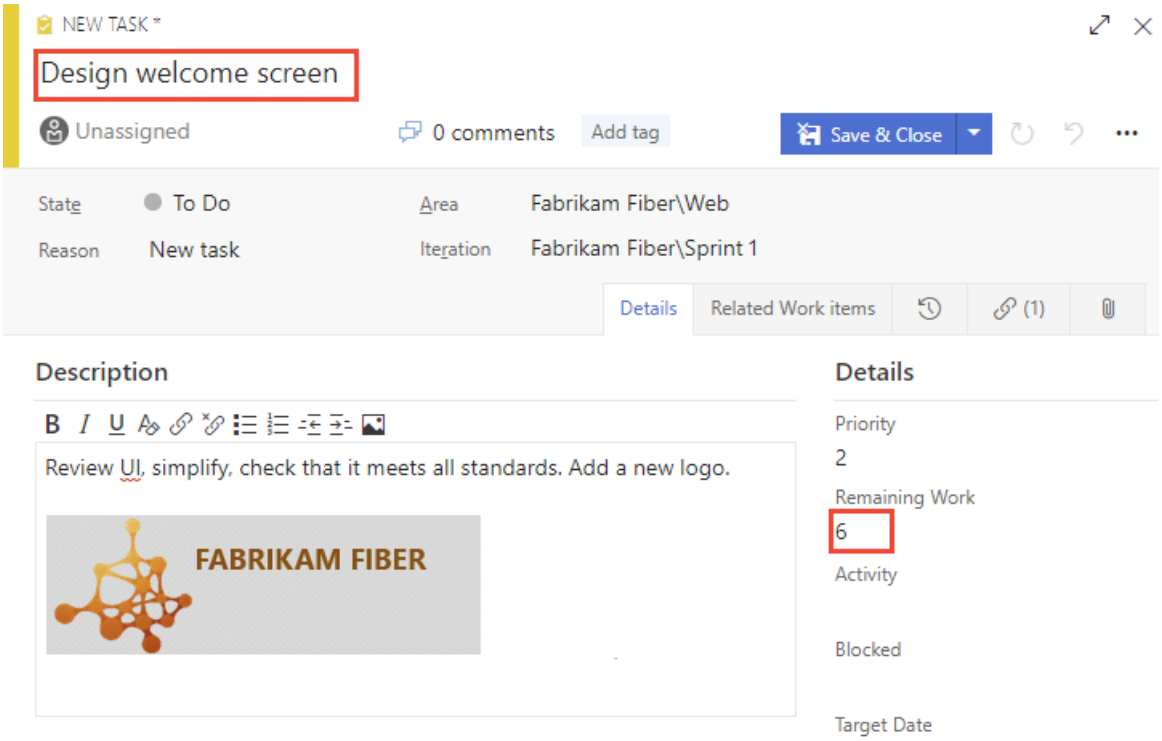
TIP

You can quickly add tasks by just specifying the title of the work item. To show fields on the card, see Customize a sprint Taskboard.

To interactively filter sprint views, choose **Filter** , and then specify a keyword or select a value for a field or tag. To learn more, see Interactively filter backlogs, boards, queries, and plans.

Fill out the task form

Name the task and enter an estimate for *Remaining Work*. Also, if you know who will do the work, go ahead and assign the task to that team member.



![New task form](media/sp-scrum-task-form.png)

At the planning stage, *Remaining Work* corresponds to an estimate of how long it will take to complete the task.

A good rule of thumb is to size tasks to take no more than a day to complete. If a task is too large, the team should break it down. In some cases, you might not estimate some tasks effectively until other tasks have been completed. Create the task now, but estimate it when you have enough information.

During the sprint, team members update remaining work to continually reflect the time required to complete the task. This value can actually increase after work begins. For example, after working 4 hours on a task that was estimated to take 8 hours, you realize you need 16 hours over what you estimated. You would update the *Remaining Work* field with 20 (8-4+16). As you complete a task, you might find that more time is required. Always update the task with your best estimate of remaining work. That way, you help accurately reflect the total amount of work remaining in the sprint.

Field

Usage

Original Estimate

The amount of approximate work required to complete a task. Typically, this field doesn't change after it's assigned.

You can specify work in hours or in days. There are no inherent time units associated with this field.

Remaining Work

The amount of work remaining to complete a task. As work progresses, update this field. It's used to calculate capacity charts and the sprint burndown chart. You can specify work in any unit of measurement your team chooses.

Completed Work

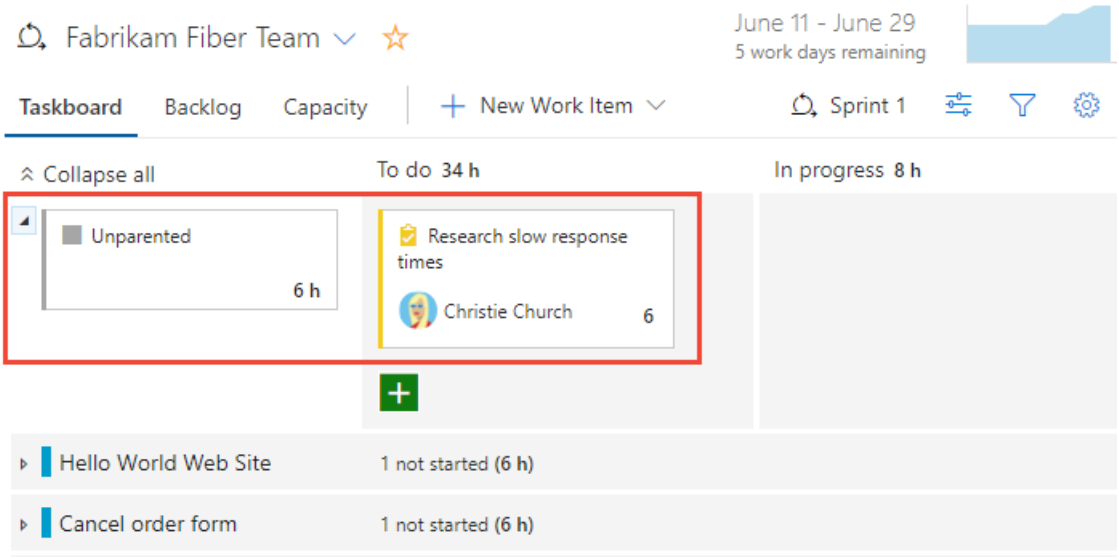
The amount of work spent implementing a task.

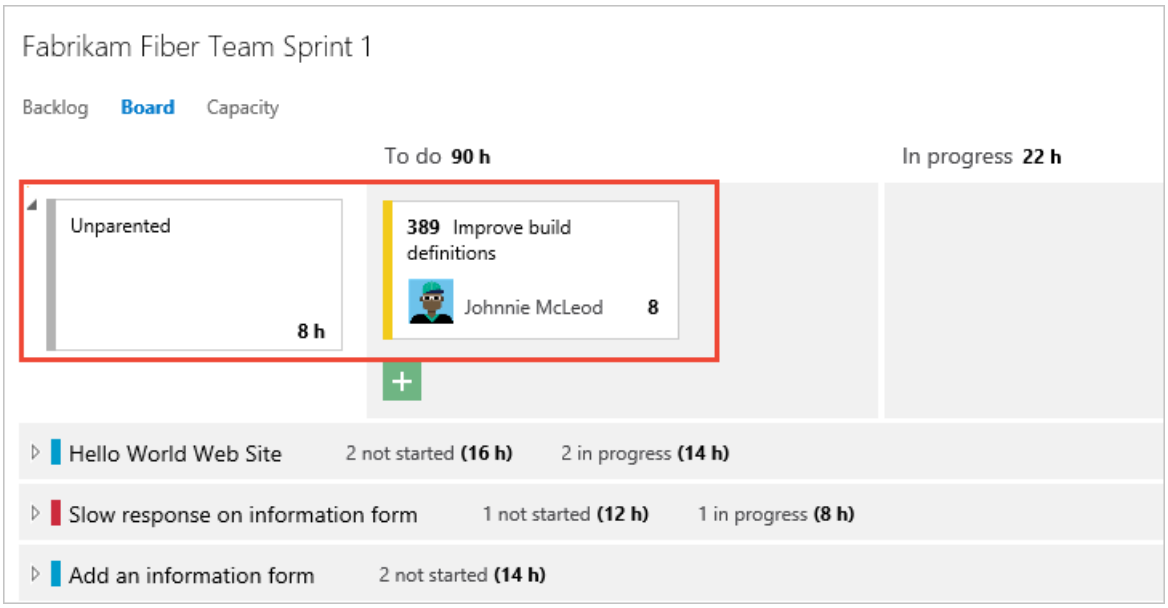
Activity

Select the type of activity this task represents when your team plans sprint capacity by activity.

Unparented tasks

Tasks without links to parent backlog items or user stories appear at the top of the taskboard. You can track unparented tasks in similar ways to other tasks. You can also drag them to an existing backlog item to parent them. The unparented card tracks the total of remaining work defined for all unparented tasks. However, it isn't associated with any work item.





Next step

3. Set sprint capacity

Related articles

- Assign backlog items to a sprint
- Setup your Backlogs & Boards
- Define iteration (sprint) paths and configure team iterations
- Interactively filter backlogs, boards, queries, and plans

3. Determine and set sprint capacity in Azure Boards

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As a next step, you'll want to determine your team's actual capacity. While velocity correlates to how your team estimates requirements, capacity correlates to actual task time. Time is calculated in either hours or days. Capacity takes into consideration the variation in work hours by team members. It also considers holidays, vacation days, and non-working days.

Because days off and time available for each team member may vary from sprint to sprint, set capacity for each sprint. The capacity tool helps you make sure your team isn't over or undercommitted for the sprint. Also, as you work day-to-day, you'll see if your team is on track.

- Set team capacity for a sprint
- Copy capacity from the previous sprint to the current sprint
- Track capacity when performing multiple activities
- Add or remove user accounts from capacity planning for a sprint
- Track capacity when working on more than one team

If you haven't set up sprints yet for your team, see the Manage sprint timelines while working in Scrum article.

Prerequisites

- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To view or set capacity, you must be granted **Basic** access or higher. Users with **Stakeholder** access can't view or set capacity. For details, see Stakeholder access quick reference.
- To set capacity, you must be a member of the team. For details, see Add users to a project or team.

About the Activity or Discipline pick-list items

The values displayed for **Activity** (Agile, Basic, or Scrum) or **Discipline** (CMMI) reflect a union of all values defined for the Activity or Discipline fields in all projects within the organization.

To change the **Activity** or **Discipline** menu selections, see Add and manage fields.

The values displayed for **Activity** (Agile, Basic, or Scrum) or **Discipline** (CMMI) reflect a union of all values defined for the Activity or Discipline fields in all projects within the project collection. Or, if your project has been customized, with the field assigned to `type="Activity" within the ProcessConfiguration file`.

To change the **Activity** or **Discipline** menu selections, see Add or modify a field, customize a picklist.

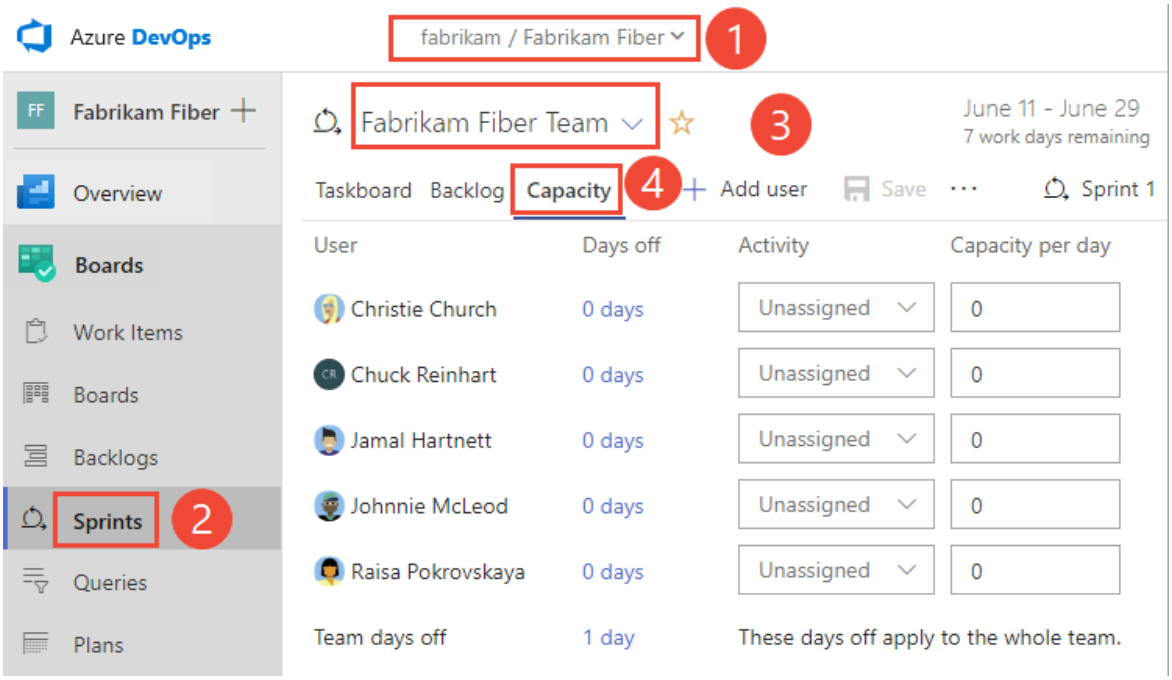
Capacity per day entries

Most teams specify capacity in hours. You can also specify it in days or any other units your team chooses. For example, 0.5 days would correspond to 4 hours for a typical 8 hour day. Choose the same unit your team uses to

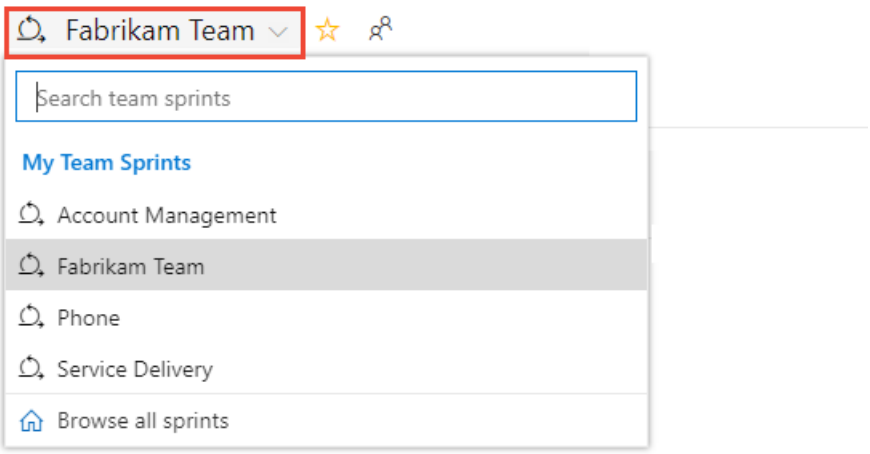
estimate and track their time. For example, the entries they'll make to the Original Estimate or Remaining Work fields.

Open a sprint backlog for a team

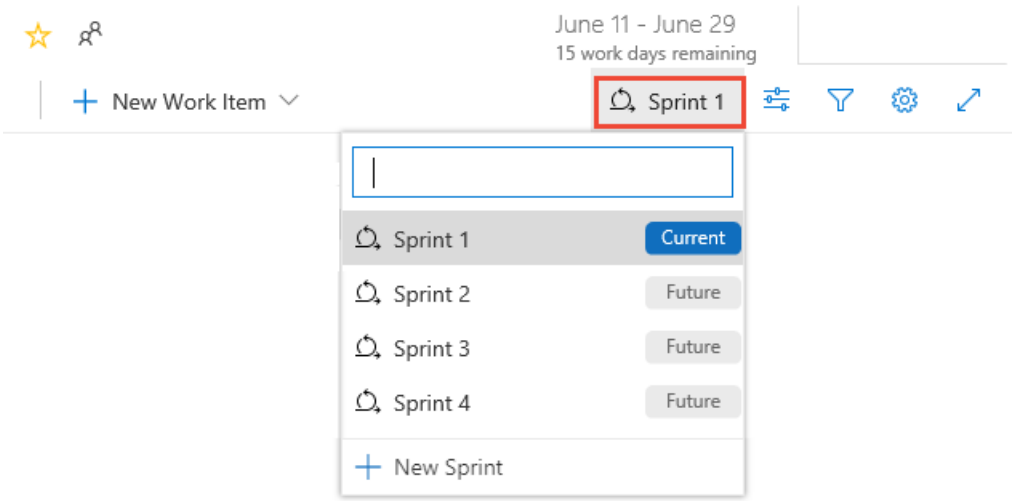
1. From your web browser, open your product backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Capacity**.



   To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

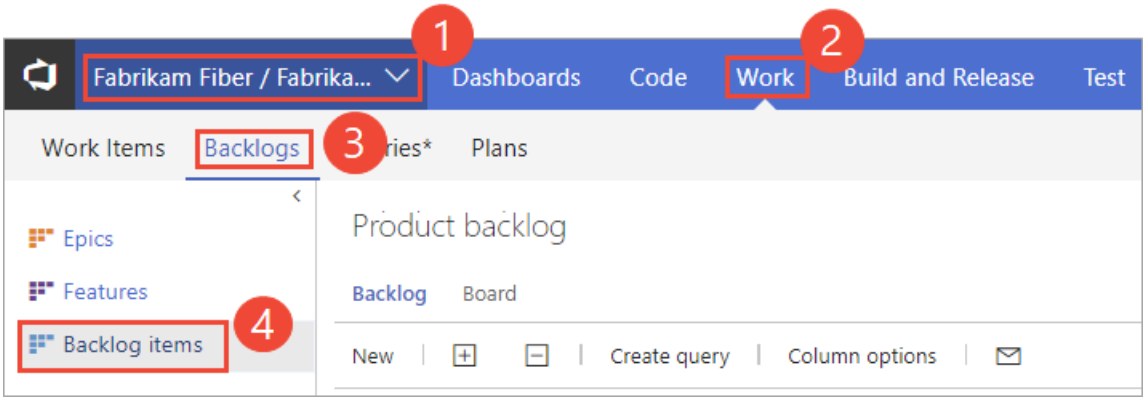


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

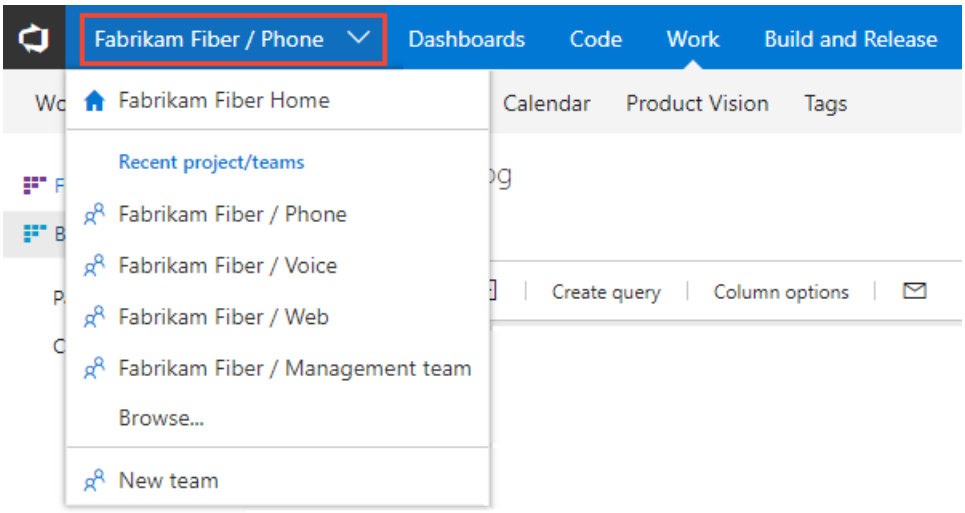


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

1. From your web browser, open your team's product backlog. (1) Select the project/team from the project/teams selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

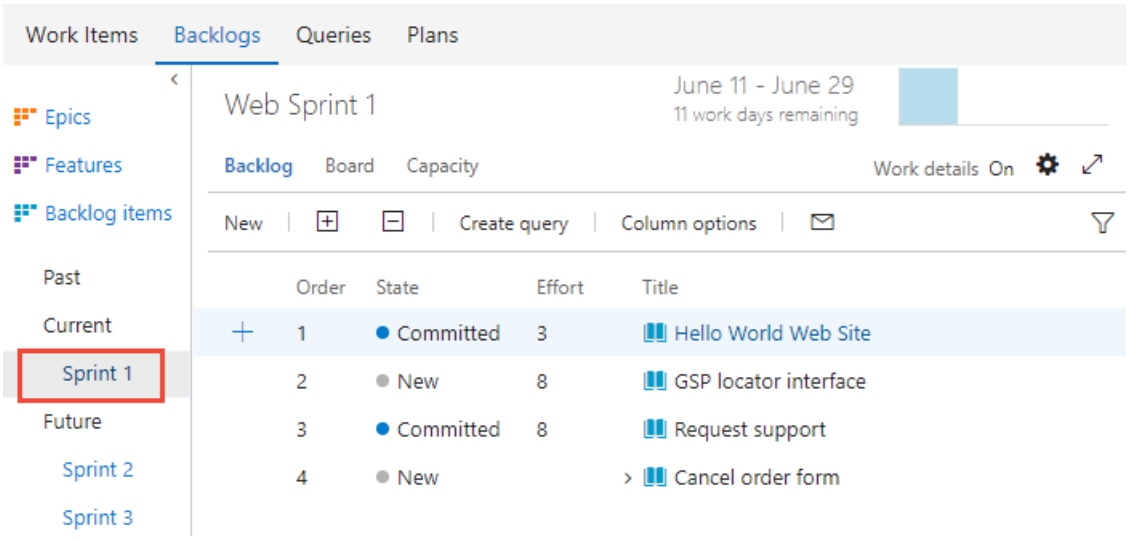


To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. Choose the sprint you want to plan.

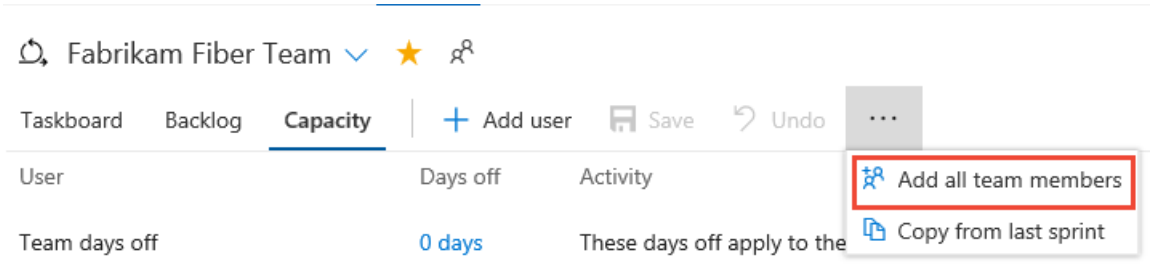


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then see Define iteration (sprint) paths.

Set capacity for the team and team members

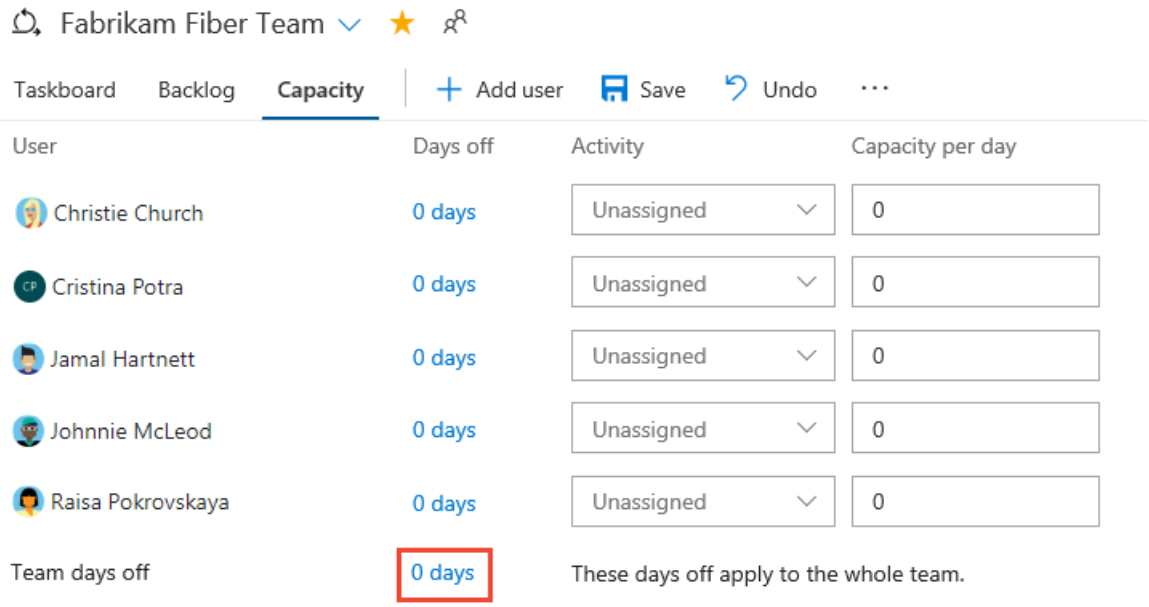
From the **Capacity** page, you can add team members, enter the team time off, and set capacity and days off for each team member.

1. If you don't see your team members listed, add them. Choose the ··· action icon and select **Add all team members**. For this feature to work, team members will have been added to the team.



> **NOTE**
> The **Add all team members** action retrieved a maximum of 100 team members. If you have more team members to add, then you can add them one by one by choosing **Add user**.

2. If you need to add other contributors to your project, choose the + **Add user**.
3. Next, set any time off that a team member will take. For the entire team days off, choose the **0 days** link as shown.

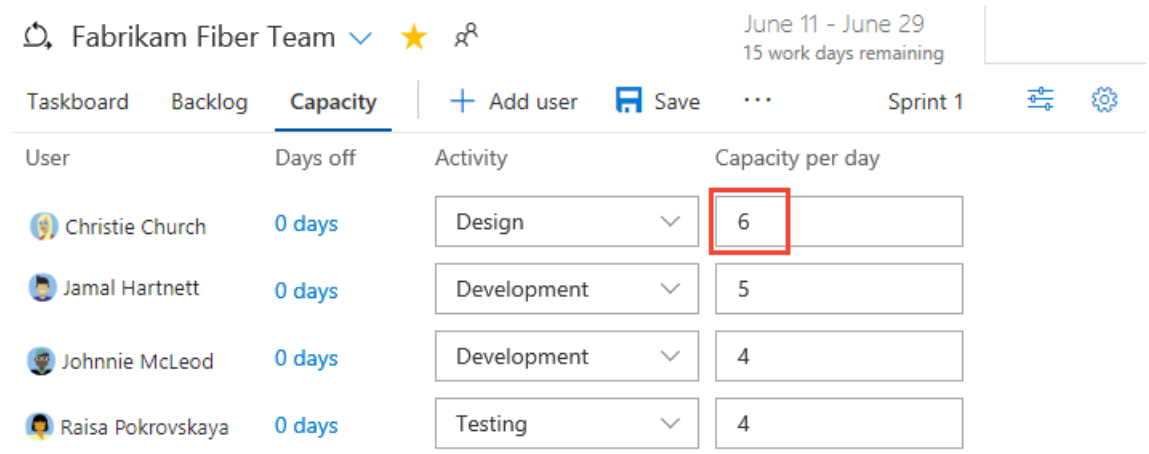


In the **Days off for** dialog, select the start and end days during the sprint that the team member or team will take off.

> **NOTE**
> Your sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown. You only have to indicate planned days off for the team. You set weekend days or other recurring days off under your team's Settings, Working days page.

4. Now, set the **Activity/Discipline** and **Capacity** per day for each team member. If you track capacity simply by team member, you can leave the Activity or Discipline selection unassigned.

   For example, Christie Church's capacity is 6 hours/day for design work.

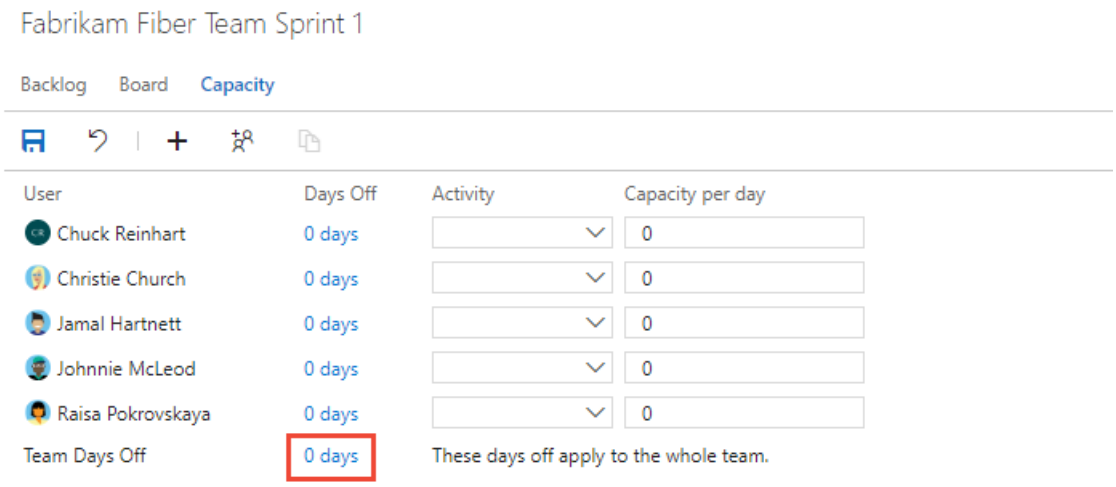


1. If you don't see your team members listed, add them. Choose the **Add missing team members** icon. For this feature to work, team members will have been added to the team.



2. If you need to add other contributors to your project, choose the + **Add user** icon.

3. Next, set any time off that a team member will take. For the entire team days off, choose the **0 days** link as shown.

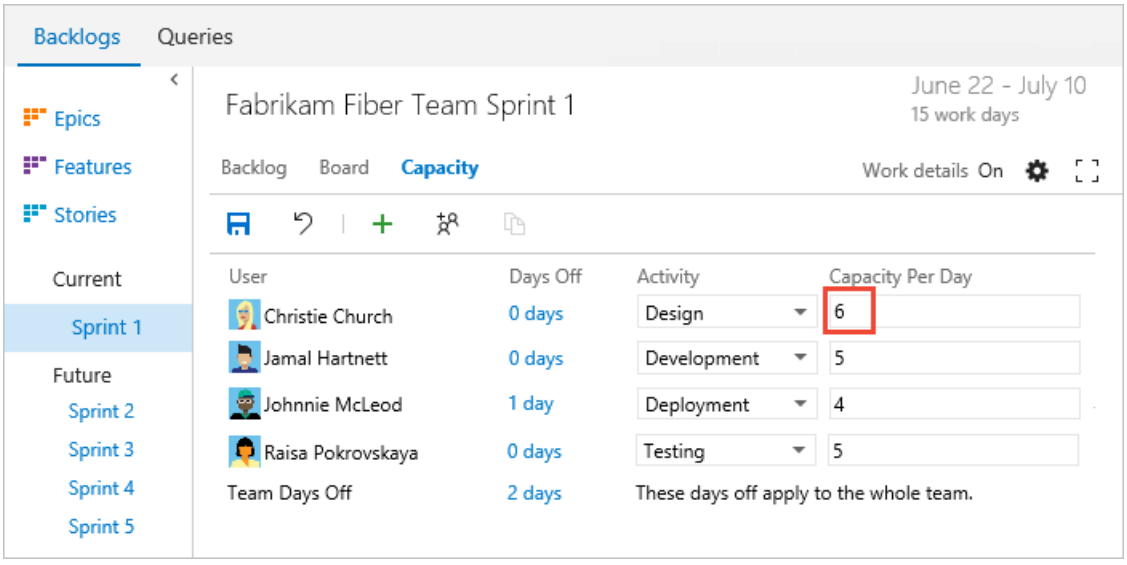


In the Days off for the entire team dialog, select the start and end days during the sprint that the team will take off.

> **NOTE**
> Your sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown. Leave those days of the week that your team doesn't work unchecked in your team's Settings, Working days page.

4. Now, set the **Activity/Discipline** and **Capacity** per day for each team member. If you track capacity simply by team member, you can leave the Activity or Discipline selection unassigned.

   For example, Christie Church's capacity is 6 hours/day for design work.

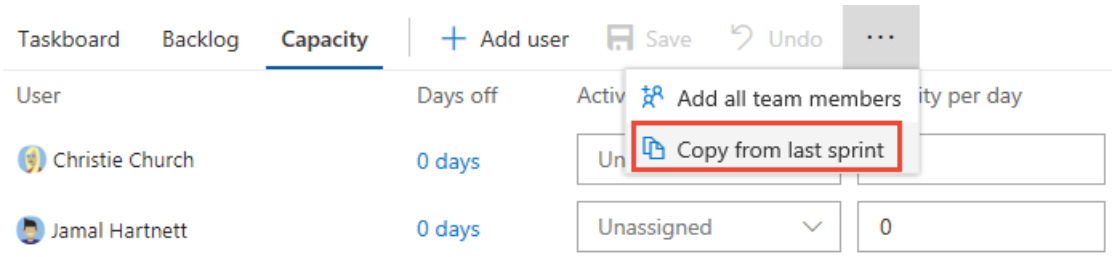


Copy capacity planning from the previous sprint

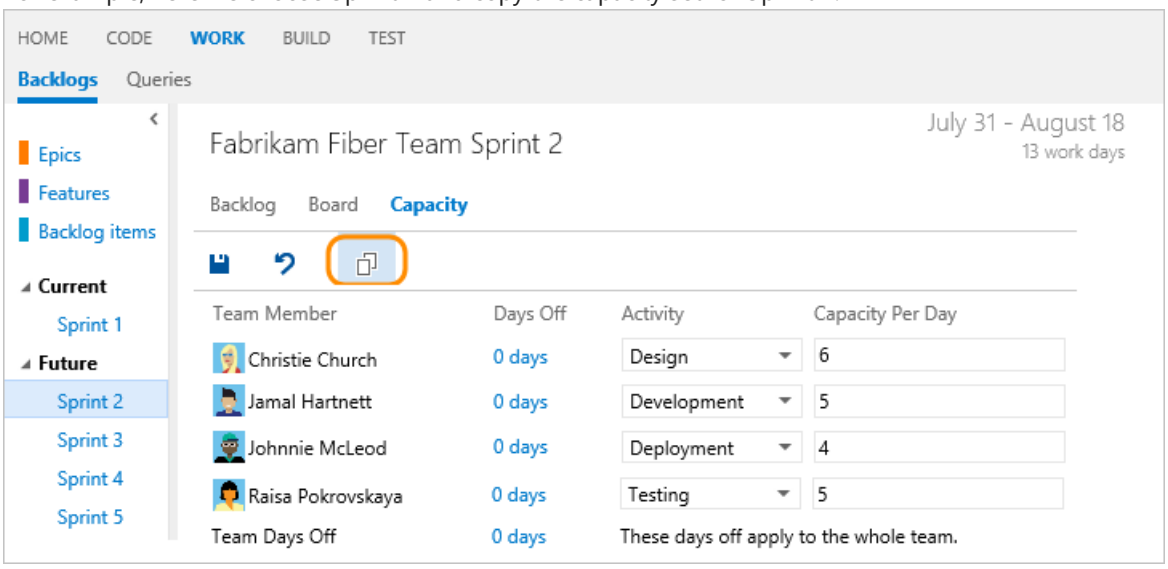
By copying the capacity from the previous sprint, you save time. With the basics defined, all you have to do is adjust the capacity based on individual and team days off and capacity allocation per activity.

> **NOTE**
> This feature is available from TFS 2015.1 and later versions.

Notice that only the capacity-per-day value and activity value are copied over. Individual and team days off remain unset. The copy operation always copies the latest updates made to the previous sprint. So you can repeat the copy operation if you've made changes to the previous sprint that you want to copy to the latest sprint.

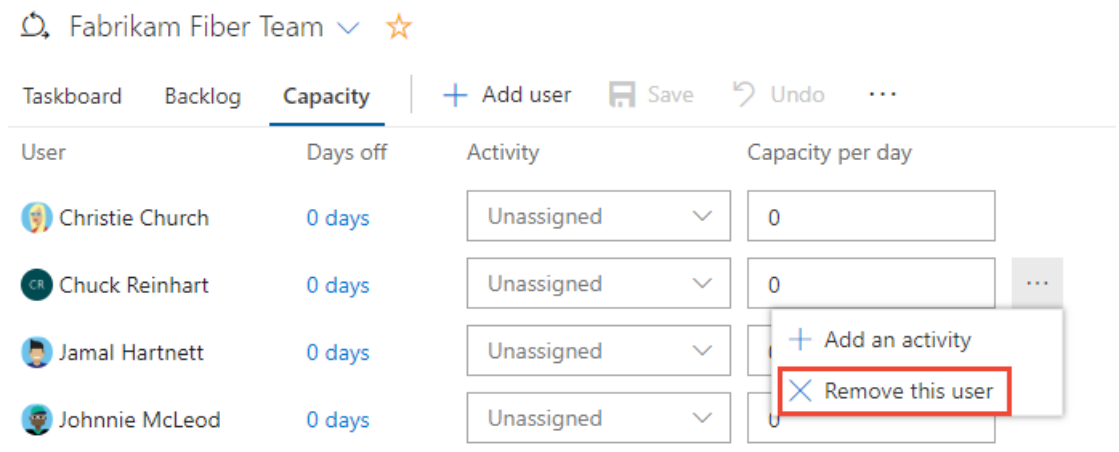


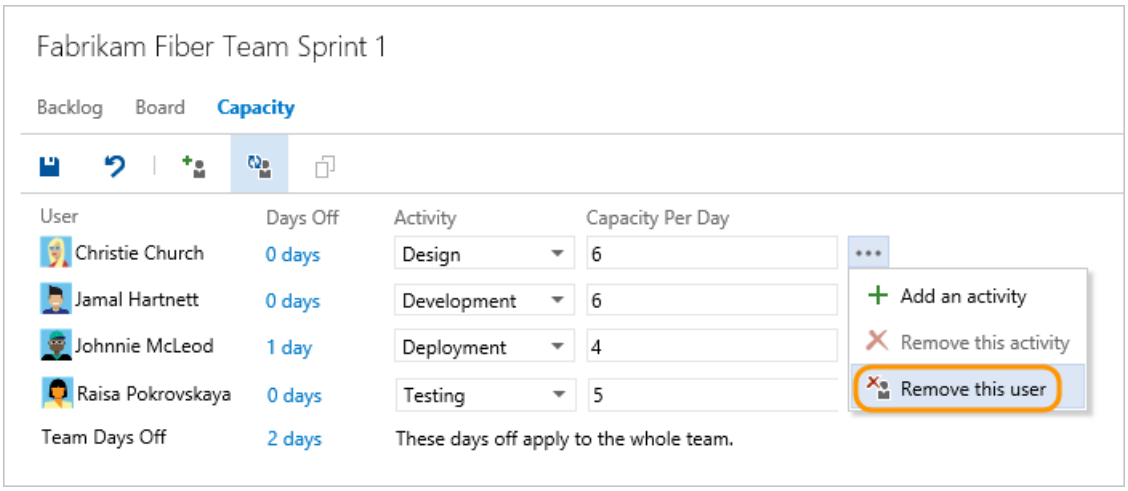
For example, here we choose Sprint 2 and copy the capacity set for Sprint 1.



Remove a user from capacity

To remove a user, choose the option from the users ··· action menu. This action won't remove the user from the team.

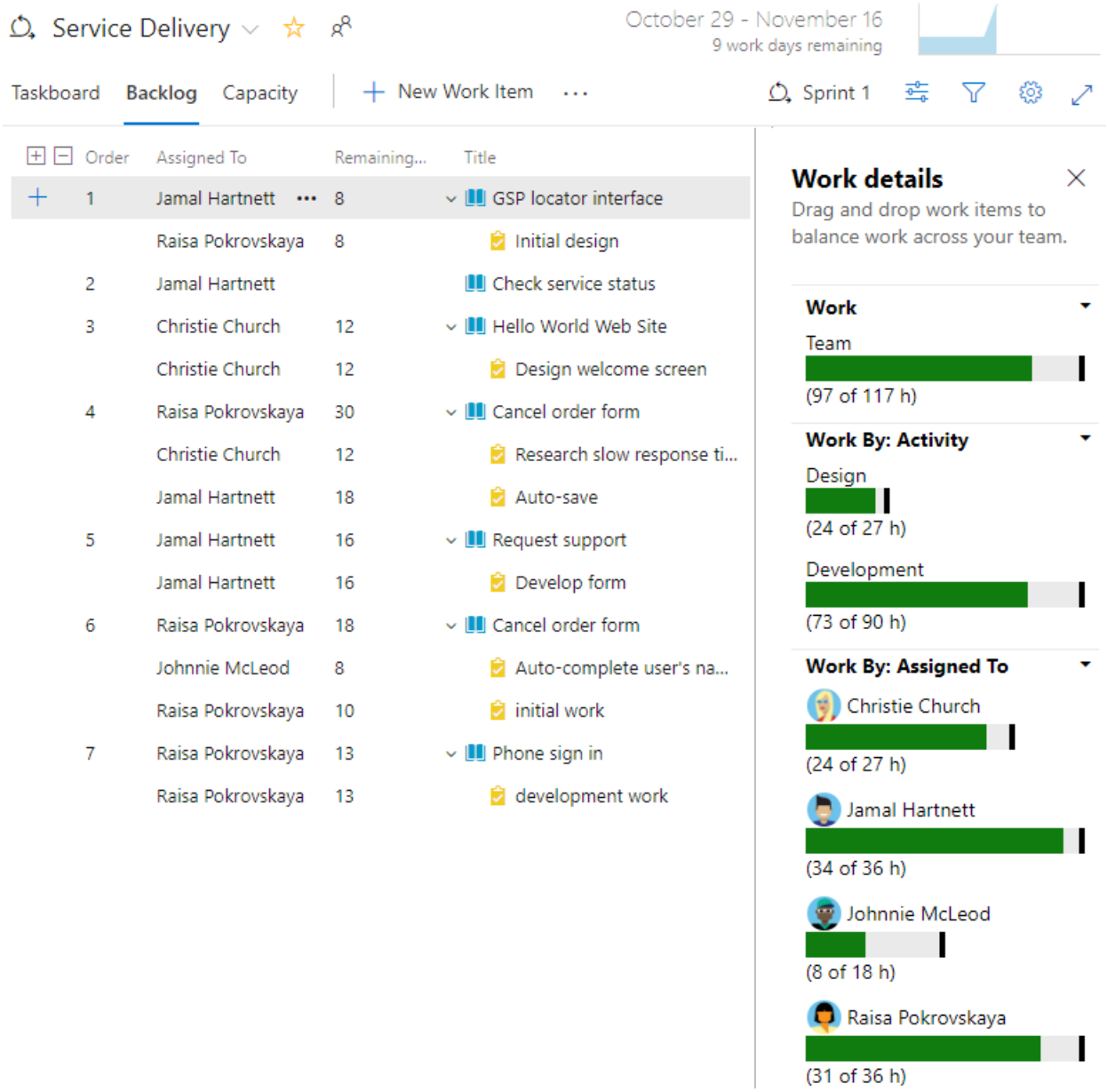


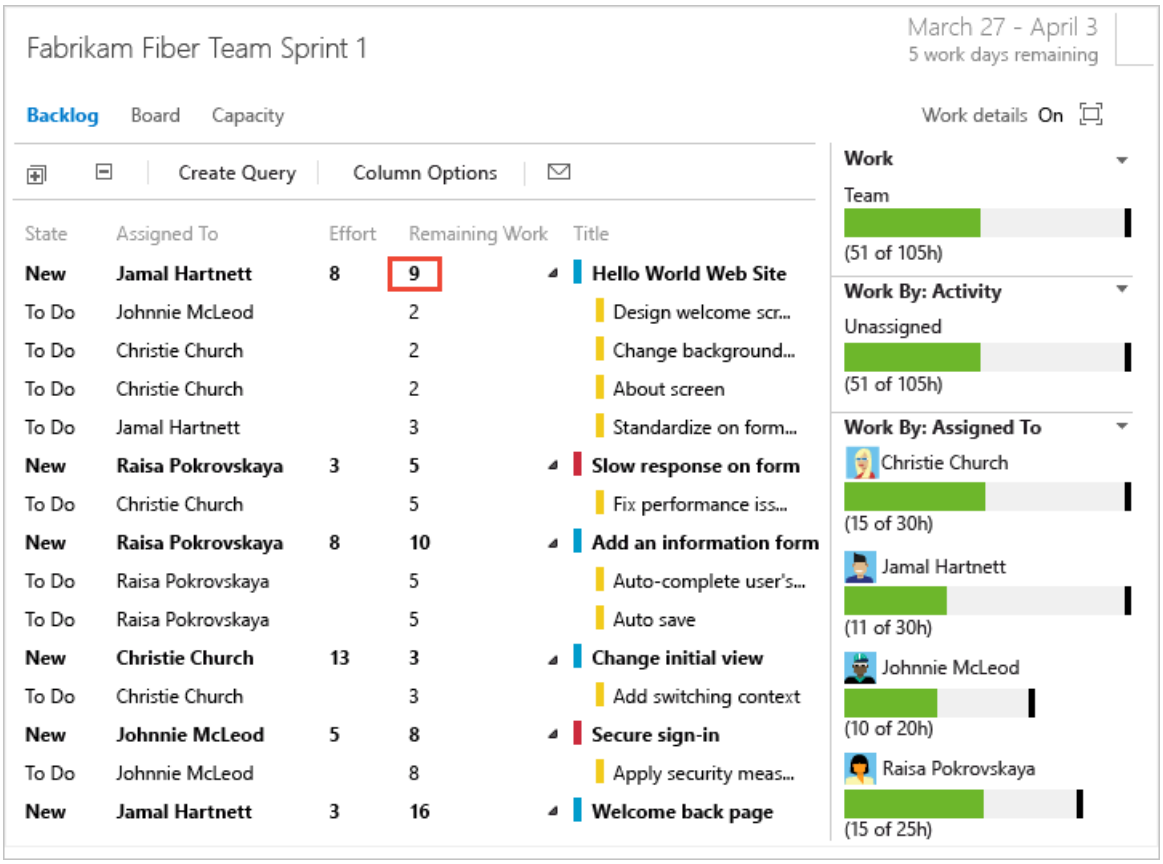


Review capacity charts

As you define tasks and estimate the work, you'll see capacity charts start to fill in for each team member. Capacity bars track the remaining work against the capacity for each team member and the entire team.

You'll also see a roll-up of the remaining work required to complete each requirement or bug.





From this view, you can easily see which individuals are at or near capacity. Teams can determine if work needs to be moved out of the sprint or to reassign tasks.

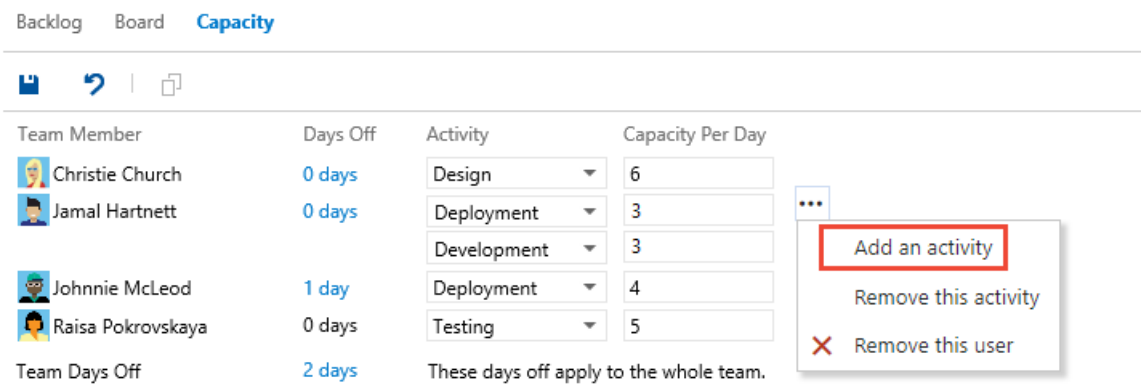
> **TIP**
> Define tasks that take a day or less to complete. This helps mitigate the risks that come from poor estimates.
>
> Also, don't divide tasks into subtasks. If you do divide a task into subtasks, specify Remaining Work only for the subtasks, as the system rolls up summary values to the parent task.

> **TIP**
> Define tasks that take a day or less to complete. This helps mitigate the risks that come from poor estimates.
>
> Also, don't divide tasks into sub-tasks as taskboards only show leaf node tasks. If you do divide a task into sub-tasks, specify Remaining Work only for the sub-tasks, as the system rolls up summary values to the parent task.

Track capacity when completing multiple activities

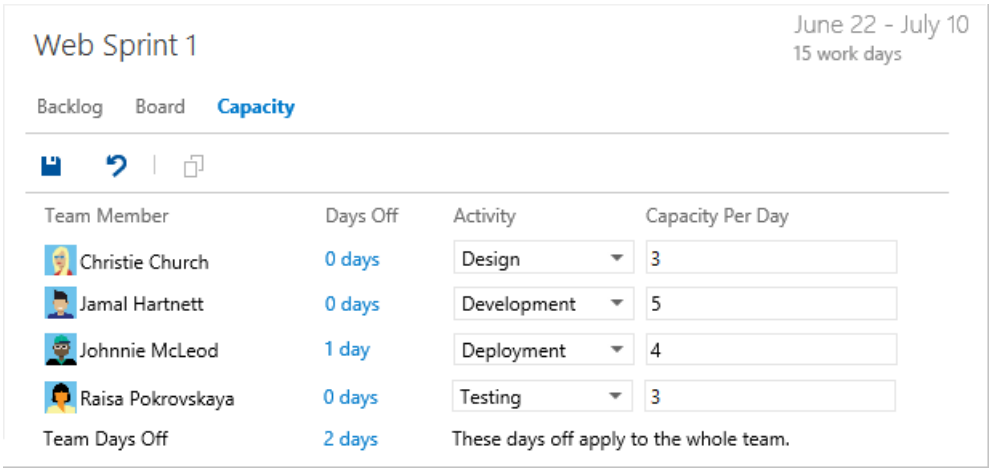
Because individual team members have different sets of skills and duties, you can track their activity and capacity for each activity and for each sprint.

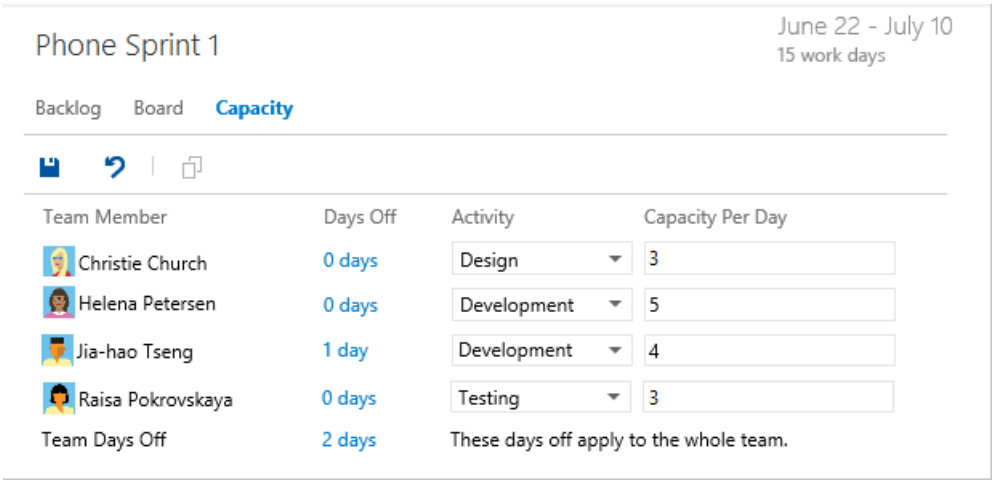
Here, Jamal divides time between Deployment and Development.



Track capacity when working on more than one team

If you work on more than one team, you'll want to specify your sprint capacity for each team. For example, both Christie and Raisa split their time between the Web and Phone teams. As such, give 3 hours a day to the Web team, and 3 hours a day to the Phone team.





If your name isn't listed in the capacity view, you need to be added as a team member.

Next step

4. Adjust work

Related articles

Setting capacity and estimating remaining work for each task provides you with the tools you need to track the amount of work and resources you have given sprint over sprint.

- Sprint burndown
- Velocity
- Forecasting
- Manage teams and configure team tools

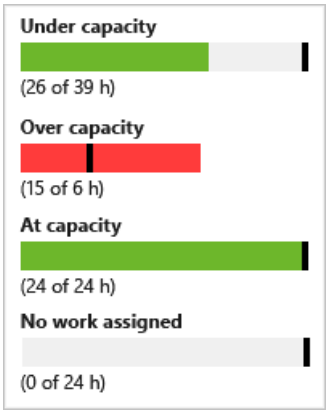
Adjust work to fit sprint capacity

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Check your team's capacity after you've defined all the tasks for all the sprint backlog items. You can consider adding more items onto the sprint if your team is under capacity. If over capacity, you'll want to remove items out of the backlog.

Next, check whether any team member is under, at, or over capacity. Or, if someone hasn't even been assigned any work. Use the capacity bars to make these determinations. If you haven't yet set capacity for your team, do that now.



Use this article to learn how to:

- Adjust your sprint plan if your team is over or under capacity
- Load balance work across your team
- Quickly reassign tasks to another team member

Prerequisites

Backlogs are automatically created when you create a project or add a team. Each team has access to their own product, portfolio, and sprint backlogs as described in About teams and Agile tools.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

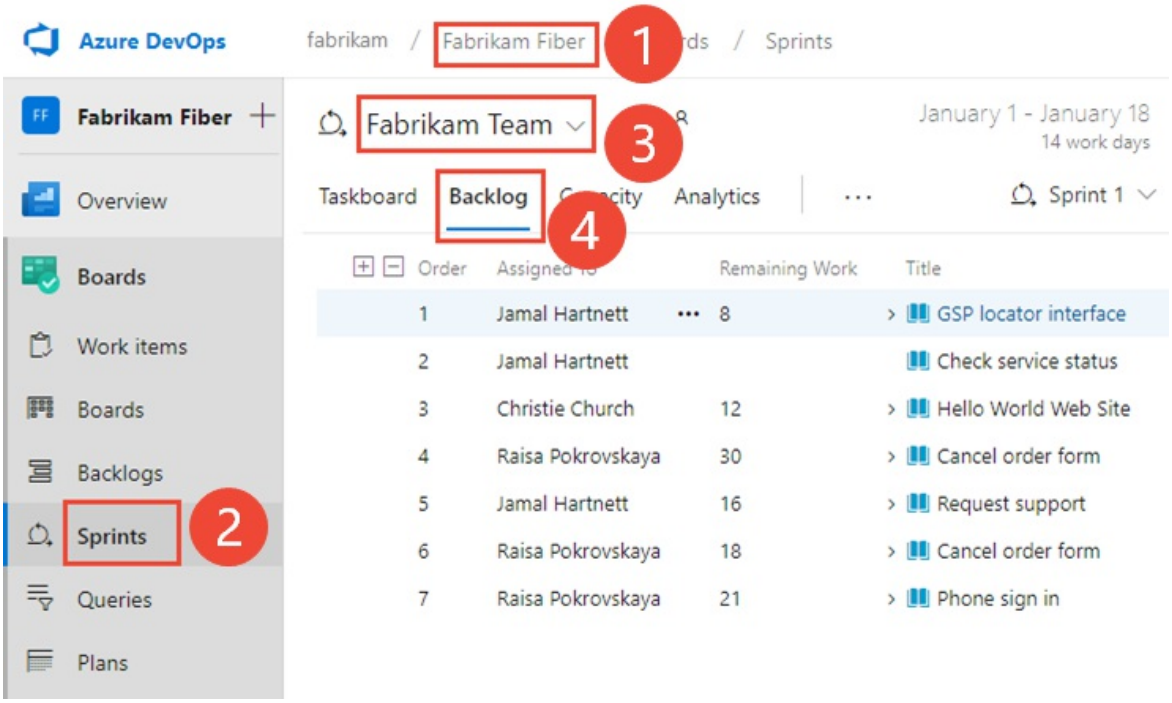
NOTE

Users with **Stakeholder** access for a public project have full access to backlog and board features just like users with **Basic** access. For details, see Stakeholder access quick reference.

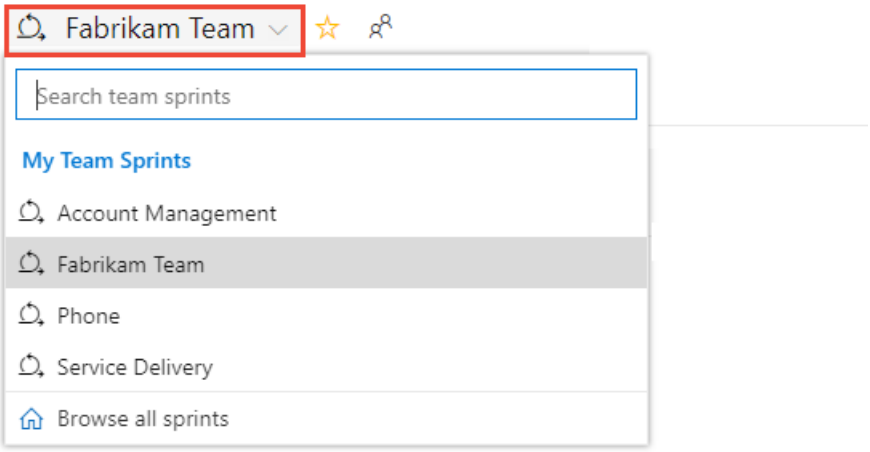
- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To use the **Planning** pane, the sprints that you want to assign work to must have been selected for your team by a team administrator. To learn more, see Define iteration (sprint) paths and configure team iterations.

Open a Sprint backlog for a team

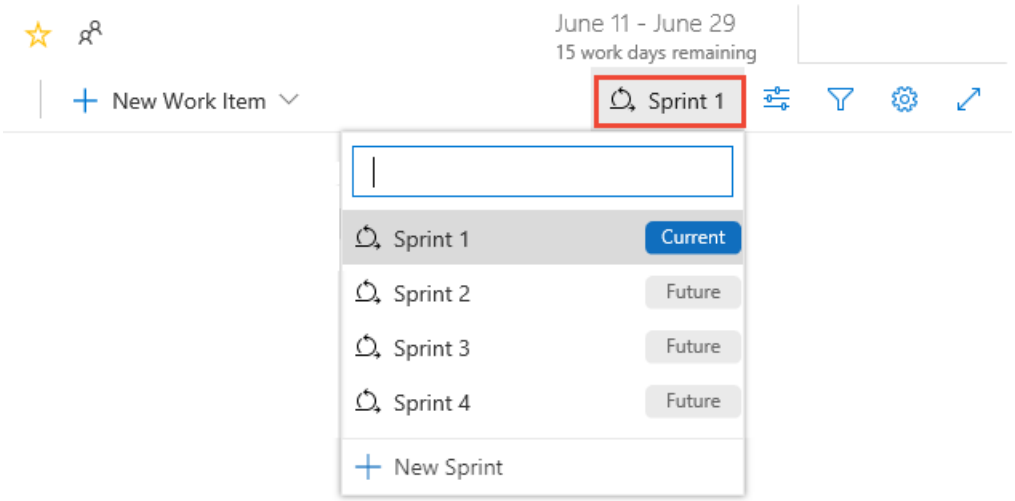
1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

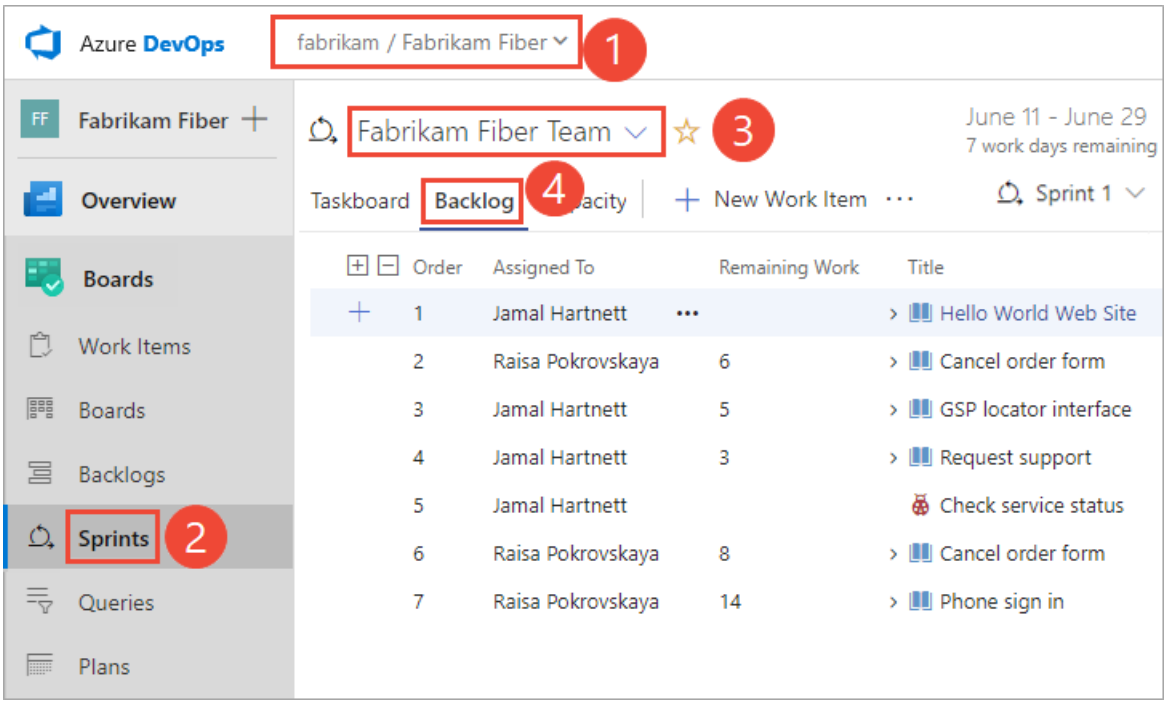


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

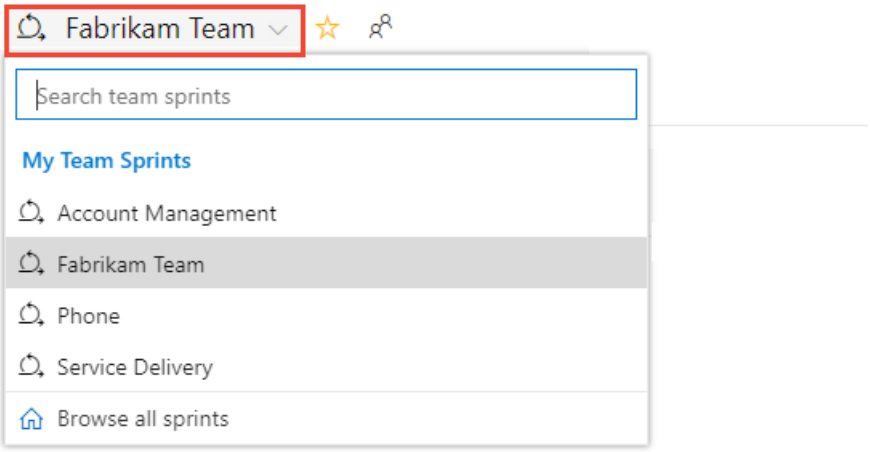


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

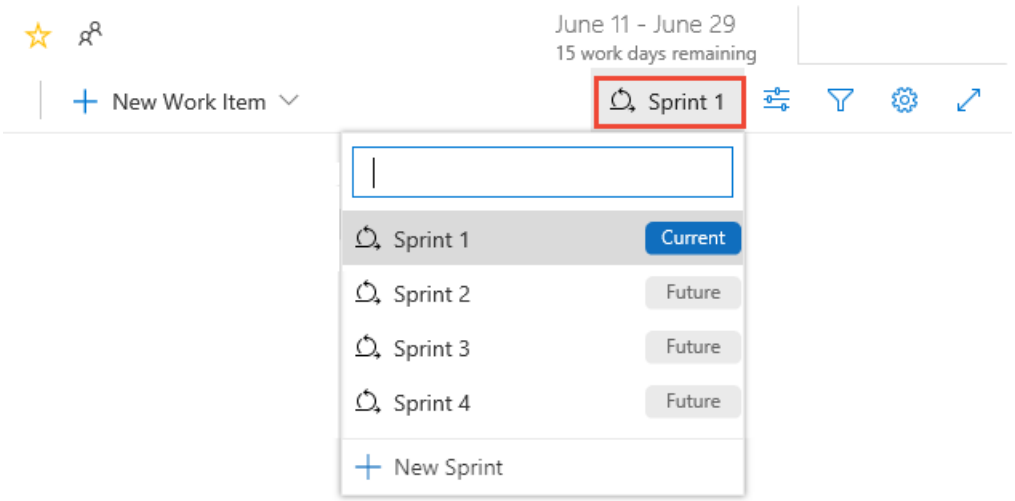
1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



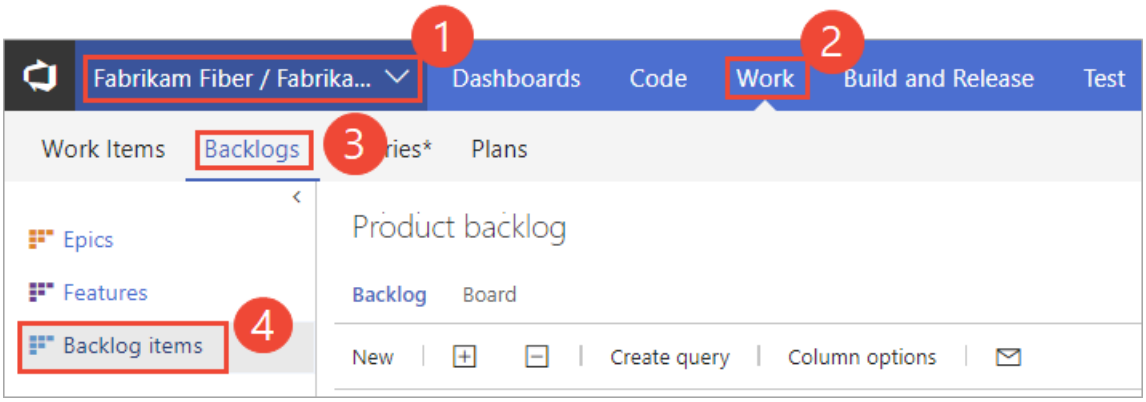
2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.



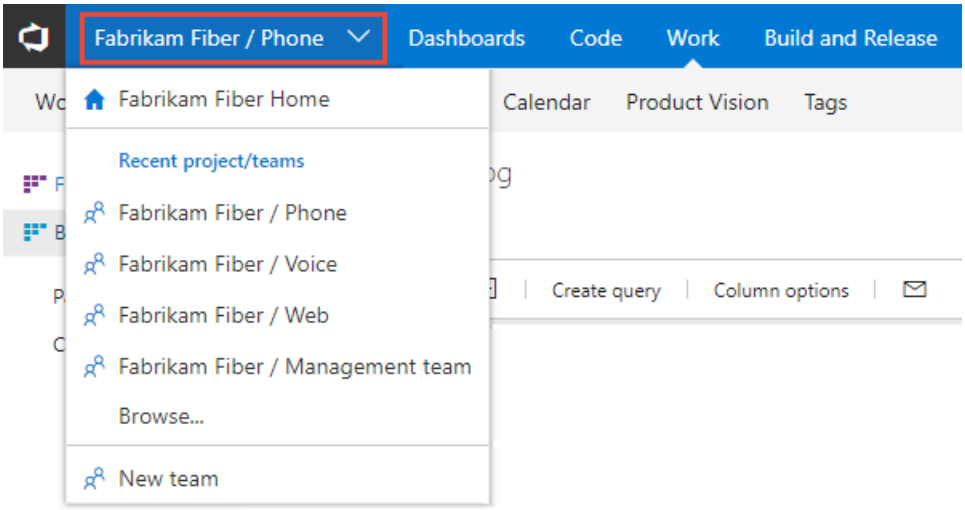
The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing**

iteration. For details, see Define iteration (sprint) paths.

1. From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

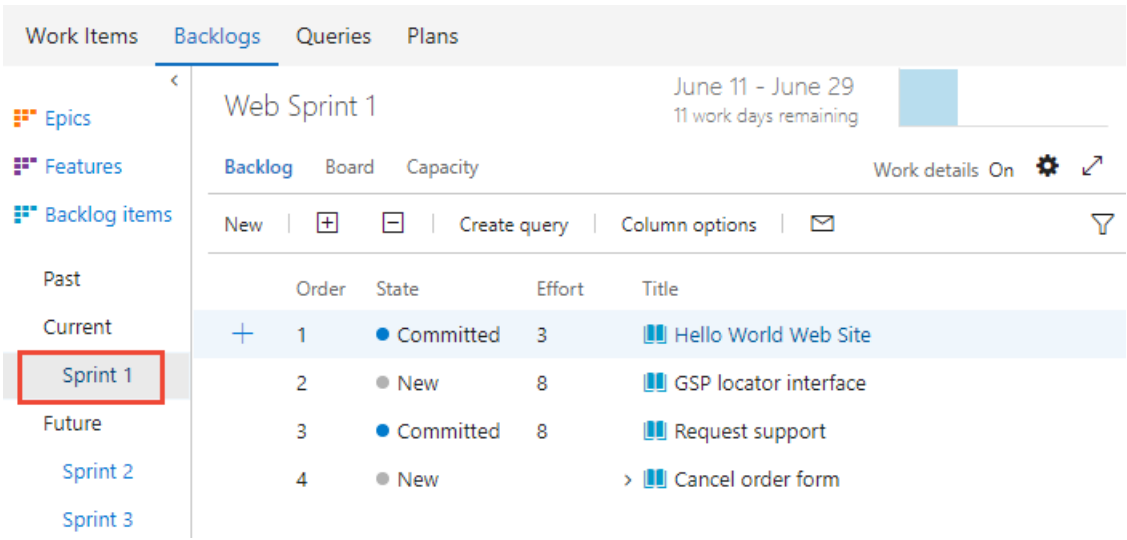


To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. Choose the sprint you want to plan.

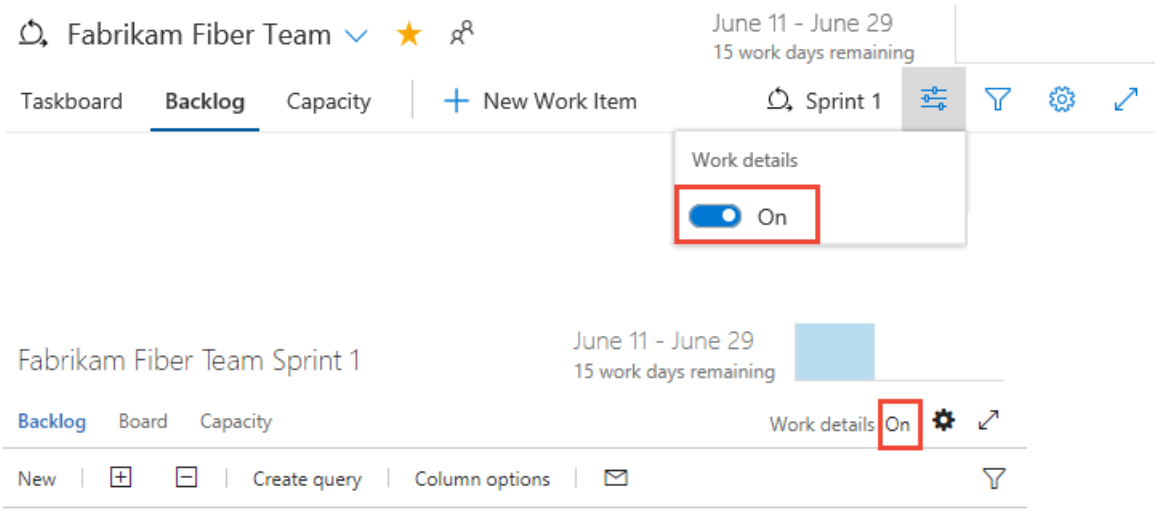


The system lists only those sprints that have been selected for the current team focus. If you don't see the

sprints you want listed, then see Define iteration (sprint) paths.

Check your team capacity

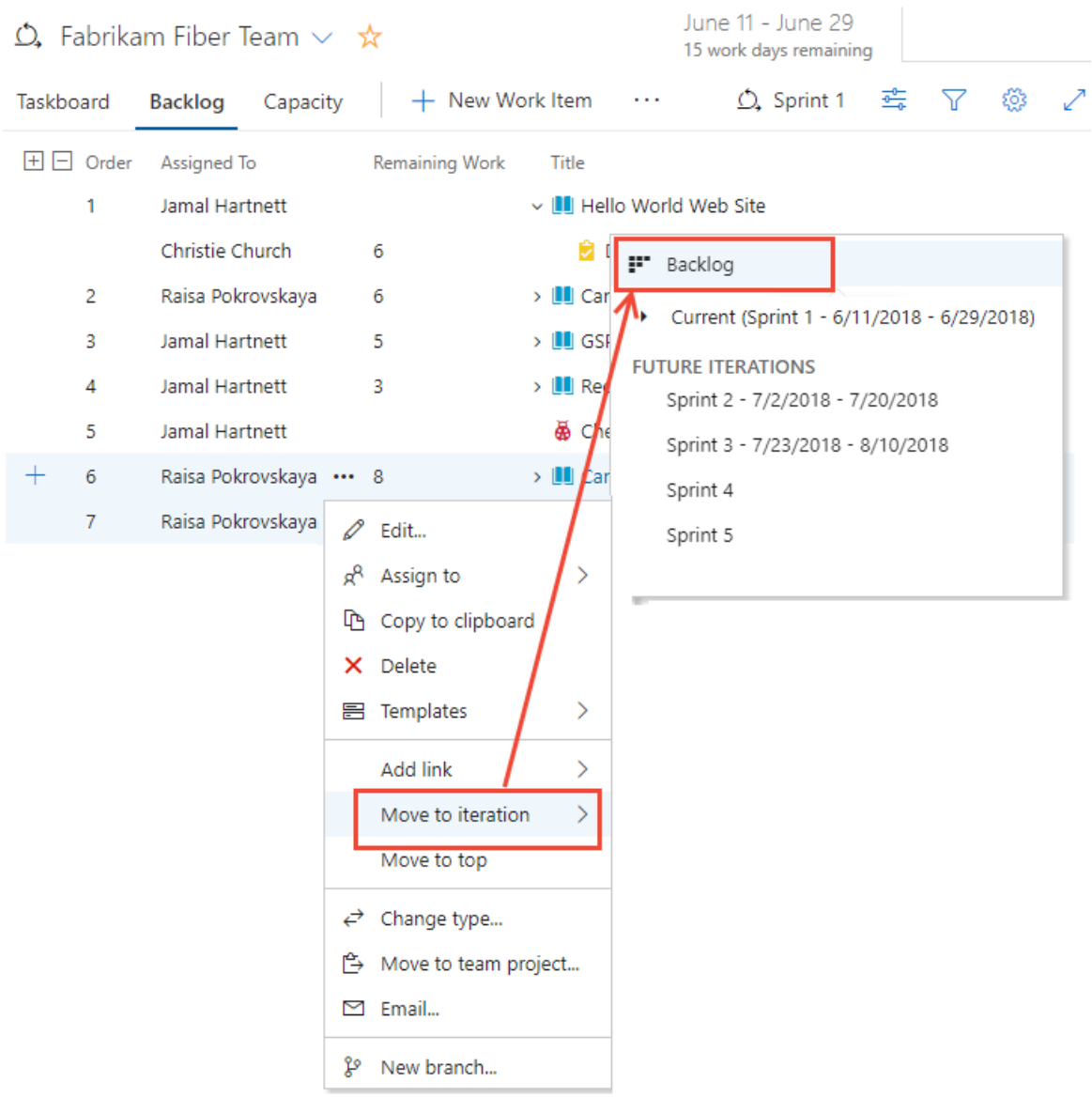
To view capacity charts, you'll want to turn **Work details** on for a sprint.



Move items out of a sprint

Move items from the sprint backlog back to the product backlog if your team is over capacity. This action will reset the Iteration Path to the default set for your team. Or, you can move the item into the next sprint your team will work in. All the tasks that you've defined for that item will move with the backlog items.

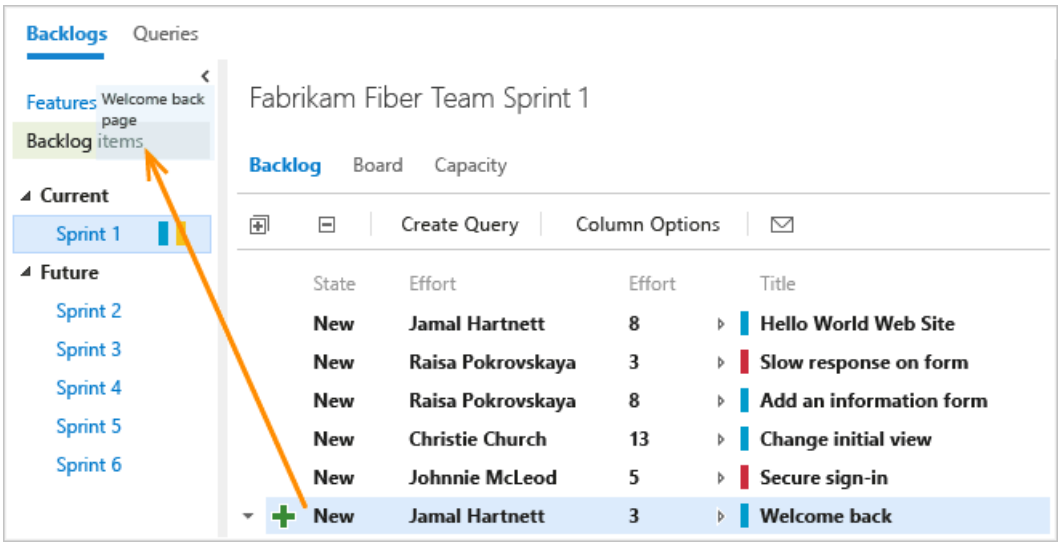
Here we select two items at the bottom of the sprint backlog, open the ··· action icon for one of the items, choose **Move to iteration**, and then select **Backlog**.



TIP

Optionally, you can open the **Planning** pane and drag a work item to the backlog or another sprint which will reassign all child tasks to the same iteration path. See Assign work to a sprint. Also, you can multi-select several items and drag them to the backlog or another sprint. Users with **Stakeholder** access can't drag-and-drop work items.

Here we select the last item in the sprint backlog and drag it to the product backlog.



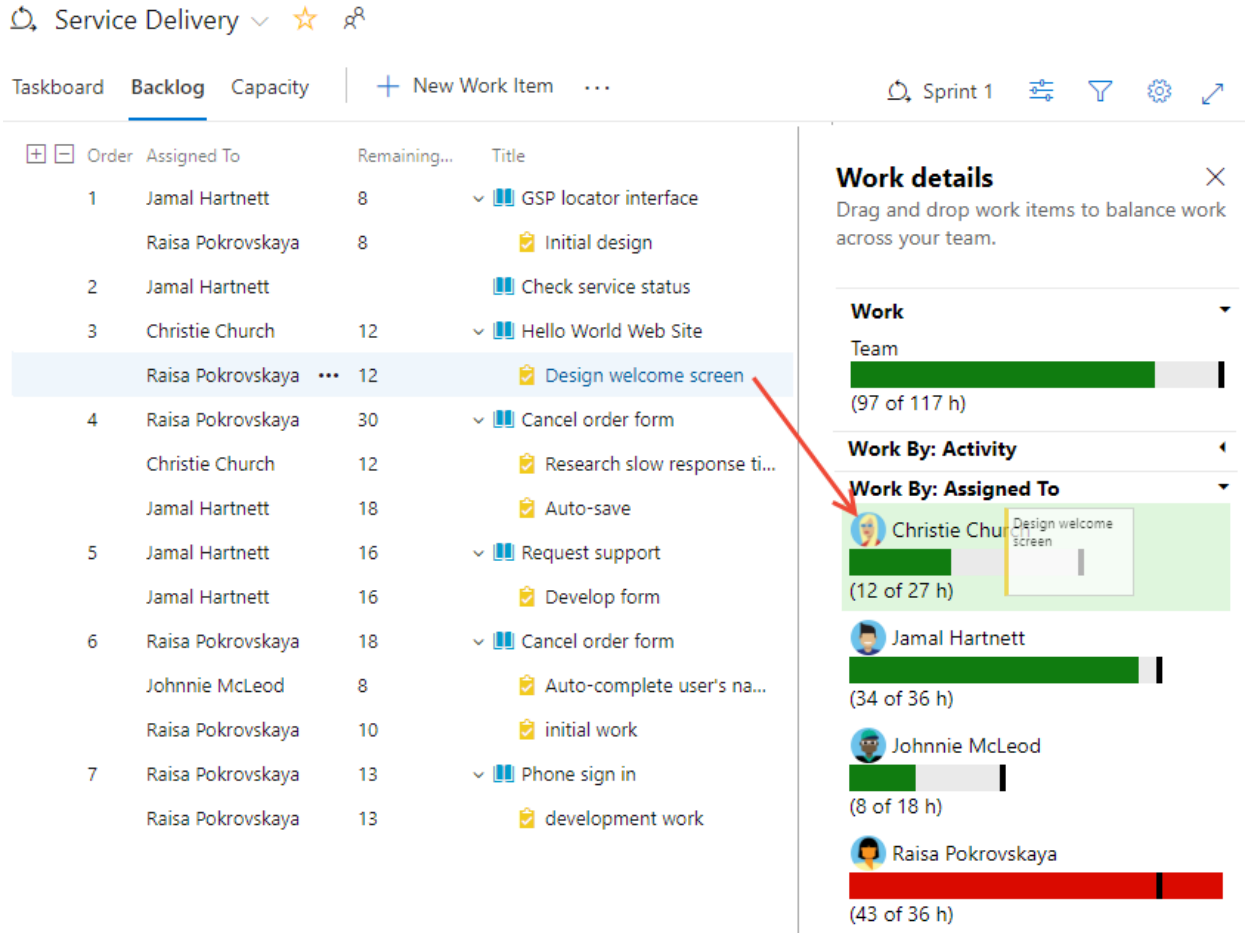
TIP

Dragging a backlog item to the backlog or another sprint reassigns all child tasks to the same iteration path. Also, you can multi-select several items and drag them to the backlog or another sprint. Users with **Stakeholder** access can't drag-and-drop work items.

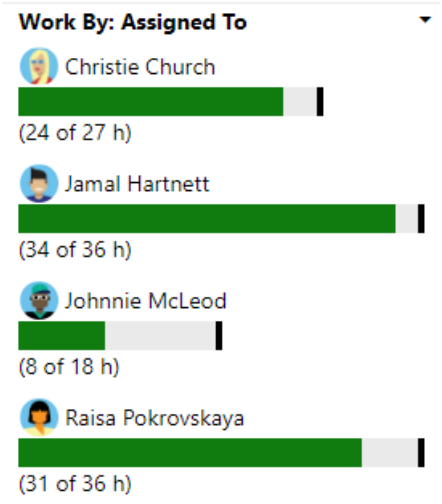
Balance work across the team

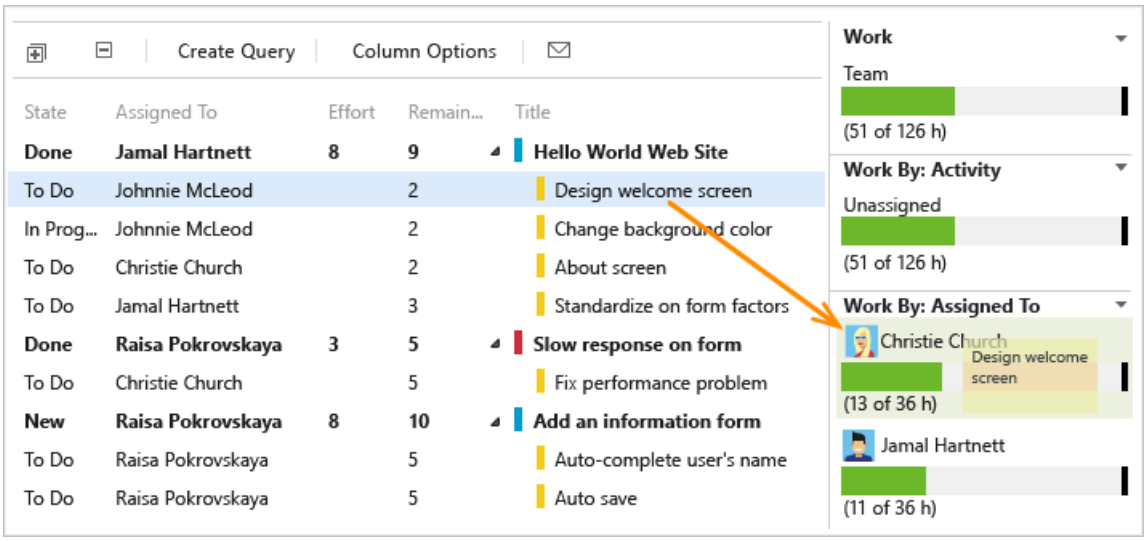
To quickly reassign tasks, drag the task onto the new assignee's capacity bar.

For example, here we reassign work from Raisa Pokrovskaya to Christie Church.



As you reassign tasks, capacity bars automatically update.





Next step

5. Share your sprint plan

5. Share your sprint plan

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

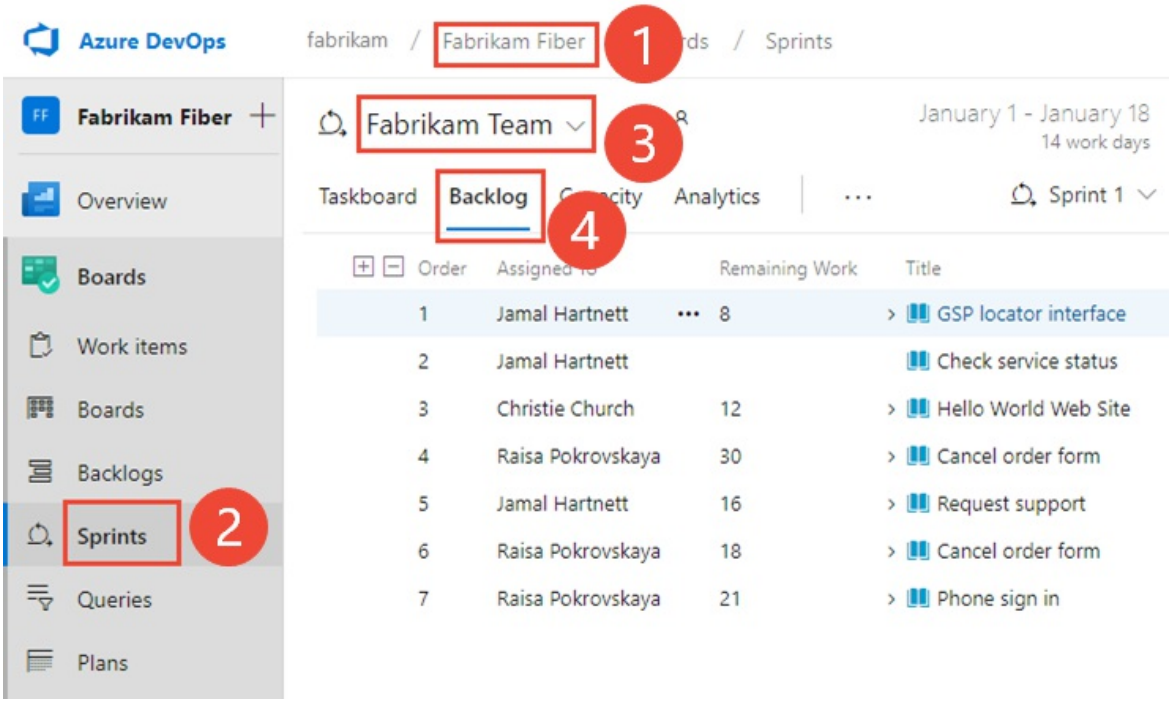
Once you've completed your sprint plan, you can easily share it with other members of your team or organization. This article shows you how to:

- Create a query from your sprint plan
- Email your sprint plan

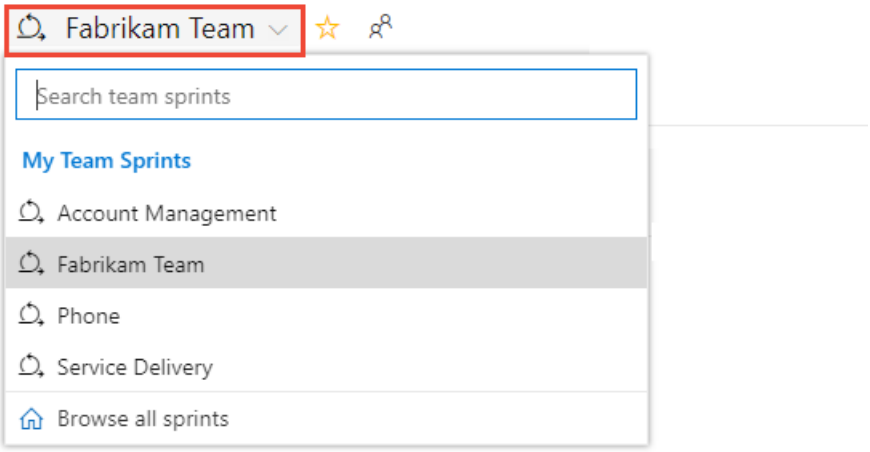
Any stakeholder on your team (someone with permissions to connect to your project) can view your sprint plan. Send them the URL of your sprint backlog page. But also, you can share it with them through email or print a version.

Open a sprint backlog for a team

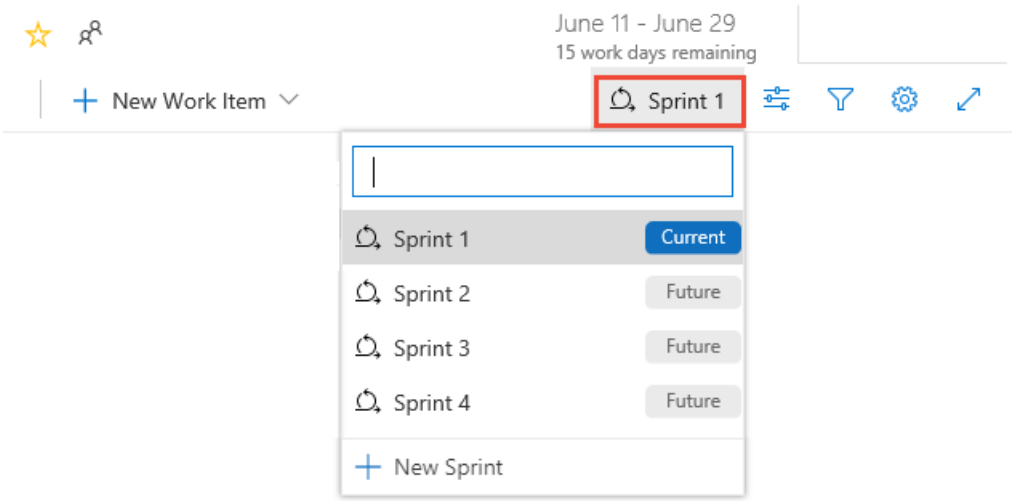
1. From your web browser, open your product backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

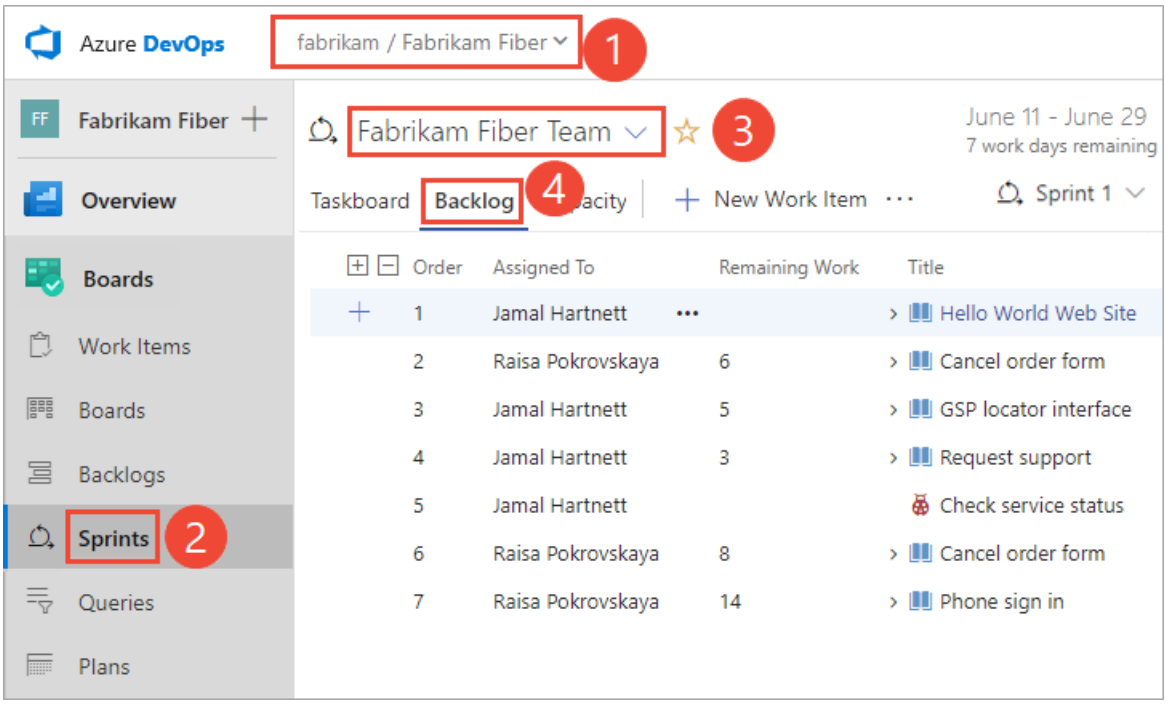


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

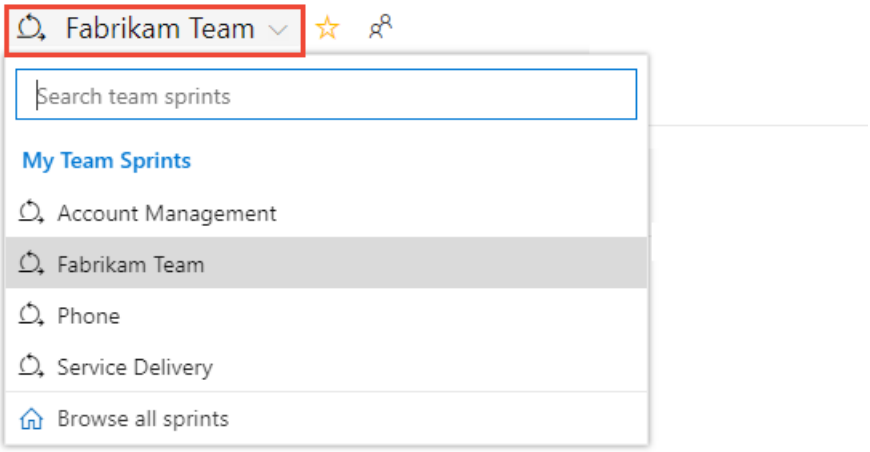


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

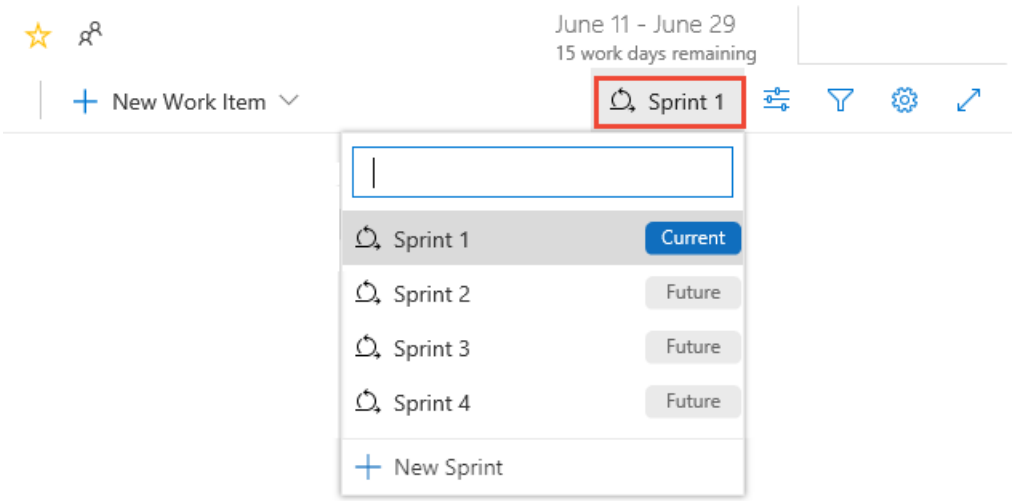
1. From your web browser, open your product backlog. (1) Check that you've selected the right project, (2) choose **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose **Backlog**.

2. 

To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



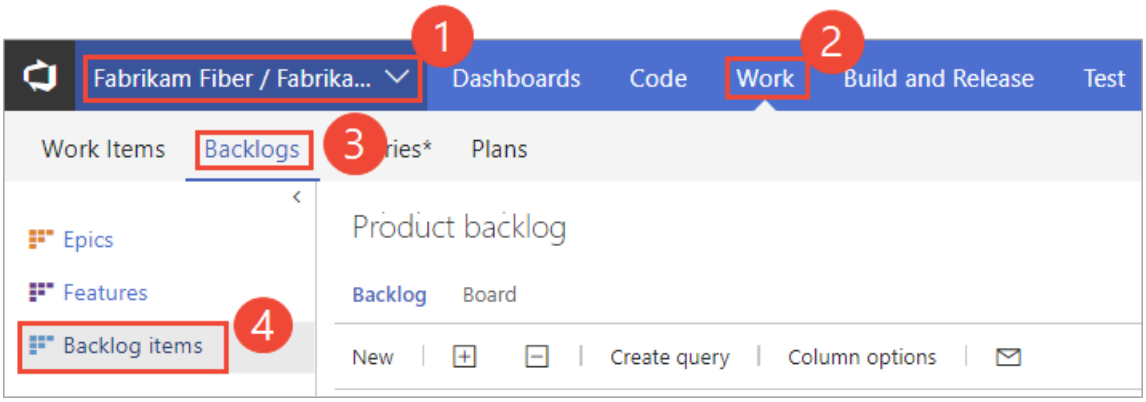
3. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.



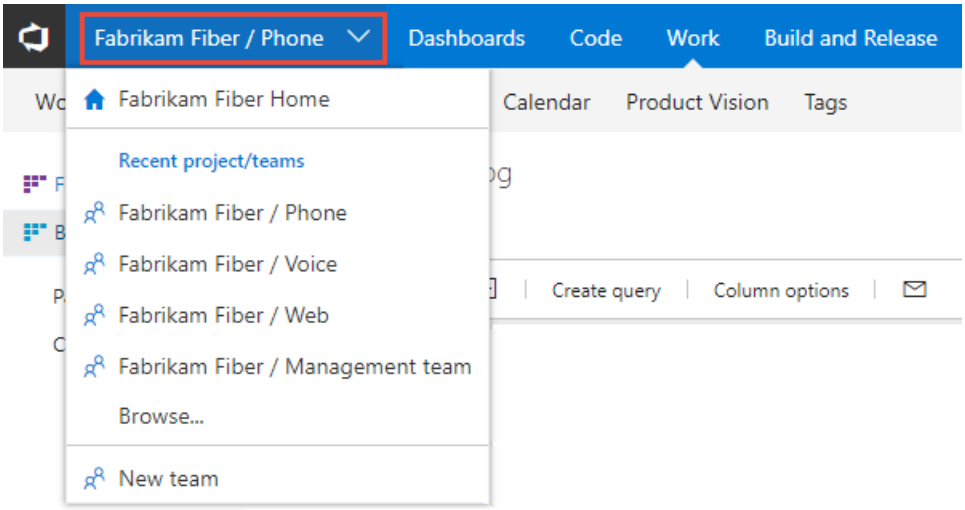
The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing**

iteration. For details, see Define iteration (sprint) paths.

1. From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

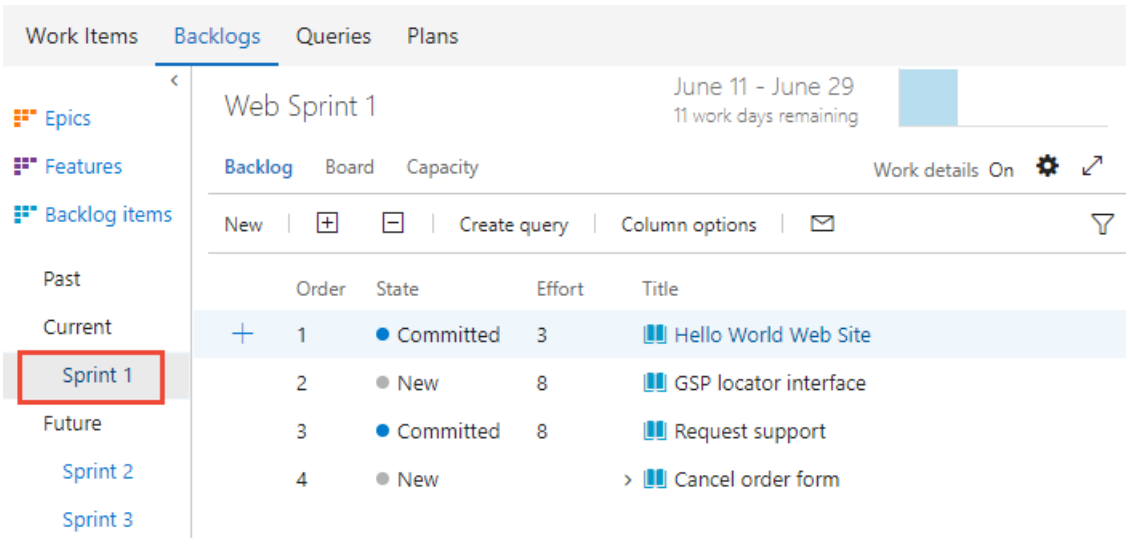


To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. Choose the sprint you want to plan.

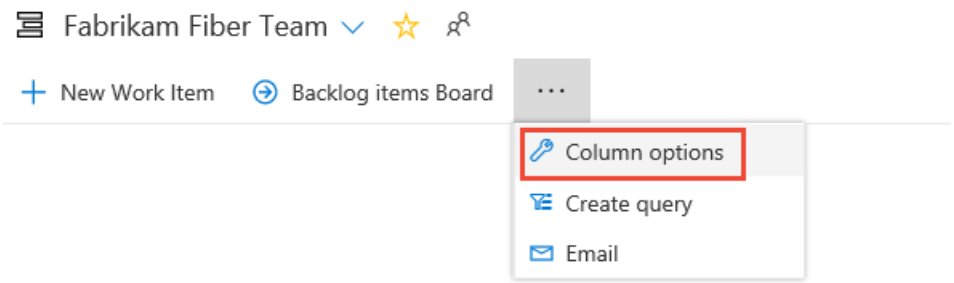


The system lists only those sprints that have been selected for the current team focus. For more

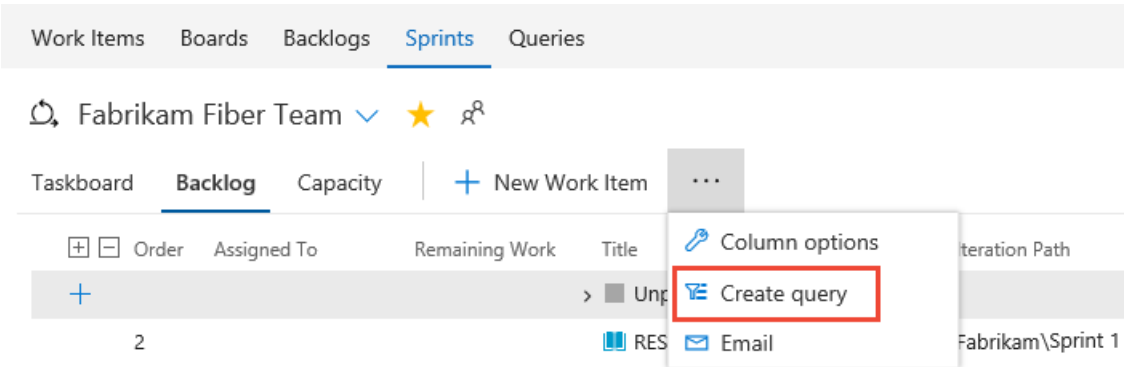
information, see Define iteration (sprint) paths.

Create query for your sprint plan

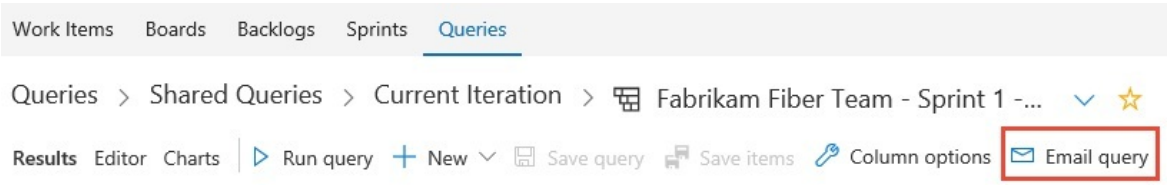
1. (Optional) To choose which columns should display and in what order, choose the ··· actions icon and select **Column options**. To learn more, see Change column options.



2. To email your sprint plan, create and save the query for the sprint backlog.



3. Then, open the query and choose the email icon.

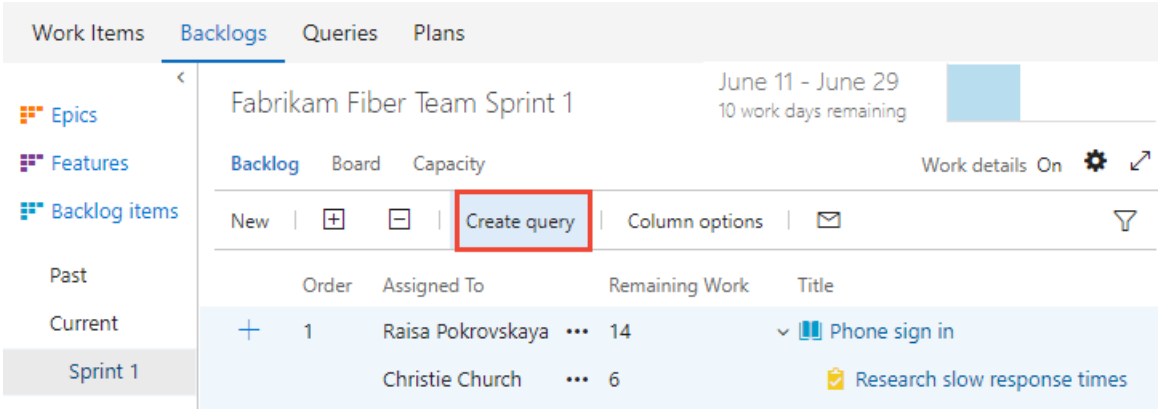


4. In the form that appears, enter the name(s) of valid users (ones who have access to the project).

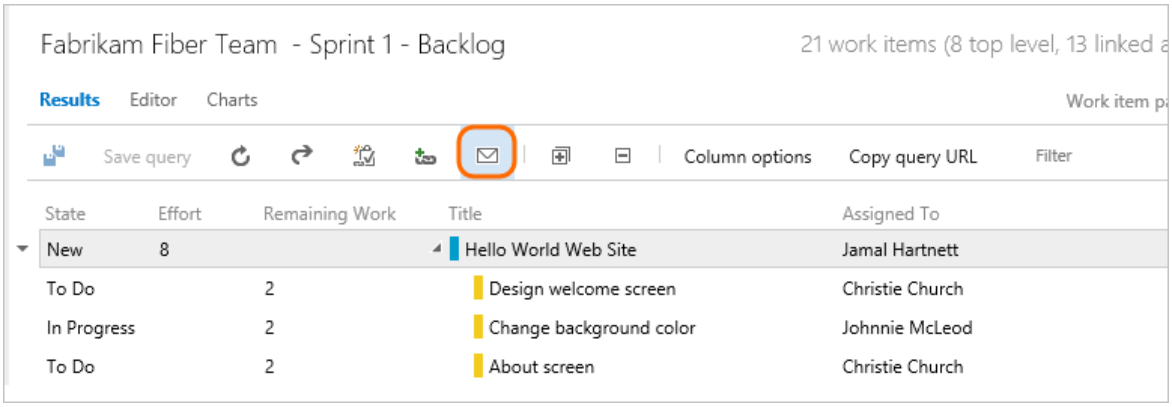
> **IMPORTANT**
> You can only send the email to individual address for a project member that is recognized by the system. Adding a team group or security group to the to line isn't supported. If you add an email account that the system doesn't recognize, you receive a message that one or more recipients of your email don't have permissions to read the mailed work items.

1. (Optional) To choose which columns should display and in what order, choose **Column options**. To learn more, see Change column options.

2. To email the sprint plan, create and save the query for the sprint backlog.



3. Then, open the query and choose the email icon.



4. In the form that appears, enter the name(s) of valid users (ones who have access to the project).

> **IMPORTANT**
> You can only send the email to individual address for a project member that is recognized by the system. Adding a team group or security group to the to line isn't supported. If you add an email account that the system doesn't recognize, you receive a message that one or more recipients of your email don't have permissions to read the mailed work items.

Or, you can select all the items in the list, choose **Copy as HTML**, and paste the formatted list into an email form or Word document. See Copy a list of work items.

Next step

6. Update the taskboard

Related articles

- Email or print work items
- Share information in work items and social tools

6. Update and monitor your Taskboard

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Once you have your sprint plan in place, you'll execute that plan during the sprint. In your daily Scrum meetings, your team can view progress made to backlog items and tasks from the sprint **Taskboard**.

Your **Taskboard** provides a visualization of flow and status of each sprint task. With it, you can focus on the status of backlog items and work assigned to each team member. It also summarizes the total amount of **Remaining Work** to complete for a task and within a column.

In this article you'll learn how to:

- Open the sprint **Taskboard** for your team
- Customize your **Taskboard**
- Use your **Taskboard** to review progress during daily scrum meetings
- Filter and group work items on your **Taskboard**
- Update the status of tasks through drag-and-drop
- Update remaining work
- Close out a sprint

If you haven't yet added tasks to your sprint backlog, do that now.

> **NOTE**
> Your **Taskboard** is one of two types of boards available to you. For an overview of the features supported on each backlog and board, see Backlogs, boards, and plans.

Prerequisites

- Connect to a project. If you don't have a project yet, create one.
- Get added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To add work items and exercise all board features, you must be granted **Basic** access or higher. Users granted **Stakeholder** accesses have limited access to features. For details, see Stakeholder access quick reference.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.

> **NOTE**
> Users assigned **Stakeholder** access can't exercise these **Taskboard** features: update fields displayed on cards or use the **Planning** pane to change the sprint assignment.

NOTE

Users with **Stakeholder** access can't exercise these **Taskboard** features: add tasks, update fields displayed on cards, drag-and-drop tasks to update status, or use the **Planning** pane to change the sprint assignment.

Open the sprint Taskboard for your team

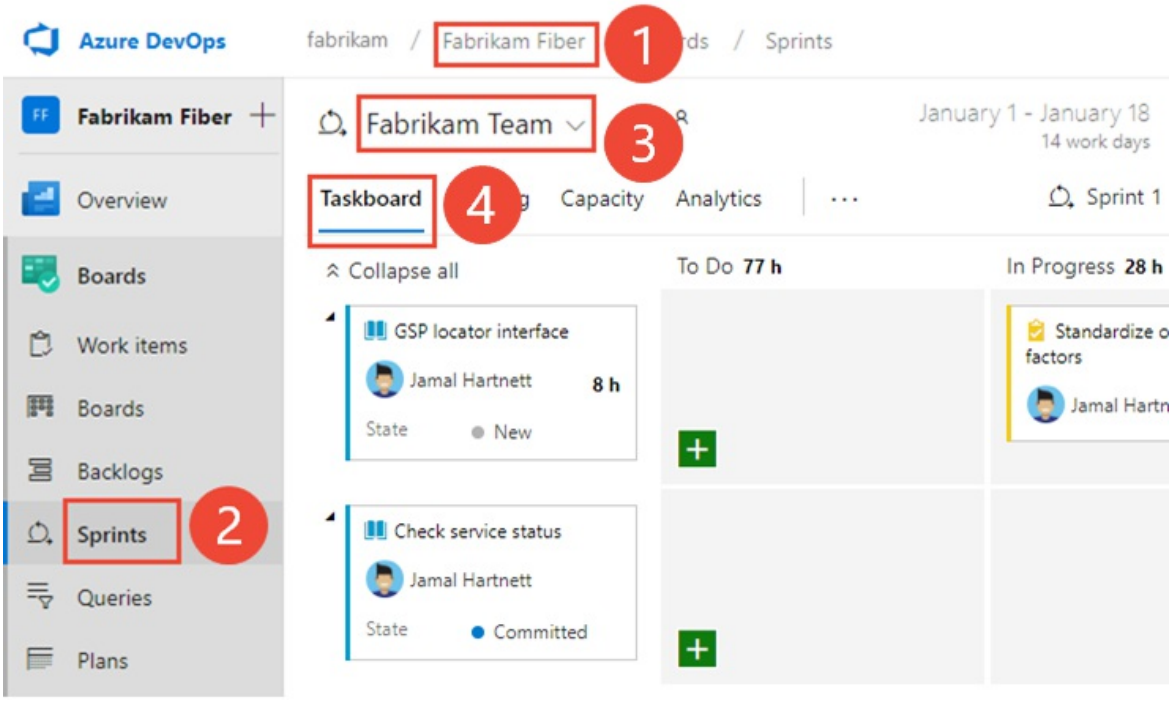
1. From your web browser, open the sprint backlog for your team.

   a. Check that you've selected the right **Project**

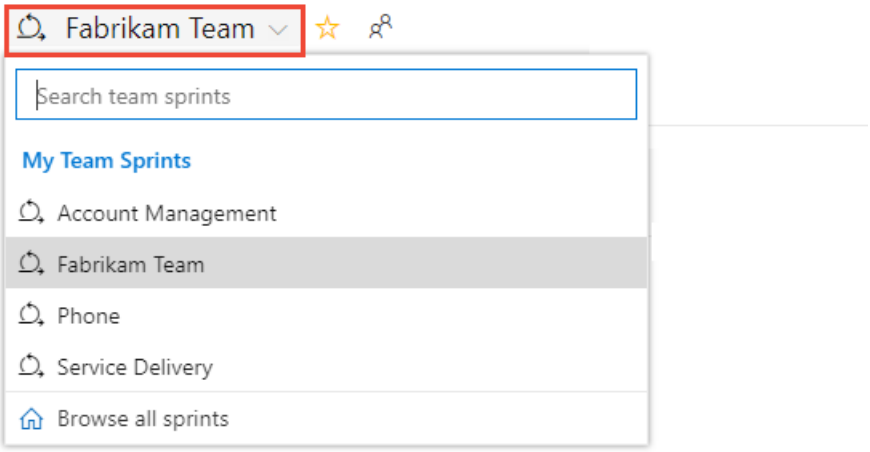
   b. Choose **Boards > Sprints**

   c. Select the correct **Team** from the team selector menu

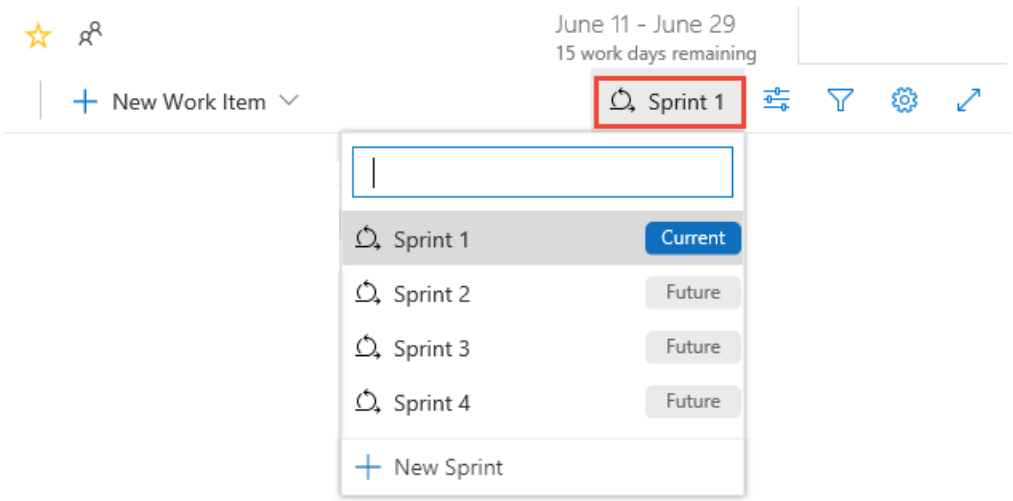
   d. Choose **Taskboard**



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

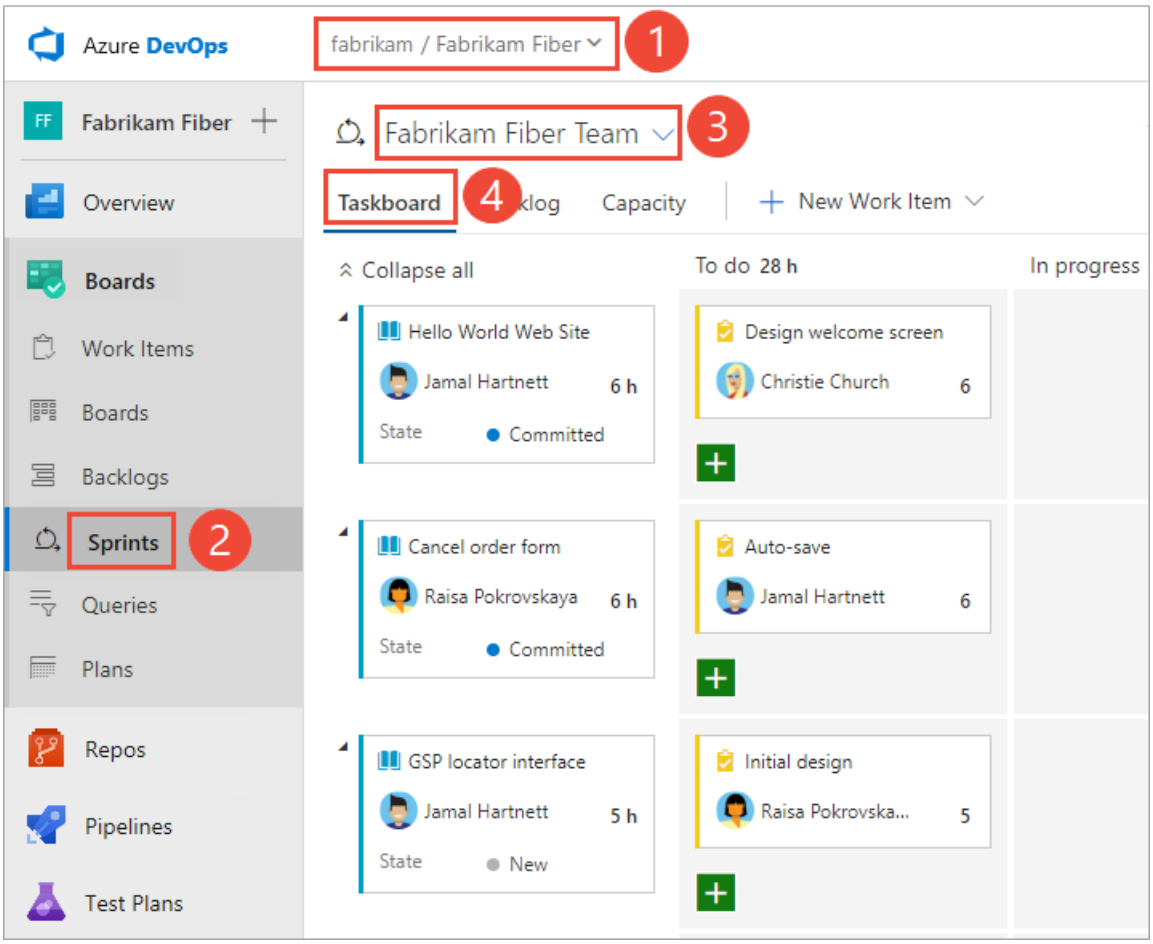


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

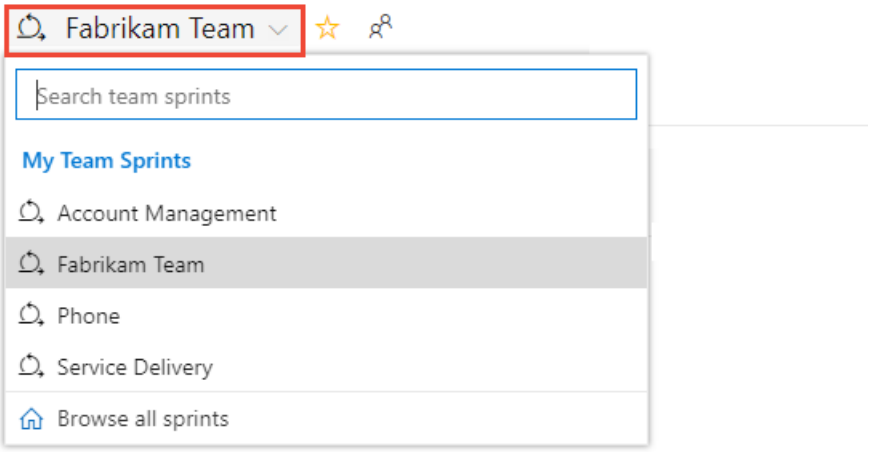


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

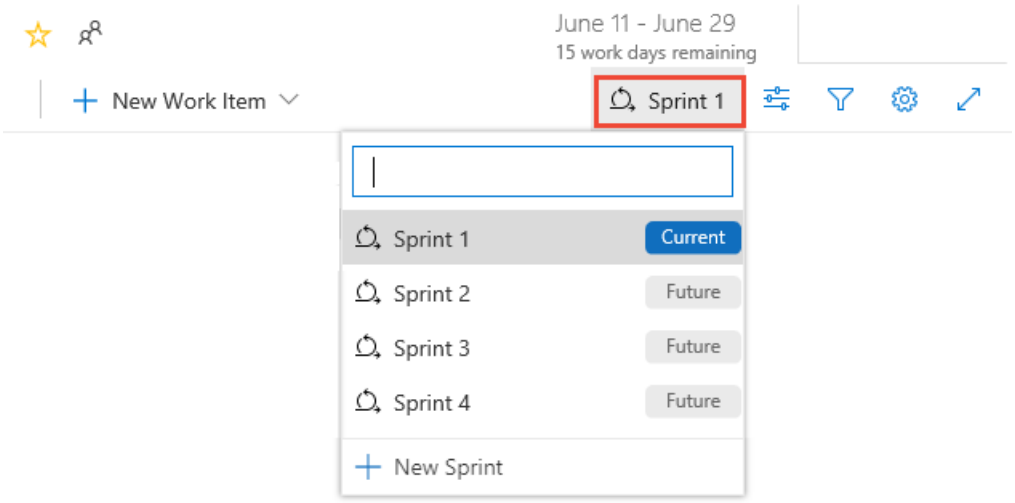
1. From your web browser, open the sprint backlog for your team. (1) Check that you've selected the right project, (2) choose **Boards>Sprints**, (3) select the correct team from the team selector menu, and lastly (4), choose (4) **Taskboard**.



To choose another team, open the selector and select a different team or choose the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

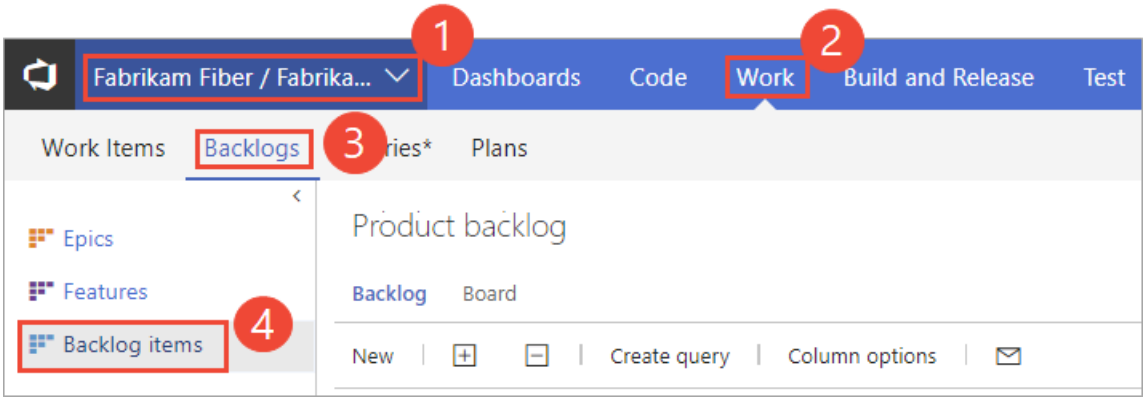


2. To choose a different sprint than the one shown, open the sprint selector and choose the sprint you want.

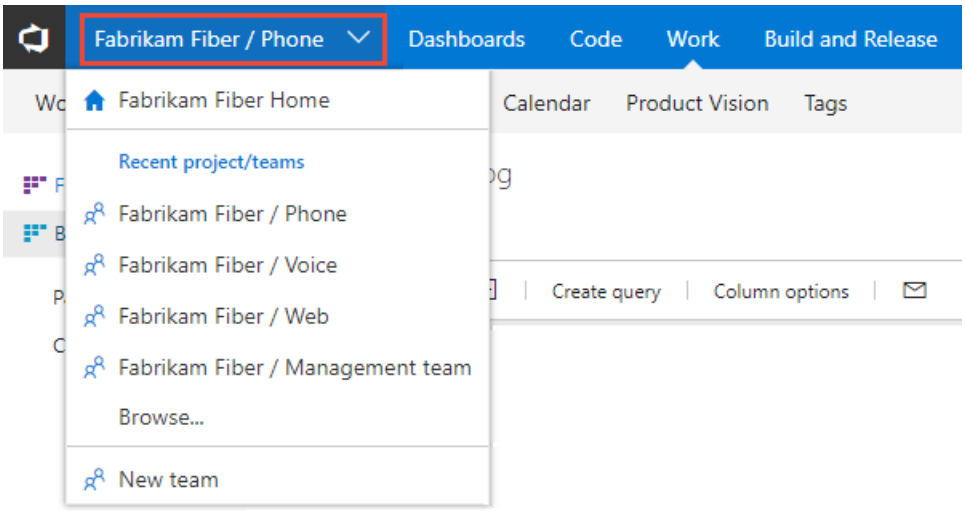


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then choose **New Sprint** from the menu, and then choose **Select existing iteration**. For details, see Define iteration (sprint) paths.

1. From your web browser, open your team's product backlog. (1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

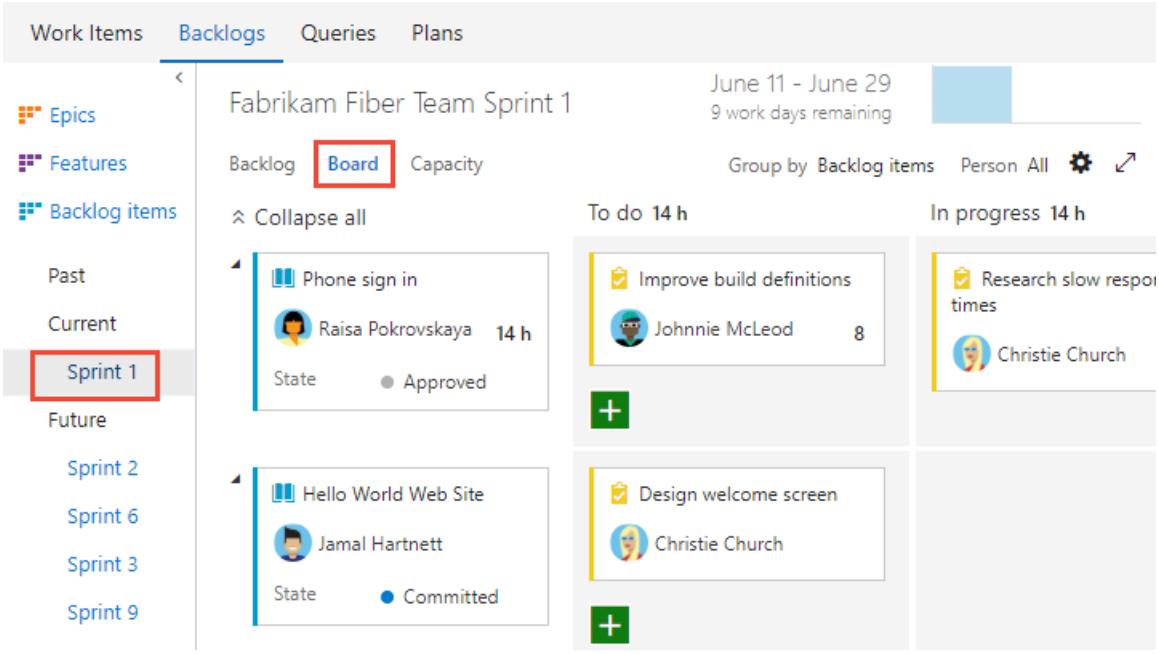


To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

2. Choose the sprint you want to plan, and then choose **Board**.



The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then see Define iteration (sprint) paths.

Customize the Taskboard

Each team can customize their **Taskboard** in the following ways:

- Add or rename columns.
- Customize cards to show another fields or change card color based on specified field criteria.
- Show bugs on the Taskboard. Your team can choose to manage bugs similar to product backlog items, as shown in this article, or manage them similar to tasks. When you track bugs similar to tasks, they'll show up on your sprint backlogs and **Taskboards** at the same level as tasks.

- Customize cards that appear on the **Taskboard** to show other fields.
- Show bugs on the Taskboard. Your team can choose to manage bugs similar to product backlog items, as shown in this article, or manage them similar to tasks. When you track bugs similar to tasks, they'll show up on your sprint backlogs and **Taskboards** at the same level as tasks.

Unlike the Kanban board for a product backlog, you can't add another columns to the **Taskboard** via a team configuration setting. Instead, you need to modify the workflow definitions for the task work item type used by the project. Workflow customizations update the columns for the **Taskboard** for all teams within the project.

An administrator can customize the **Taskboard** for all teams in the following ways:

- Add a custom workflow state to the task WIT for a process
- Add a custom work item type to the Taskboard for a process

An administrator can customize the **Taskboard** for all teams in the following ways based on the process model selected for the project:

Inherited process model:

- Add a custom workflow state to the task WIT for a process
- Add a custom work item type to the Taskboard for a process

On-prem XML process model:

- Add a custom workflow state to the task WIT for a process
- Add a custom work item type to the Taskboard for a process
-

An administrator can customize the **Taskboard** for all teams in the following ways:

- Modify the workflow for the task WIT definition.
- Add a work item type to a backlog or board.

Review progress in daily scrum meetings

During your daily Scrum, you can filter your **Taskboard** to help focus on items of interest.

- Group by Backlog items or Group by stories to monitor progress of your product backlog items, stories, requirements, or bugs.
- Group by People when you want to monitor progress of individual team members.

> **NOTE**
> Your Taskboard automatically refreshes when changes occur. There isn't any live updates control, it simply happens in the background. As other team members move or reorder cards on the taskboard, the Taskboard automatically updates with these changes. You don't need to press F5 to see the latest changes.

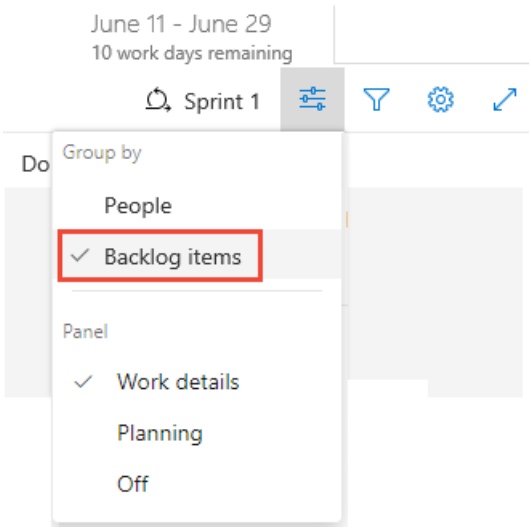
Use the Person filter when you want to focus on work assigned to individual team members.

> **TIP**
> If you're seeing tasks that don't belong to your team, check that you've selected the correct team.

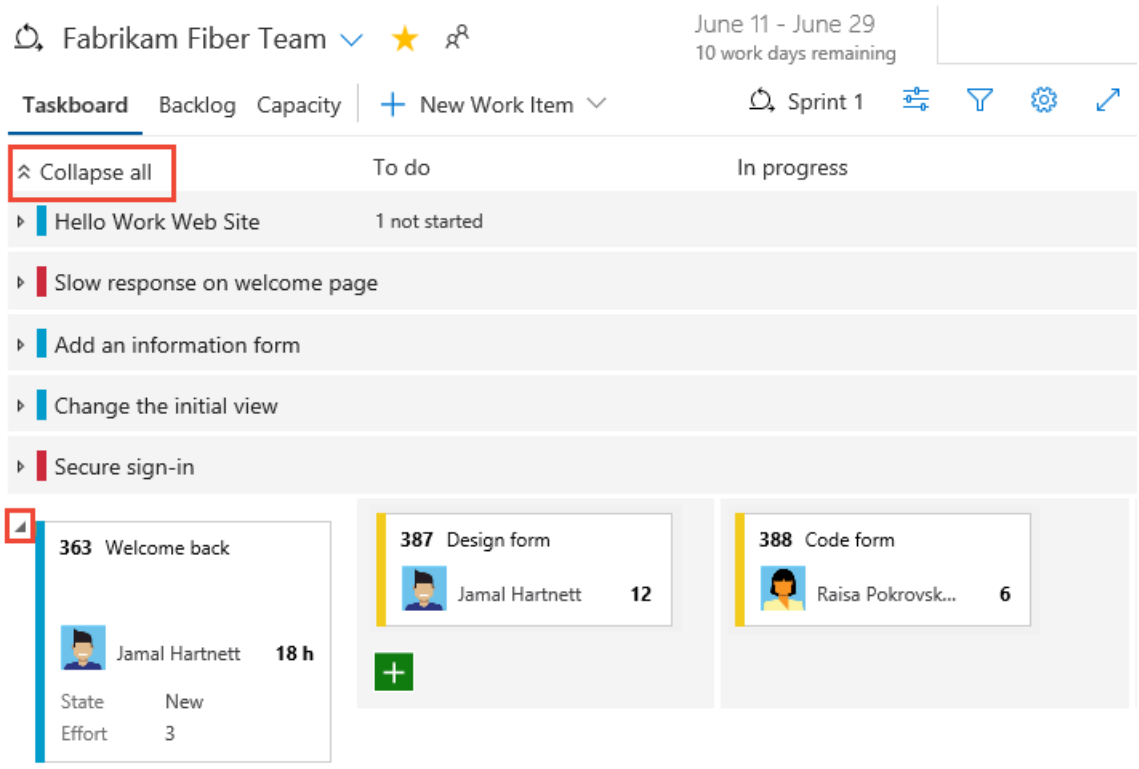
Show progress on items

With this view, you can quickly see which items are nearing completion and which have yet to be started.

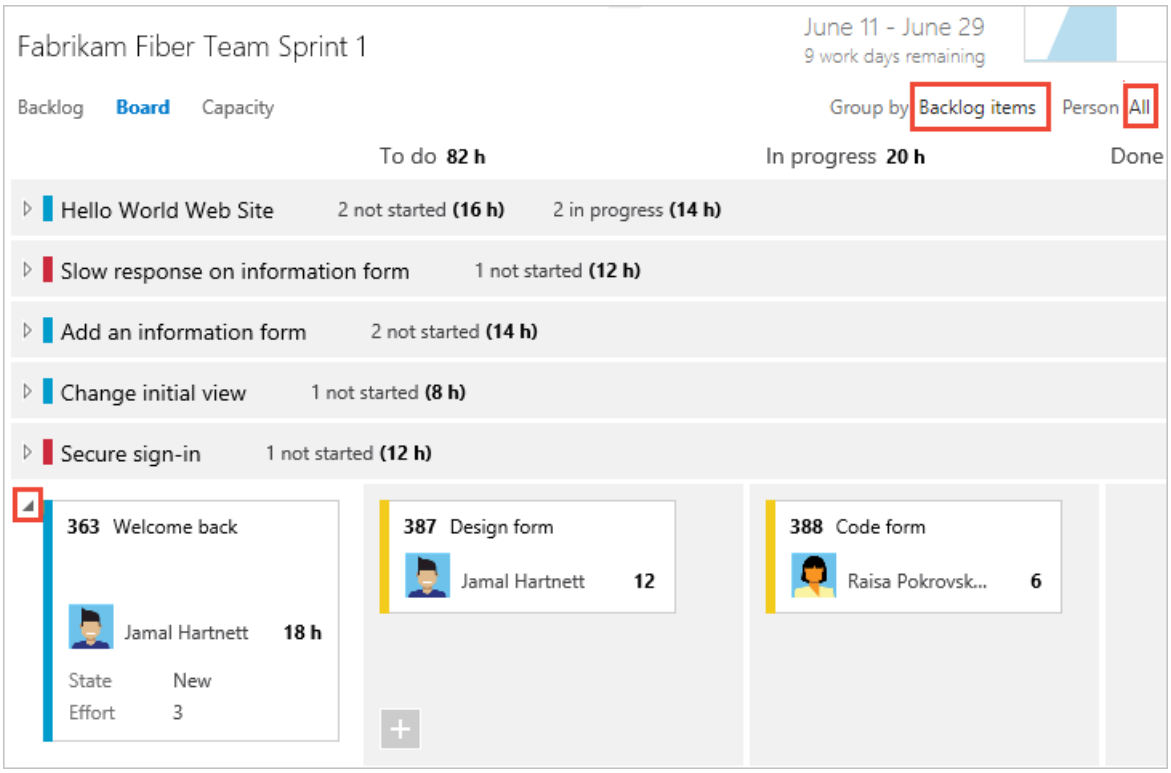
1. To show cards based on their backlog-to-task groupings, choose the view options icon and select **Backlog items** (for Scrum), **Stories** (for Agile), and **Requirements** (for CMMI).



2. You can **Collapse All** or **Expand All** rows, and selectively expand ▷ and collapse ◢ a row to focus on a particular item and its tasks.



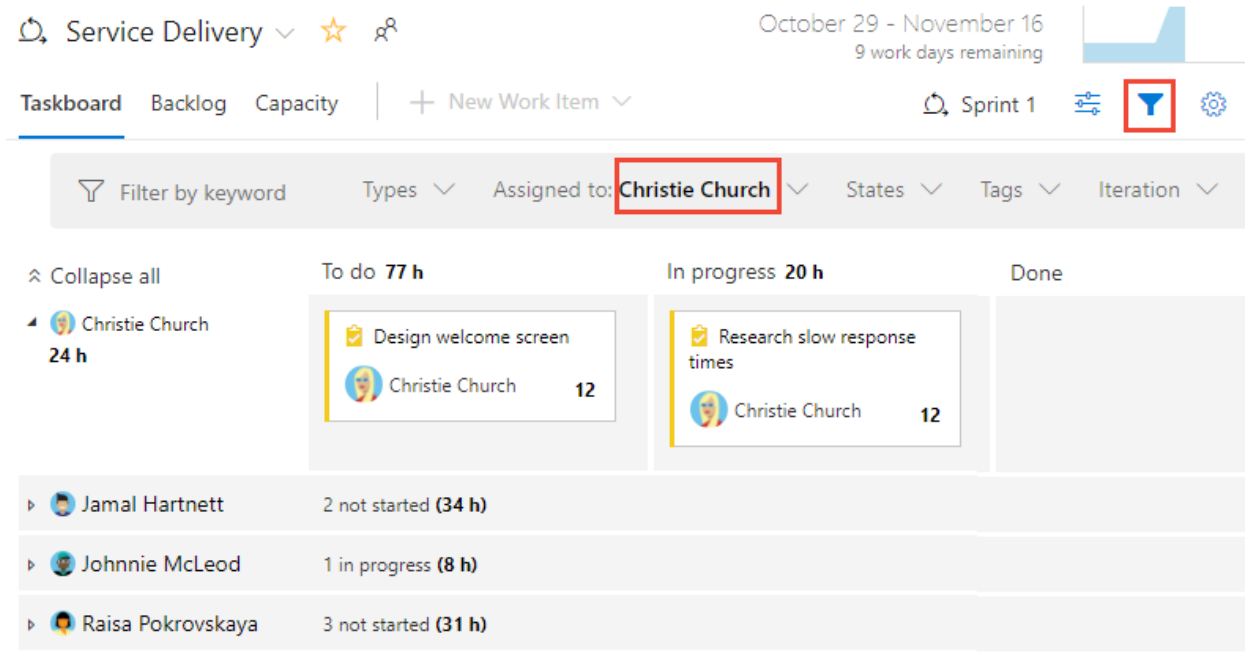
You can expand ▷ and collapse ◢ a row to focus on a particular item and its tasks.



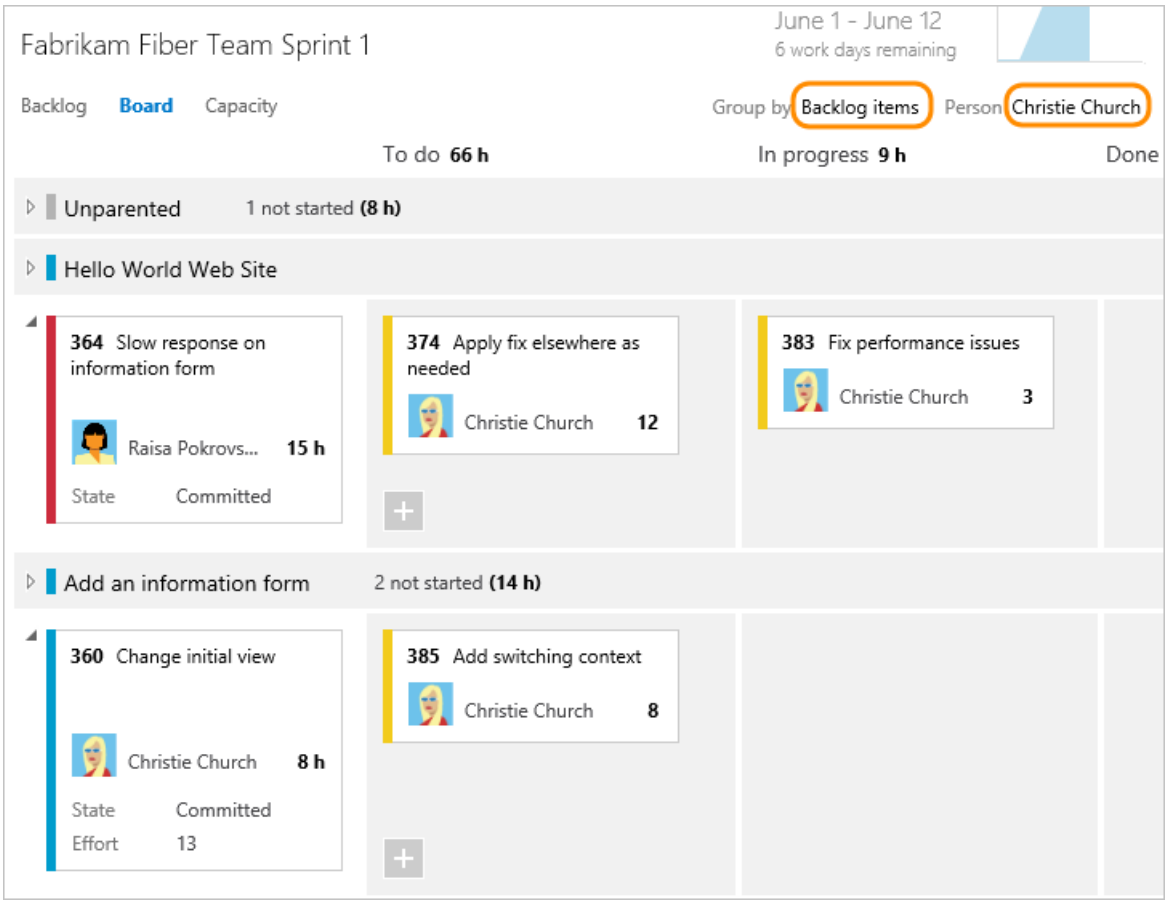
Show a team member's progress

With this view, you can focus on the work completed and the work remaining for each individual team member. You can quickly see who may need help with completing their sprint tasks. This view shows items and tasks assigned to the selected team member.

To filter on the tasks for a specific team member, choose the filter icon, and then select their name from the **Assigned to** filter box.



Choose the **Group by People** option, and then select a specific team member, or **All**.

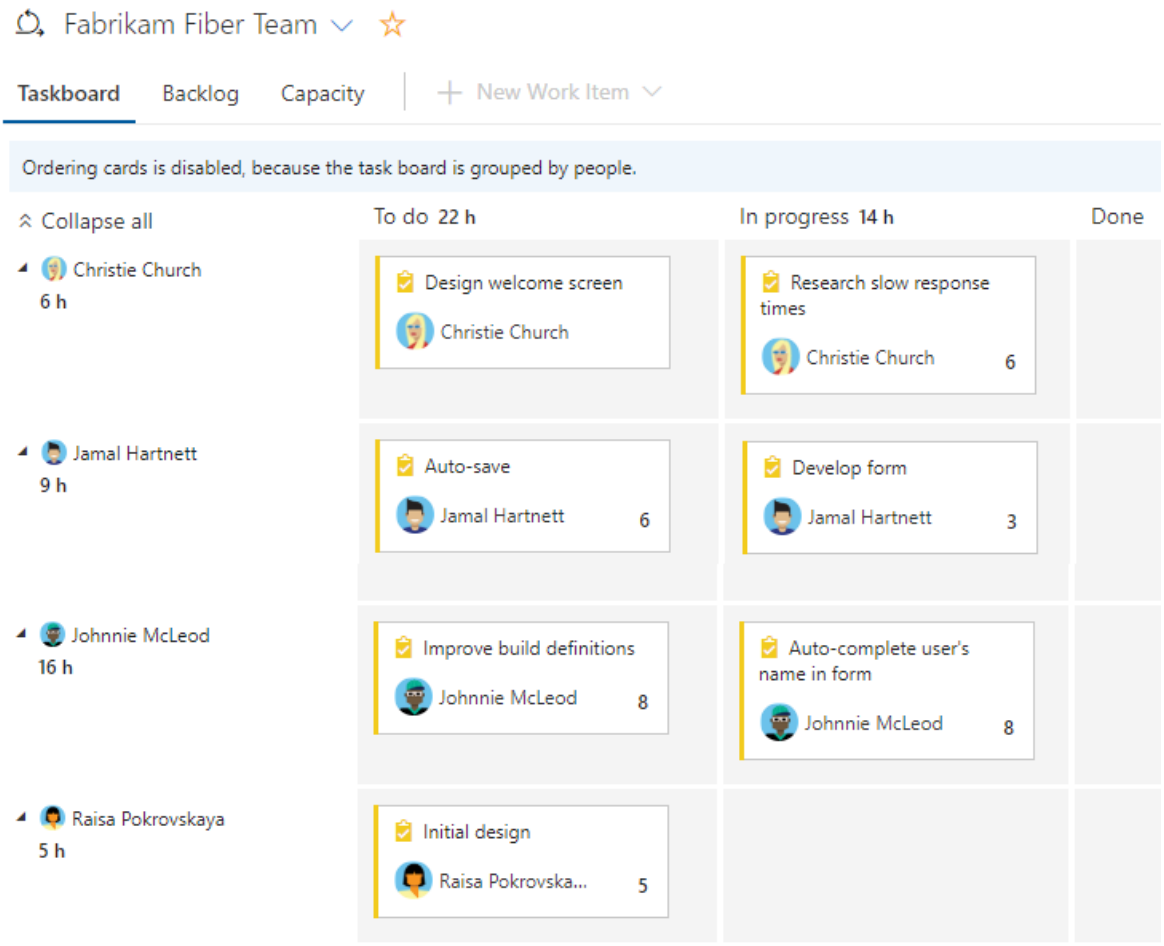


Group tasks by team members

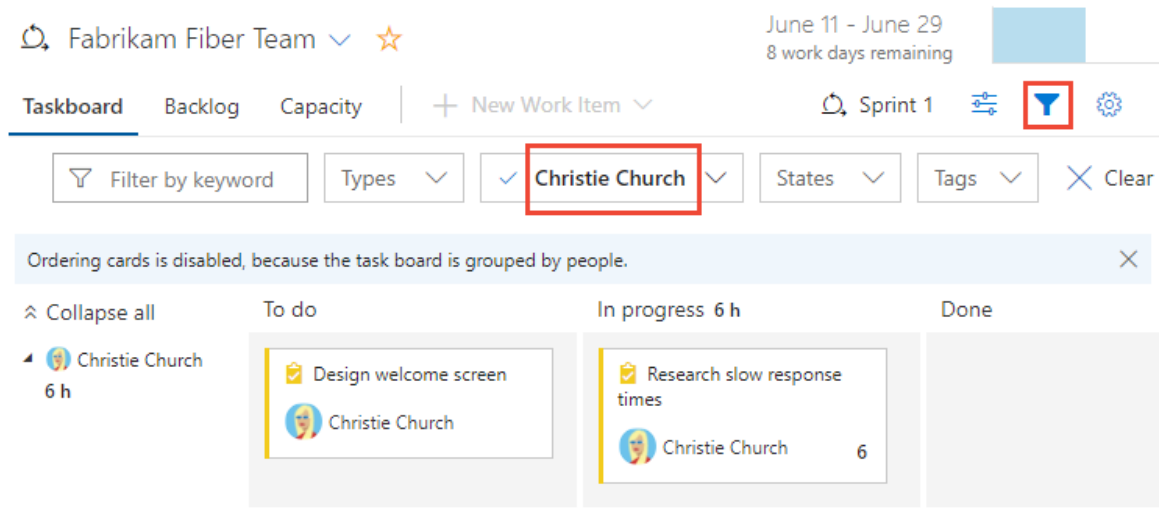
With this view, you can quickly see all the tasks associated with each team member. Backlog items don't appear in this view, only the tasks associated with each individual.

1. Choose the view options icon and select **People**.

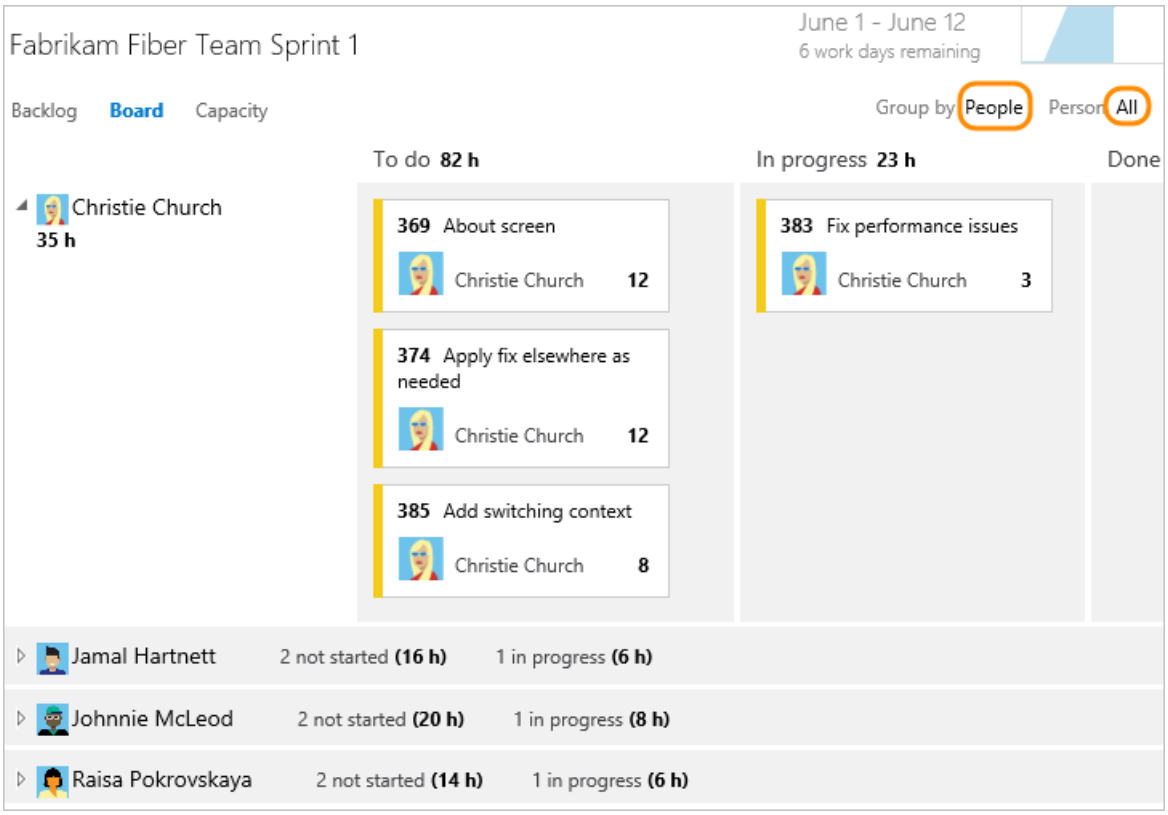
   Only those team members with tasks assigned to them are listed. All their tasks are shown as cards under their column state.



2. To filter on the tasks for a specific team member, choose **Filter** , and then select their name from the **Assigned to** filter box. To learn more, see Interactively filter backlogs, boards, queries, and plans.



Choose the **Group by People** option, and then select a specific team member, or **All**.

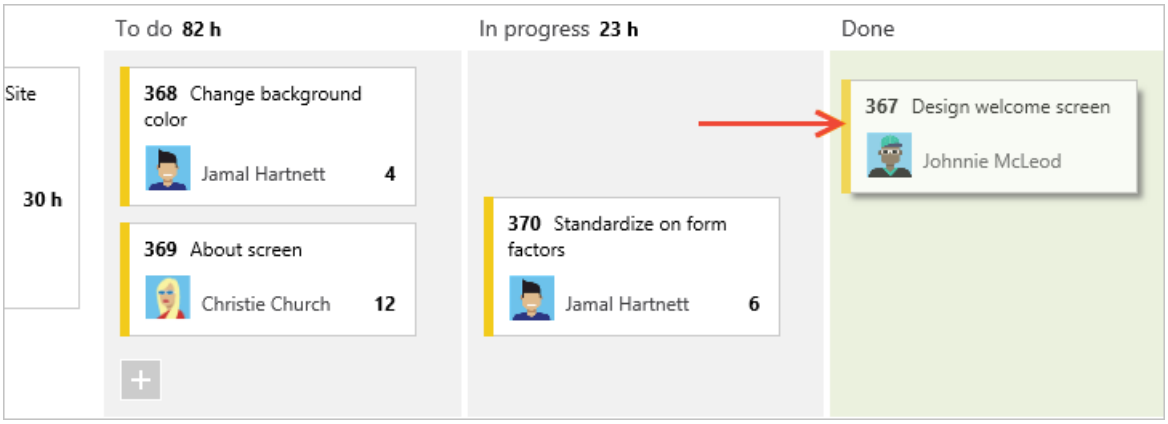


Update tasks during the sprint cycle

The **Taskboard** makes quick work of updating both task status and remaining work.

Update a task's status

Drag tasks to a downstream column to reflect if they are in progress or completed.

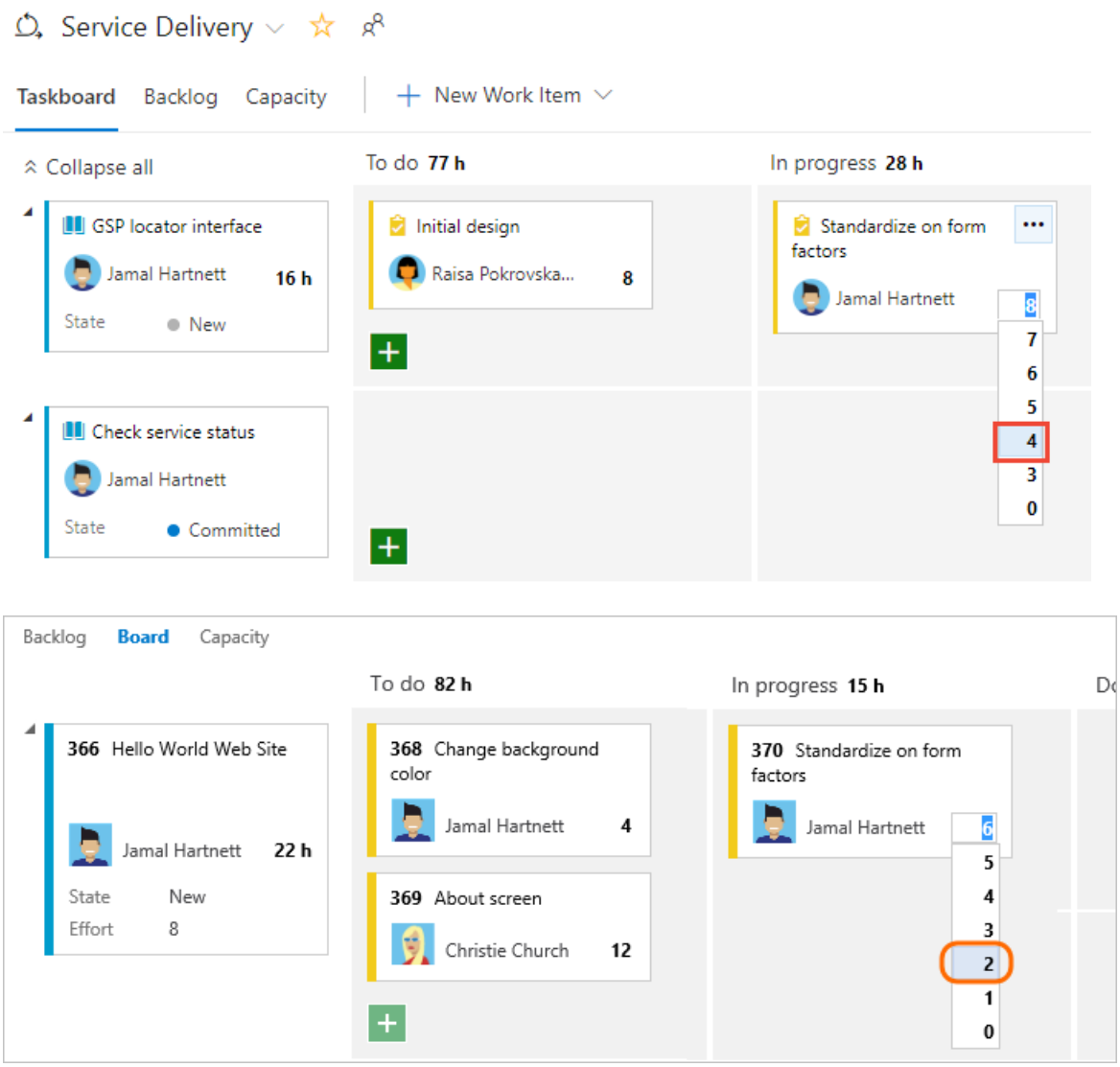


When you move a task to the **Done** or **Completed** column, the system automatically updates the **Remaining Work** field to 0 in all processes, except CMMI. If you discover more work is remaining, change the State back to **In progress** or **To do**, and enter a value for the **Remaining Work**.

Update Remaining Work

Updating **Remaining Work**, preferably before the daily Scrum meeting, helps the team stay informed of the progress being made. It also ensures a smoother burndown chart.

Each team member can review the tasks they've worked on and estimate the work remaining. If they've discovered that it's taking longer than expected to complete, they should increase the **Remaining Work** for the task. **Remaining Work** should always reflect exactly how much work the team member estimates is remaining to complete the task.



Close out a sprint and update your Taskboard

At the end of the sprint, you'll want to complete these final tasks:

- Zero out *Remaining Work* of all completed tasks
- Update the status of all completed backlog items
- Drag incomplete backlog items and tasks to the next sprint or back to the product backlog.

Drag an incomplete item to the product backlog or to a future sprint updates the Iteration Path of all unfinished child tasks to correspond to the product-backlog iteration path or future sprint.

You can drag-and-drop work items onto a sprint from any backlog or board.

Next step

Work with sprint burndown charts to monitor progress, manage scope creep, and mitigate risks.

Related articles

As you can see, the Taskboard provides support for your Scrum activities. For related articles, see:

- Assign backlog items to a sprint
- Interactively filter backlogs, boards, queries, and plans
- Scrum best practices

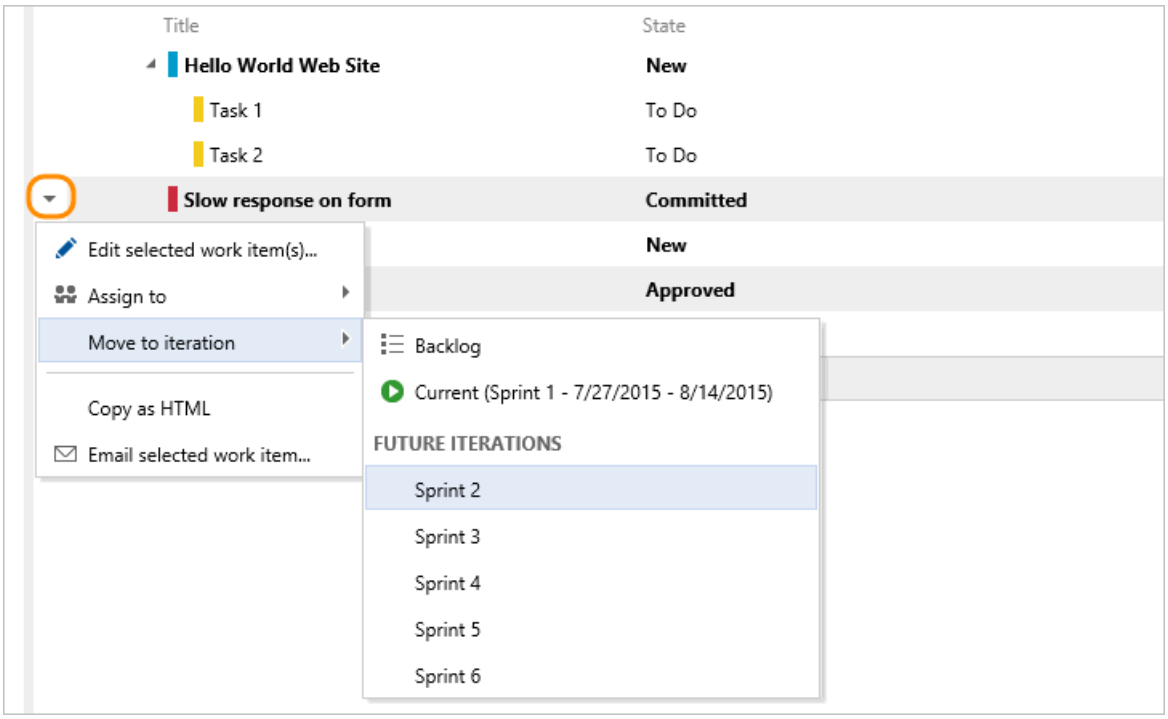
- Sprint planning
- Schedule sprints
- Customize a sprint Taskboard
- Capacity planning

Reduce the number of items on the taskboard

If you exceed the number of items allowed on your taskboard, you'll receive a message indicating that you need to reduce the number of items. The maximum number of items includes work item types included in the Requirement and Task categories.

You can reduce the number of items on the taskboard by moving them to the backlog or another sprint. When you move a parent PBI or user story, all active child tasks (State not equal to Done or Closed) automatically move with the parent item.

- From the taskboard, drag the PBI or user story from the first column onto the backlog or future sprint. All child tasks automatically move with the parent item.
- From the sprint backlog, multi-select the items to move and then select the context menu for an item. Then, select the iteration to move them to.



Or, you can increase the maximum number of allowed items.

Customize cards on a sprint taskboard in Azure Boards

10/4/2022 • 12 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Sprint Taskboards are similar to Kanban boards because they show work items as cards instead of as lists. They're different in the ways summarized in Backlogs, Boards, and Plans. Similar to the Kanban boards, you can customize cards and add columns.

Sprint Taskboards are similar to Kanban boards in that they show work items as cards instead of as lists. They're different in the ways summarized in Backlogs, Boards, and Plans. Similar to the Kanban boards, you can customize cards. To change column names or add columns, you need to customize the workflow.

> **NOTE**
> This article addresses customization of a sprint Taskboard. For information on customizing a Kanban board, see Manage and configure team tools.

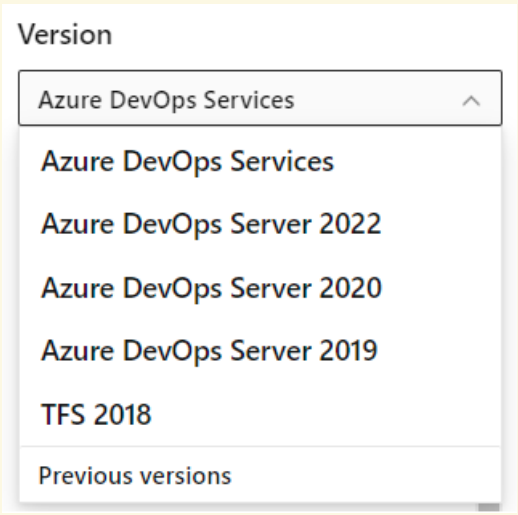
Prerequisites

- You must have a sprint Taskboard you want to configure. When you add a team, you add a Taskboard for every sprint that you select for your team. To learn more, see About teams and Agile tools.
- To add or rename columns, or customize cards, you must be added to the team administrator role for the team's settings you want to modify, or be a member of the **Project Administrators** security group. To get added, see Add a team administrator or Change project-level permissions.

Taskboard customization options

To add or remove columns, choose **Column Options**. You customize all other options through the **Settings** dialog for the Taskboard.

IMPORTANT



To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

Option	Use to...
Show bugs	Manage bugs on Taskboard similar to tasks.
Customize columns	Add or remove columns shown on Taskboard.
Fields	Add or remove fields from cards. Includes adding the **Parent** field to cards.
Fields	Add or remove fields from cards.
Styles	Add styling rules to change card color and title style based on field criteria.

NOTE
Each team can customize their Taskboard columns and cards. Taskboard settings are not inherited from other teams that they may share portions of area paths.

Understand the Taskboard customization sequence

Before you configure your Taskboard, you'll want to make sure the following tasks are as complete as possible. Otherwise, you'll find yourself revisiting your configuration.

Process Administrator:

1. Add custom work item types that you want to appear on your Taskboard. For details, see Add and manage work item types.
2. Customize your iteration backlog to ensure all work item types you want to have will appear on the Taskboards. For details see Customize backlogs & boards.
3. Customize each work item type to have any custom fields you want to show. For details, see Customize a workflow.

> **NOTE**
> You can customize a work item type which is different then customizing the card displayed on the Taskboard. You customize a WIT by adding fields, changing the workflow, adding custom rules and more. You can also add custom work item types and custom backlog levels. For details, see Customize an inheritance process.

> **NOTE**
> You can customize a work item type which is different then customizing the card displayed on the Taskboard. You customize a WIT by adding fields, changing the workflow, adding custom rules and more. You can also add custom work item types and custom backlog levels. For details, see Customize the On-premises XML process model.

Team Administrator:

1. Meet with your team and determine how the team wants to manage bugs, similar to requirements or tasks.
2. Add any tags to work items that you want to use to support styling rules.

Show bugs on your taskboard

If you want bugs to appear on your taskboard, change your team settings for show bugs on the backlogs and boards.

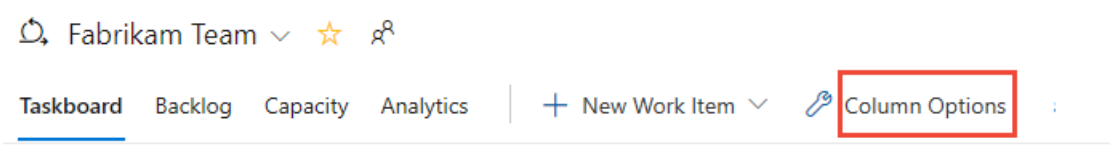
Add columns

You can add columns or rename columns that appear in your Taskboard. You'll see different column titles and choices based on the process used to create your project and whether your team has chosen to treat bugs like requirements or like tasks.

> **NOTE**
> Columns you add to a Taskboard aren't supported with corresponding fields such as the Kanban board columns you add which is supported with Board Column field.

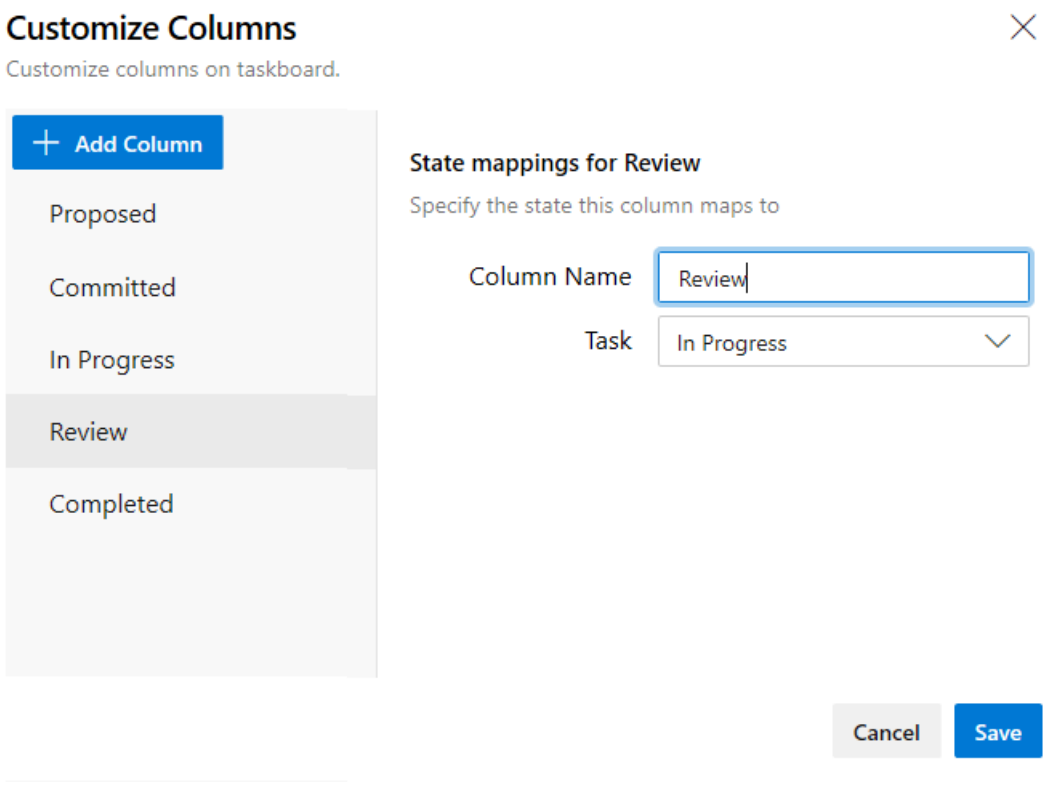
The changes made apply to all sprint taskboards for the selected team.

1. From your web browser, open your team's sprint Taskboard as described in Update and monitor your Taskboard. Remember, only team or project administrators can customize the taskboard.
2. Choose **Column Options**.



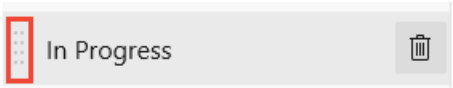
3. In the Customize Columns dialog, choose the column you want to rename, or choose **Add Column**.

   In this example, we add a column named Review and set the **Task** to *In Progress*.

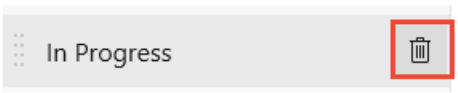


   Similar to the Kanban board, each column must map to a category state. There are four category states: *Proposed*, *Committed*, *In Progress*, and *Completed*. At least one column must map to *Proposed* and one column must map to *Completed*. To learn more about each state, see Workflow states and state categories.

4. To change the column order, hover over the column and choose the grabber icon and drag it up or down within the list of columns.



5. To delete a column, first make sure that the column doesn't contain any work items. If it does, move the items to another column. Then, hover over the column and choose the delete icon.



Add fields to cards on a sprint taskboard

Scrum teams use the Taskboard to burn down work and report on progress during daily stand ups. Each sprint Taskboard shows cards that correspond to both requirements and tasks.

Provide information-rich cards

Information rich cards not only provide at-a-glance information of interest to you and your team. They also provide a way for you to update a field without opening the work item. And, with style rules, you can highlight those work items with select colors based on the criteria you set.

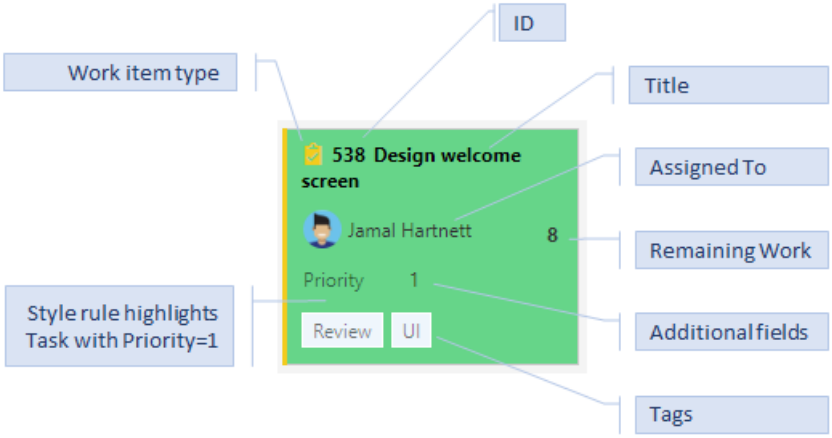
Each card corresponds to a work item that you use to share information, track status, and assign work.

> **NOTE**
>
> To understand the differences between backlogs, boards, taskboards, and Delivery plans, see Backlogs, boards, and plans.
> If your backlog or board doesn't show the work items that you expect or want, see Set up your backlogs and boards.

In this example, the bug work item type (WIT) shows all the core fields. It also shows three other fields and tags. To make severity 1 bugs stand out, a styling rule has been added to cause the card to display as yellow.

In the card shown below, the following customizations have been set for the task work item type (WIT):

- Show all core fields: ID, Assigned To, Remaining Work, Tags
- Show three additional fields: Priority
- Apply styling rule to display tasks with Priority=1 as green



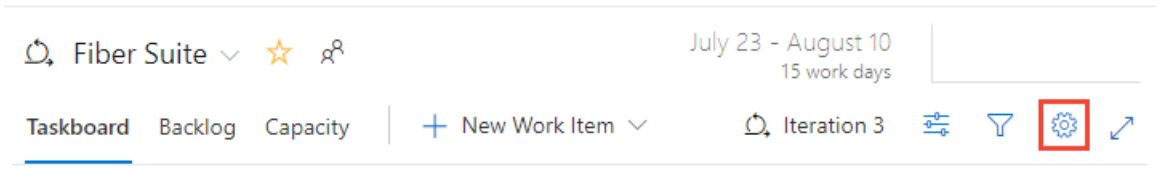
You can either increase or simplify the information that displays on your cards. It all depends on what's of interest to your team. Does your team like to refer to work items by their ID? Do they want to see estimates? Do they want to highlight work items according to set criteria? Or, will just the bare bones of title and assignment suffice?

Your best bet is to show fields on cards based on what your team frequently refers to. Or, you can show fields based on updates when using the Taskboards. Also, add fields with information that you can use to filter the board. If you're new to working with these tools, see Sprint planning.

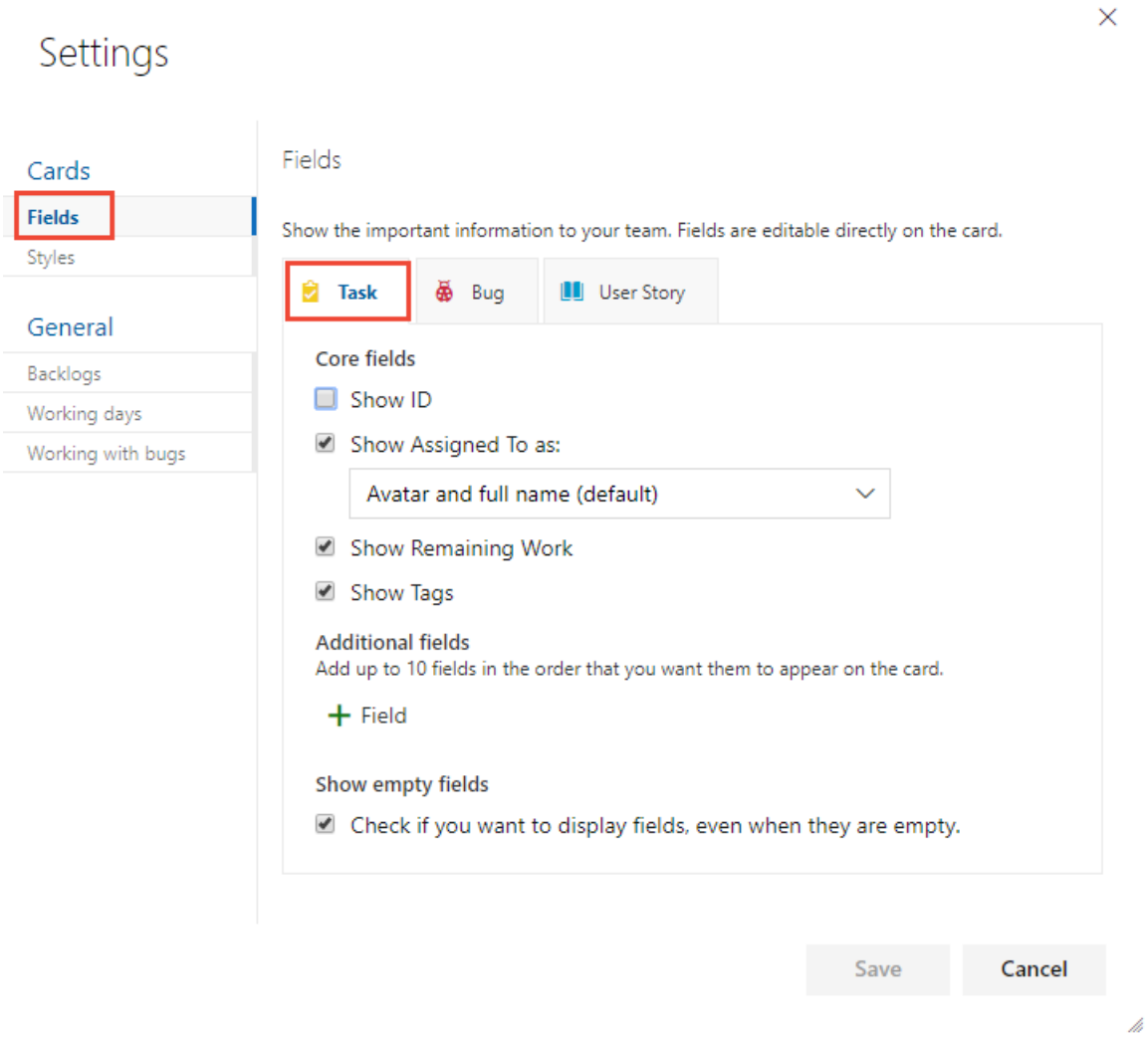
Add or remove fields from cards on the Taskboard

You change the way cards appear on the Taskboard in the same way you change the appearance of cards on Kanban boards. Only here, you start from the Taskboard.

1. Open the taskboard for the sprint you want to customize. Remember, only team or project administrators can customize the taskboard.
2. Choose the gear icon to open the Settings dialog.



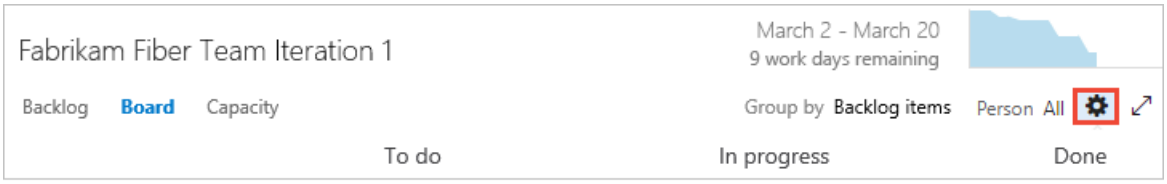
3. Choose **Fields** and then a work item type to see all the settings you can modify.
4. Place a check mark in the check box for those fields you want to have appear on the board.



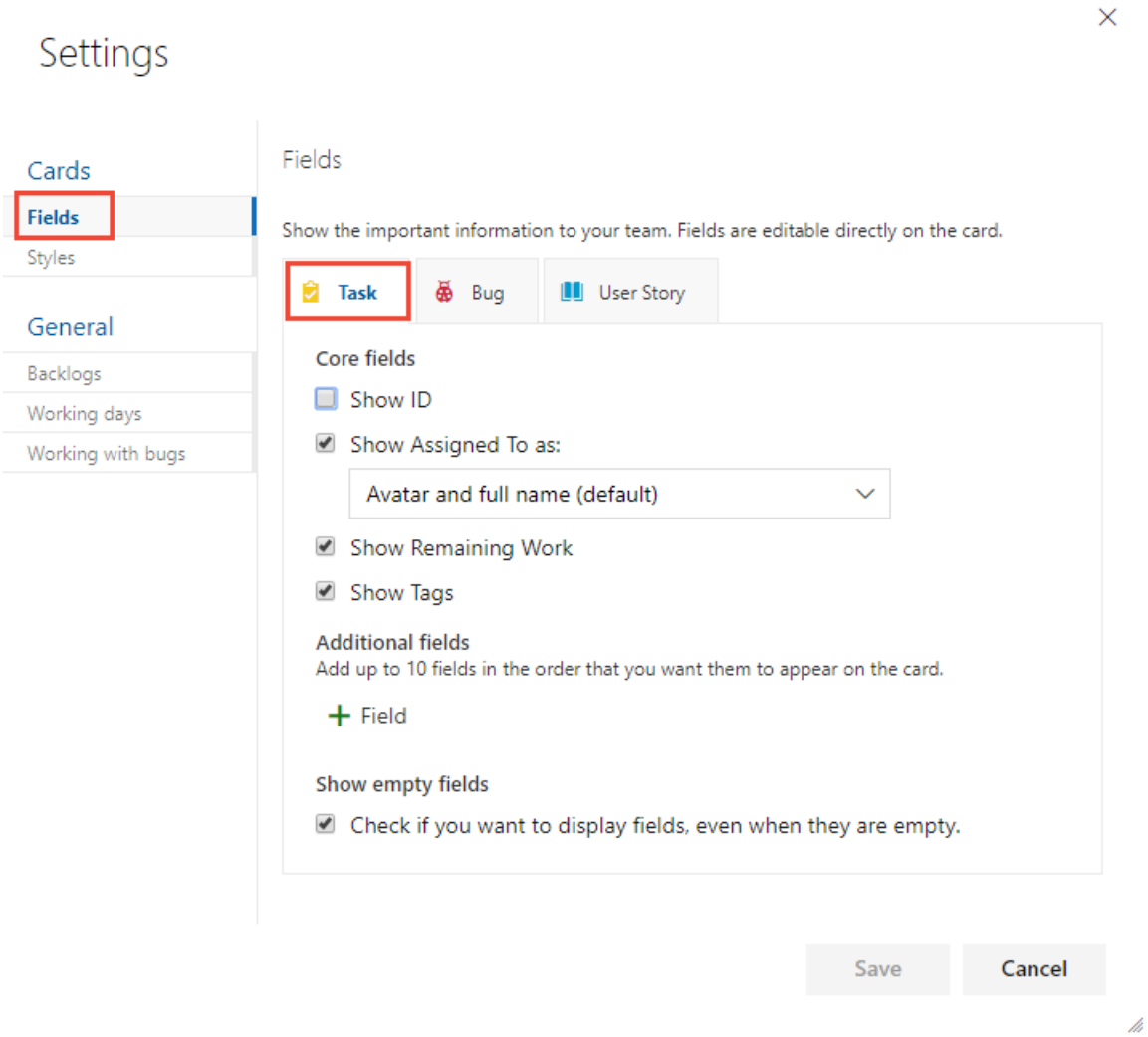
Repeat this step for each work item type you want to change. Don't be surprised if the options change when you choose a different work item type. For example, **Show Remaining Work** only applies to tasks and perhaps bugs, but not to user stories or product backlog items.

5. To add a field, choose the + plus icon and enter the name of a field you want to add.
6. To remove a field, choose the ✕ delete icon next to the field.
7. When done with your changes, choose **Save**.

1. Open the taskboard for the sprint you want to customize. Remember, only team or project administrators can customize the taskboard.
2. Choose the ⚙ gear icon to open the Settings dialog.



3. Choose **Fields** and then a work item type to see all the settings you can modify.
4. Place a check mark in the check box for those fields you want to have appear on the board.



Repeat this step for each work item type you want to change. Don't be surprised if the options change when you choose a different work item type. For example, **Show Remaining Work** only applies to tasks and perhaps bugs, but not to user stories or product backlog items.

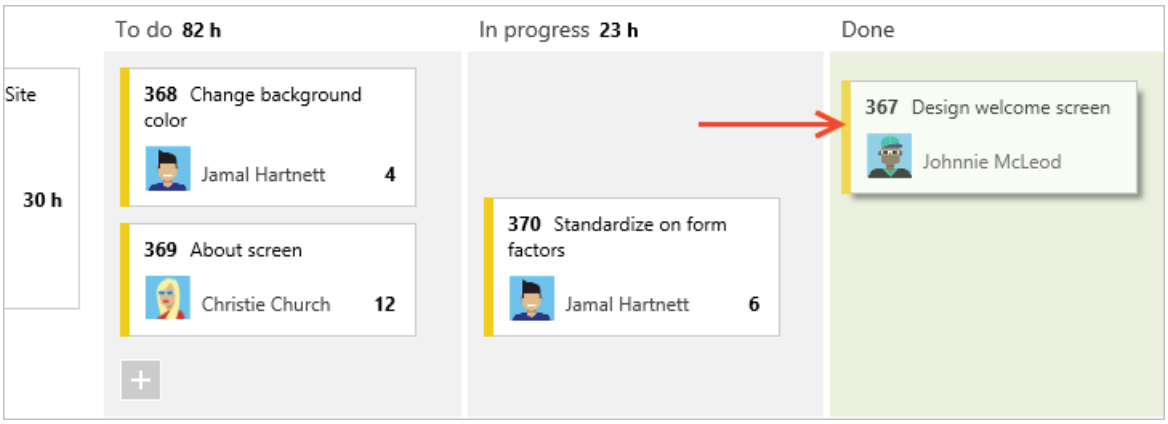
5. To add a field, choose the + plus icon and enter the name of a field you want to add.
6. To remove a field, choose the ✕ delete icon next to the field.
7. When done with your changes, choose **Save**.

Update fields from cards

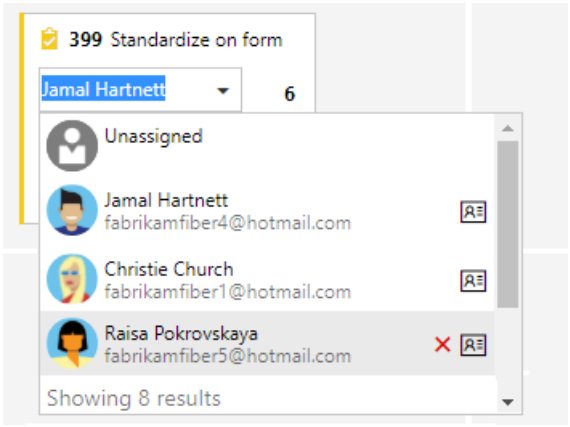
Using the board views provides you with quick and easy ways to update work items as work progresses. Making daily or frequent updates helps everyone on your team stay in sync with what's been done and what needs doing next.

To update status of a work item, you simply drag-and-drop cards to a different column. To change the order or stack ranking of a work item, you drag a card up or down within a column.

Moving the card from In Progress to the Done column on the Task board, for example, updates the corresponding State field. In this case, the State field updates from Doing to Done.

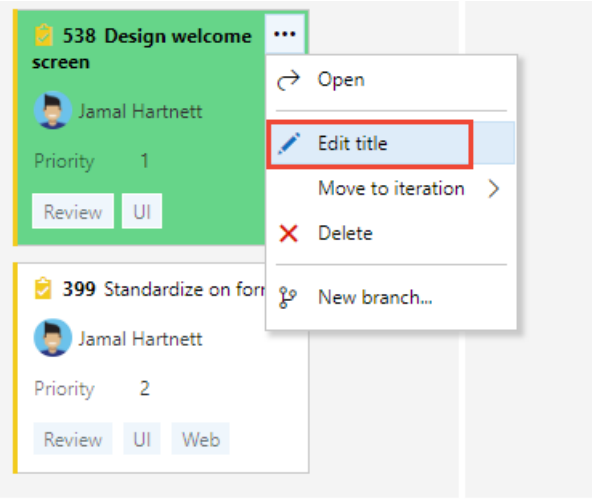


Another handy feature is to update a field without having to open the work item. You can update most fields shown on the card. Here we reassign a task.



This quick update feature is useful when you need to update many work items at once. For example, you can add estimates or update Remaining Work.

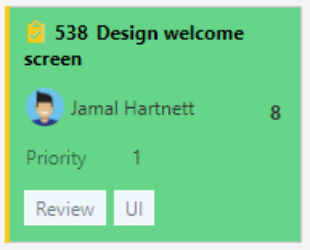
To change the title, choose the ··· actions icon, and then choose **Edit title**.



To add tags, double-click the work item to open it. And, just a reminder, you can't change the IDs for a work item, not from the card and not from within the form.

Define style rules, highlight cards

With styling rules, you can cause cards to change color when their corresponding work items meet criteria that you set. Here, we highlight Priority 1 tasks by having the cards display as green.



Example styling rules

What rules should you apply to highlight work items? Here are a few examples and their associated criteria.

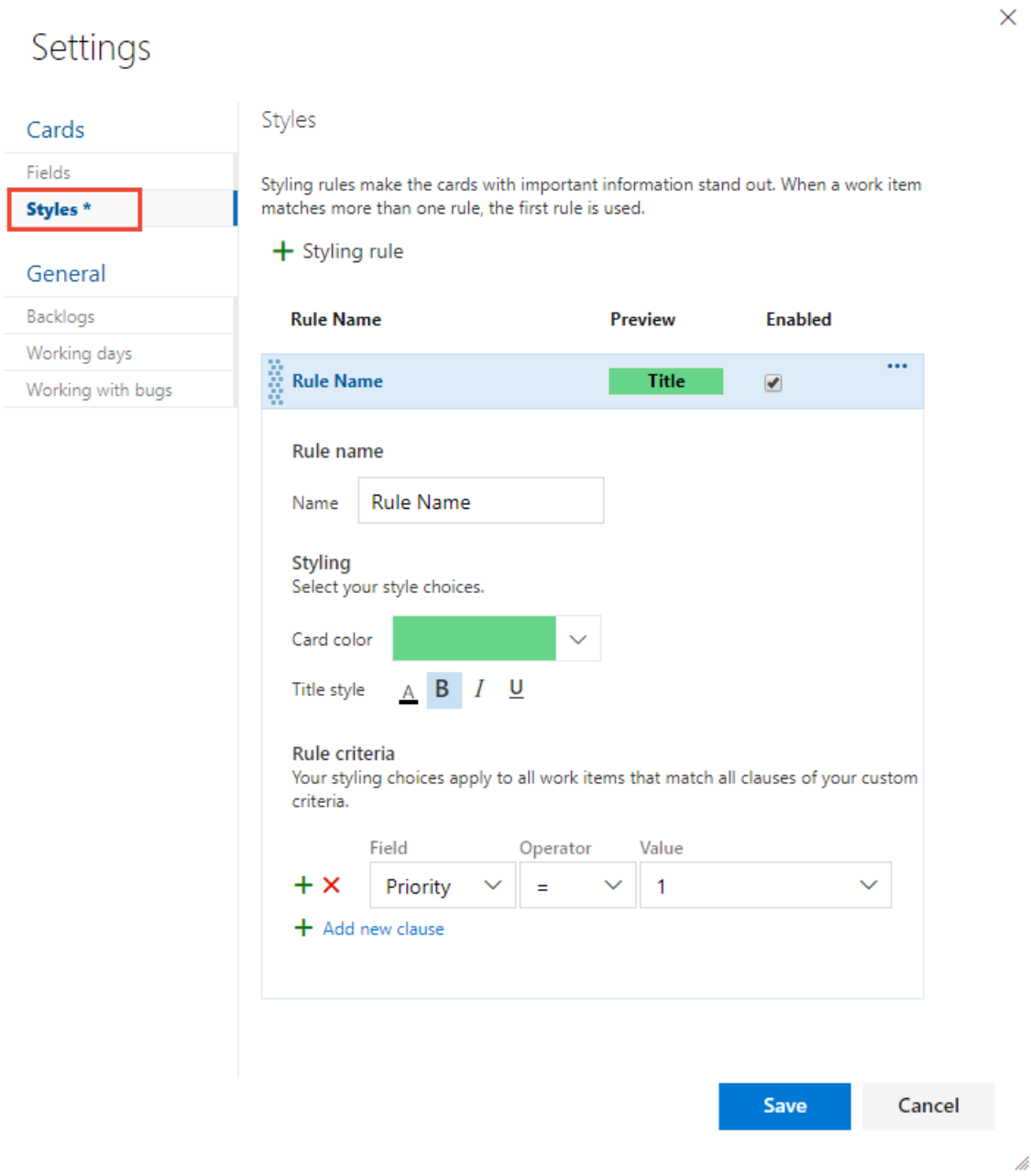
WORK ITEMS	CRITERIA
High priority items	`Priority = 1`
High effort items	`Remaining Work>=12`
Stale items unchanged in the last 5 days	`Changed Date @Today-5`
Title contains a key word	`Title Contains Yes`
Severity 1 bugs	`Severity = 1 - Critical AND Work Item Type = Bug`
High value business items	`Business Value 50`
Items assigned to specific feature area	`Area Path Under Fabrikam Fiber\Phone`
Contains specific tag	`Tags Contain RTM`
Blocked tasks (Scrum process only)	`Blocked = Yes`

Add or remove a style rule

You can apply style rules to change the color of Taskboard cards based on specified field criteria.

1. Open the Taskboard that you want to customize.
2. Choose the gear icon to open the **Settings** dialog.
3. Choose **Styles** to specify a style rule. Choose the plus icon to add a style. Select the color to apply to the card and define the criteria for the style rule.

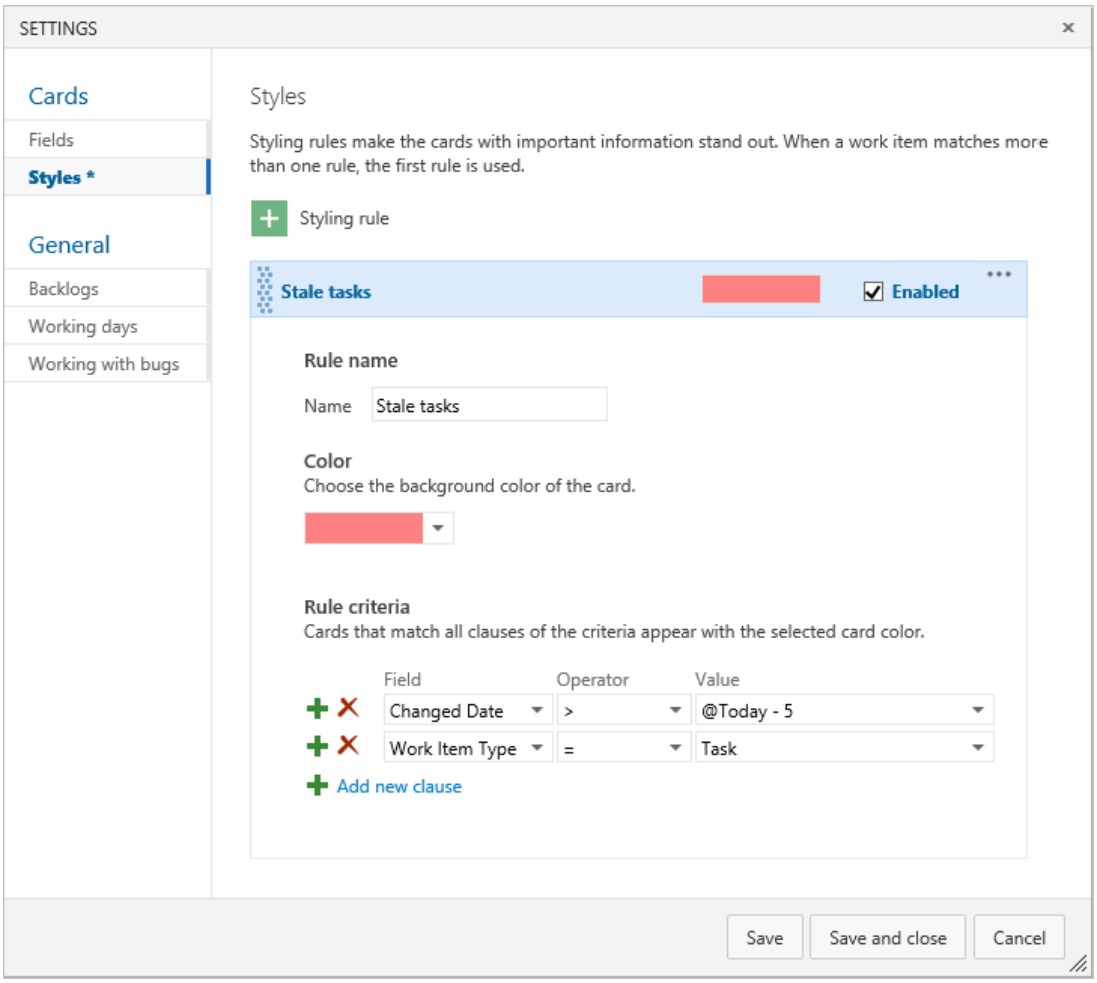
   In this example, we show the **Styles** dialog for the taskboard.



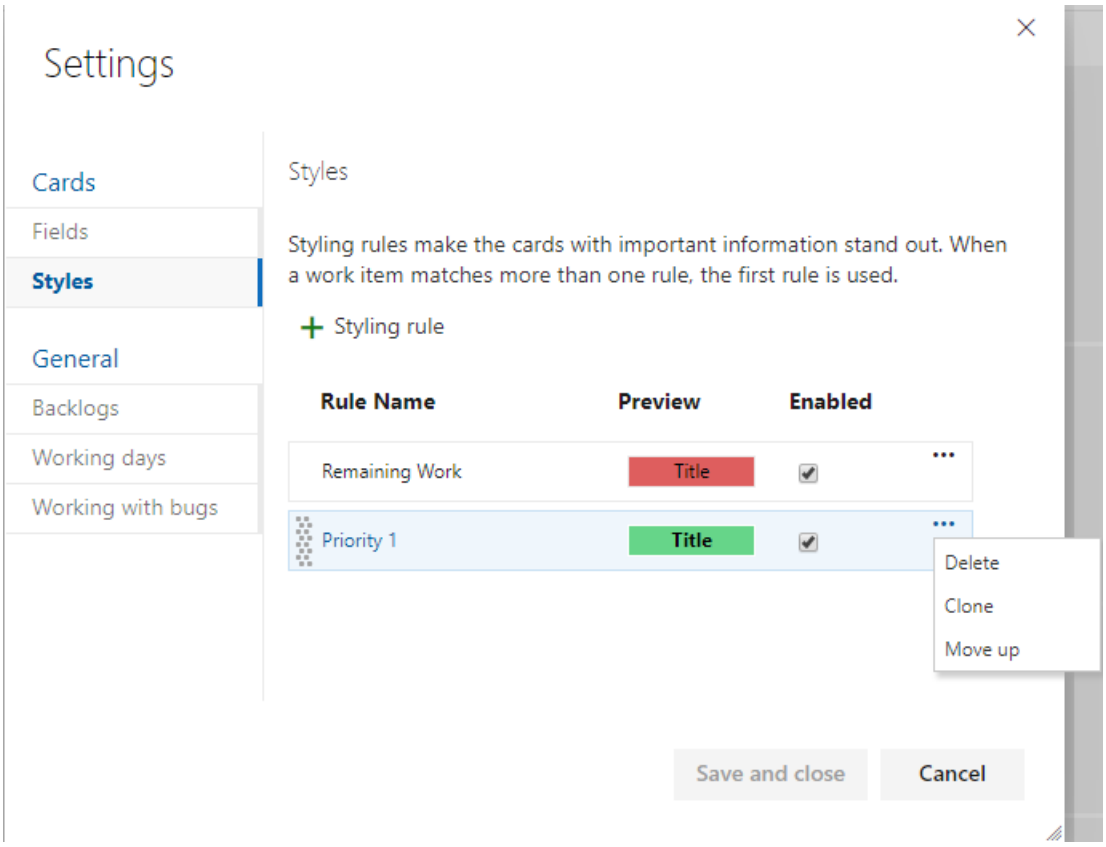
Follow these rules when creating and ordering your style rules:

- The criteria you specify works in a similar fashion as when constructing a query.
- All clauses are considered AND clauses, grouping clauses isn't supported.
- Card rules apply to all work items that meet the rule criteria
- Rule color applies to work items based on the order in which rules are listed. If you add more than one style rule, make sure that you move them in the order of most importance. Drag them into the order you want them applied.
- You can quickly enable and disable a style rule.

  Here we add a Stale tasks rule that highlights tasks that haven't changed in the last five days.



4. To copy or delete a style rule, choose the ··· actions icon and select **Clone** or **Delete**, respectively.

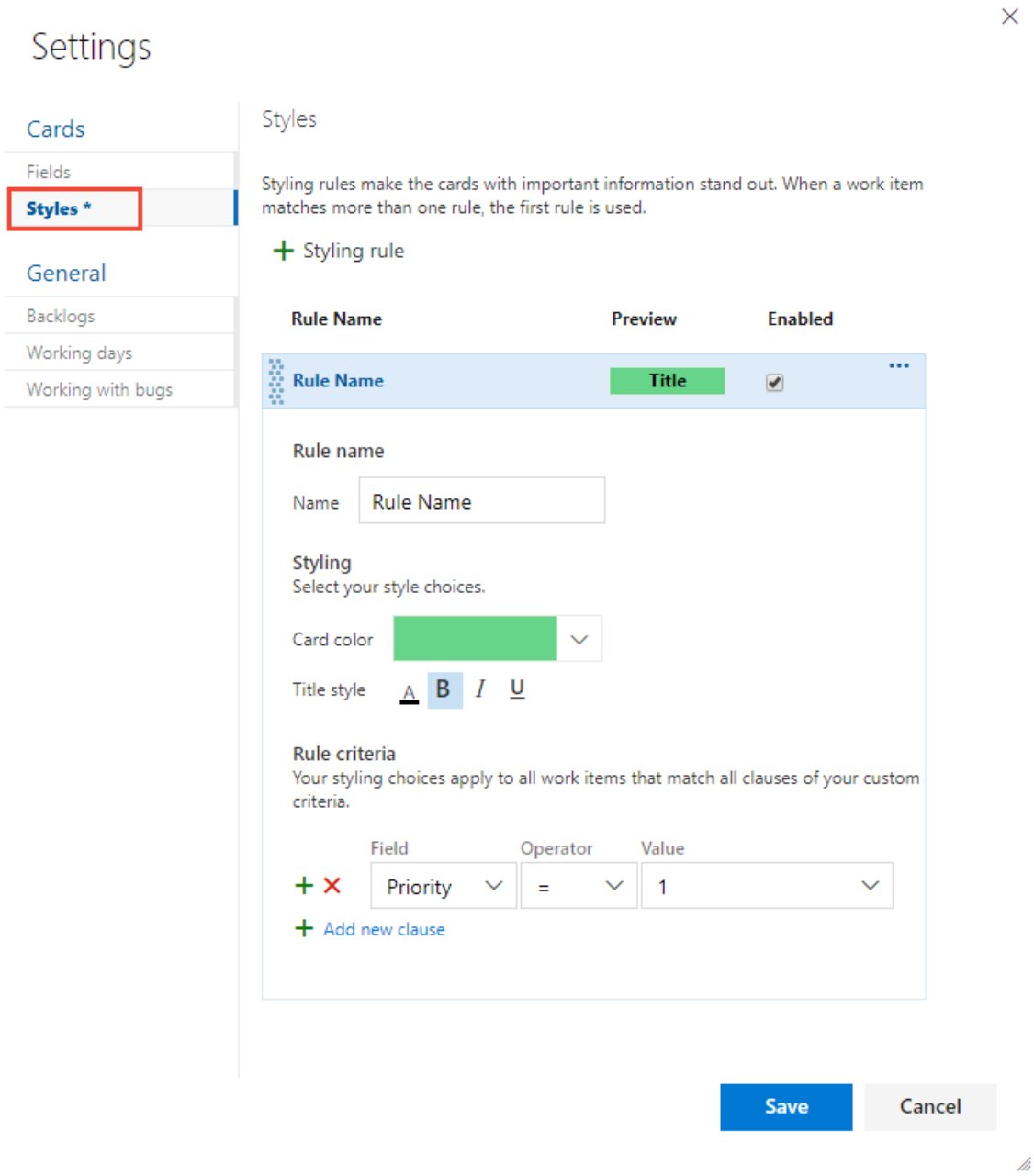


5. When done with your changes, choose **Save**.

1. Open the Taskboard that you want to customize. If you're not a team admin, get added as one. Only team and project admins can customize boards.

2. Choose the gear icon to open the **Settings** dialog.
3. Choose **Styles** to specify a style rule. Choose the plus icon to add a style. Select the color to apply to the card and define the criteria for the style rule.

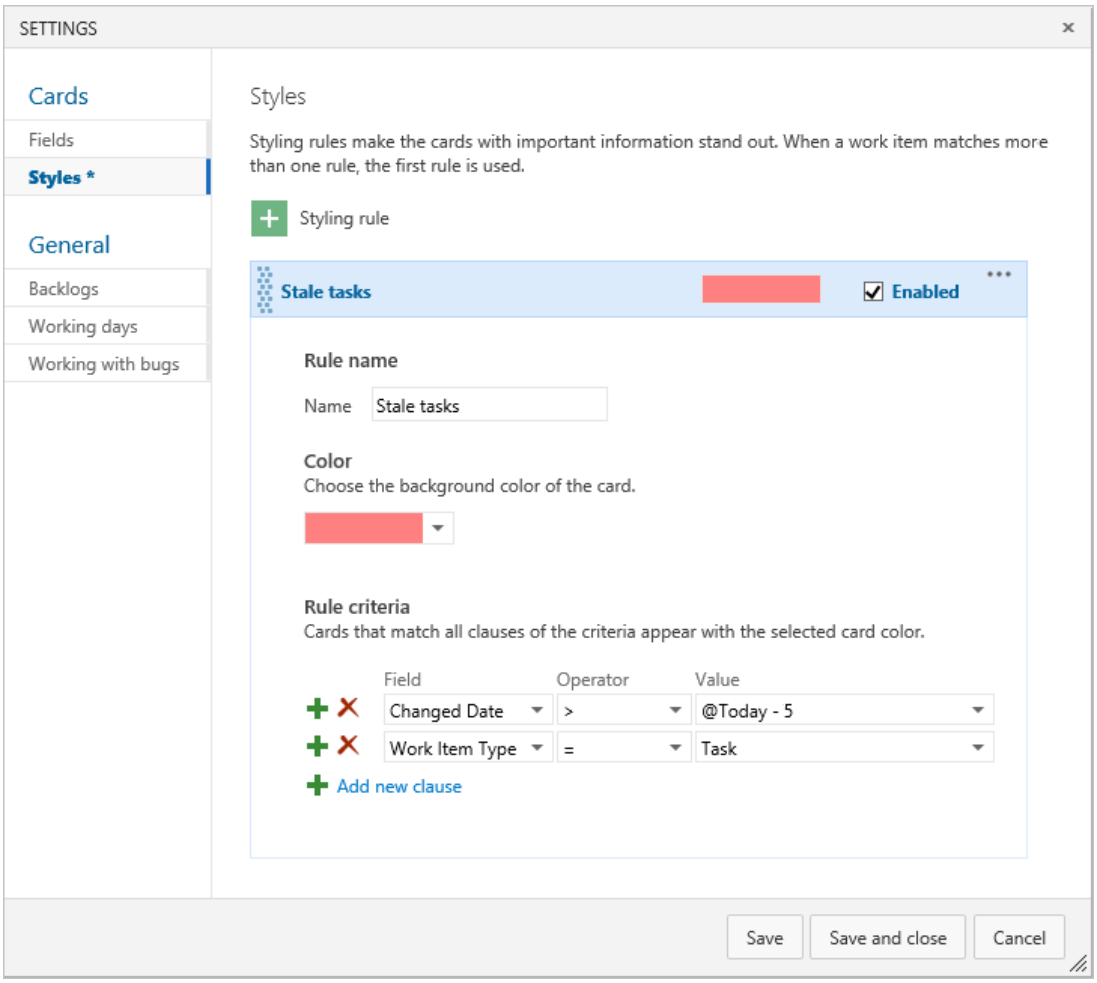
   In this example, we show the **Styles** dialog for the taskboard.



Follow these rules when creating and ordering your style rules:

- The criteria you specify works in a similar fashion as when constructing a query
- All clauses are considered AND clauses, grouping clauses isn't supported
- Card rules apply to all work items that meet the rule criteria
- Rule color applies to work items based on the order in which rules are listed. If you add more than one style rule, make sure that you move them in the order of most importance. Drag them into the order you want them applied.
- You can quickly enable and disable a style rule

  Here we add a Stale tasks rule that highlights tasks that haven't changed in the last five days.



4. To copy or delete a style rule, choose the ··· actions icon and select **Clone** or **Delete**, respectively.

5. When done with your changes, choose **Save**.

Changes cause automatic taskboard updates

Your Taskboard automatically refreshes when changes occur. There isn't any live updates control, it simply happens in the background. As other team members move or reorder cards on the taskboard, the Taskboard automatically updates with these changes. You don't need to press F5 to see the latest changes.

Related articles

- Manage and configure team tools
- Setup your backlogs and boards
- Show bugs on backlogs and boards
- Set working days

Manage columns in a work item list in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Each column corresponds to a work item field. You can add and remove columns within work item lists to show the fields of interest to you. Or, you can drag a column to a new position. Your settings persist for each page you customize and are only valid for your views.

Specifically, you can do the following actions from the following list views:

Action	Backlogs	Sprint backlogs	Queries	Work items
Add or remove a column field	Yes	Yes	Yes	Yes
Add or remove the Parent field	Yes	Yes	Yes	Yes
Add or remove a rollup column	Yes	No	No	No
Sort on a column	No	No		

Yes

Yes

> **TIP**
> Unlike a query result, you can't sort a backlog by a column. However, you can use the **Create Query** link on each backlog to create a query that you can sort on any field column you choose from the **Sorting** tab of the Column options dialog. While you may be able to add a field to sort on, not all fields are supported. For example, selection of the **Parent**, **History**, **Description**, or other rich-text field will result in the display of an error message as you can't sort on these fields.

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. You can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. If your project uses the Inherited process model, you can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project.

About persistence of column options

Once you set the column options for a specific view, your settings persist until you change them. The following notes apply to specific views.

- Column options you set for a backlog apply only to the active team and backlog. Options you set for the product backlog differ from the options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query.
- Column options you set for one of the supported Work items views persists for a specific view only, such as **Assigned to me**, **Following**, **Mentioned**, and so on.

- Column options you set for a backlog apply only to the active team and backlog. Options set for the product backlog differ from those options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query. For Shared queries, your settings can get overwritten when someone else saves different column option settings.

> **NOTE**
> You can't set column options for other members of your team, nor can you set default column options.

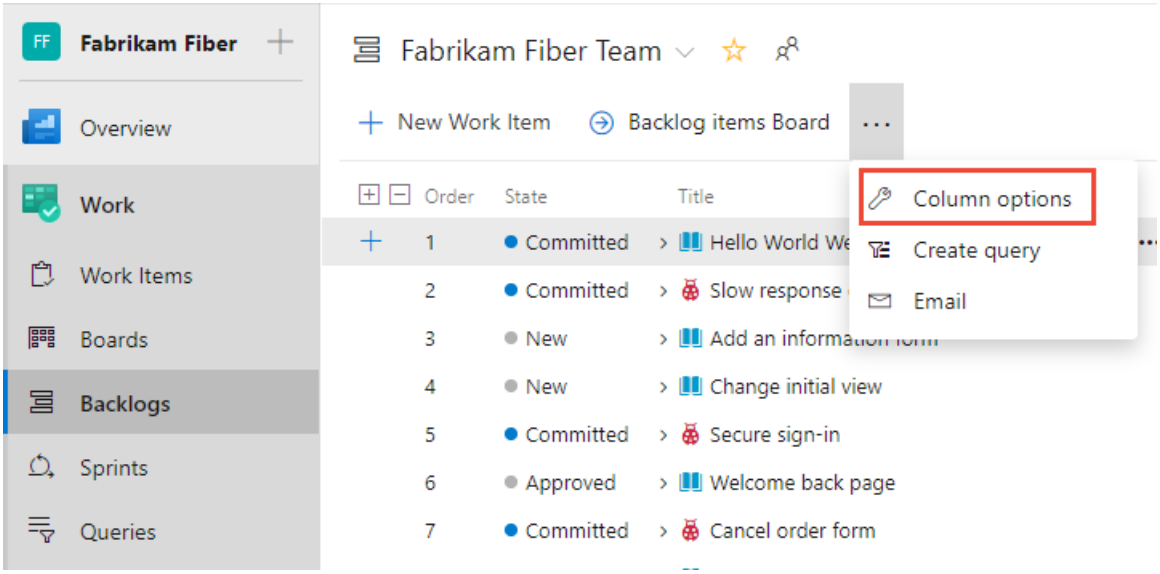
> **NOTE**
> You can't set column options for other members of your team. Also, for projects that use the Inheritance process model, you can't set default column options. For projects that use the On-premises XML process model, you can set the default column options for product, portfolio, and sprint backlogs. To learn how, see Process configuration XML element reference.

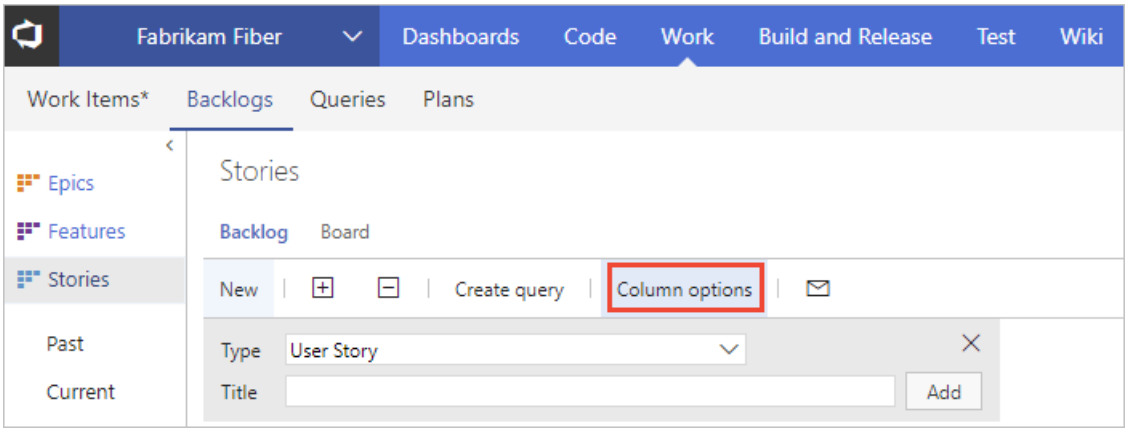
NOTE

You can't set column options for other members of your team. For projects that use the On-premises XML process model, you can set the default column options for product and portfolio backlogs. To learn how, see Process configuration XML element reference.

Open the Column options dialog

Start by opening the **Column Options** dialog. If you don't see the option, choose the ... and choose from the options provided.





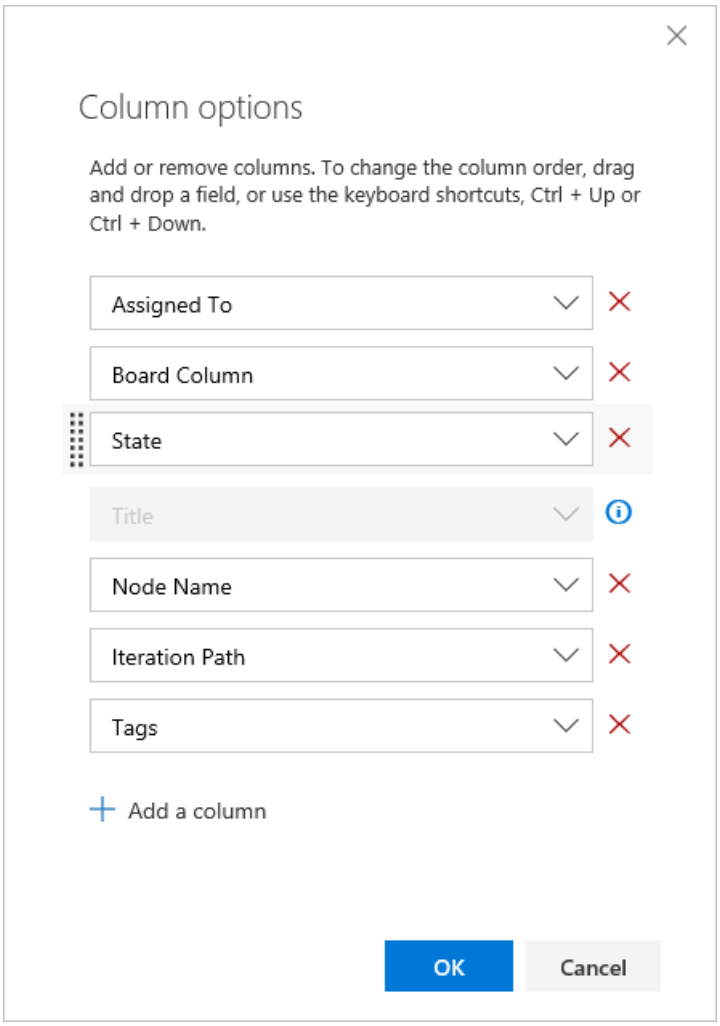
Add or remove columns

- Browser
- Visual Studio

In the Column options dialog, choose **Add a column** to add a field that isn't shown. To change the order of the fields, drag-and-drop the field where you want it within the set of selected fields. And, to remove a field, choose the ✕.

NOTE

The following dialog is available from TFS 2018.2 and later versions.

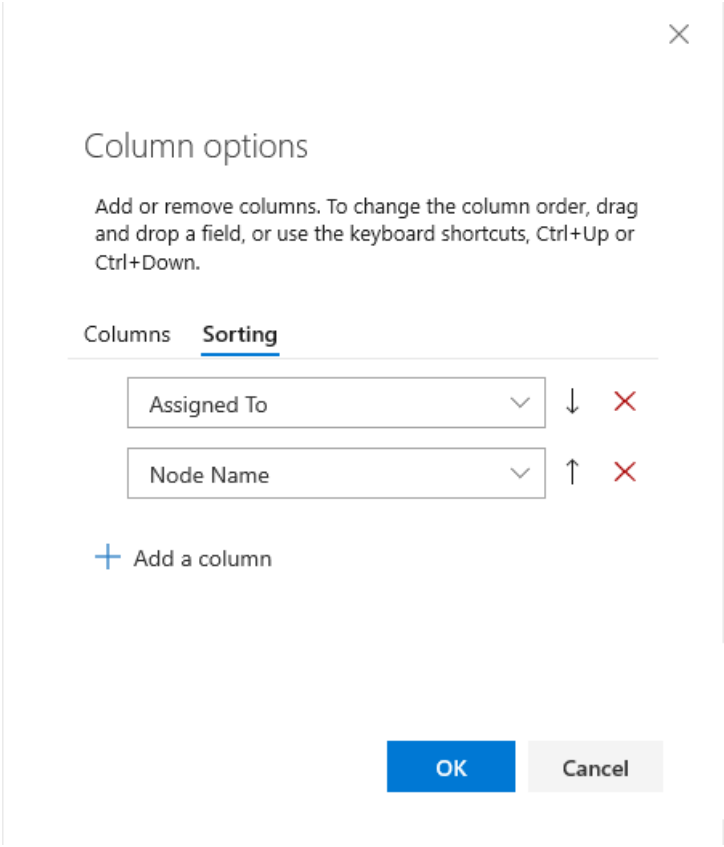


Add or remove rollup columns

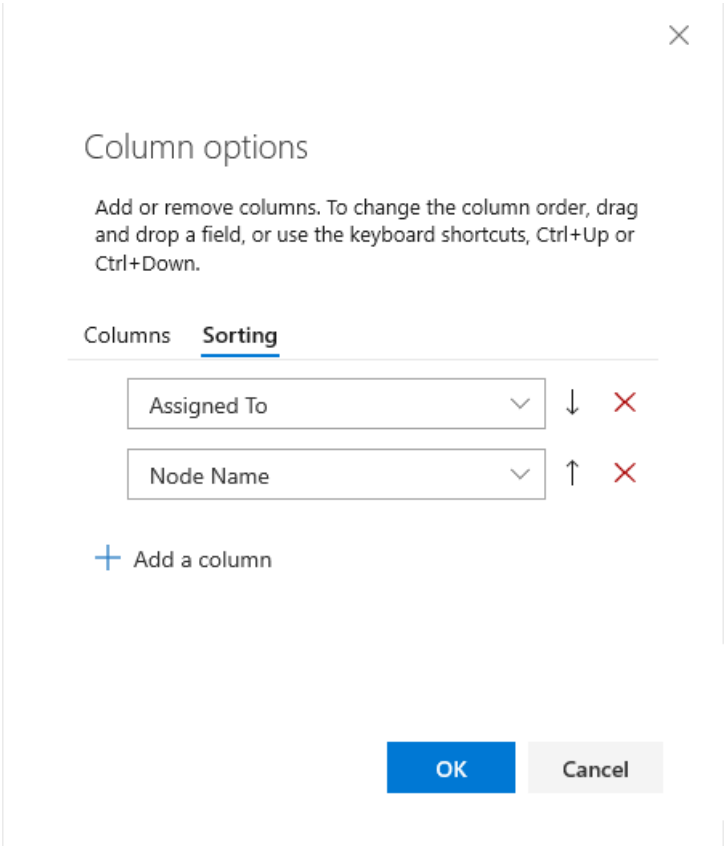
Rollup columns can display progress bars or the sum of numeric fields of child items. You can add them to any product or portfolio backlog. To learn more, see Display rollup progress or totals.

Sort on a column

You can sort query results and Work items views. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



You can sort query results. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



Use keyboard shortcuts to change the column order, column width, or sort options

You can change the column order, column size, or sort options by using the following keyboard commands:

- To change the column order, choose the field and drag it to a new location
- To resize a column, choose the column divider to the right of the field and drag to a new location
- For query results:
  - Add the field as a column in order to sort by that field
  - To sort by a column, hold down the SHIFT key and select on the field
  - To reverse the sort order, SHIFT+click on the field
  - To sort by multiple columns, SHIFT+click on each column in the order you want to sort

For other keyboard shortcuts, enter **?** to display available shortcuts based on the page you're on.

Related articles

- Display rollup progress or totals
- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- View, run, or email a work item query
- Create managed queries
- Customize a sprint Taskboard

- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- Create managed queries

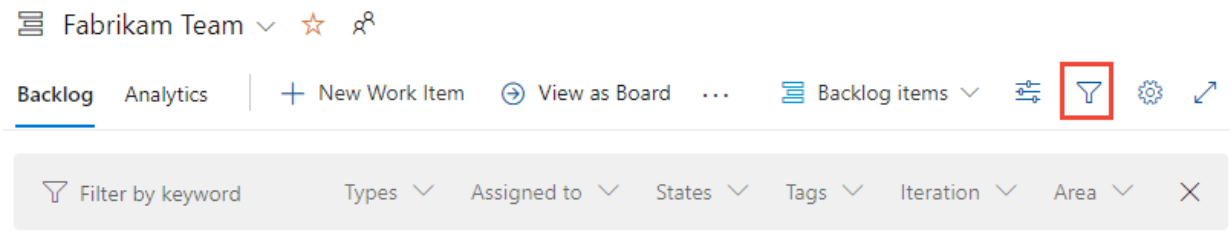
Interactively filter backlogs, boards, queries, and plans in Azure Boards

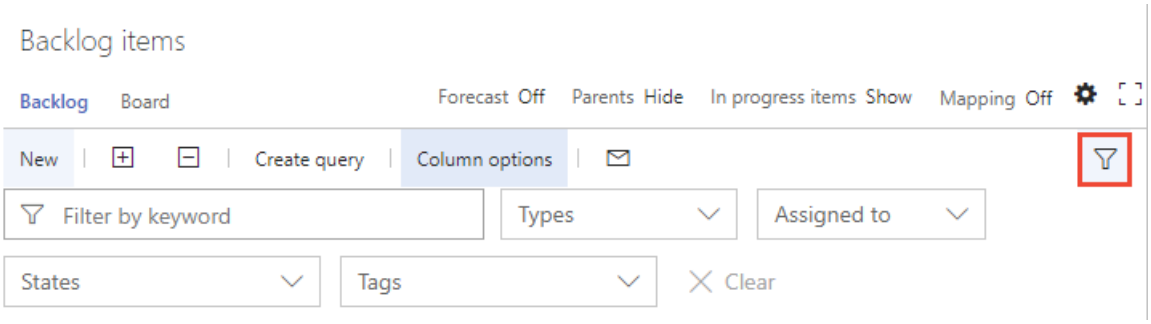
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

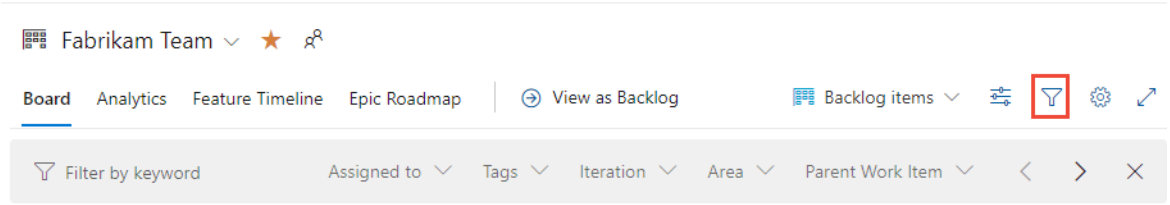
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

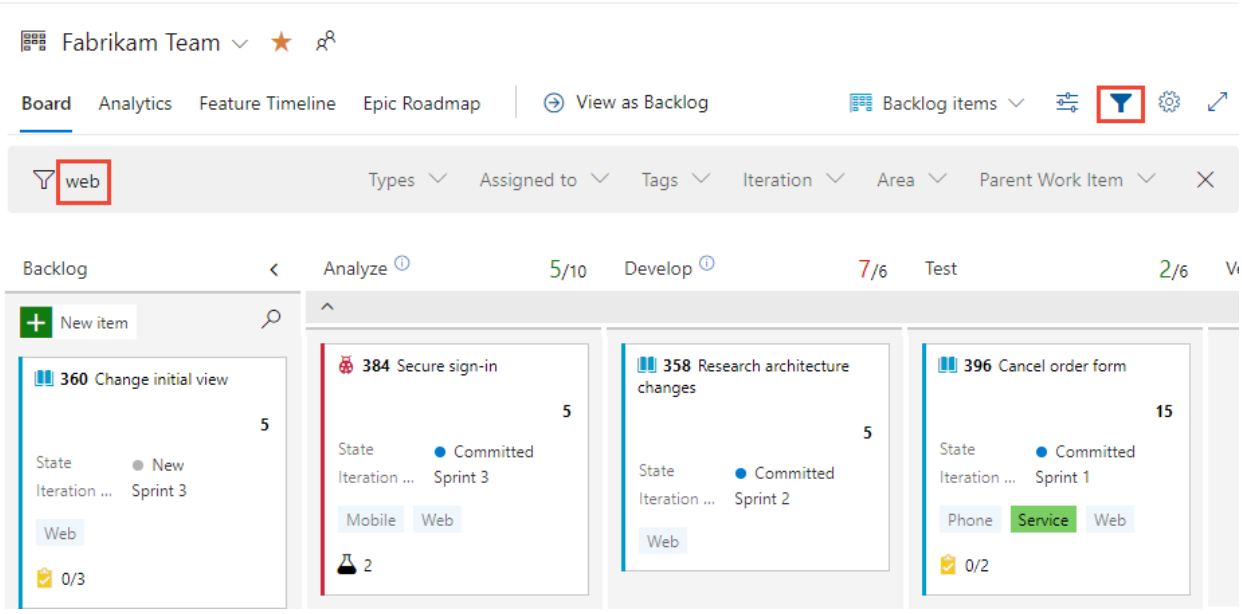
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

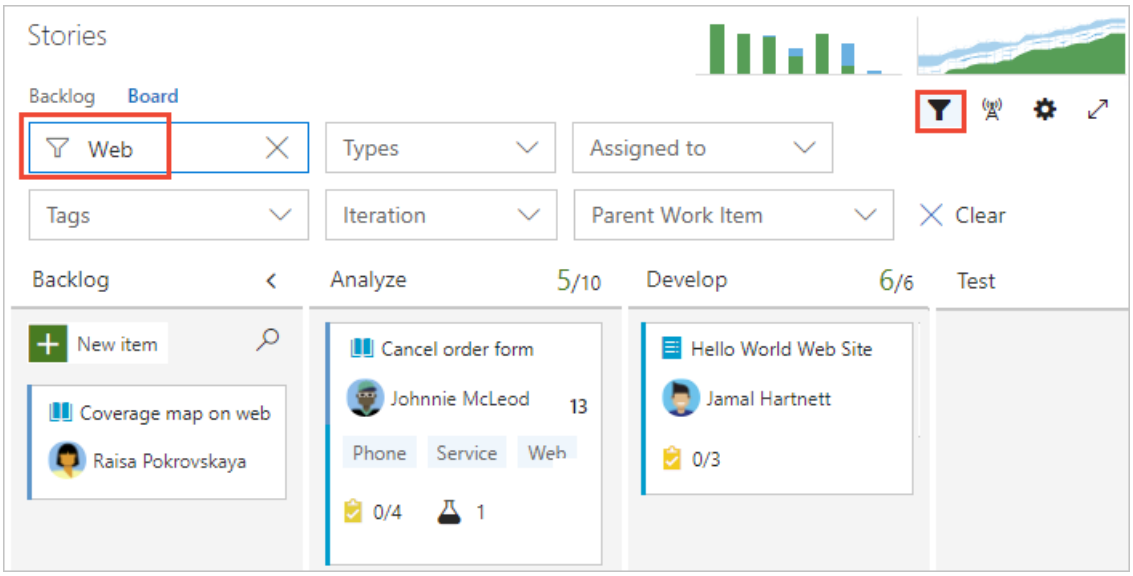
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

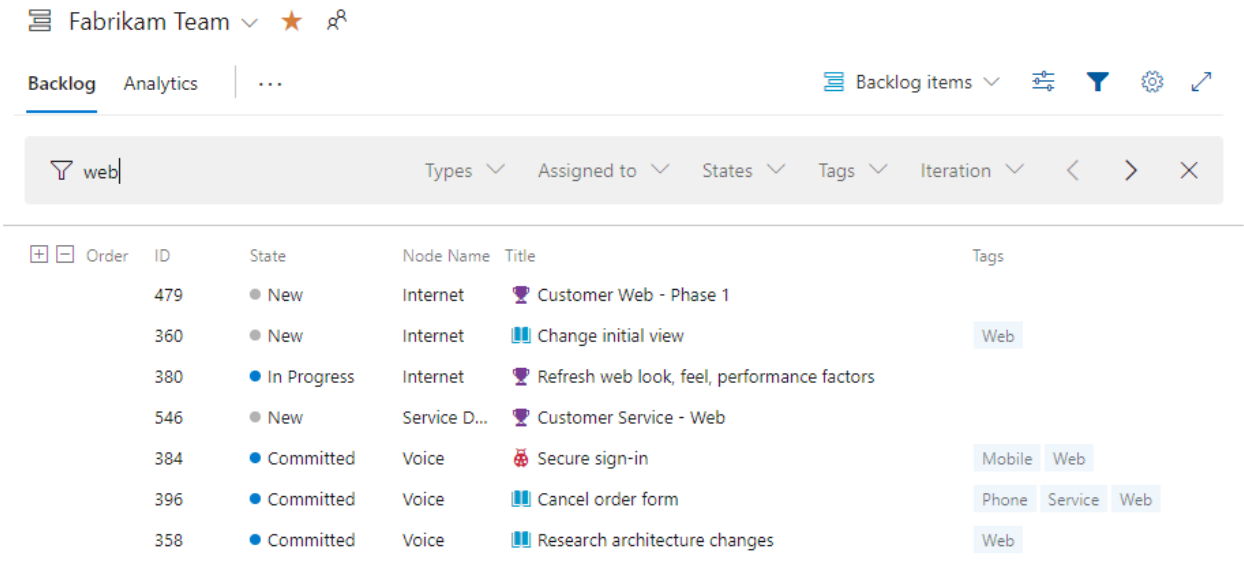


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

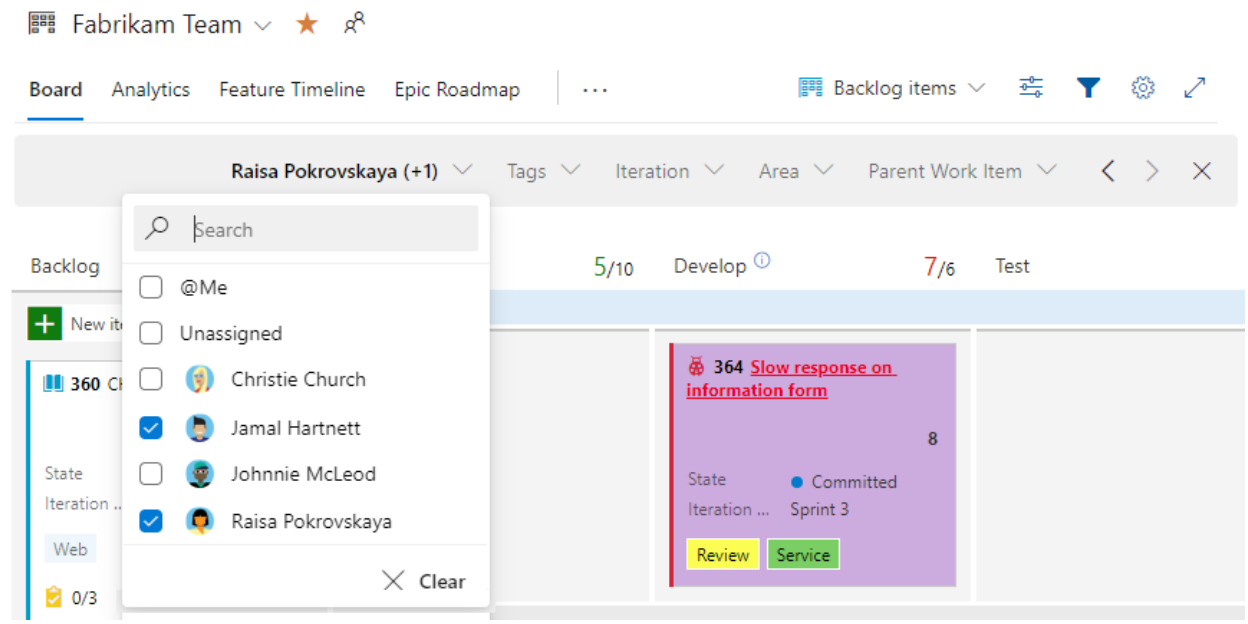
Features board

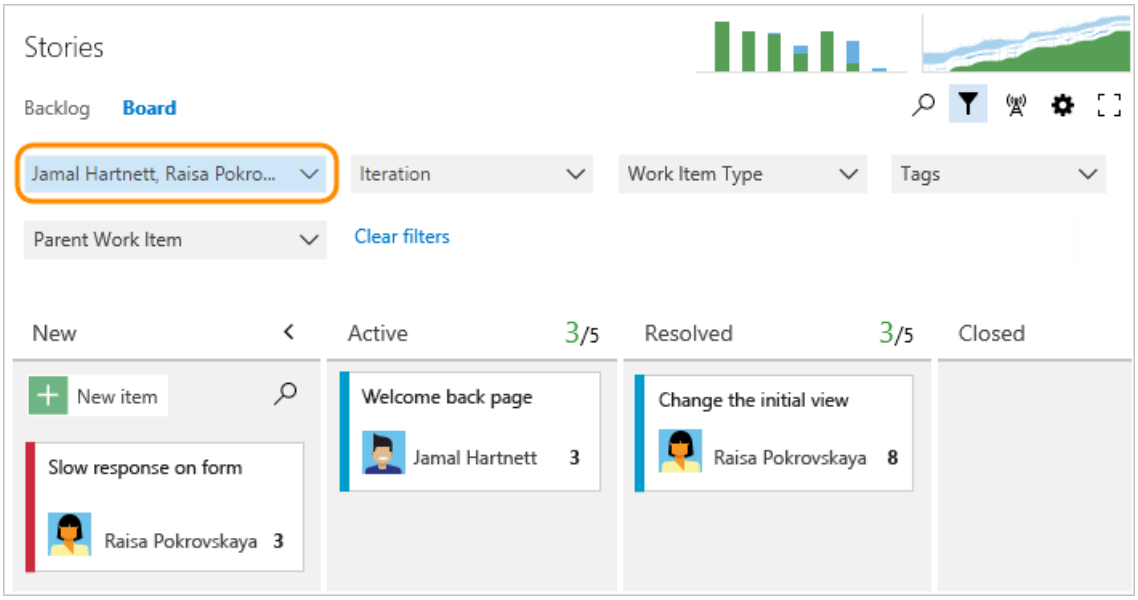
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

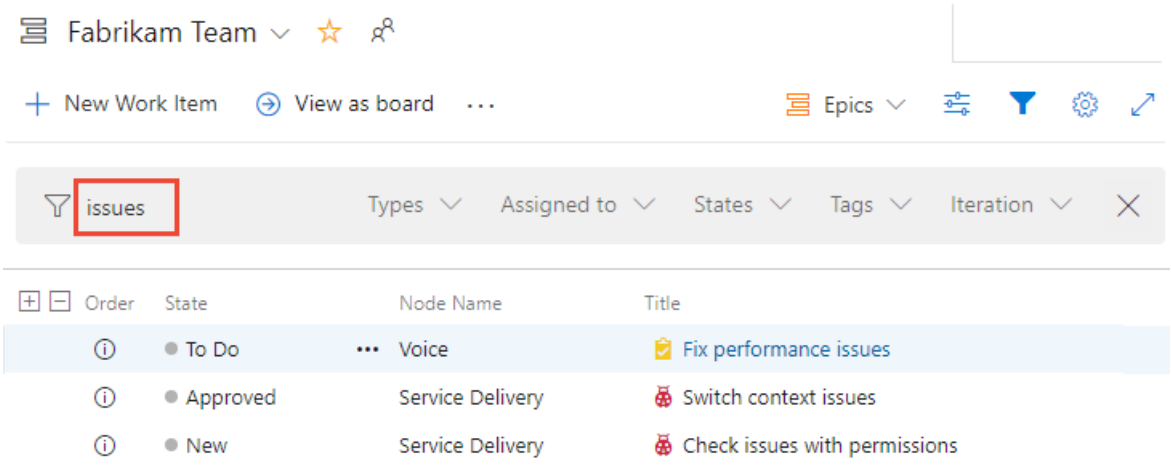
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

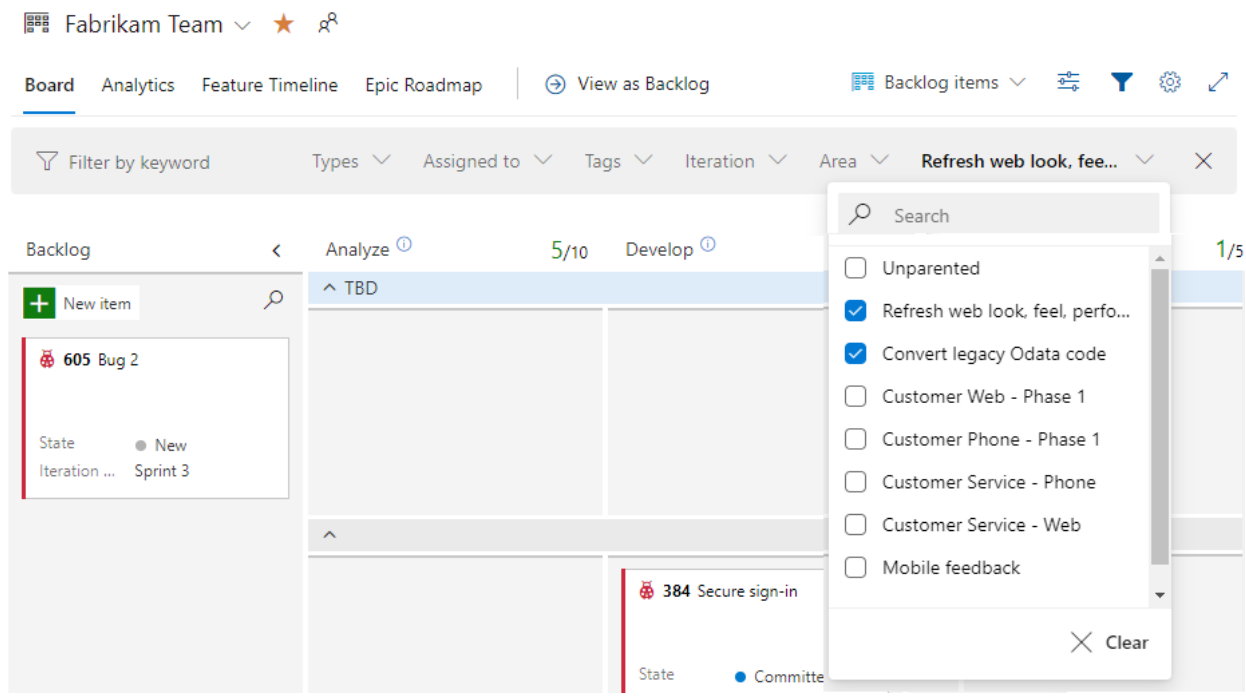
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

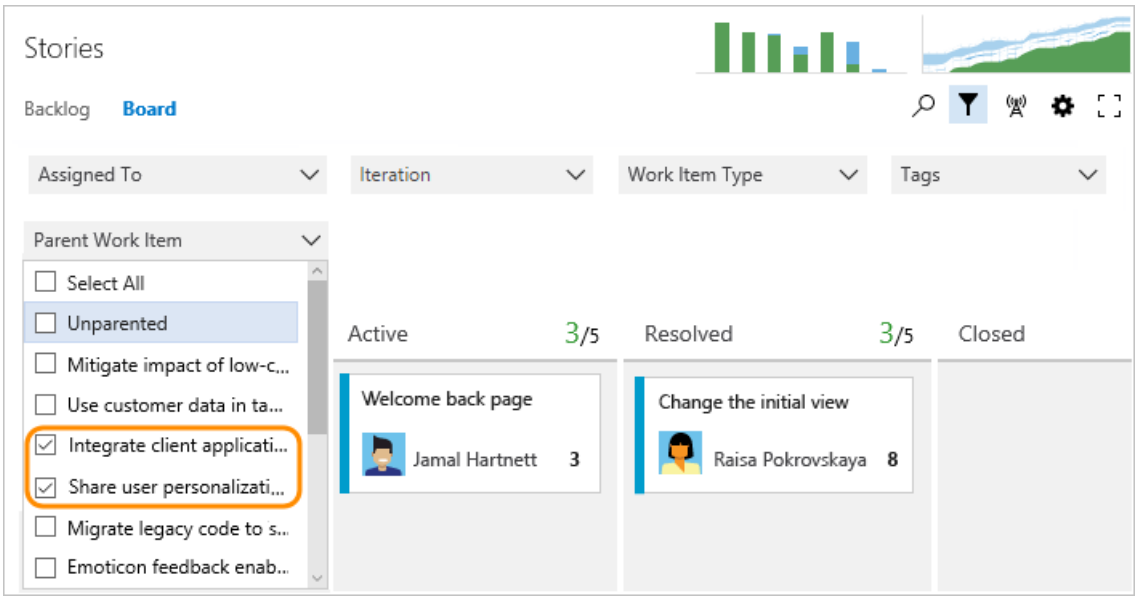
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



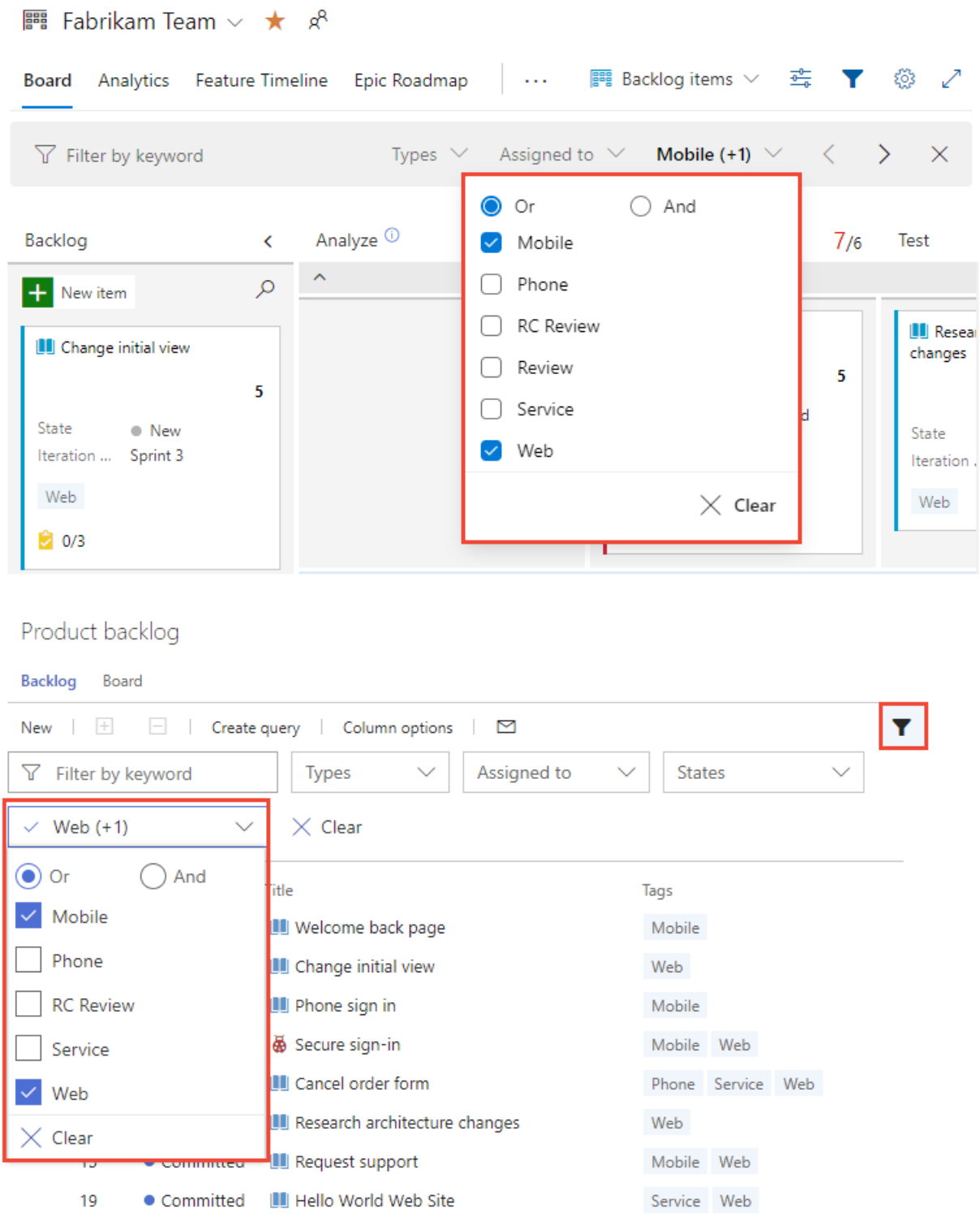


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

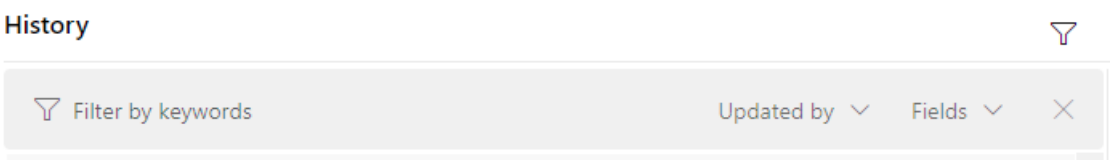


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Configure working days

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Configure the days that your team works. Your team's working days aid in capacity planning purposes and when you're using sprint and scrum methods in Azure Boards. Each team's sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown. Leave those days of the week that your team doesn't work unchecked.

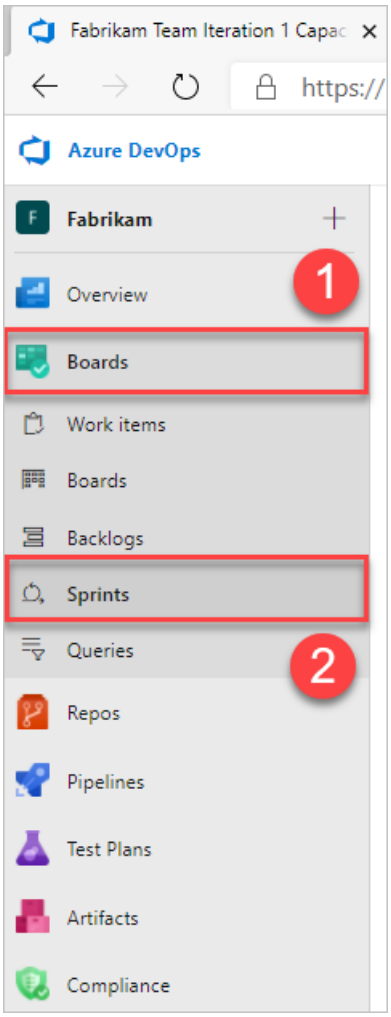
> **NOTE**
> The **Working days** setting specifies the days that teams regularly take off each week. To specify additional non-working days off, such as holidays or a team day off, set these through the **Capacity** page as described in Set sprint capacity, Set capacity for the team and team members.

Prerequisites

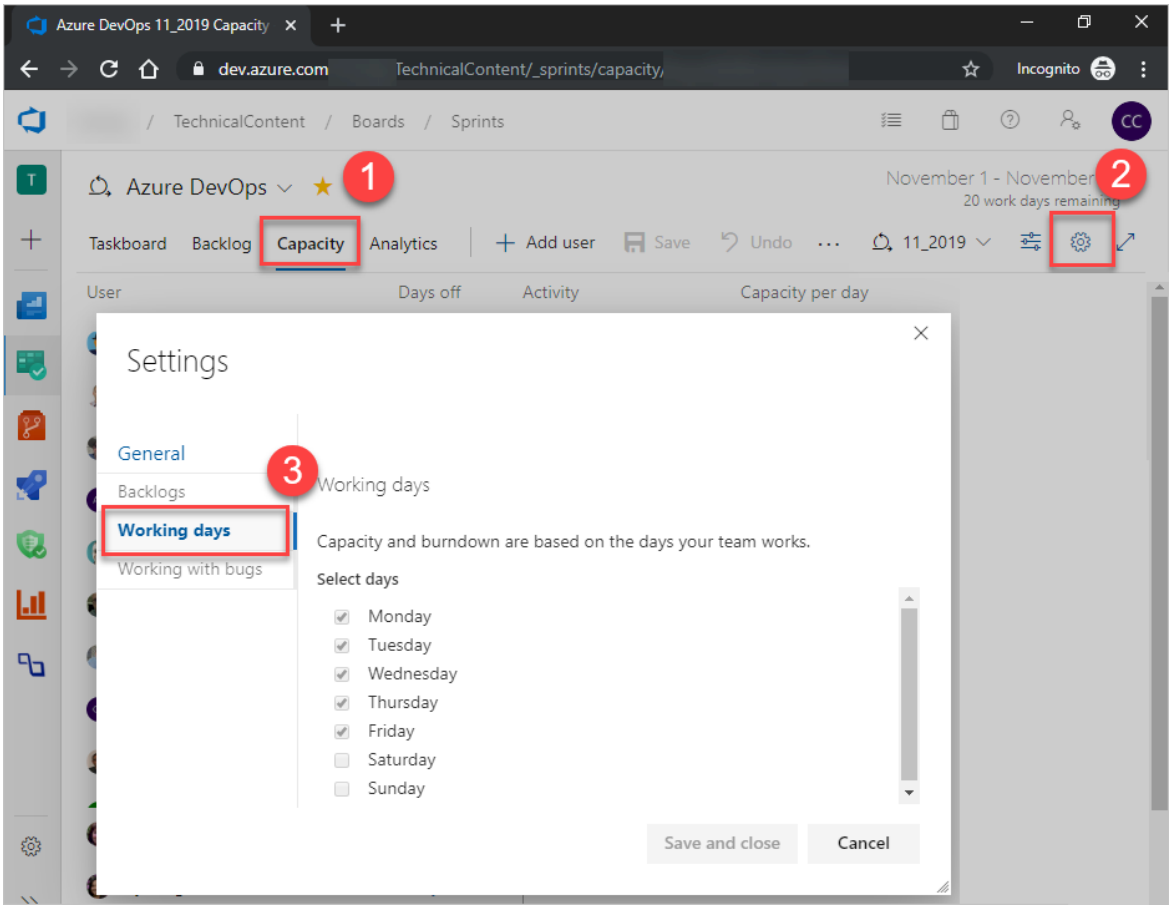
- To configure team settings, you must be added as a team administrator or be a member of the **Project Administrators** group. See Change project-level permissions.

Set working days

1. In **Azure Boards**, select the **Sprints** tab.



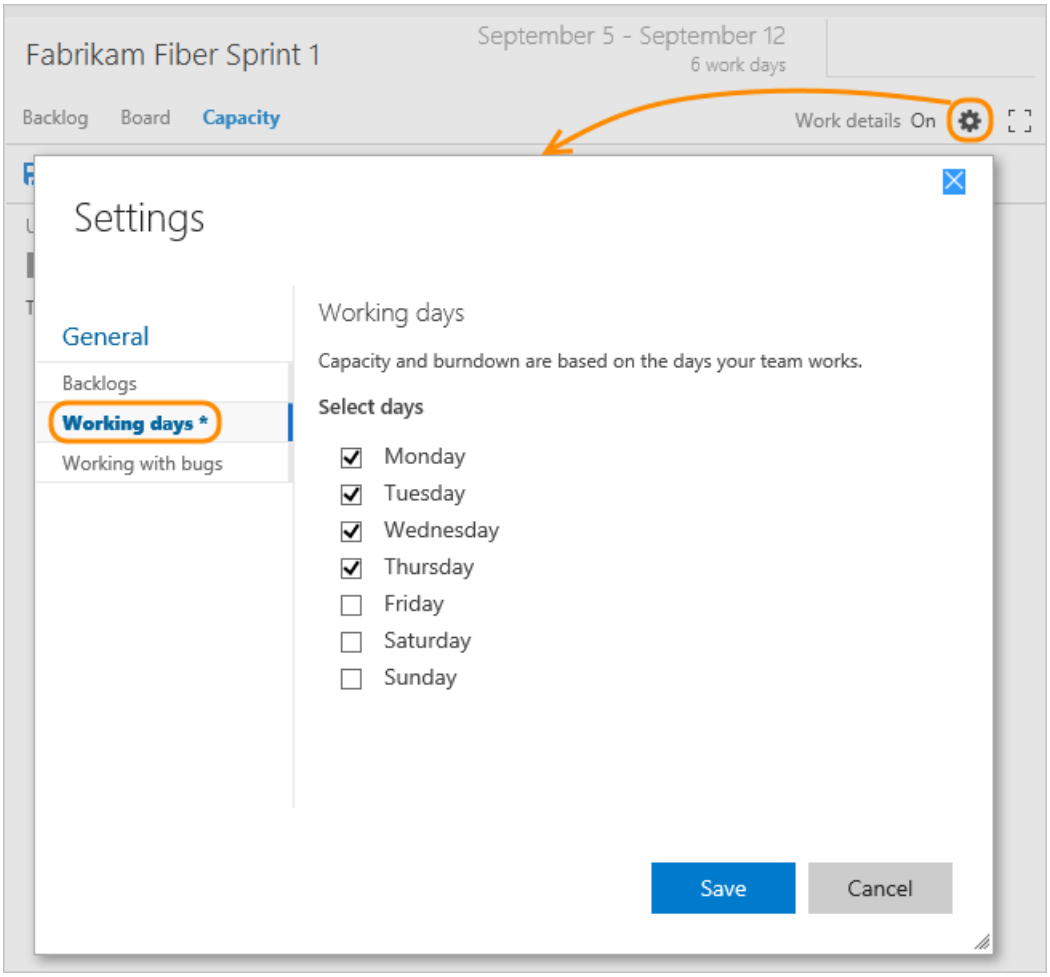
2. Select **Capacity** > Settings > **Working days**.



3. Select the checkbox next to the appropriate working days, and then select **Save and close**.

Your team's working days are updated.

1. From Azure Boards, select **Capacity** > ⚙ > **Working days**.



2. Select the checkbox next to the appropriate working days, and then select **Save and close**.

   Your team's working days are updated.

Related articles

- About Sprints, Scrum and project management
- Scrum and sprint planning tools
- Manage teams and configure team tools

Scrum and best practices

10/4/2022 • 12 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Sprint planning meetings

Much of sprint planning involves a negotiation between the product owner and the team to determine the focus and work to tackle in the upcoming sprint. It's useful to time-box the planning meeting, restricting it to 4 hours or less.

In the first part of the meeting, your product owner meets with your team to discuss the user stories that might be included in the sprint. Your product owner will share information and answer any questions that your team has about those stories. This conversation might reveal details such as data sources and user interface layout. Or it might reveal response time expectations, and considerations for security and usability. Your team should capture these details within the backlog items form. During this part of the meeting, your team learns what it must build.

As you plan your sprints, you may discover other requirements to capture and add to your backlog. Before your sprint planning meeting, you'll want to refine your backlog to make sure it's well defined and in priority order.

Also, setting a sprint goal as part of your planning efforts can help the team stay focused on what's most important for each sprint.

After you've planned your sprint, you may want to share the plan with key stakeholders.

You can learn more from these resources:

- What is Scrum?
- Sprint planning white paper
- The Scrum Guide
- Build and manage the product backlog white paper

Set sprint goals

Scrum teams use sprint goals to focus their sprint activities. They often set this goal during their sprint planning meeting. The goal summarizes what the team wants to accomplish by the end of the sprint. By explicitly stating the goal, you create shared understanding within the team of the core purpose. The sprint goal can also help guide the team when conflicts arise around priorities.

Tips from the trenches: Define sprint goals

The sprint goal defines what the product owner and the team consider as the ultimate target to accomplish that sprint. It's not a random selection of backlog items that don't really have a relationship, but a short piece of text that captures what the team should try to accomplish. Normally the product owner comes up with the sprint goal before selecting many items for the next sprint. The items for that sprint should all fit that common goal.

Sprint goals can be feature oriented, but might also have a large process component such as deployment automation or test automation.

For example:

- This sprint we'll focus on a simple user story. We'll use it to prove that the proposed solution will work.
- This sprint will revolve around implementing the security features that will properly secure the administration section of the website.
- This sprint will be about integrating the most important payment gateways so that we can start collecting money.

Setting the sprint goals helps the team to stay focused. It will make it easier to define priority of tasks within a sprint and it will probably help limit the number of stakeholders and end users that are involved.

During the sprint review, the most important question you should ask yourself is whether you managed to achieve the sprint goal. How many stories you completed comes second. If the goal is accomplished, the sprint succeeds, even if not all stories were finished.

Contributed by Jesse Houwing, Visual Studio devops Ranger and a senior consultant working for Avanade Netherlands.

Tips for successful triage meetings

Fixing bugs represents a trade-off with other work. Use your triage meeting to determine how important fixing each bug is against other priorities related to meeting the project scope, budget, and schedule.

- Establish the team's criteria for evaluating which bugs to fix and how to assign priority and severity. Bugs associated with features of significant value (or significant opportunity cost of delay), or other project risks, should be assigned higher priority and severity. Store your triage criteria with other team documents and update as needed.
- Use your triage criteria to determine which bugs to fix and how to set their State, Priority, Severity, and other fields.
- Adjust your triage criteria based on where you are in your development cycle. Early on, you may decide to fix most of the bugs that you triage. However, later in the cycle, you may raise the triage criteria to reduce the number of bugs that you need to fix.
- Once you've triaged a bug, assign it to a developer. The developer can then investigate and determine how to implement a fix.

Manage your technical debt

Consider managing your bug bar and technical debt as part of your team's overall set of continuous improvement activities. You may find these resources of interest:

- Good and Bad Technical Debt (and how TDD helps) by Henrik Kniberg
- Managing Technical Debt posted by Sven Johann & Eberhard Wolff

Tips from the trenches: Bug management

Agile Bug Management: Not an Oxymoron
by Gregg Boer, Principal Program Manager, Visual Studio Cloud Services at Microsoft

Address any known bug debt every sprint

Every sprint, the team looks at any bugs remaining in the bug backlog and dedicates capacity to get that known set of bugs down to zero, or near-zero. Whether this is one day, one week, or the entire sprint, they fix the bugs first. Bugs found later, within the sprint, aren't considered part of that initial commitment. Unless they're high priority, they're put on the bug backlog for the next sprint.

Many teams work in a commitment-based organization. Often, management places a high value on a team's ability to meet their commitments. Doing capacity planning against a known set of bugs makes sprint planning more deterministic, increasing their chance to meet commitments. Any new bugs discovered during the sprint

aren't a part of the initial commitment, and will be tackled next sprint.

Manage bug debt across an enterprise

An organization transitioning to a culture where debt is continually eliminated likely is dealing with the following question: How do you get teams to reduce their bug count without telling them exactly what to do? Leadership wants the team to change, yet gives the team autonomy to determine how they change. One option is to use a bug cap.

For example, consider a bug cap of three bugs per engineer. As such, a team of 10 people shouldn't have more than 30 bugs in its bug backlog. If the team is over its cap, it's expected to stop work on new features and get under the bug cap. A team is expected to be under its cap always, but the team decides how it wants to do that. The bug cap ensures that teams don't carry bug debt for too long. The team can learn from the mistakes that causes the bugs to be injected in the first place.

Remember that the bug cap represents the bugs in the bug backlog. It doesn't include bugs found and fixed within the sprint in which a feature is developed. Those bugs are considered undone work, not debt.

While bugs contribute to technical debt, they may not represent all debt.

Poor software design, poorly written code, or short-term fixes can all contribute to technical debt. Technical debt reflects extra development work that arises from all these problems.

Track work to address technical debt as PBIs, user stories, or bugs. To track a team's progress in incurring and addressing technical debt, you'll want to consider how to categorize the work item and the details you want to track. You can add tags to any work item to group it into a category of your choosing.

Role of the Scrum Master

Scrum Masters help build and maintain healthy teams by employing Scrum processes. They guide, coach, teach, and assist Scrum teams in the proper employment of Scrum methods. Scrum Masters also act as change agents to help teams overcome impediments and to drive the team toward significant productivity increases.

Core responsibilities of Scrum Masters include:

- Support the team to adopt and follow Scrum processes. For example, you shouldn't let the daily Scrum meeting become an open discussion that lasts 45 minutes.
- Guard against the product owner or team members from adding work after the sprint begins.
- Clear blocking issues that prevent the team from making forward progress. This work might require you to complete small tasks, such as approving a purchase order for a new build computer or resolving a conflict within your team.
- Help the team work to resolve conflicts and issues that arise and learn from the process.
- Ask questions that reveal hidden issues and confirm that what people are communicating is well understood by the entire team.
- Identify and safeguard the team from potential issues becoming major issues. Just as it's cheaper to fix a bug soon after it's discovered, it's also easier and less disruptive to fix a team issue when it's small and manageable.
- Prevent the team from presenting incomplete user stories during a sprint review meeting.
- Gather, analyze, and present data to business stakeholders to demonstrate how the team is improving and growing. For example, if your team has increased the amount of value that it has delivered while generating fewer bugs, make the value visible through regular communications to business stakeholders.

Good Scrum Masters have or develop excellent communication, negotiation, and conflict resolution skills. They actively listen to the words that people say and write. They also listen to how they deliver their messages, for example their body language, facial expressions, vocal pace, and other nonverbal communication.

Just as it's cheaper to fix a bug soon after it's discovered, it's also easier and less disruptive to fix a team issue when it's small and manageable before it grows into a major issue.

Daily Scrum meetings

Daily Scrum meetings help keep a team focused on what it needs to do the next day. Staying focused helps the team maximize their ability to meet sprint commitments. Your Scrum Master should enforce the structure of the meeting and ensure that it starts on time and finishes in 15 minutes or less.

Three aspects of successful Scrum meetings are:

- Everyone stands up. Standing up helps to keep the meetings focused and short.
- They start and end on time and occur at the same time in the same location each day
- Everyone participates, each team member answers the three Scrum questions:
  - *What have I accomplished since the most recent Scrum?*
  - *What will I accomplish before the next Scrum?*
  - *What blocking issues or impediments might affect my work?*

> **NOTE**
> The focus for scrum meetings is on the status of work that needs to be passed from one team member to another team member.

Team members should strive to answer their questions quickly and concisely. For example:

> *"Yesterday, I updated the class to reflect the new data element that we pull from the database, and I got it to appear in the interface. This task is complete. Today, I will ensure that the new data element is correctly calculating with the stored procedure and the other data elements in the table. I believe I will accomplish this task today. I will need someone to review my calculations. I have no impediments or blocking issues."*

This response conveys what the team member accomplished, what the team member will accomplish, and that the team member would like some help looking at the code.

Contrast with this next example:

> *"Yesterday, I worked on the class, and it works. Today, I will work on the interface. No blocking issues."*

The team member doesn't provide enough detail about what class they worked on nor about the interface components they'll complete. In fact, the word accomplished never came up.

It's important that no one interrupts during report outs. Each person must have sufficient time to answer the three questions.

More in-depth and follow-up discussions should take place after the meeting, as people return to their desks or, if a significant amount of conversation is necessary, in a follow-up meeting.

Many teams delay discussions by using the "virtual parking lot" method. As subjects come up that a team member thinks needs further discussion, they can quietly walk to a whiteboard or flipchart and list the subject in the parking lot. At the end of the meeting, the team determines how they'll handle the listed items.

Sprint review meetings

Conduct your sprint review meetings on the last day of the sprint. Your team demonstrates each product backlog item that it completed in the sprint. The product owner, customers, and stakeholders accept the user stories that meet their expectations and identify any new requirements. Customers often understand their needs more fully after seeing the demonstrations and may identify changes that they want to see.

Based on this meeting, some user stories will be accepted as complete. Incomplete user stories will remain in the product backlog. New user stories will be added to the backlog. Both sets of stories will be ranked and either estimated or re-estimated in the next sprint planning meeting.

After this meeting and the retrospective meeting, your team will plan the next sprint. Because business needs change quickly, you can take advantage of this meeting with your product owner, customers, and stakeholders to review the priorities of the product backlog again.

Sprint retrospective meetings

Retrospectives, when conducted well and at regular intervals, support continuous improvement.

The sprint retrospective meeting typically occurs on the last day of the sprint, after the sprint review meeting. In this meeting, your team explores its execution of Scrum and what might need tweaking.

Based on discussions, your team might change one or more processes to improve its own effectiveness, productivity, quality, and satisfaction. This meeting and the resulting improvements are critical to the agile principle of self-organization.

Look to address these areas during your team sprint retrospectives:

- Issues that affected your team's general effectiveness, productivity, and quality.
- Elements that affected your team's overall satisfaction and project flow.
- What happened to cause incomplete backlog items? What actions will the team take to prevent these issues in the future?

  For example, consider a team that had several tasks that only one individual on the team could do. The isolated expertise created a critical path that threatened the sprint's success. The individual team member put in extra hours while other team members were frustrated they couldn't do more to help. Going forward, the team decided to practice eXtreme Programming to help correct this problem over time.

As a team, work to determine whether to adapt one or more processes to minimize the occurrence of problems during the sprint.

Your team may need to do some work to implement an improvement. For example, a team that found themselves negatively affected by too many failed builds decided to implement continuous integration. Because they didn't want to disrupt process, they took a few hours to set up a trial build before turning it on in their production build. To represent this work, they created a spike and organized that work against the rest of the product backlog.

Related articles

- What is Scrum?
- Agile Retrospectives: Making Good Teams Great

Sprints and Scrum key concepts in Azure Boards

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This article provides a short dictionary of terms and available tools used in tracking work using Sprints and Scrum methods. See also:

- Agile glossary
- Work item field index
- Project management and navigation glossary

Agile tools

A suite of web-based tools used to track work and support Agile methodologies. Agile tools support the core Agile methods—Scrum and Kanban—used by software development teams today. Learn more: About Agile tools and Agile project management.

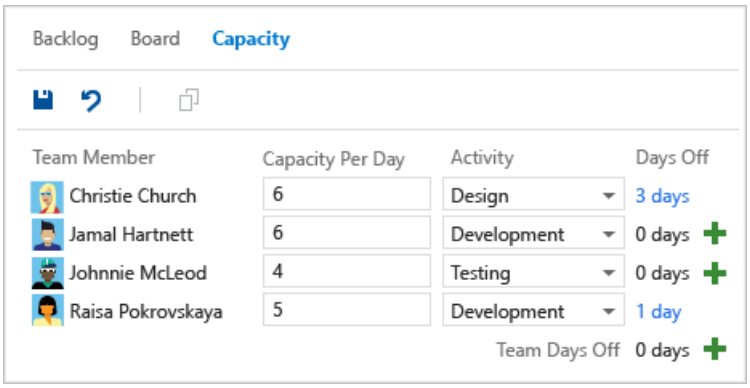
Bugs

A type of work item that records a potential source of dissatisfaction with the product. The common name of a work item type for tracking code defects. Each team can choose how they want to manage bugs. Some teams like to track bugs along with requirements on the backlog. Other teams like to track bugs as tasks performed in support of a requirement. The bugs then appear on their Taskboard. Learn more: Manage bugs.

Team and individual capacity

Capacity correlates to actual task time, either hours or days, that an individual or a team has to work. Azure DevOps provides a Capacity tool for each team's sprint to set capacity. Teams typically set capacity when they plan to create tasks and estimate the time it takes to complete a task.

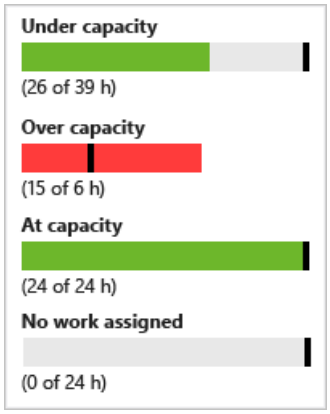
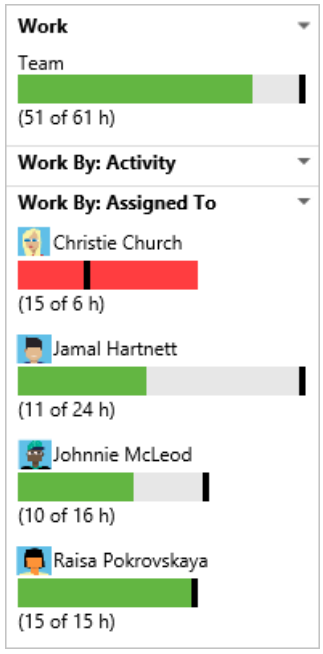
By setting team capacity, the team knows exactly the total number of work hours or days the team has for each sprint. With this tool, you set individual team member capacity and days off. Setting capacity for each team member working during a sprint causes the capacity bar for that individual to appear. Learn more: Set sprint capacity.



Capacity bars

With capacity bars, you can quickly see who is over, at, or under capacity. Capacity bars update with each of these activities:

- Tasks are assigned with non-zero remaining work
- Change in remaining work
- Date change within the sprint cycle. Individual and team capacity always reflects their capacity from the current day until the end of the sprint.

CAPACITY COLORS	CAPACITY BARS
	

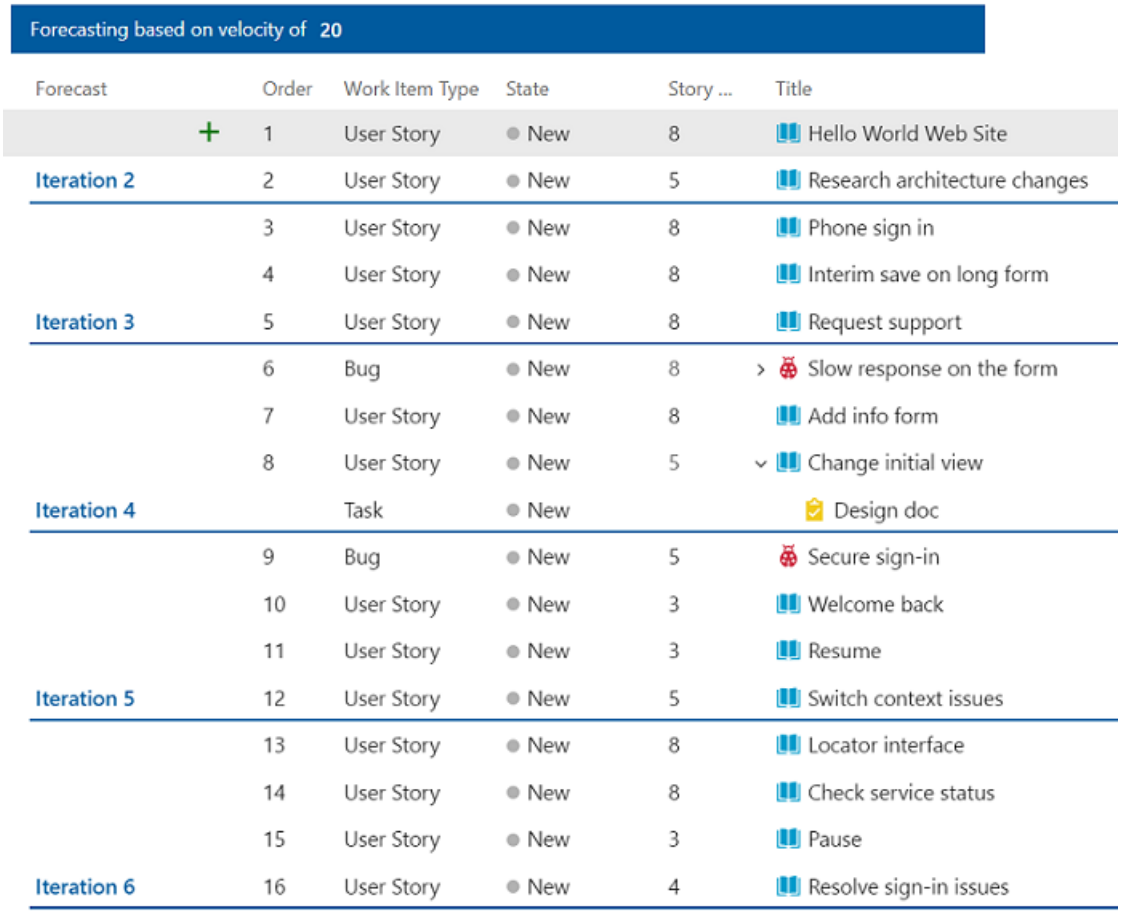
Learn more: Adjust work to fit sprint capacity.

Daily scrum meetings

Daily Scrum meetings help teams stay focused on what they need to do to maximize their ability to meet their sprint commitments. The team's Scrum Master should enforce the structure of the meeting and ensure that it starts on time and finishes in 15 minutes or less. Learn more: Scrum best practices, Daily scrum meeting.

Forecast

The forecast tool helps teams plan their sprints. The tool shows teams the backlog items that can be completed in future sprints based on work item estimates and a set velocity. As shown here, a velocity of 20 indicates that it will take five sprints to complete the work shown. Learn more: Forecast your product backlog.



Iteration paths (aka sprints)

A time period, usually two to three weeks, used to group work items to be completed during that time period. Sprints are used in Scrum methods to support sprint planning, sprint burndown, and other Scrum processes. Iteration paths allow you to group work into sprints, milestones, or other event-specific or time-related period. Learn more: About area and iteration paths.

Product backlog

An interactive list of work items that corresponds to a team's project plan or roadmap for what the team plans to deliver. The product backlog supports prioritizing work, forecasting work by sprints, and quickly linking work to portfolio backlog items. You can define your backlog items and then manage their status using the Kanban board.

Each product backlog can be customized by a team. Learn more: Create your backlog.

Product backlog item

A type of work item that defines the applications, requirements, and elements that teams plan to create. Product owners typically define and stack rank product backlog items which are defined with the Scrum process. Learn more: Scrum process work item types and workflow.

Product owner role

The role of product owners is to act as the interface between customers and the team. A product owner can reduce the need for detailed specifications. They reduce the need by being more responsive to the team's questions about implementation details. Also, they clearly define acceptance criteria within each requirement. Learn more: Refine your backlog, Role of the product owner.

Scrum Master role

Scrum Masters help build and maintain healthy teams by employing Scrum processes. They guide, coach, teach, and assist Scrum teams in the proper employment of Scrum methods. Scrum Masters also act as change agents to help teams overcome impediments and to drive the team toward significant productivity increases. Learn more: Scrum best practices, Role of the Scrum Master.

Sprints (also known as iterations)

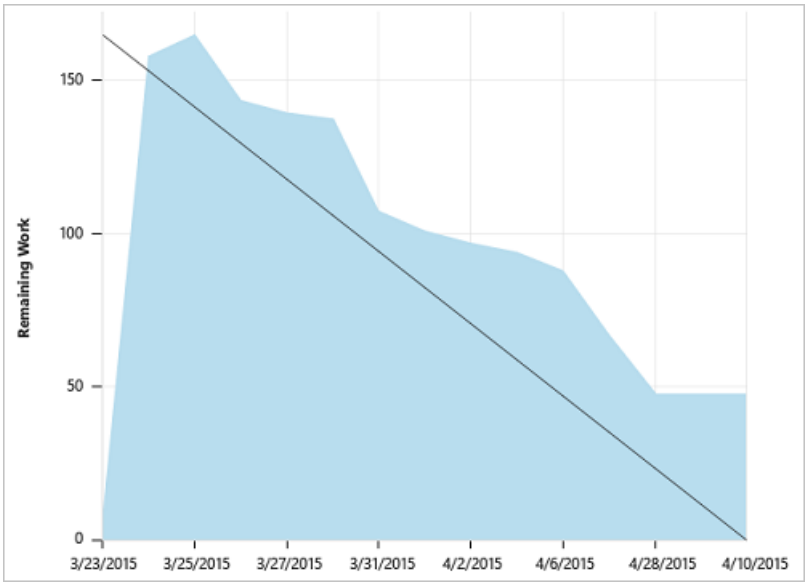
A sprint is a time period of usually two to three weeks that's used to group work items to be completed during that time period. Sprints are used in Scrum methods to support sprint planning, sprint burndown, and other Scrum processes. Sprints are defined via iteration paths. To learn more, see About area and iteration paths (aka sprints).

Sprint backlog

An interactive list of work items that have been assigned to the same sprint or iteration path for a team. The sprint backlog supports teams that use Scrum methodologies. Learn more: Sprint planning.

Sprint burndown chart

The sprint burndown chart reflects the progress made by a team in completing all the work they estimated during their sprint planning meeting. Team's monitor it to mitigate risk and check for scope creep throughout their sprint cycle. The ideal trend line always indicates a steady burndown. The blue area, as shown in the following chart, represents what's actually going on. It shows the buildup of work as team members add tasks and the reduction of work as team members complete those tasks. Learn more: Monitor sprint burndown.



Sprint goals

Sprint goals are used to focus sprint activities. The goal summarizes what the team wants to accomplish by the end of the sprint. Learn more: Scrum best practices, Set sprint goals.

Sprint planning

The Sprint planning meeting occurs at the start of a sprint and is when the product owner and team agree on a set of sprint goals and work. Learn more: Scrum best practices, Sprint planning meetings.

Sprint retrospective meetings

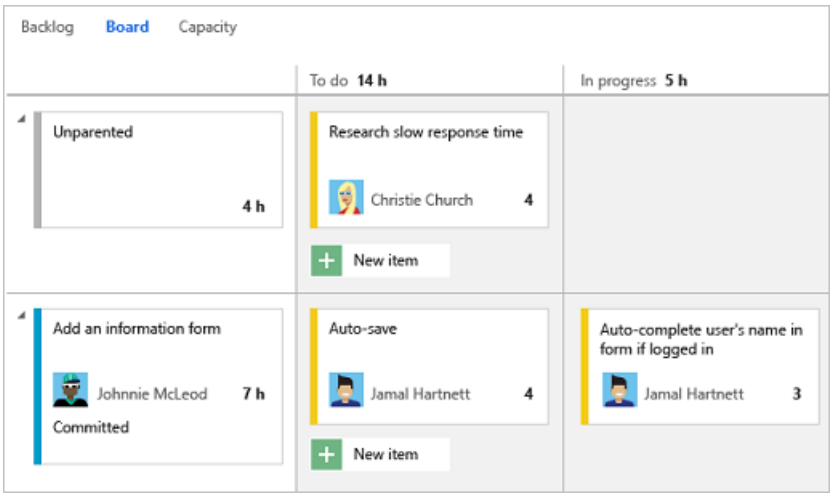
The Sprint review or retrospective meeting occurs at the end of a sprint. This meeting is when the team demonstrates the work that they completed during the sprint. The product owner, customers, and stakeholders accept the user stories that meet their expectations and identify any new requirements. Customers often understand their needs more fully after seeing the demonstrations and may identify changes that they want to see. Learn more: Scrum best practices, Sprint retrospective meeting.

Task

A task is a type of work item used to track estimated and remaining work. In Scrum, a task is defined to range between four and twelve hours. Defining tasks are essential for monitoring sprint burndown, working with team capacity, and using the Taskboard. Tasks are linked to their parent product backlog items or user stories. Learn more: Add tasks to backlog items.

Taskboard

A taskboard provides an interactive progress board for work required to complete a team's sprint backlog. During your sprint, you'll want to update the status of tasks and the remaining work for each task. Updating tasks daily or several times a week yields a smoother sprint burndown chart. Learn more: Taskboard.



Teams

A team corresponds to a selected set of project members. With teams, organizations can subcategorize work to better focus on all the work they track within a project. Each team gets access to a suite of Agile tools. Teams can use these tools to work autonomously and collaborate with other teams across the enterprise. Each team can configure and customize each tool to meet their work requirements. To learn more, see About teams and Agile tools.

Team member

A member who has been added to a project or organization who has been added to a specific team. Project members can be added to several teams. Several Agile tools, such as capacity planning, team alerts, and dashboard widgets are team-scoped. That is, they automatically reference the users that have been added as members of a team to support planning activities or sending alerts.

To add users to a team, see Add users to a project or specific team.

Technical debt

Technical debt includes anything the team must do to deploy production quality code and keep it running in production. Examples are bugs, performance issues, operational issues, accessibility, and others. Learn more about how to minimize technical debt: What is Agile Development?.

Triage meetings

Triage meetings are used to review and organize the backlog and bugs assigned to a team. Other details, such as estimates, acceptance criteria, and more may be added to the work items. Typically, a product owner runs triage meetings, and team leads, business analysts, and other stakeholders who can speak about specific project risks attend them. Learn more: Triage work items.

User story

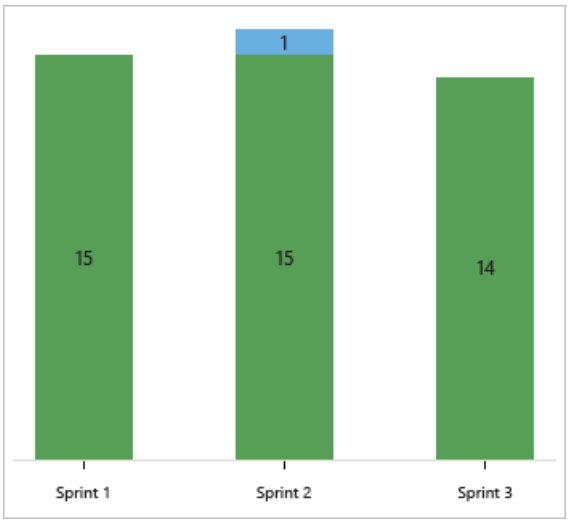
A type of work item that defines the applications, requirements, and elements that teams plan to create. Product owners typically define and stack rank user stories. User story is defined with the Agile process. Learn more: Agile process work item types and workflow.

Velocity and velocity chart

Velocity provides a useful metric for gaining insight into how much work your team can complete during a sprint cycle. After your team has worked several sprints, they can use the velocity chart and forecast tool to estimate work that can be accomplished in future sprints.

Velocity is a measure of how much work a team can complete based on their sprint cadence. The built-in velocity chart measures velocity by summing the Story Points (Agile), Effort (Scrum), or Size (CMMI) defined for a sprint.

For example, in the chart shown below the green bar indicates the total estimated effort (story points) of the user stories completed within each sprint. Blue corresponds to the estimated effort of items not yet completed. Learn more: View and work with the built-in team velocity chart.



Along with the built-in Velocity chart, you can add a Velocity widget to your team dashboard. You can configure this widget to sum a count of work items or the sum of effort. Learn more: Configure the Velocity widget.

Each team is associated with one and only one velocity chart. Velocity will vary depending on team capacity, sprint over sprint. However, over time, the velocity should indicate a reliable average that can be used to forecast the full backlog. By minimizing the variability of backlog item size—effort or story points—you gain more reliable velocity metrics. Learn more: Add tasks to backlog items.

Related articles

- Refine your backlog

- Scrum best practices.
- About Sprints and Scrum
- What is Scrum?

Configure and monitor sprint burndown

10/4/2022 • 17 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Throughout your sprint, you can monitor the sprint burndown report to determine if your team is on track to complete its sprint plan. There are two sprint accessible burndown charts: the in-context Burndown Trend report viewable from a team sprint backlog and the Sprint Burndown widget you can add to a dashboard.

Both the report and the widget derive data from Analytics. They support monitoring burndown based on a count of work items or a sum of Story Points/Size/Effort, Remaining Work, or other numeric field.

You can add either the report or widget to a dashboard. Also, you can monitor progress using the Analytics-based burndown or burnup widgets. They provide more configuration options.

Throughout your sprint, you can monitor the sprint burndown report to determine if your team is on track to complete its sprint plan. The in-context sprint burndown report supports tracking burndown based on Remaining Work assigned to sprint tasks. If you don't track tasks or Remaining Work, then you can use the Analytics-based burndown and burnup widgets. They provide more configuration options.

Throughout your sprint, you can monitor the sprint burndown chart to determine if your team is on track to complete its sprint plan. Both the in-context sprint burndown report and the Sprint Burndown widget support tracking burndown based on Remaining Work assigned to sprint tasks.

> **NOTE**
> You can't add an in-context report to a dashboard. However, you can add the Sprint burndown widget or the Analytics-based burndown or burnup widgets to a dashboard.

> **NOTE**
> You can't add an in-context report to a dashboard. However, you can add the Sprint burndown widget to a dashboard.

Use this article to learn about:

- Metrics tracked in the sprint burndown report and widget
- Team activities required to track tasks and Remaining Work
- How to set interactive controls to personalize your view of the sprint burndown report
- How to add the in-context Burndown Trend report to a dashboard
- How to configure the Sprint Burndown widgets
- How to view current and past sprint burndowns

For an overview of all burndown/burnup charts available to you, see Burndown and burnup guidance.

- Metrics tracked in the sprint burndown report and widget
- Team activities required to track tasks and Remaining Work
- How to configure the Sprint Burndown widget
- How to view current and past sprint burndowns

NOTE

Sprint burndown reports are derived from data tracked by a team during a sprint or iteration. To learn more, see About teams and Agile tools.

The in-context Burndown Trend report

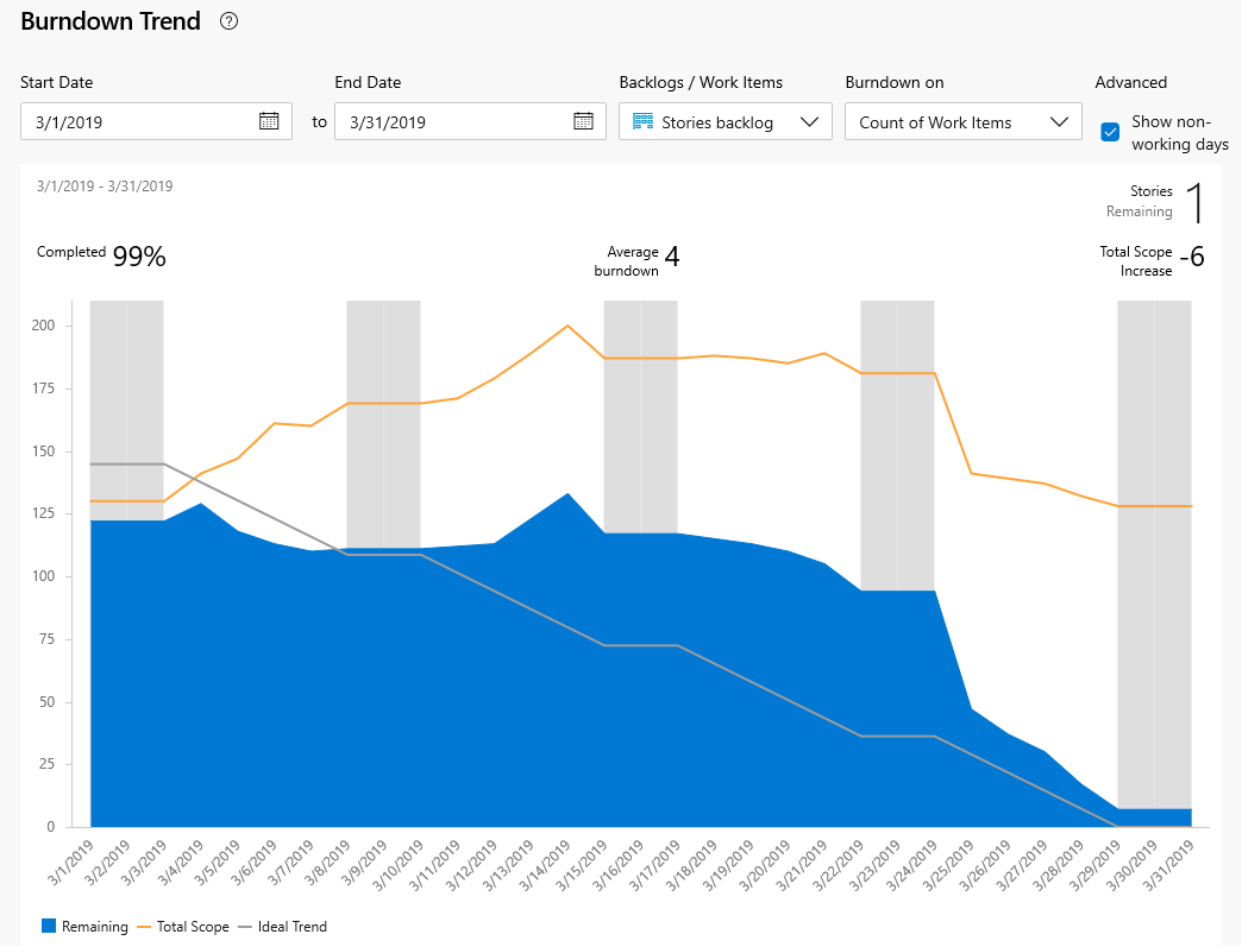
The Burndown Trend report is based on either a count of tasks or remaining work estimates, or other numeric fields that you define and update throughout the sprint cycle. For details, see Sprint planning. To open this report, jump to the section Open a Sprint backlog.

A healthy sprint burndown report looks something like the image shown below. Typically, there will be a stair-case burndown as individual team members may only update their work items once a week or every few days. The **Total Scope** line indicates the number of work items added after the sprint starts. The **Ideal Trend** line is calculated based on the number of work items, days in the sprint, and number of working days.

The blue area indicates the number of work items active or in progress each day of the sprint. As shown in this example, one work item is still active at the end of the sprint.

NOTE

The **Show non-working days** shades those days set through the team's **Working days** setting as well as the team's days off set through the **Capacity** page.



NOTE

The **Total Scope** line reflects the number of work items added to the sprint. If the team's default iteration is the **@CurrentIteration**, then new work items are added to the current iteration. The scope decreases as the Iteration Path is modified to another sprint, or work items are completed.

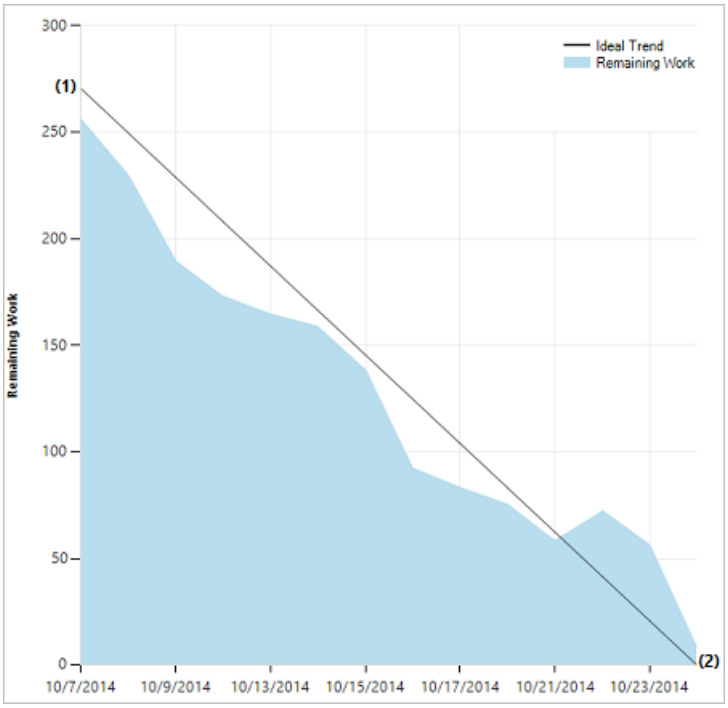
The in-context sprint burndown report is based on the tasks and Remaining Work estimates you define and update throughout the sprint cycle. For details, see Sprint planning and taskboard. To open the sprint burndown chart, jump to the section Open sprint burndown chart.

A healthy sprint burndown chart will look something like this. The *Ideal Trend* line connects the two points:

- **(1)** Team's total capacity at the start of the sprint.
- **(2)** 0 Remaining Work at the end of the sprint.

The slope represents the rate at which the team needs to burn down work to finish the sprint on time.

The actual graph, the blue area, represents the total amount of planned sprint work and how it changes throughout the course of the sprint. The blue area corresponds to the sum of all Remaining Work set for all sprint tasks, and possibly bugs, that have the current sprint as their iteration path.

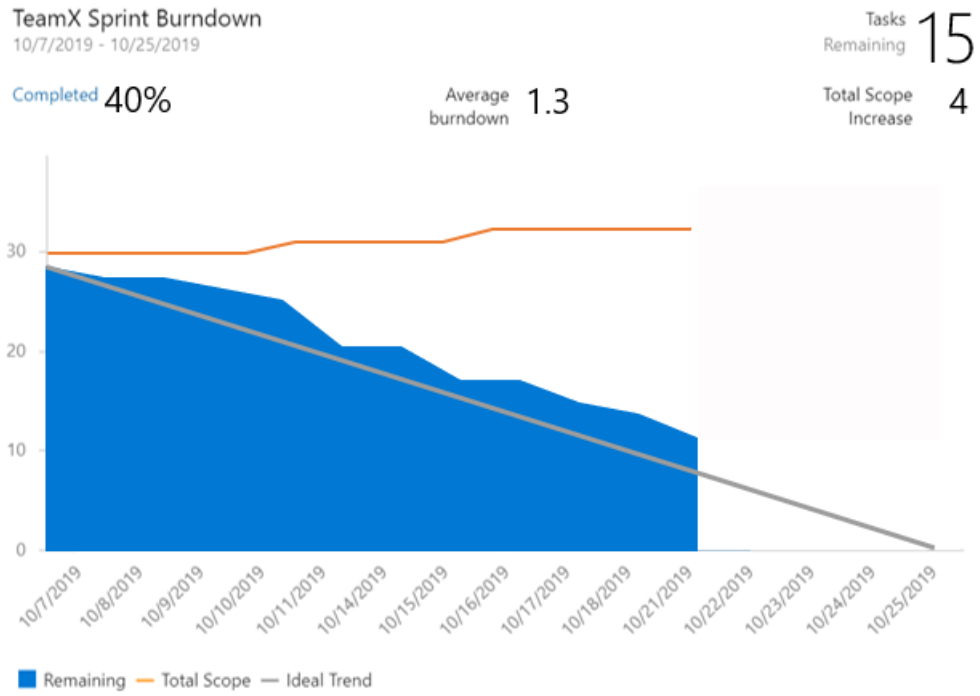


The Sprint Burndown widget

In the widget catalog, you'll find two versions of the Sprint Burndown widget: the Analytics-based **Sprint Burndown** and **Sprint Burndown (Legacy)** which is built from the work tracking data store.

Sprint Burndown widget

The Analytics-based Sprint Burndown widget provides an easy way to monitor progress for a team by showing work remaining for a given sprint. Work remaining is the vertical axis and time is the horizontal axis. You can define remaining work based on Stories or Tasks, and by counting the work items or summing a field.

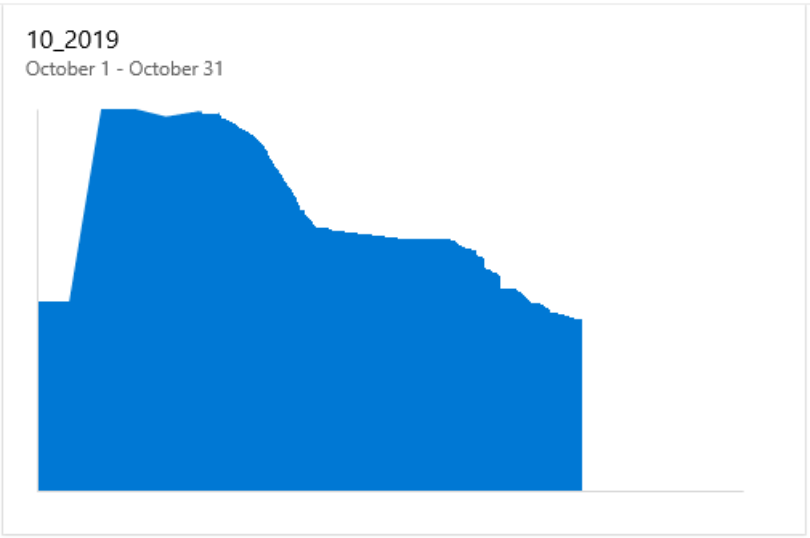


The charts provide useful metrics to help you answer the question: *Are we on track to complete this set of work by the end a sprint?*

- Percentage work complete
- Number of work items not estimated (if using a field other than Remaining Work)
- Average burndown
- Total scope increase

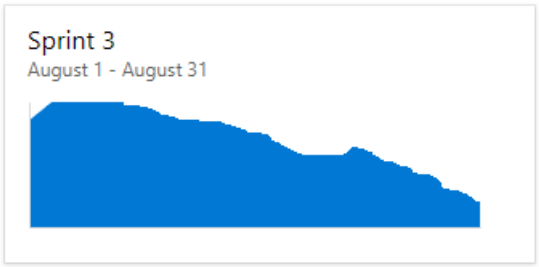
Sprint Burndown (Legacy) widget

The **Sprint Burndown (Legacy)** widget adds a chart based on Remaining Work defined for tasks in a team's current sprint. Select this version when you don't have access to Analytics. Configuration options include team selection, and widget size.



If your dashboard already has a legacy version available, you can easily upgrade the widget by editing the widget's configuration, and checking **Try the new version now**. You can always go back to the legacy version by unchecking the box.

The **Sprint Burndown** widget adds a chart based on Remaining Work defined for tasks in the team's current sprint. There are no configuration options for this widget.



Prerequisites

- You must be a member of a project. If you don't have a team project yet, create one.
- If you haven't been added as a project member, get added now.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Have enabled or installed Analytics. You must be an account owner or a member of the **Project Collection Administrators** group to add extensions or enable the service.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin.

Team activities to track tasks and Remaining Work

To monitor sprint burndown, your team must schedule sprints and assign work to those sprints.

If you want to monitor sprint burndown based on tasks and Remaining Work, your team must carry out these extra actions.

To monitor sprint burndown, your team must schedule sprints and assign work to those sprints. For sprint burndown charts to show meaningful data, your team must carry out these extra actions.

Required activities:

- Define and estimate tasks for each product backlog item you're working on in the sprint. If you work from your team's backlog and taskboard, the items you create will automatically be assigned to the current sprint (Iteration) and to your team's default Area Path.
- Update Remaining Work for each sprint task as work progresses.

Recommended activities:

- Define tasks that take a day or less to complete to lessen the effect of poor estimates.
- Don't divide tasks into subtasks. If you divide a task into subtasks, specify hours only for the subtasks. These

hours are rolled up as summary values for the parent task.

- Update Remaining Work daily or several times within a week to support monitoring and achieve a smoother burndown chart.
- At the end of the sprint, update the task status of completed tasks and determine how to handle incomplete tasks.

Empty sprint burndown chart

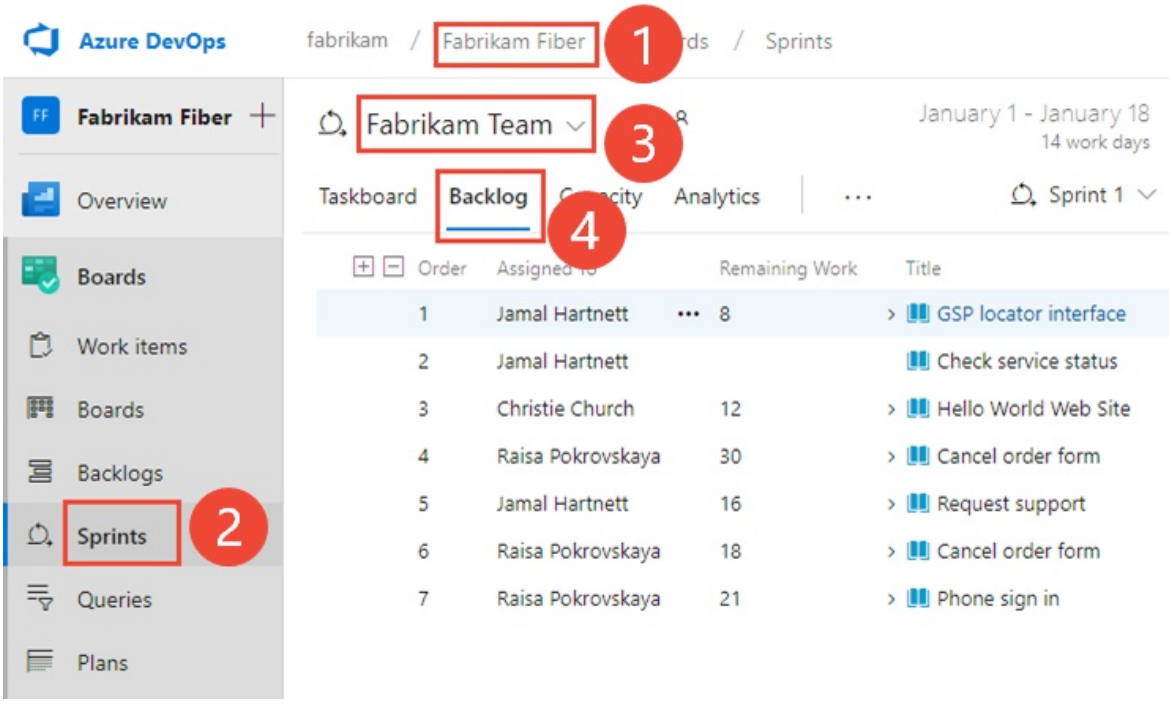
If your sprint burndown chart appears empty, check these points:

- Have you assigned tasks to the sprint associated with the chart?
- Have you assigned Remaining Work to the tasks assigned to the sprint?
- Are the parent work items of the tasks assigned to the same sprint? If not, the tasks may appear in another sprint associated with the parent item.

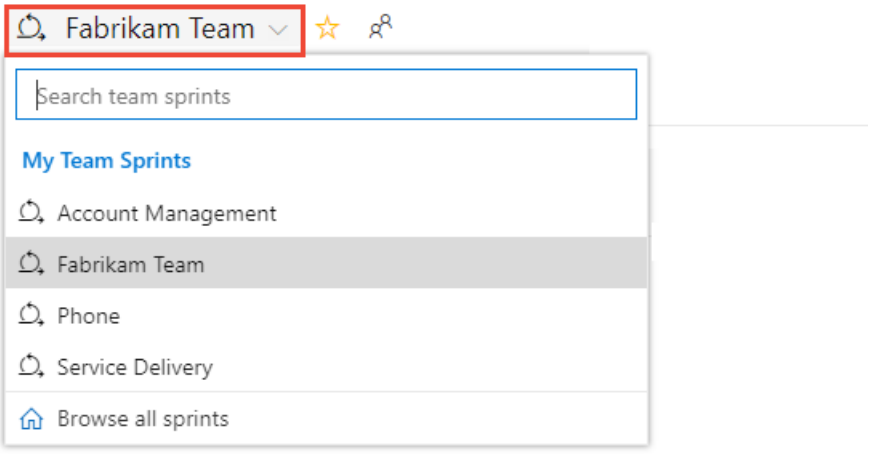
Open a Sprint backlog

You view the in-context sprint burndown report from a team's Sprint backlog.

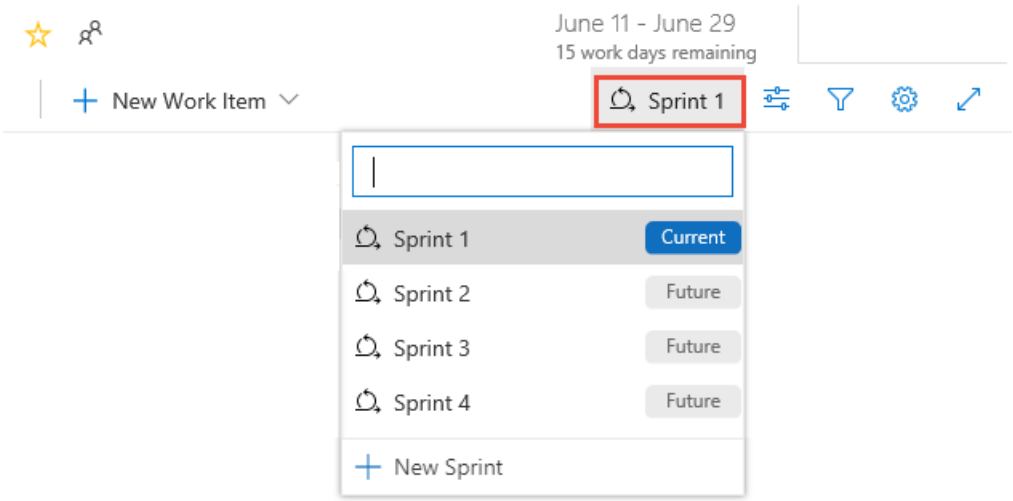
1. From your web portal, open your team's sprint backlog. (1) Check that you've selected the right project, (2) select **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), select **Backlog**.



To select another team, open the selector and select a different team or select the **View Sprint directory** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

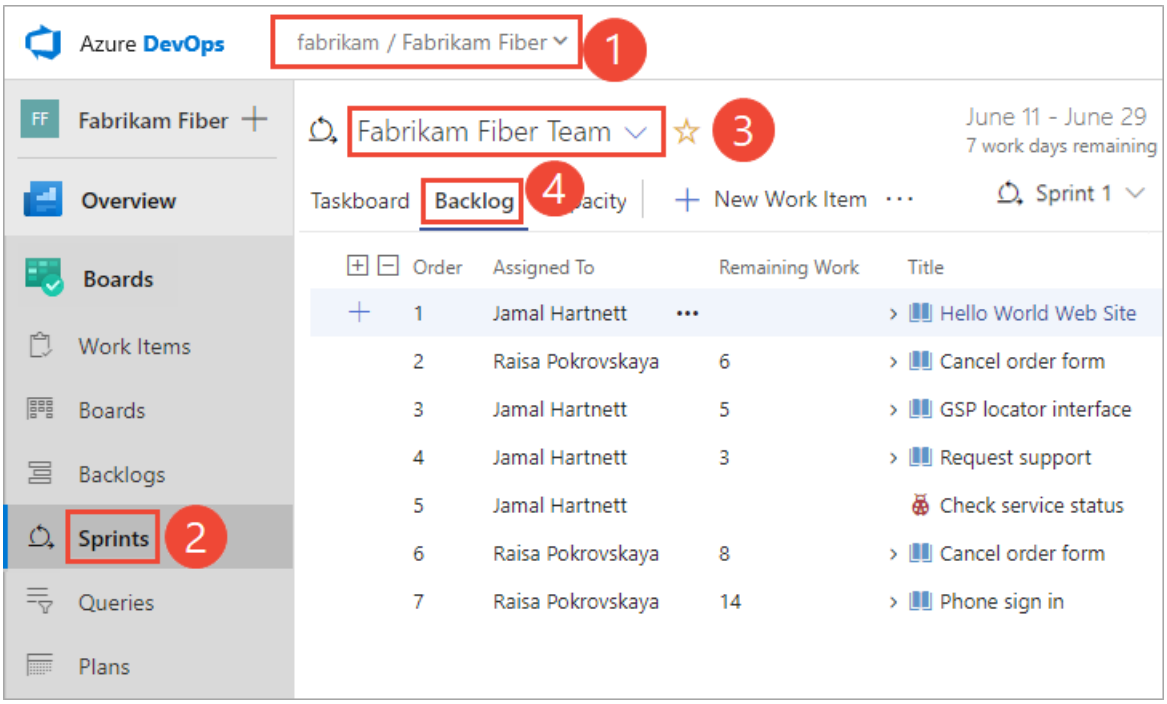


2. To select a different sprint than the one shown, open the sprint selector and select the sprint you want.

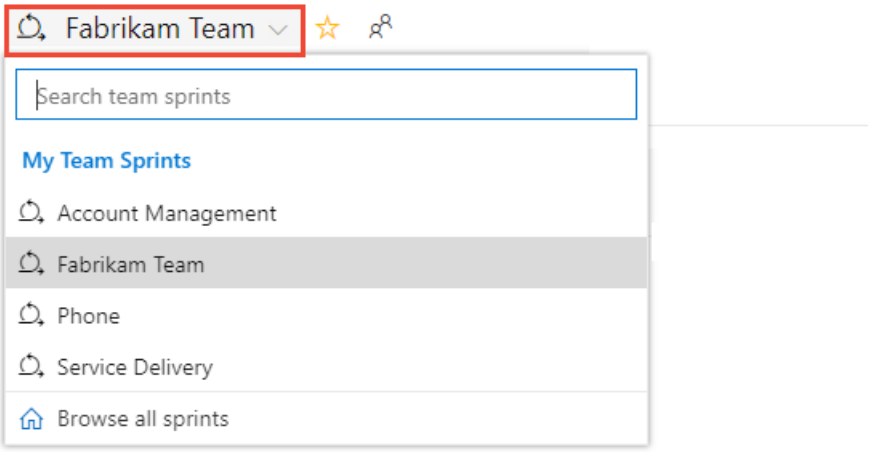


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then select **New Sprint** from the menu, and then select **Select existing iteration**. For details, see Define iteration paths.

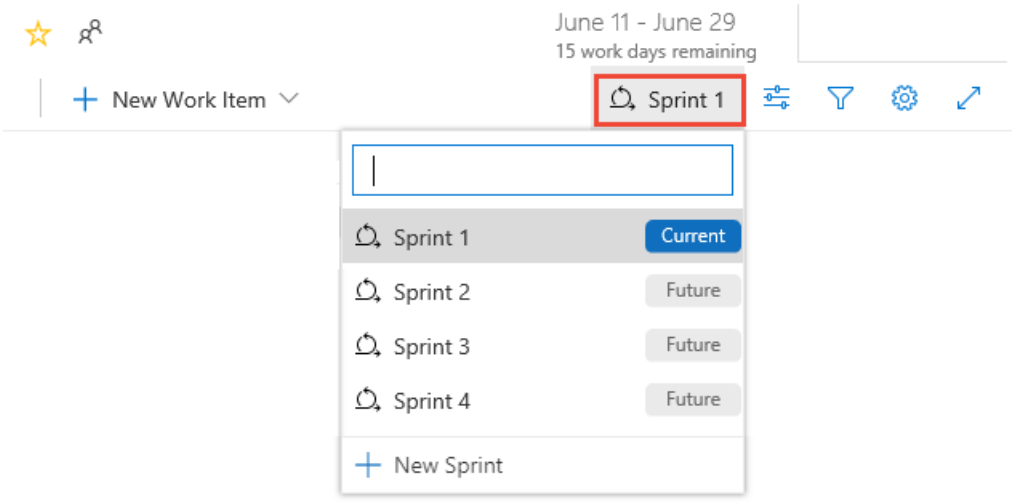
1. From your web browser, open your team's sprint backlog. (1) Check that you've selected the right project, (2) select **Boards > Sprints**, (3) select the correct team from the team selector menu, and lastly (4), select **Backlog**.



To select another team, open the selector and select a different team or select the **Browse all sprints** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

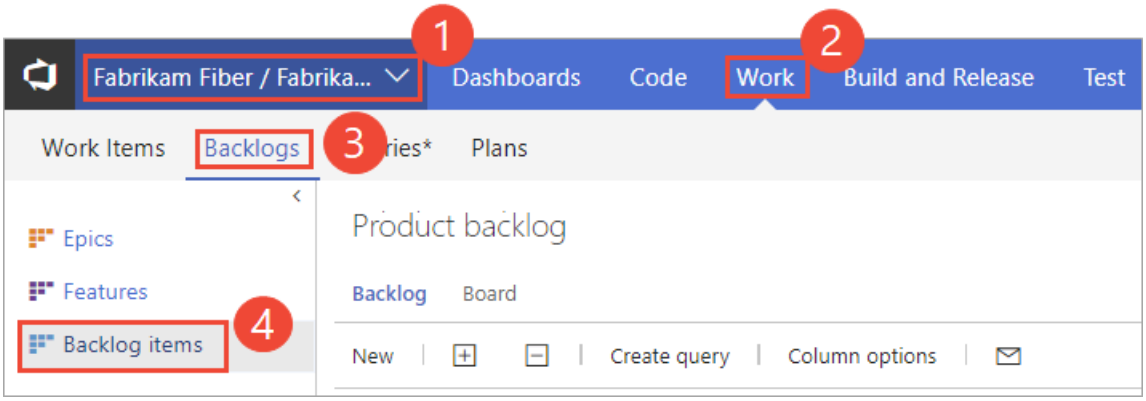


2. To select a different sprint than the one shown, open the sprint selector and select the sprint you want.

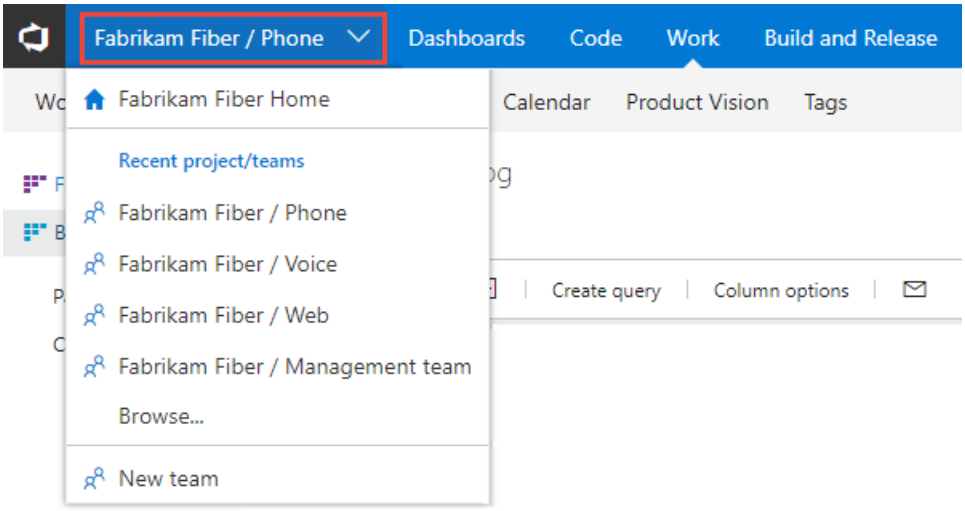


The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then select **New Sprint** from the menu, and then select **Select existing iteration**. For details, see Define iteration paths.

1. From your web browser, open your team's sprint backlog. (1) Select the team from the project/team selector, select (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

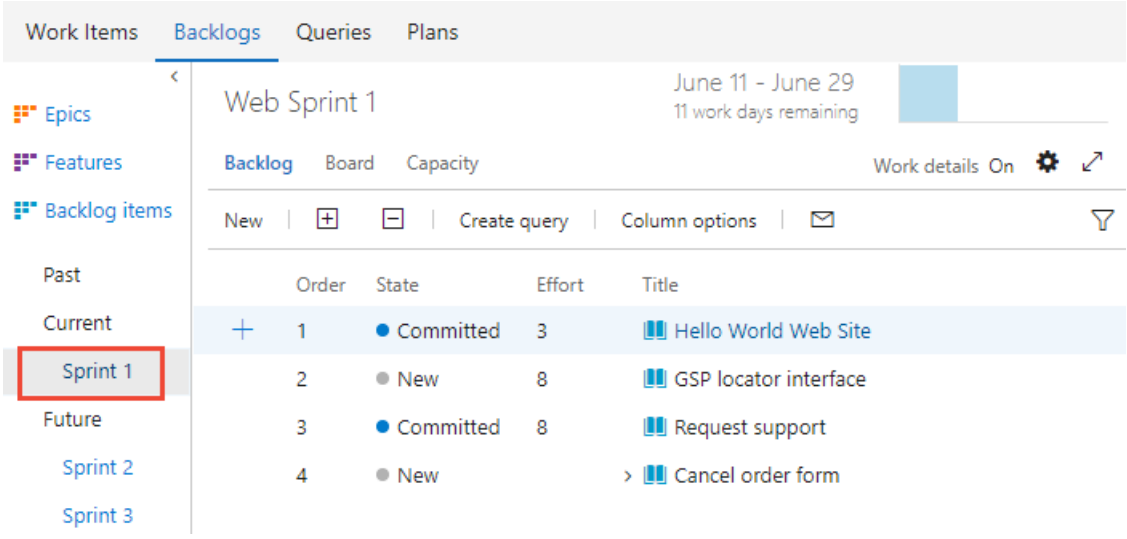


To select another team, open the project/team selector and select a different team or select the **Browse** option.



The set of sprints selected for your team appears in the left pane. If you don't see any sprints listed, you can add sprints or select existing sprints for your team's use. To learn how, see Define sprints.

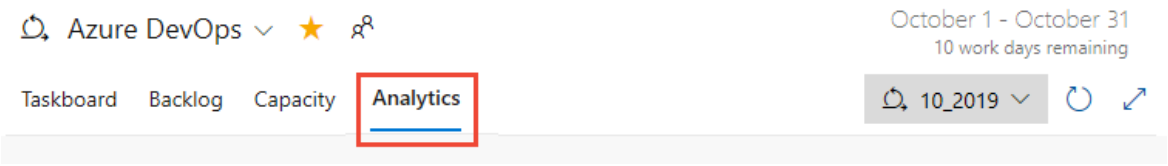
2. Select the sprint whose burndown chart you want to view.



The system lists only those sprints that have been selected for the current team focus. If you don't see the sprints you want listed, then see Define iteration paths.

View the in-context Burndown Trend report

1. To open the Sprint burndown report, select **Analytics**.



2. Use the interactive controls to select from the following options:

   a. The **Start Date** and **End Date** of the sprint. These dates will default to the team's current iteration sprint dates.

   b. The **Backlogs/Work Items** to burn down on, either the product backlog—Stories, Issues, Product Backlog Items, or Requirements—or Tasks backlog to use. Your selection impacts the options available for the **Burndown on** menu.

   c. The **Burndown on** field to use to calculate burndown, either a Count of Work Items or a sum of a field, such as Story Points, Effort, or Size.

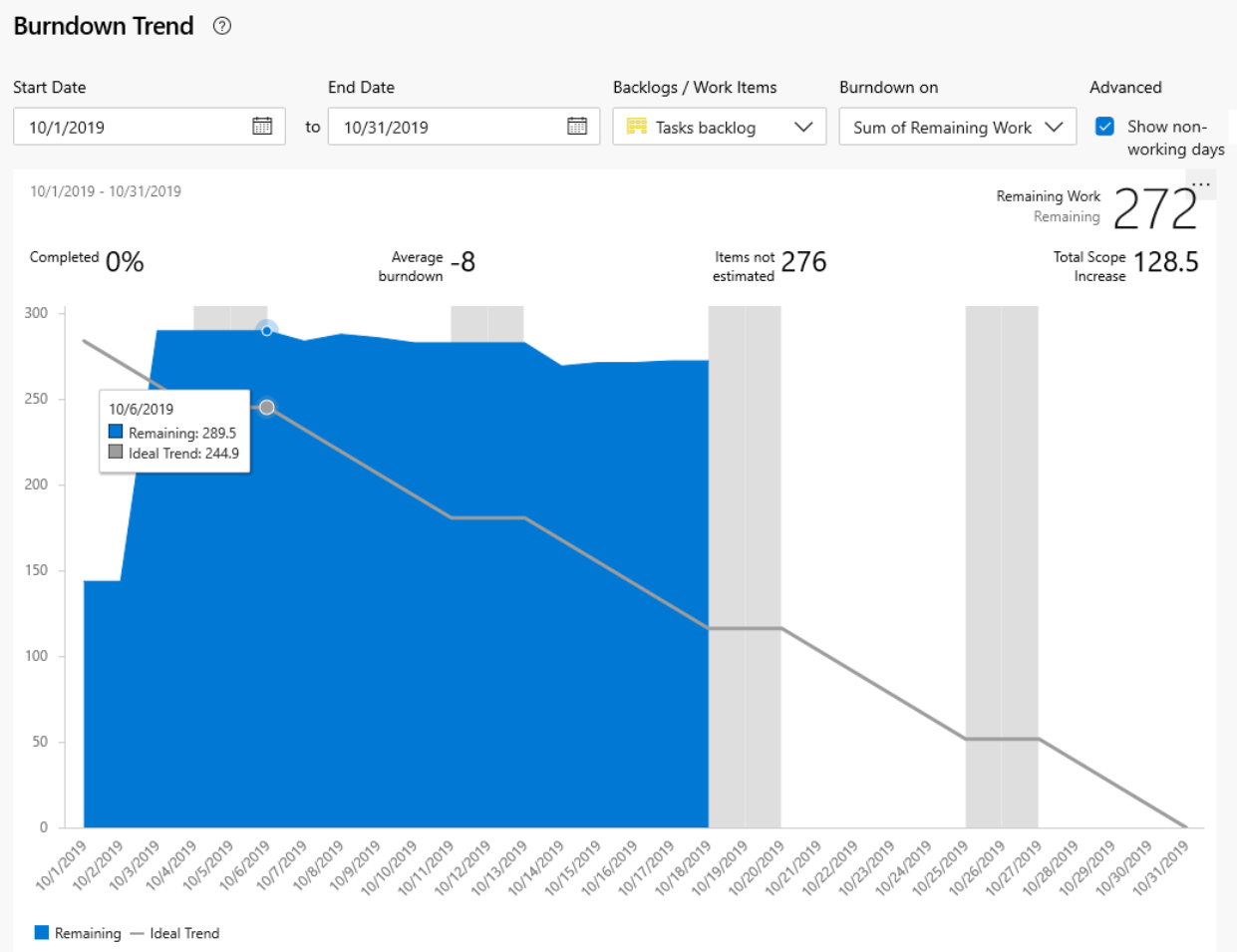
   d. Check or uncheck **Show non-working days**. Non-working days appear as gray bars in the background when enabled. Default non-working days are set for a team and for a team's sprint through the capacity page. See Set working days and Set sprint capacity.

      Select **Reset** to reset the controls to the default options. By default, the dates are set to the selected sprint. Changes to the start and end dates don't change any sprint date definitions.

3. If you don't track Remaining Work in tasks, you can view burndown based on a count of work items/tasks. Hover over any point on the chart to show a summary of the data for a specific day.

- Sum of Remaining Work
- Count of Work Items
- Sum of Story Points

When you choose to view the **Tasks backlog** and **Sum of Remaining Work**, the blue area shows the sum of Remaining Work per day for those tasks that are still active or in progress. As the Remaining Work is updated, the chart indicates the rate of burndown. The **Scope** trend line indicates the addition of Remaining Work after the start of the sprint. The **Ideal** trend line indicates the ideal burndown rate for the sprint. **Capacity** lines are only shown when the team has configured capacity.



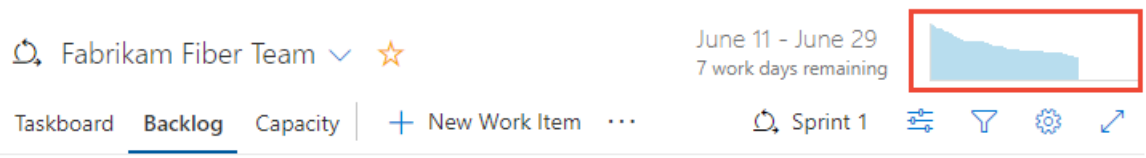
NOTE

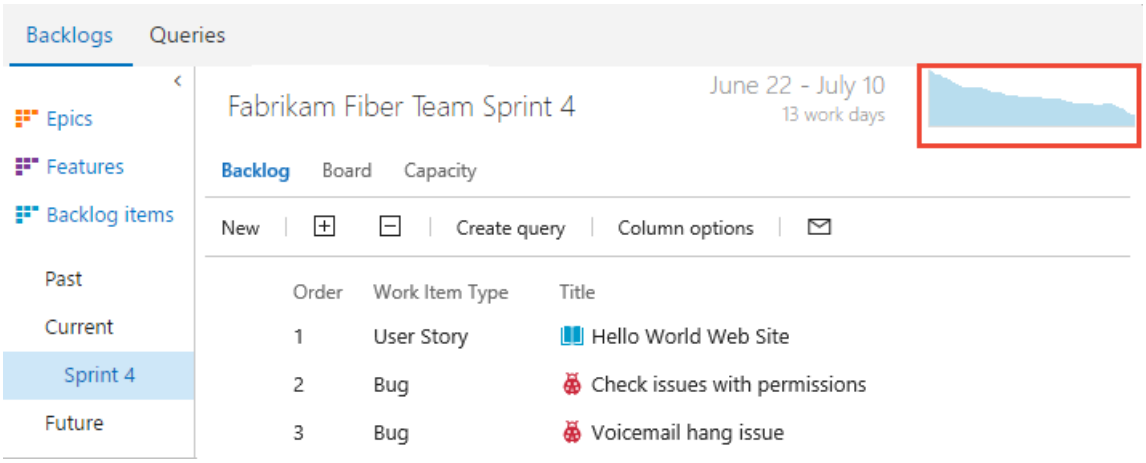
The options for the sum fields depend on the numeric fields defined for task and requirement category work item types. The most common fields used to show the burndown trend are:

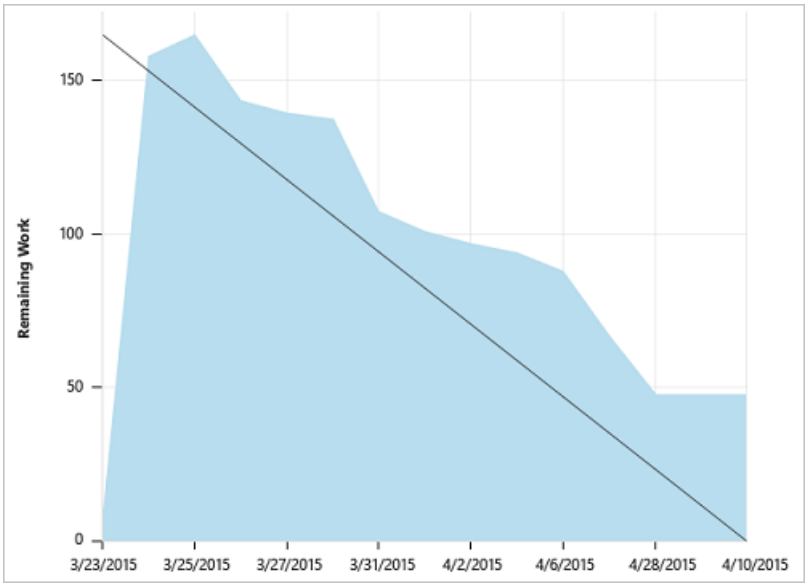
- Count of work items
- Sum of Story Points, Effort, or Size
- Sum of Remaining Work.

The selections you make are only set for you, and persist across sessions until you change them.

Select the chart to display it in a larger view.

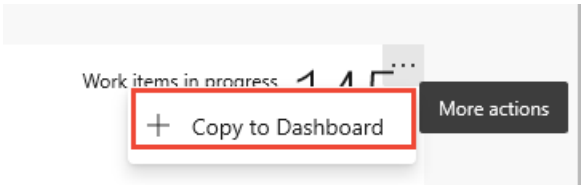






Add the report to a dashboard

1. To add the report to a dashboard, select the ··· actions icon and select **Copy to Dashboard**.



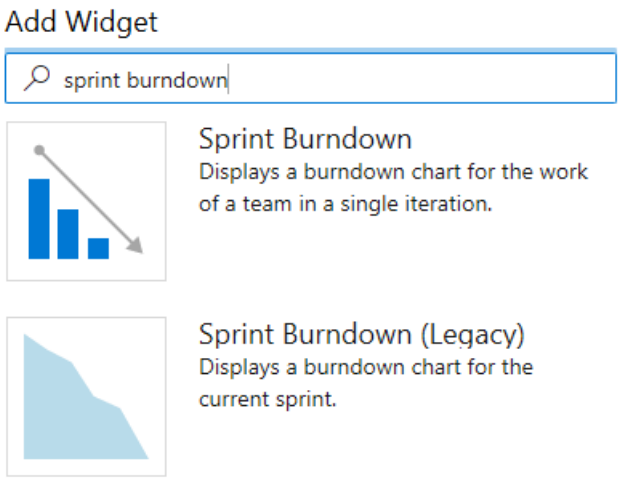
2. Select the dashboard and select **OK**.

Add the Sprint Burndown widget to a dashboard

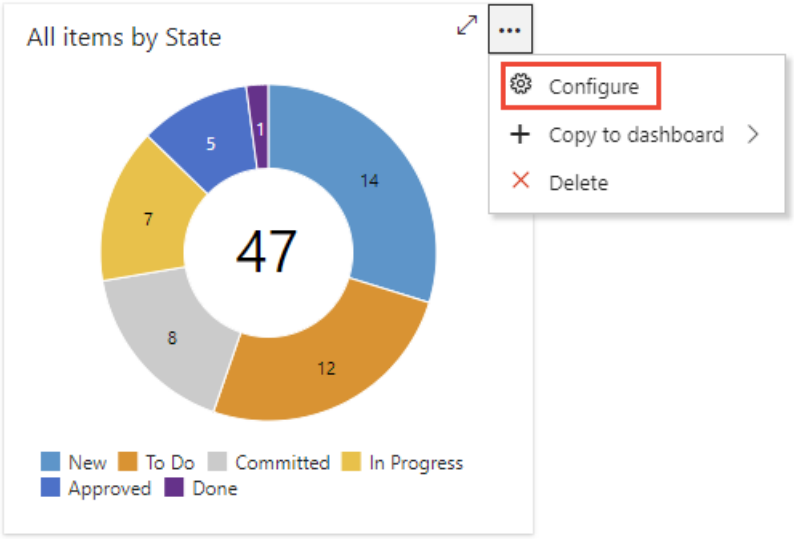
You can add the Sprint Burndown widget to a dashboard and select the team whose progress you want to monitor. You configure these widgets for one or more teams.

1. If you haven't yet added the Sprint Burndown widget to your dashboard, do that now.

   You can filter the **Add** widget dialog with "sprint burndown" to quickly locate the two widgets available to you.



2. To configure the widget, select the ··· actions icon and select the **Configure** option.

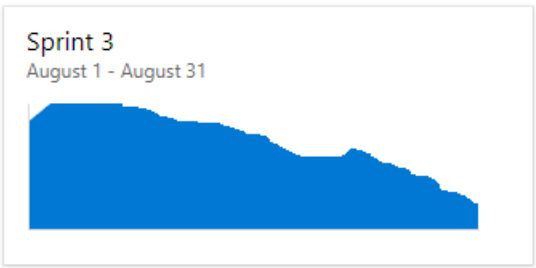


1. Select **Edit** to add the Sprint burndown widget to your team dashboard.

   The widget catalog automatically opens. Drag the Sprint Burndown widget onto the dashboard.

2. When you're finished with your additions, select **Done Editing**.

   The sprint burndown chart for the team's current sprint is added to the dashboard. There's no configuration option associated with this widget.

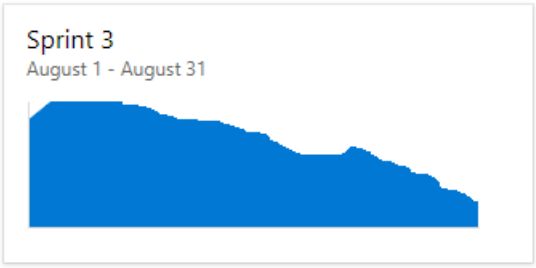


1. Select  to modify a dashboard. Select  to add a widget to the dashboard.

   The widget catalog automatically opens. Add all the widgets that you want and drag their tiles into the sequence you want.

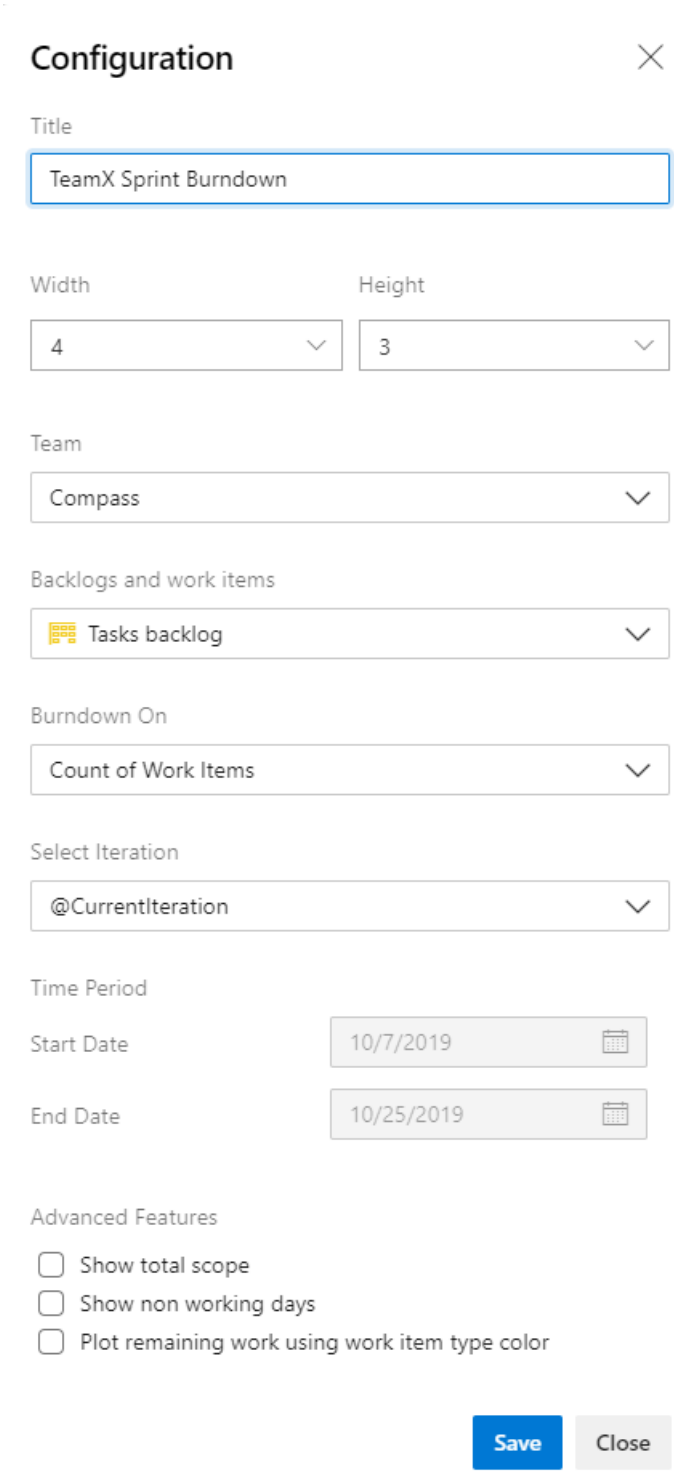
2. When you're finished with your additions, select **Done Editing**.

The sprint burndown chart for the team's current sprint is added to the dashboard. There's no configuration option associated with this widget.



Configure the Analytics-based Sprint Burndown widget

1. To configure the widget, select the ··· actions icon and select the **Configure** option.



2. Modify the **Title** of the widget and select your preferred **Size**. The Sprint Burndown widget can scale up to 10x10.

3. Make the following selections:

   - **Team** - Select the **Team** you want to track.
   - **Backlogs and work items** - Select the work items to include in your burndown. You can select to any backlog or a specific work item type.
   - **Burndown on** - Choose how you want to burndown. You may burndown by count of work items or a sum based on a selected field.
   - **Select iteration** - You may select **@CurrentIteration**, or a specific iteration.
   - **Time period** - If you selected **@CurrentIteration**, these dates aren't editable, as they'll be automatically to the start/end date of the current iteration. If you selected a specific iteration, you may customize the start/end date for the burndown chart.

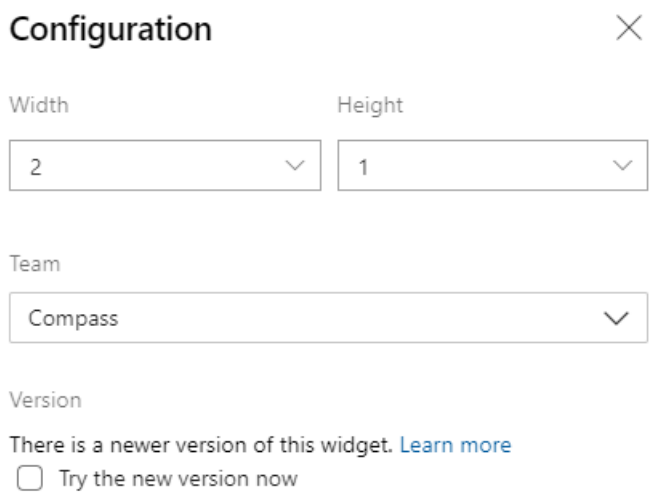
4. **Advanced features**: Check the boxes of the following options that you want to add to your chart.

   - **Show total scope**: Displays both the historical and projected scope increase.
   - **Show non working days**: Displays non working days on the burndown. When displayed, non working days are shaded.
   - **Plot remaining using work item type color**: Displays remaining work based on the work item type color, rather than the default blue color. If multiple work items are included, then it stacks colors by work item type.

> **NOTE**
> The **Show non-working days** shades those days set through the team's **Working days** setting as well as the team's days off set through the **Capacity** page.

Configure the Sprint Burndown (Legacy) widget

1. To configure the widget, select the ··· actions icon and select the **Configure** option.

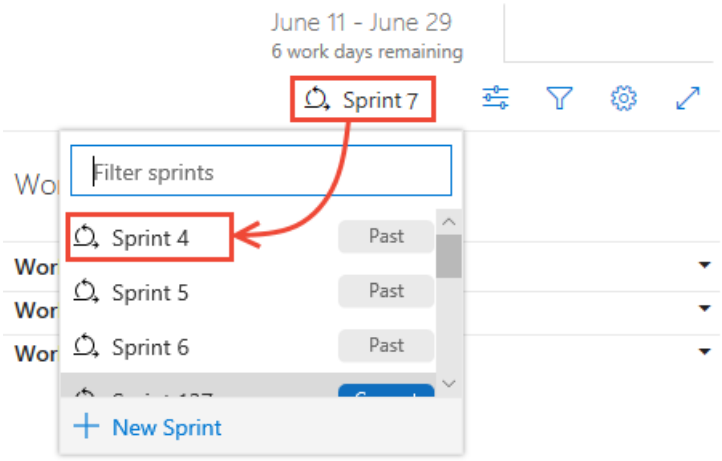


If your dashboard already has a legacy version available, you can easily upgrade the widget by editing the widget's configuration, and checking **Try the new version now**. You can always go back to the legacy version by unchecking the box.

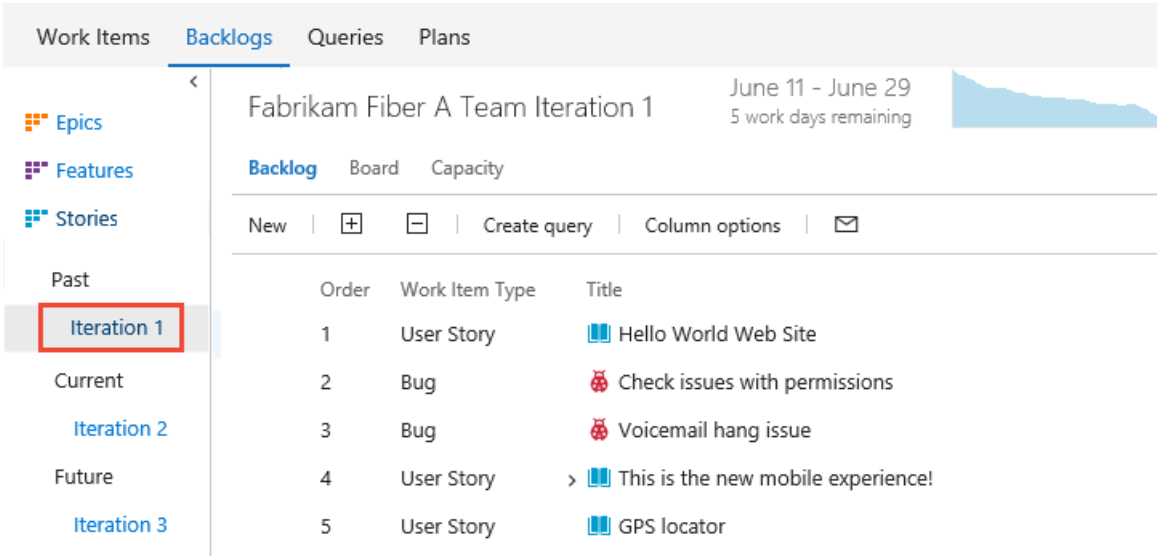
Current and past sprint burndown charts

As you complete each sprint, the system maintains a history of your activity.

To view a past sprint and its burndown chart, select the sprint from the Sprint selector.

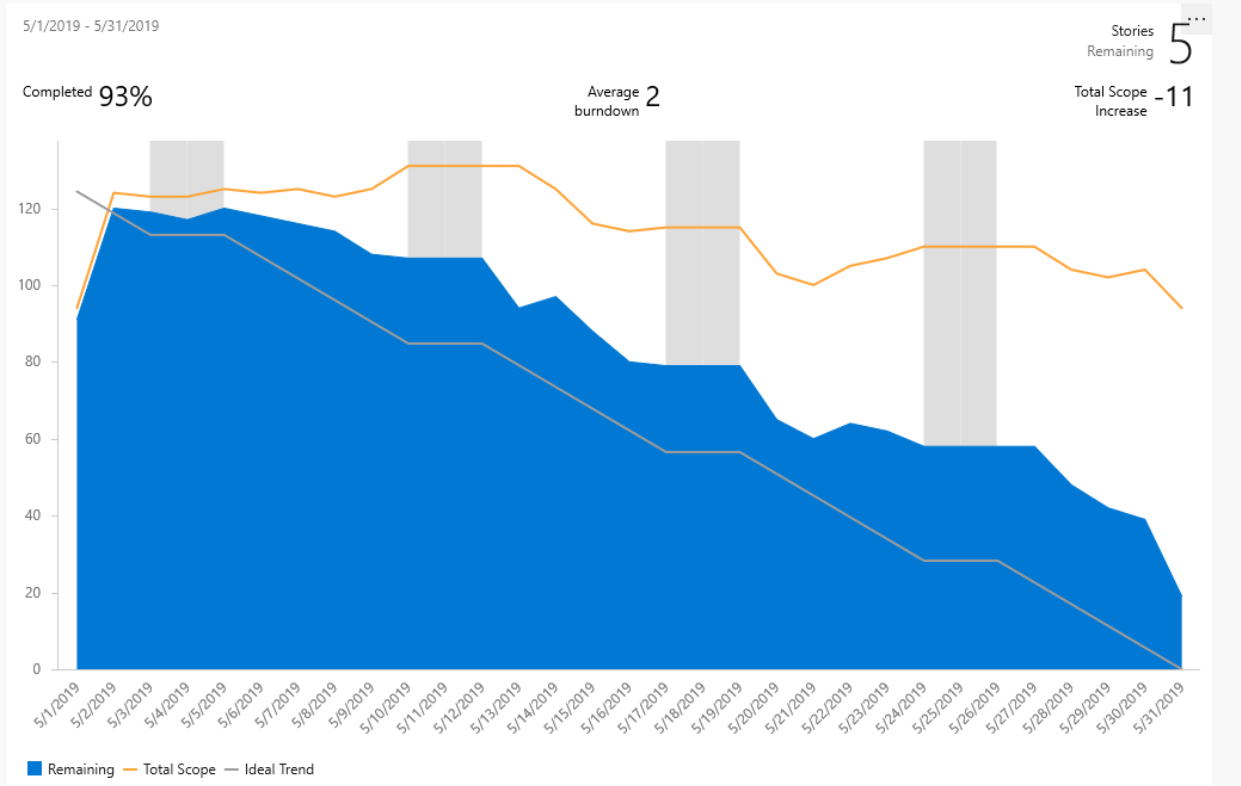


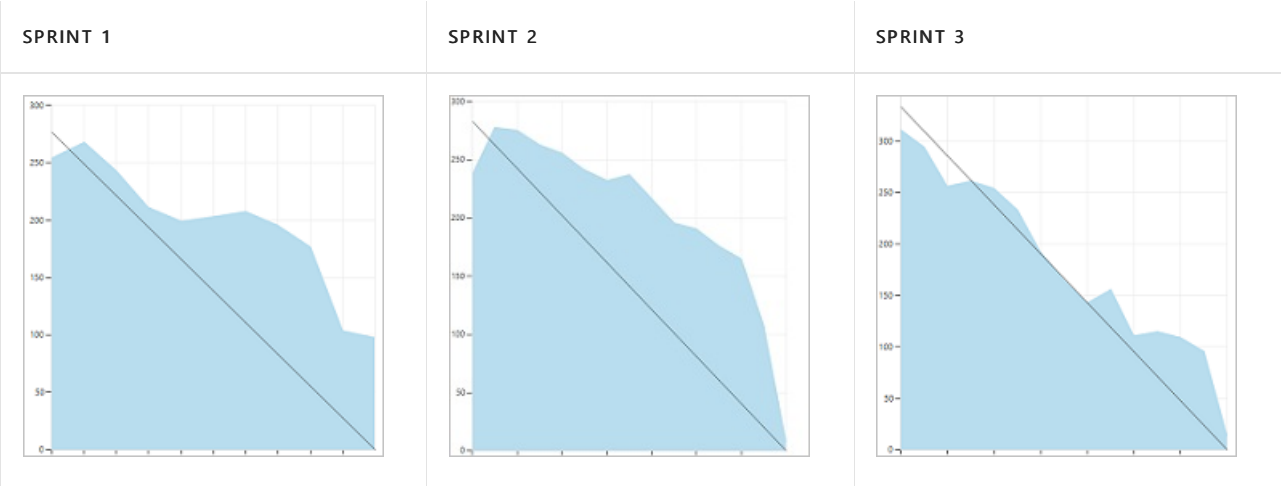
To view a past sprint and its burndown chart, select the sprint listed under the **Past** section of the sidebar.



You can review sprint burndown in-context reports to show the team patterns in execution. The burndown charts maintain a record of the team's ability to plan and estimate.

- May
- June
- July





Teams may find it useful to review these reports periodically during their sprint retrospectives. It may spark useful discussions and lead to setting one or more sprint goals, such as:

- How does our projected velocity match up to our actual velocity?
- How can we more accurately determine how much we can accomplish in a sprint?
- How can we complete work at a more regular pace throughout the sprint?

Next steps

Burndown and burnup guidance

In addition to the sprint burndown chart, teams can review the velocity at which they work sprint over sprint. The velocity chart tracks how many backlog items your team works on in a sprint. You can use your team velocity as input into the forecast tool to help plan your sprints.

Related articles

You can learn more about defining, planning, and executing your sprints from these articles:

- Define iteration paths and configure team iterations

- Sprint planning
- Add tasks to backlog items
- Update and monitor your Taskboard
- Scrum and best practices

And, from these industry resources:

- Understanding the Scrum Burndown Chart

For projects that use the On-premises XML process model, you can specify the format that appears—**h** for hours or **d** for days—for the remaining work field.

Configure a Burndown or Burnup widget

10/4/2022 • 14 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

The Burndown and Burnup widgets provide the flexibility to create charts for:

- Any type of scope
- Any number of teams
- Within specified time periods.

Burndown charts focus on remaining work within a specific time period, while burnup charts focus on completed work.

Both chart types help answer the question: *Are we on track to complete this set of work by the end date?*

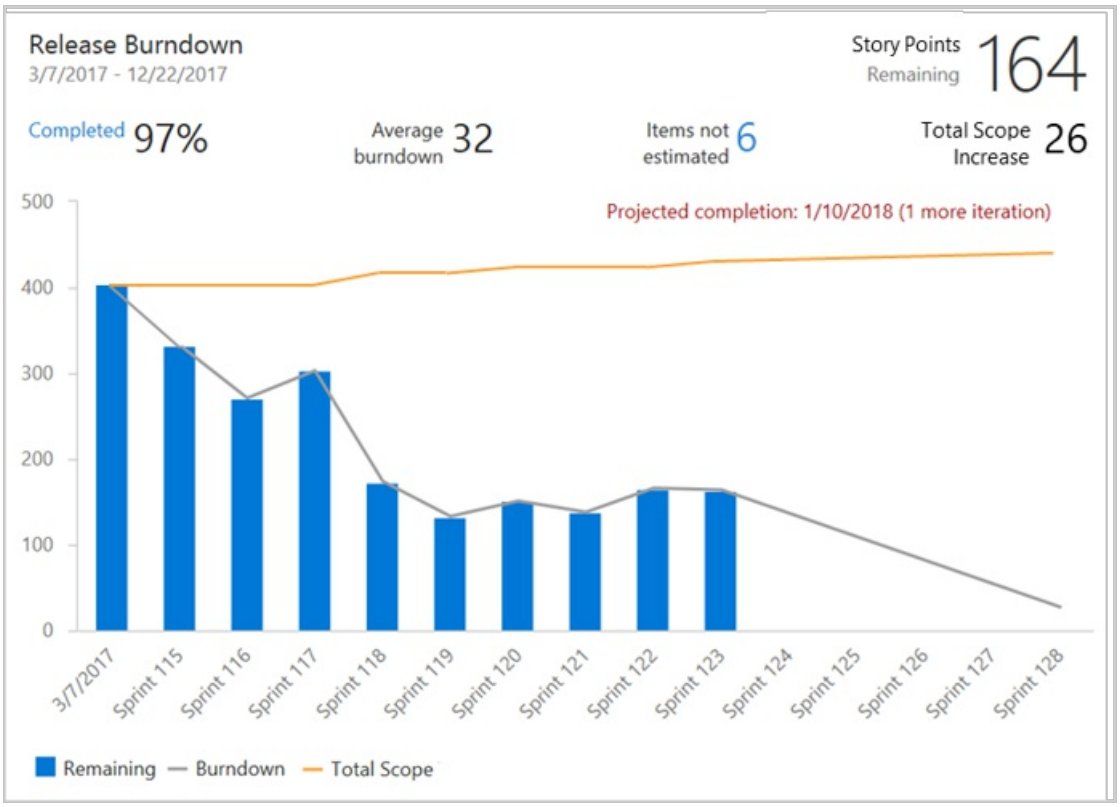
Use this article to learn how to:

- Interpret a Burndown or Burnup widget
- Configure the Burndown or Burnup widgets
- Use burndown metrics
- Work with a burndown chart
- Configure a sprint burndown

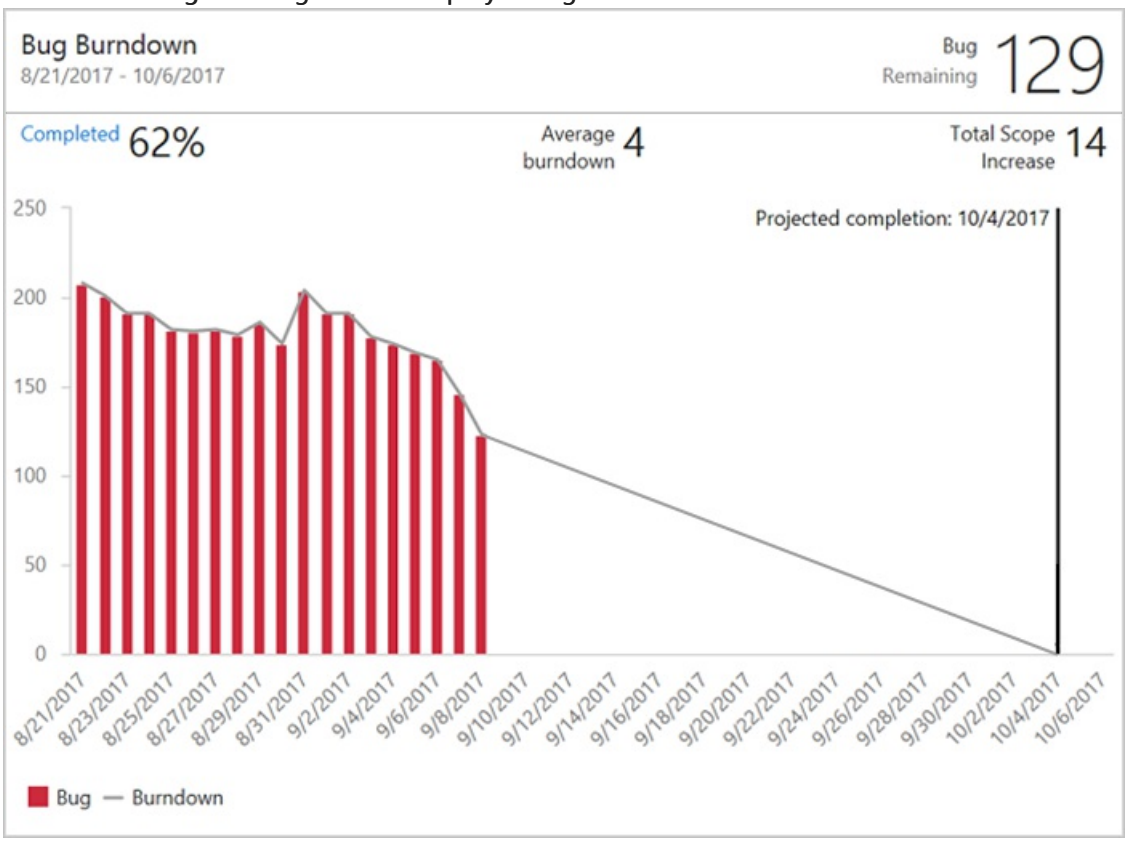
For an overview of all burndown/burnup charts available to you, see Burndown and burnup guidance.

Use the burndown chart to track completion of a predefined scope of work over a predefined period of time. For example, a sprint burndown tracks the sprint backlog completion by end of the sprint. A release burndown tracks the release backlog completion by the end of the release. You can define a bug burndown chart to track completion of a set of bugs by a certain date.

Burndown widget configured to display a Release Burndown



Burndown widget configured to display a Bug Burndown



Metrics

Burndown and burnup charts provide an easy way to monitor progress across teams and sprints by showing work remaining over time. Work remaining is the vertical axis and time is the horizontal axis. Remaining work is calculated as a sum of a particular field like Story Points, or count of a type of work item like User Stories. Also, each chart calculates and displays the average burndown or burnup rate and added scope over the course of the project.

Based on historical burndown and scope increase, the Burndown chart shows a projected work completion date.

Using burndown, teams can stay on top of their progress and see the immediate effect of their work on their delivery date.

These charts provide the following useful metrics:

- Percentage work complete
- Average burndown rate
- Total scope increase
- Number of work items not estimated with Story Points, or whichever field you're burning down on
- Projected burndown, based on historical burndown rate
- Projected scope increase, based on historical scope increase rate
- Projected completion date, based on historical burndown and scope increase rates.

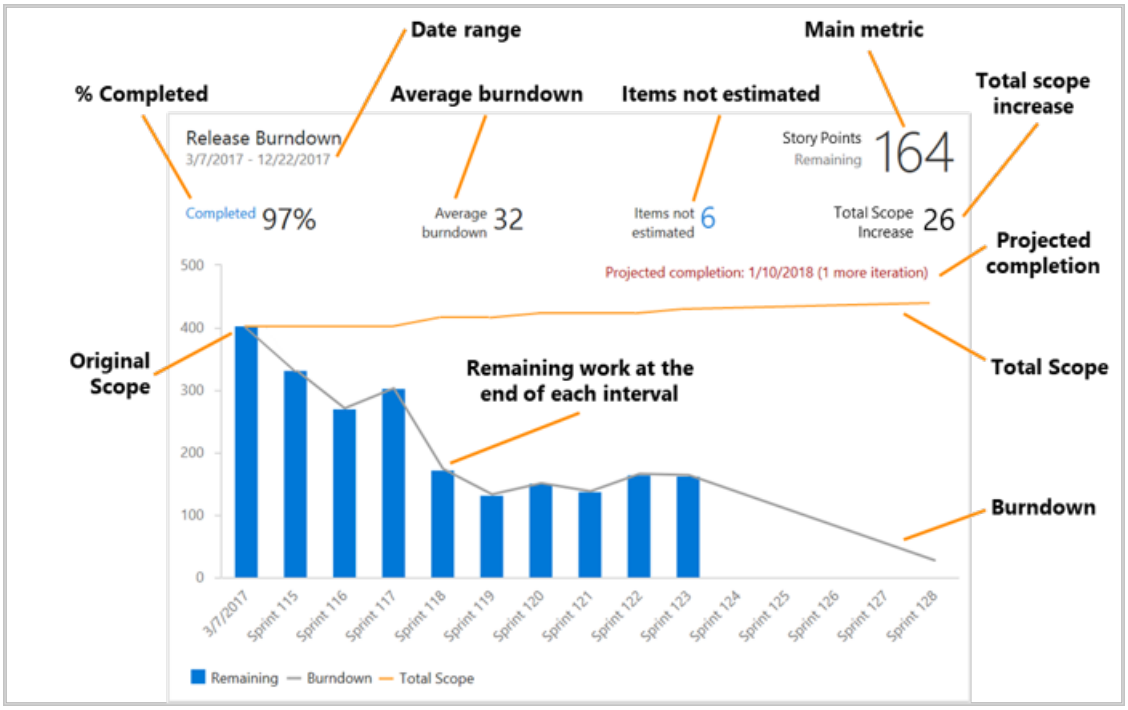
Interpret a Burndown or Burnup chart

Looking at the burndown chart, a team can not only get immediate insight as to their progress, but also learn about their rhythm and behavior. Most burndown lines aren't straight lines. The team never moves at exactly one fixed velocity. Scope increases occur over time. For example, if your projected completion date moves, you may want to ask one of these questions:

- *Are we adding too much scope?*
- *Is the average burn rate changing, and if so, why?*

Burndown charts also help teams understand risks to their release. If the projected end date exceeds the release target date, teams may need to reduce scope or lengthen the project. Burndown can also indicate that progress is greater than expected, providing the uncommon, but wonderful option of adding scope.

As the following diagram shows, charts based on the Burndown and Burnup widgets provide many calculated elements.



ELEMENT	DESCRIPTION
Date range	The start and end date of the burndown. When burndown is plotted by iterations, the end date is the end of the last iteration.

ELEMENT	DESCRIPTION
Main metric	Current remaining work based on the selected burndown method.
% Completed	The percentage of work completed based on original scope. Select **% Completed** to see the full list of completed work items.
Average burndown	Average work completed per interval or iteration.
Items not estimated	Shows only when burning down on the Sum of a field. It represents the current number of items that don't have a value in the selected **Burndown on** field. Select the number to see a full list of work items without estimates.
Total Scope Increase	show how much work was added to the original scope since the burndown started.
Projected completion	Calculates the projected completion date based on the remaining work and historical burndown and scope increase rates. If the projected completion date is before the specified **End Date**, it draws a vertical line on the interval when the work should complete. If the projected completion date is after the specified **End Date**, it displays the number of other intervals/iterations needed to complete the work.
Original Scope	Original scope is all remaining work since the specified **Start Date**. The chart burns down from the original scope. **% Complete** and **Total Scope Increase** are calculated based on your original scope.
Total Scope	Represents to the total scope of the burndown. The plotted points include both completed and remaining work. The total scope line indicates the scope change of your project. For past data points, the plotted total scope represents actual total scope as of the end of each interval/iteration. For future data points, the plotted total scope represents a projected scope change, based on past scope changes.
Burndown	Represents the burndown. The burndown line tells you how fast you're burning down the work. For past data points, the plotted burndown represents actual burndown as of the end of each interval/iteration. For future data points, the plotted burndown represents a projected burndown, based on past burndown.

Prerequisites

- You must be a member of a project. If you don't have a team project yet, create one.
- If you haven't been added as a project member, get added now.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Have enabled or installed Analytics. You must be an account owner or a member of the **Project Collection Administrators** group to add extensions or enable the service.
- To add a widget to a team dashboard, you need to be a member of the team. You must have **Basic** access or greater, have dashboard permissions, or be a team admin or project admin. Default settings provide all team members with permissions.
- **Boards** must be enabled. If disabled, none of the work tracking Analytics widgets will display. To re-enable it, see Turn an Azure DevOps service on or off.

Add the widget to your dashboard

The Configuration dialog for the Burndown and Burnup widgets is the same. You configure these widgets for one or more teams. To learn more about teams, see Add teams.

1. If you haven't yet added the Burndown widget to your dashboard, do that now.
2. To open the configuration dialog, select ··· **More actions** and select the **Configure** option.

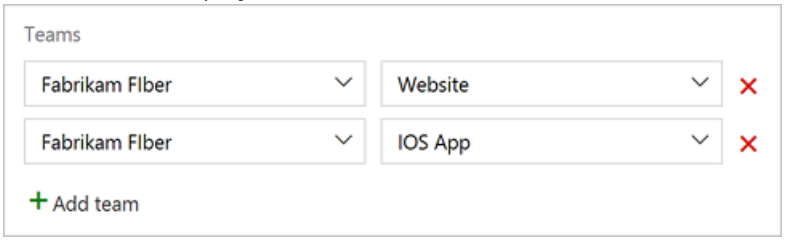
1. If you haven't yet enabled or installed Analytics], do that now.
2. If you haven't yet added the Burndown widget to your dashboard, do that now.
3. To open the configuration dialog, select ··· **More actions** and select the **Configure** option.

Choose the teams and work items to chart

1. Modify the **Title** of the widget and select your preferred **Size**. The Burndown widget can scale up to 10x10.

2. Select the **Teams** you want to track. Select at least one **Project** and one **Team**.

   If you want to track progress across teams, add more teams using the team selector. You may also select teams from other projects.

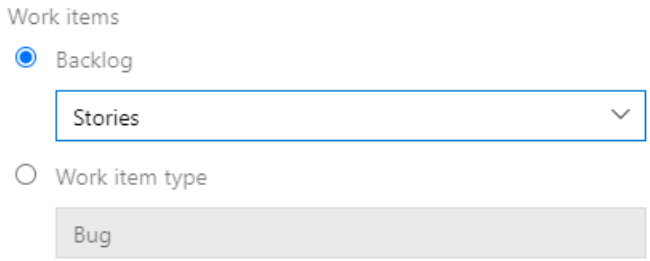
   

   The Burndown chart will display the burndown of remaining work for all selected teams.

   > **NOTE**
   > While you can select teams from other projects, all of the available configuration options—**Work items**, **Field criteria**, and **Burndown on** will show selections from your **current project**.
   > The list of selectable backlogs, work item types, and fields are based on your current project.
   >
   > For example, if you select a work item type that doesn't exist in another project, the burndown will not include work items from that project. If you select a field that doesn't exist in another project, that field will be considered blank for the burndown. Therefore, a burndown created across multiple projects will only work if the Process for those projects are the same, or at least very similar.

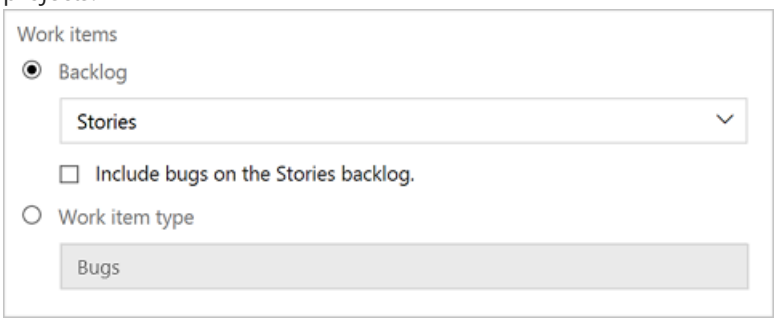
3. Choose a backlog or the work item type that you want to monitor. The burndown can include work based on items in your **Backlog** or by **Work item type**.

   Choosing a **Backlog** includes all the work item types configured for that backlog.



Choosing a **Backlog** includes all the work item types configured for that backlog.
If you select the **Stories** backlog, you have another option: **Include bugs on the Stories backlog**.
Place a checkmark in the box to include bugs along with user stories in your burndown.

This option is available for the PBI Backlog for Scrum projects and the Requirements backlog for CMMI projects.



> **NOTE**
> If your project has been customized using a Hosted XML process and has created a customized bug work item category name, then the Burndown and Burnup widgets won't be able to query for work items within that category. To query for bugs, the customized bug work item type must belong to the default **Bug Category**, reference name `Microsoft.BugCategory`.

Choose **Work item type** to monitor burndown or burnup on a specific work item type. In the list, you'll find all the project's work item types including custom work item types.

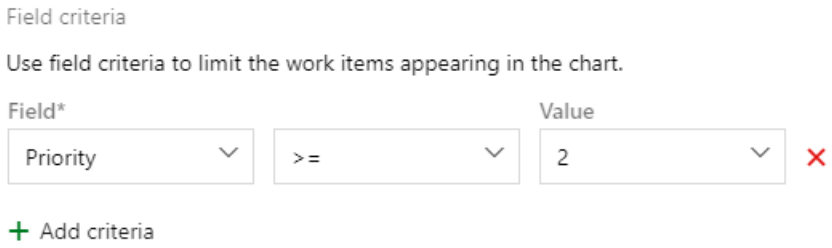


4. (Optional) Select field criteria to limit the work items that appear in the chart. Filtering is based on values assigned to fields as defined for each work item on the date within the tracking period.

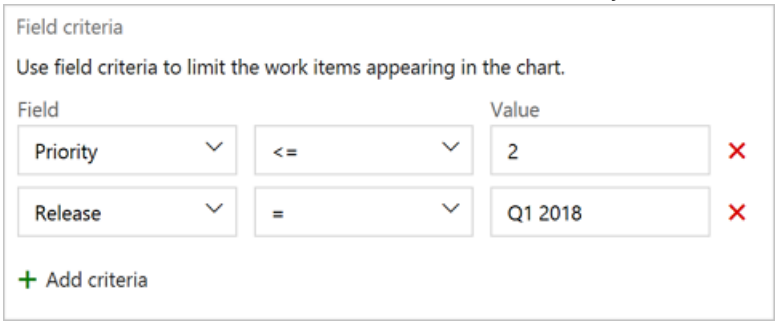
> **NOTE**
> When setting filters in this step or the following step, it is important to understand how filters are applied to historical data. Read Filters applied to historical data for more information.

- You can filter by any field available in your project, even a specific tag.
- Boolean fields aren't available for selection.
- No Date, HTML fields are available for filtering

- All field criteria are AND-ed together. That is, work items must match all the field criteria to be included in the burndown or burnup chart.
  For example, here we filter on top priority items by adding a filter **Priority** >=2.

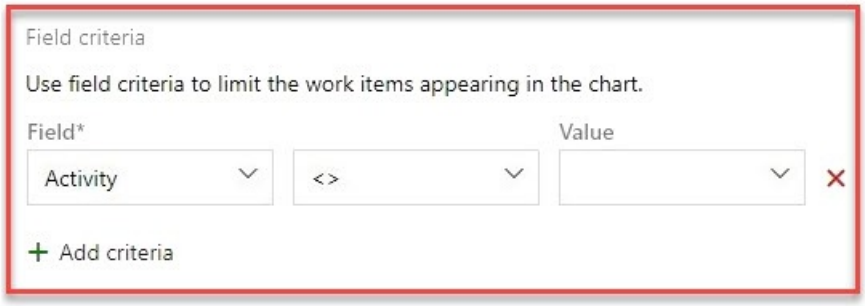


You may add multiple field criteria, by selecting **Add criteria**. For example, you can also select a custom field such as Release, to create a burndown chart of only those items assigned to a specific release.



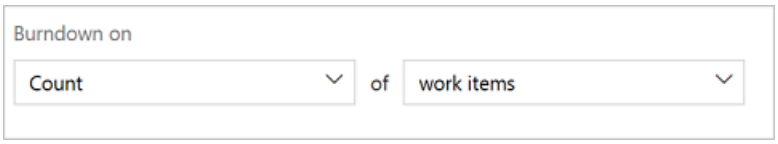
> **NOTE**
> Analytics-based charts are built based on the `WorkItemsSnapshot` EntityType. Snapshot entity types are modeled as daily snapshots. Data is aggregated based on assignments made as of the date they are assigned. What this means is that if you want to filter a Burndown/Burnup widget based on field or tag assignments, you must assign those prior to the period you want to monitor. Otherwise, they aren't registered by the widget until the date on which they are applied.

You can even filter on a null value for the **Field Criteria**. This behavior is consistent with a query using the same field criteria. Here we select to filter on work items whose **Activity** value isn't defined.

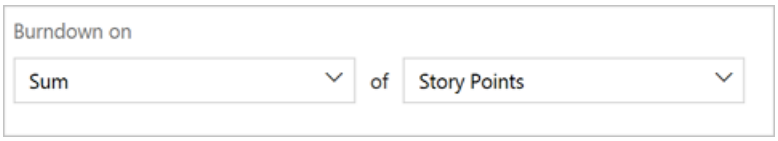


Choose your burndown/burnup metric to track

- Select how you want to calculate burndown or burnup: by **Count** of work items or by **Sum** based on a selected field.
  Here, we choose to base the burndown on a count of work items.



And, here we select a sum based on **Story Points**.



You can select from standard or custom fields of integer or decimal data type, such as **Story Points**, **Effort**, or **Remaining Work**.

> **NOTE**
> Burndown works best when aggregating size fields like Story PPoints. If you choose to Burndown on fields that change during the sprint, like Remaining Work for Tasks, the calculation of "Items not Estimated" will grow as items are closed.

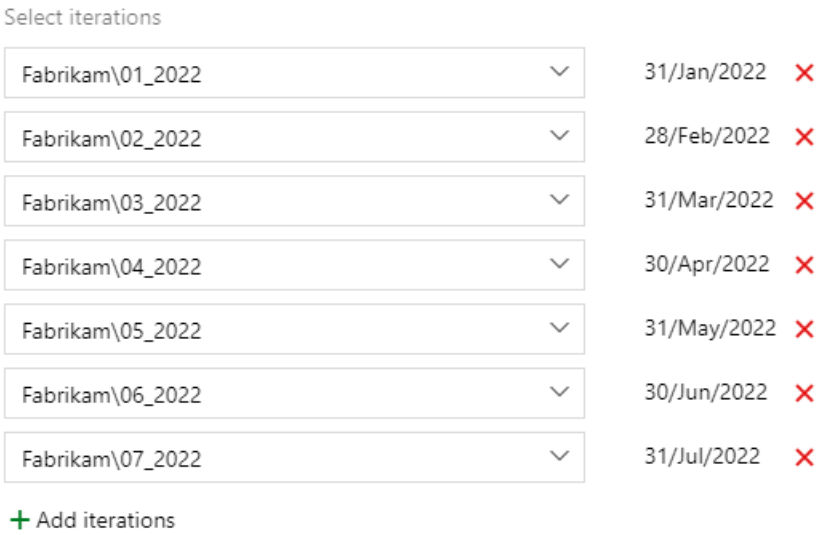
Choose the time period and plotting interval

- Select the time period. You can select from one of the following options to define your time period:

OPTION	PURPOSE FOR BURNDOWN
Start Date	Determines the original scope baseline. The chart burns down from the original scope. **% Complete** and **Total Scope Increase** are calculated based on your original scope.
End Date	Specifies the target date of completion. Your goal is to burndown the original scope of work by the **End Date**.
Plotting Interval	Here you select the intervals to plot between the **Start Date** and **End Date**. Average Burndown is based on the selected interval. You can plot burndown based on daily/weekly/monthly intervals or based on an iteration schedule.

Plot based on an iteration schedule

After selecting the **Start Date**, set **Plot burndown by** to **Iteration**. You can select iterations from your current project.



Add multiple iterations by selecting **Add iterations**.

TIP

The iteration selection box supports search. Type a partial name of an iteration and the closest match appears.

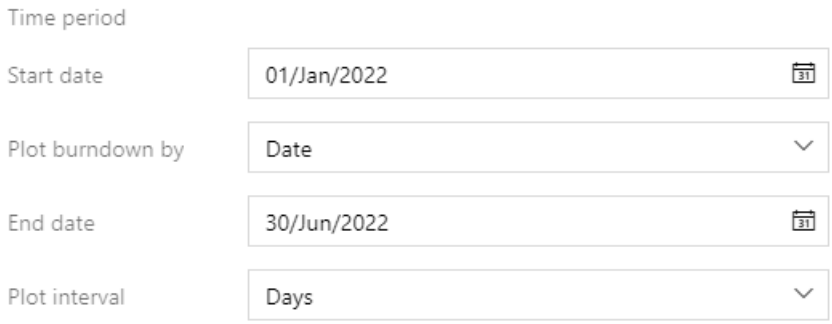
Iterations you can select are based on the **current project**, even if you selected teams from other projects. The burndown chart plots remaining work based on the end date of the iteration. It calculates remaining work across all teams and projects, based on that iteration end date. For example, if an iteration ends on 07/31/2022, the burndown chart calculates remaining work as of 07/30/2022, counting or summing all work items for every team or project. So, a cross-project burndown works when plotting by iterations, as long as all the teams have selected the same iteration schedule.

The burndown chart uses the end date of each iteration to plot the remaining work for that iteration.

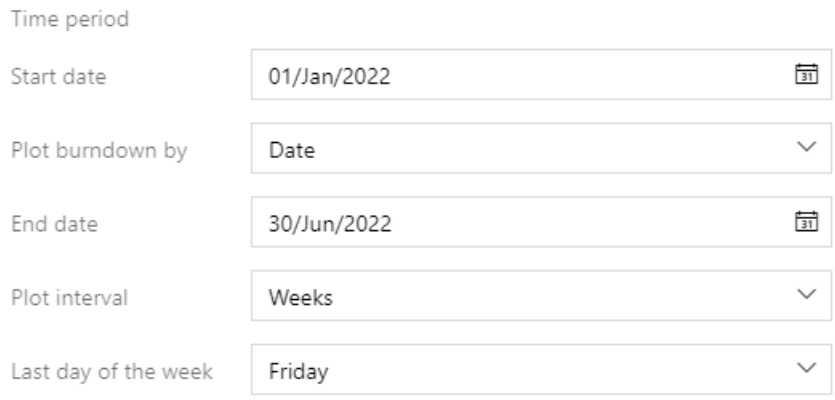
If you select to plot based on an iteration schedule, you can't select **End Date**. The burndown assumes the **End Date** is the last iteration's end date.

Plot based on a daily, weekly, or monthly interval

After selecting the **Start Date**, set **Plot burndown by** to **Date**. Specify the **End Date** for your burndown. You can set **Plot interval** to Days, Weeks, or Months.



If you select **Weeks**, then you can select the **Last day of week**. The remaining work for each interval will be calculated based on that day.



If you select **Months**, then burndown/burnup is calculated based on the last day of each month.

> **NOTE**
>
> The **Average Burndown** assumes that every interval is the same length. It does not consider months that are different lengths. Additionally, it assumes that the interval between the **Start Date** and the first month is a full month, even if the length of time between **Start Date** and the first month's end date does not match your typical length of a month. For example, a **Start Date** of 11/15/2021, would plot the first month as 10/31/2021, but would be counted as a full month for your **Average Burndown**. For best results, enter a **Start Date** that is the same as the first month's start date. **This is also true when plotting by weekly intervals.**

Other options

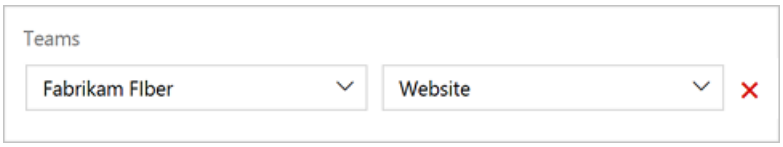
Check the boxes of the following options that you want to add to your chart.

- **Show burndown**: Displays both the historical and projected future burndown
- **Show total scope**: Displays both the historical and projected scope increase
- **Show completed work**: It displays remaining work and completed work as stack bar
- **Plot remaining using work item type color**: Displays remaining work based on the work item type color, rather than the default blue color. If multiple work items are included, then it stacks colors by work item type.

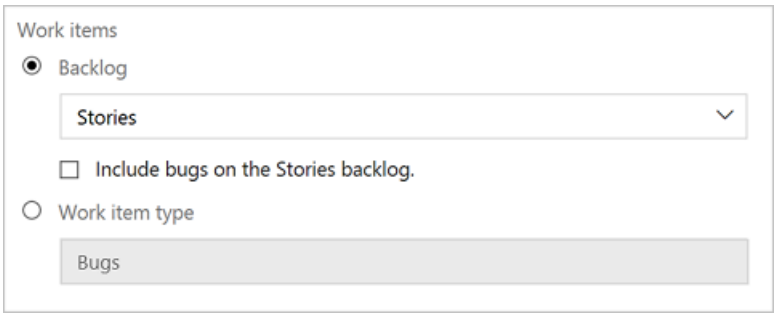
Configure a Sprint Burndown widget

One of the most common burndowns is the sprint burndown. A Sprint burndown is useful to determine if your team is on track to complete their sprint plan. You can use the following example on how to configure your Burndown widget to represent a sprint burndown. For this example, we show burndown for the Fabrikam Fiber - Website team for sprint **08_2022**. The sprint starts on 08/01/2022 and ends on 08/31/2022.

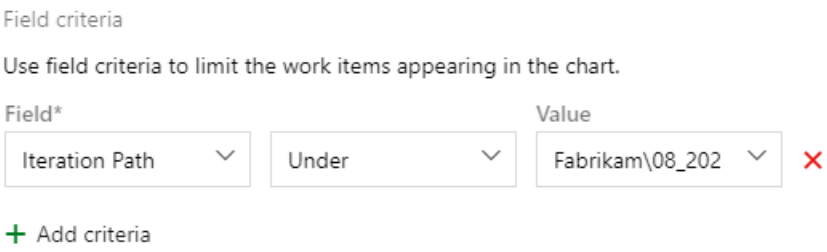
1. Select a single team to monitor their sprint burndown.



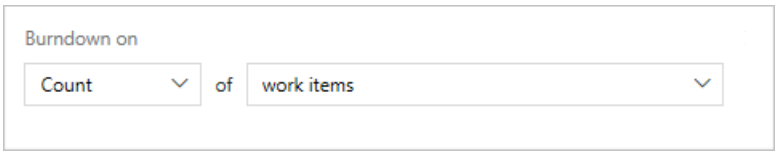
2. Choose your work items. For this example, select the **Stories** backlog.



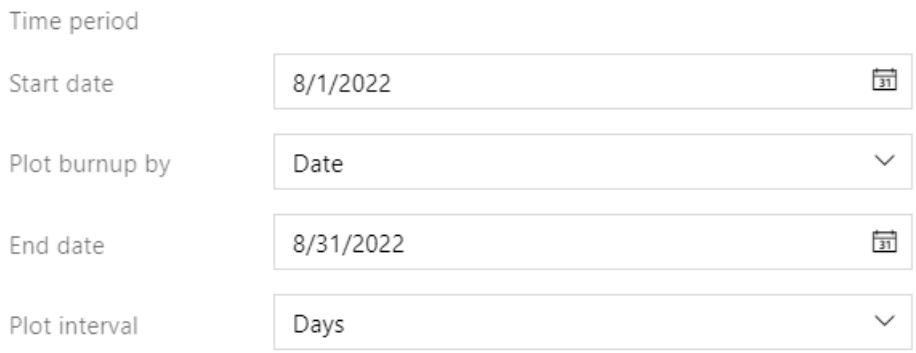
3. Select the iteration path you want to create the sprint burndown for. Add a field criteria on **Iteration Path** to match your sprint.



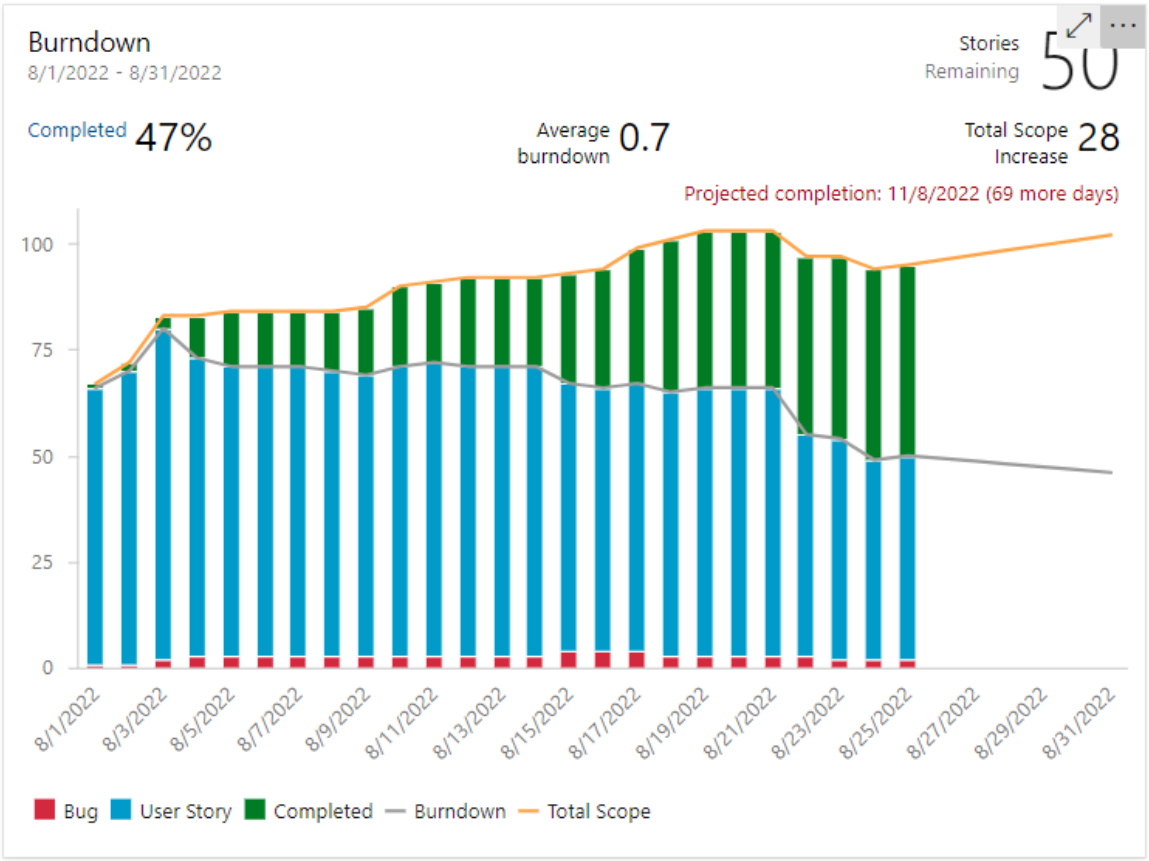
4. Select how you want to calculate burndown. You can use **Count** of work items, or **Sum** of any field.



5. Set the start date to be the first day of your sprint. For example, 08/01/2022.
6. Set **Plot burndown by** to **Date**. Set the end date to be the last day of your sprint. For example, 08/31/2022.



7. Save your configuration. This widget now shows a daily burndown of the Fabrikam Fiber - Website team for sprint **08_2022**. The burndown shows a count of work items completed per day as well as remaining stories and bugs. As the team added 28 items after the sprint began, that number is reflected in the **Total Scope Increase**.



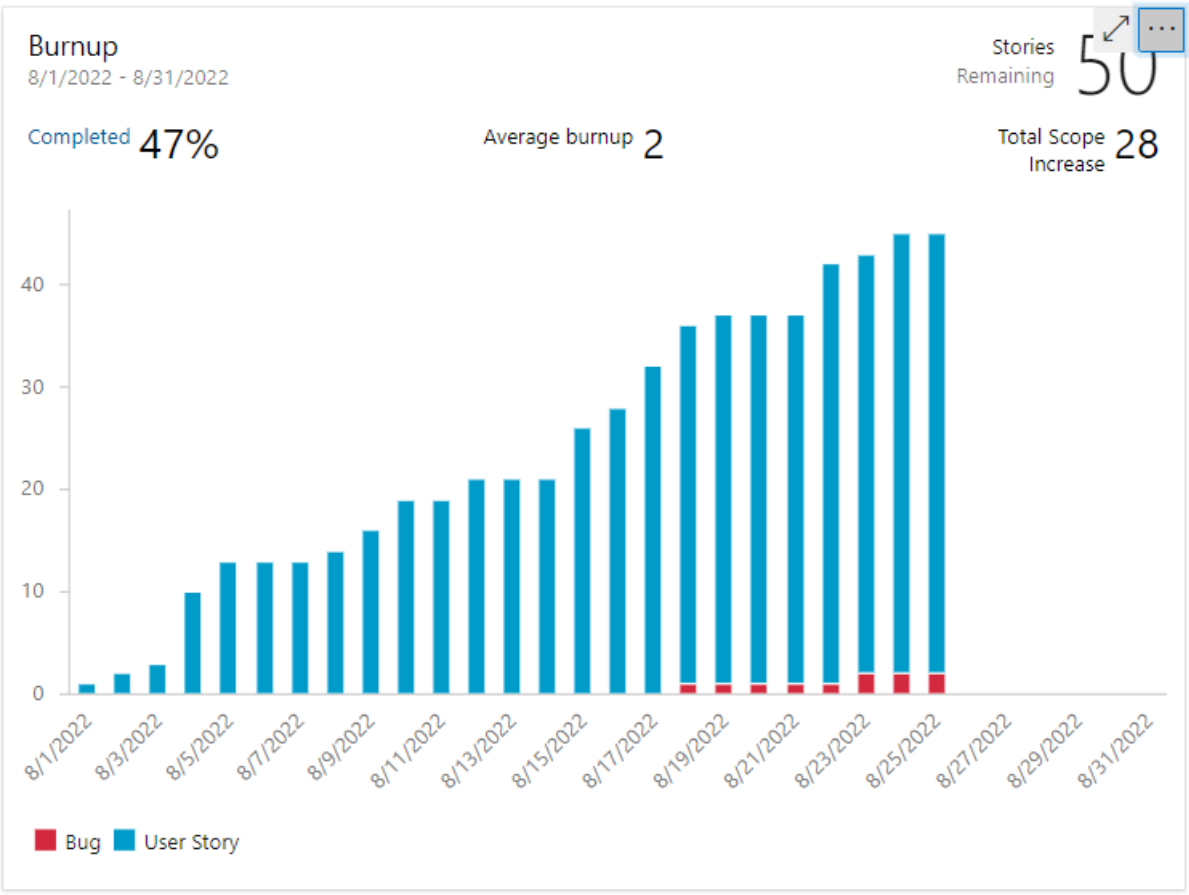
To change the sprint this widget is monitoring, for example to sprint **09_2022**, you need to manually change the widget configuration field criteria and dates.

Configure a Sprint Burnup widget

Configuring the Burnup widget is exactly like configuring the Burndown widget, except that it plots work

completed, rather than work remaining.

Burnup Widget displaying a Stories Burnup



Next steps

Burndown and burnup guidance

Related articles

- Configure and monitor sprint burndown
- Define iteration paths or sprints) and configure team iterations
- Add a custom field to a work item type
- Applying filters to historical data

Track your work by using managed queries in Azure Boards

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

List bugs, user stories, or other work items based on field criteria you specify using queries. You can then review these lists with your team, triage work, or bulk update work items. Along with managed queries, the semantic search tool provides some overlapping and different functionality worth exploring.

Use managed queries to support these operations:

- Bulk update of work items using the web portal
- Triage and update work items
- Review a hierarchy of work items
- Share a list of work items with a team member

You can create queries and query folders from the web portal or from a supported client, such as Visual Studio Team Explorer and Team Explorer Everywhere, a plug-in for Eclipse. Changes you make in one client are reflected in other clients as all changes are stored in the work tracking data store.

Get started using queries

If you're just getting started, read View, run, or email a work item query. For a quick reference to query editor tasks and sample queries, see Query quick reference.

- To find work items that are assigned to you, add the **@Me** macros as the value for the **Assigned To** field in one of the query clauses.
- All valid users with standard access can create queries and folders under the **My Queries** area. To create queries and query folders under **Shared Queries**, you must have the Contribute permission set. For more information, see Set permissions on queries.
- You can modify any query by adding criteria to focus on a product area, an iteration, or another field. To modify a query, open the query editor.
- You can open any query in Excel. You can also update the fields of one or more work items and publish your changes to the database for tracking work items.
- You can visualize status or progress by creating a pie-chart, column chart, or trend chart for flat-list queries.

Query capabilities

The following sections provide an overview of the functions supported to define and manage work item queries.

- Query filters are defined through the Query Editor.
- Query macros can be selected for specific fields to create a query clause.
- Query results and query management features are capabilities available through the Query Results page.

Query filters

The following table summarizes the query filter functions supported by each Azure DevOps version.

> **NOTE**
> Managed queries don't support proximity searches, however semantic searches do. In addition, semantic searches support both `*` and `?` as wildcard characters and you can use more than one wildcard character to match more than one character. To learn more, see Functional work items search.

Filter function	Query support	Supported versions
Text string searches (single text, multi-line text, rich text)	Searches are not case sensitive.	All versions
Wildcard searches	Wild card = `*`	All versions
Linked work item searches	Find work items based on direct links or topological/hierarchical link types. Filter linked work items based on `MODE` (WIQL syntax)	All versions
Logical clause groupings	Group and nest clauses using AND and OR Boolean operators.	All versions
Historical String and DateTime field queries	Find work items based on a field match with a previous value. Supported operator: `Was Ever` Find work items based on a value defined on a specific date. Supported operator: `ASOF` (WIQL syntax)	All versions
Query using macros or variables	Use macros to create queries relative to a date, other tools, such as team area path, team iteration, and more.	All versions
Search across projects	Find work items in one or more projects in an organization or collection. Default is the current project. Use the **Team Project** field to query on two or more projects.	TFS 2015-TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services
Field comparison searches		

Find work items based on how two fields compare with one another.
Supported operators: `=[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field]`

All versions

Query on tags

Find work items based on whether they contain or don't contain a tag. **Supported operators**: `Contains, Does Not Contain`

All versions

Query on blank or empty fields

Find work items based on empty or not empty HTML/rich text fields.
Supported operators: `Is Empty, Is Not Empty`

Azure DevOps 2019-2020, Azure DevOps Services

In and Not In Group searches

Find work whose field value matches any value in a delimited set, such as a set of work item types, workflow states, or picklist values. Separate values with the list separator that corresponds to the regional settings that are defined for your client computer. For example, you might use a comma(,).

TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services

Boolean searches

Find work items based on boolean field value.

TFS 2017-TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services

---

Query History and Discussion

Find work items based on key words or phrases added through the Discussion.

TFS 2017.2-TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services

---

Query on Kanban board fields

Find work items based on their Kanban column, swimlane, or Doing/Done status.

Azure DevOps 2019-2020, Azure DevOps Services

Interactive query filters

Filter query results based on a key word or select fields.

Azure DevOps 2019-2020, Azure DevOps Services

To bulk move, copy, or paste query clauses, install and use the WIQL editor. To learn more, see Cross-service and enhanced query operations

Supported macros

The following table summarizes the query macros or variables supported by the Azure DevOps versions. You can use some of these macros to filter notifications.

> **NOTE**
> You can use certain macros from the web portal only. These include the **@CurrentIteration**, **@CurrentIteration +/- n**, **@Follows**, **@MyRecentActivity**, **@RecentMentions**, **@RecentProjectActivity**, and **@TeamAreas** macros. These macros aren't supported when exporting a query to Excel, notification filters, or exercised from Team Explorer, or REST APIs.

For more detailed descriptions and links to examples, see Query fields, operators, and macros.

Macro	Query support	Supported versions
[Any]	Find any work item type, `Work Item Type=[Any]`, or any State, `State=[Any]`.	All versions
@Me	Find work where `Identity field=logged in user`.	All versions
@Today	Find work where `Date-Time field=today`.	All versions
@Project	Find work defined in one or more projects.	TFS 2015-TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services
@CurrentIteration	Find work defined in current iteration for a team.	All versions
@CurrentIteration +/-n	Find work defined in +/- n of current iteration for a team.	Azure DevOps 2019-2020, Azure DevOps Services
@Follows	Find work current logged in user is following, `ID In @Follows`.	TFS 2017-TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services
@MyRecentActivity, @RecentMentions, @RecentProjectActivity		

Find work items recently changed, `ID In @MyRecentActivity` See also View and add work items, Work Items page.

TFS 2018, Azure DevOps 2019-2020, Azure DevOps Services

@StartOfDay, @StartOfMonth, @StartOfWeek, @StartOfYear

Find work where the selected date-time field is within the current day, month, week, or year with a plus/minus offset, example: `Closed Date>=@StartOfDay-7` .

Azure DevOps 2019-2020, Azure DevOps Services

@TeamAreas

Find work assigned to an Area Path or Iteration Path of specified team, for examples, see Query by area or iteration path.

Azure DevOps 2019-2020, Azure DevOps Services

Query results and query management features

The following features provide support for working with query results, saving and sharing queries, and more.

- Add a query chart to a dashboard
- Apply interactive filters
- Choose column sort
- Choose query column options
- Copy query results
- Copy Query URL
- Create a query from a backlog
- Create query charts
- Create query folders

- Define a query hyperlink
- Define and edit a query
- Define WIQL syntax
- Email query results
- Favorite a query
- Filter a query

- Import/export a query (csv)
- Import/export a query (.wiq)
- Open and run a query
- Publish/refresh a query (Excel)
- Run REST API queries
- Save queries
- Set query permissions
- Triage query results

Unsupported features

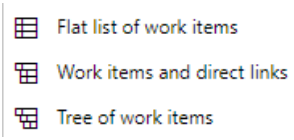
Work item queries only support querying of work items and work items linked to other work items. Here are a few of the tasks that managed queries don't support:

- Hierarchical views of Test Plans, Test Suites, and Test Cases. These items aren't linked together using parent-child link types. Instead, you can view the hierarchy through the Test>Test Plans page.
- Views showing linked objects such as builds, releases, code, or other non-work item objects.

- List work items linked from one project to another.
- Export a cross-project query to Excel. Direct links queries export to Excel as a flat-list.

Query types (flat, direct links, tree)

Azure Boards supports three query types. The icon next to each query indicates the query type. Use the following guidance to choose the query type based on what you want to accomplish with the query.



Query type

Usage guidance

Flat list of work items

- List items to do bulk updates to fields
- Triage a list of work items
- Create a query chart and add it to a dashboard
- Create a chart to get a count of items or sum a field
- Export a list of items to Excel to update fields

Work items and direct links

- List items that are dependent on other work items
- Find items related or dependent on other work items
- List linked work items to do bulk updates to fields
- Triage a list of linked work items
- List test-related linked work items
- Find orphaned backlog items, work items that have no parent

> **NOTE**
> **Work items and direct links** queries export to Excel as a flat list. Direct links queries are imported as a flat list as modifying multiple types of links isn't a supported feature in Excel.

Tree of work items

- List a tree of Parent-Child related work items, or other tree-topology link type
- Triage a hierarchical list of work items
- Export a hierarchical list of items to Excel to update fields or modify the hierarchy

To learn more about link types, see Link type reference.

My Queries, Shared Queries, and Favorites

Only you can view and run queries that you save under **My Queries** with the queries directory. Also, you can favorite one of these queries to have it appear within your query selector.

Queries you and others save under **Shared Queries** can be viewed by everyone with access to the project.

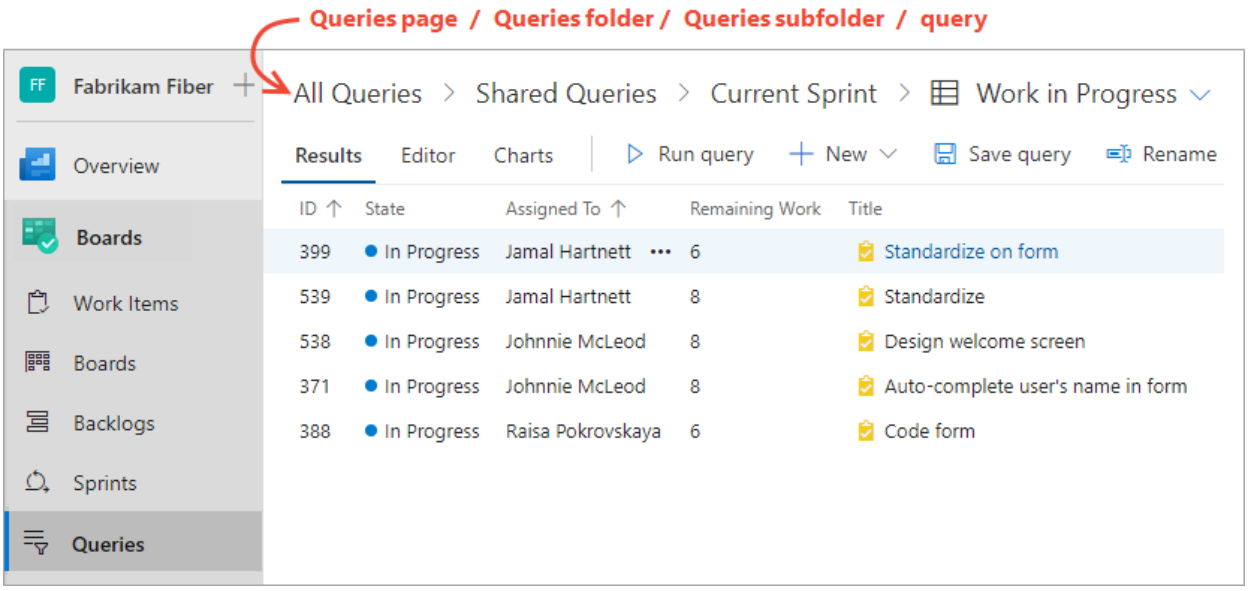
Shared queries can be organized within folders and favorited by you or for a team. Also, you can set permissions on the folders and queries to prevent others from moving or editing them.

For details, see:

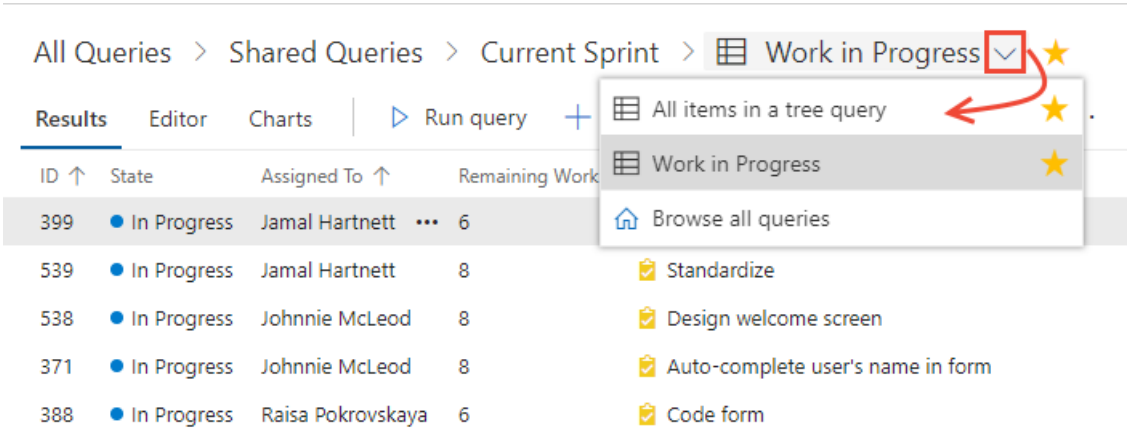
- Manage queries & query folders
- Set query permissions
- Favorite a query and Set personal or team favorites

Query directory, query folders, and breadcrumbs

The **Queries** page contains a directory-focused view that you can filter to find specific queries of interest. When working in the **Queries** pages, you can navigate to a subfolder, folder, or page.



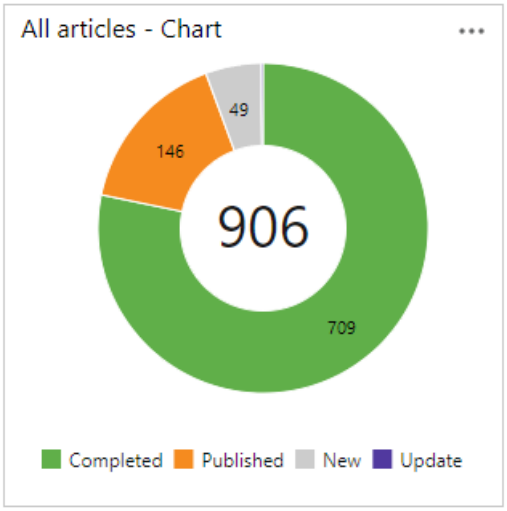
Also, you can choose a query that you've favorited from the selector menu. Or, you can choose to browse all queries that return you to the **All Queries** page.

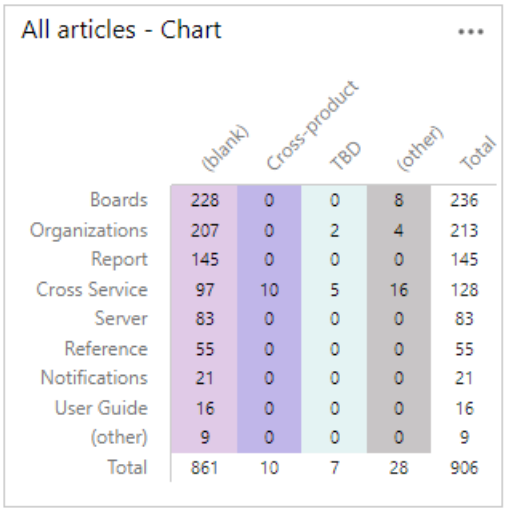


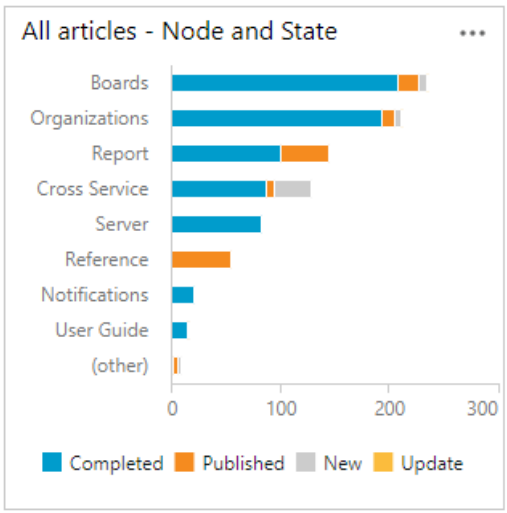
For more information, see Query FAQs, Navigate, and Folders.

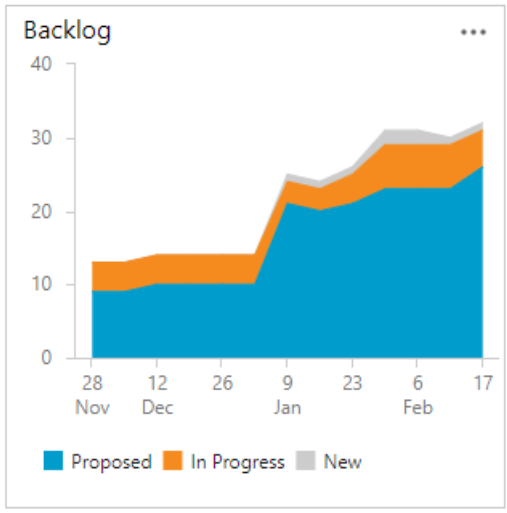
Query charts and widgets

You can quickly create pie, bar, pivot, and trend charts from a flat-list query. Queries must be flat-list and return 1000 or less work items. You can add your query charts to a dashboard, retitle, and reconfigure them.









Query-based widgets provide support for presenting query information on a dashboard. For example, the number of active bugs or a list of work items that you can interact with. To learn about query charts and widgets, see these articles:

- Chart a flat-list query
- Chart for work items widget
- Query results widget
- Query tile widget

Add a custom field to support queries

To add a custom field to support your query needs, see Customize your work tracking experience.

Taskboard versus query list items

You may notice and wonder why the contents of the taskboard differ from the contents listed with its created query? To learn more, see taskboard items versus query list items.

Full-text search queries and collation settings

To support full-text search queries, the collation settings of the SQL Server database should correspond to a language that has a word breaker registered with SQL Server. If you use an unsupported language, you could receive unexpected results when you run a work item query that specifies the **Contains** or **Contains Words** operators with text strings.

To learn more, see the following articles:

- sys.fulltext_languages (Transact-SQL)
- ALTER FULLTEXT INDEX (Transact-SQL)
- SQL Server 2008 Full-Text Search: Internals and Enhancements
- Query Fields, Operators, Values, and Variables

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Related articles

- Query FAQs
- Query quick reference
- Cross-service and enhanced query operations
- Work item field index
- Set query permissions
- Query fields, operators, and macros
- Bulk add or modify work items with Excel

Use an index to query quick reference data in Azure Boards and Azure DevOps

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use this index to quickly access example queries and information on opening, defining, and working with queries. To learn how to use the Query Editor, see Define a query. If you find that your queries take too long to return results, review the Guidance to create high-performing queries.

Example queries

You can list work items based on the following criteria...

Keywords, wildcard queries

- Keyword or phrase query with wildcards
- Undefined field value queries
- Empty or not empty HTML field queries
- Category based queries
- Items you're following
- Recent activity work items

Tag based queries

- Items containing a specific tag
- Items that don't contain a specific tag
- Items that contain two or more tags
- Items that have no tag assignments
- Chart work items and group by tags

- Items containing a specific tag
- Items that don't contain a specific tag
- Items that contain two or more tags
- Items that have no tag assignments
- Chart work items and group by tags

State, reason, or workflow change queries

- Resolved user stories
- Items I resolved in the last week
- Items failing acceptance tests
- Items closed within the last 15 days
- Items removed as they're duplicate
- Items closed and then reactivated
- Stories in the Code/Doing column
- Items in the Expedite swimlane
- Items in a swimlane containing "Test"

Date and iteration-based queries

- Items created in the last 30 days
- Items modified on a specific date
- Items resolved today
- Items closed within a specific time period
- Items whose updated status
- Items closed in the current sprint

Link and attachment count queries

- All child items of a single epic
- All related items
- Items with one or more attachments
- Items with 2 or more hyperlinks
- Items containing external links

Identity-based queries

- Active items assigned to me
- Closed items that were assigned to me
- Active items assigned to my team
- Items I've modified in the last 30 days
- Items I closed
- Items I resolved in the last week
- Team or group membership queries
- My recent activity work items

Work item count and numeric field queries

- Count of active bugs per developer
- Count of bugs by area and states
- Sum of story points and their status
- Burnup chart of user stories for a sprint
- Sum of remaining work per developer

History, Discussion, and revision-change queries

- History contains a specific word
- History doesn't contain a specific word
- Reactivated items
- Items closed within a time period
- Items you've been associated with

Team focus queries

- Assigned to a member of a team
- Assigned to a team's area path
- Assigned to a team's current sprint
- Assigned to a team's sprint window

- Assigned to a member of a team
- Assigned to a team's current sprint |

Build and test field queries

- List bugs and linked test cases
- List automated test cases
- List requirement-based test suites
- List query-based test suites

Other

- List deleted work items (Recycle bin)
- Query by field value comparisons

Query tasks

- Add a query
- Add a query chart
- Add a query to a dashboard
- Add a work item chart to a dashboard
- Add a query folder
- Add columns to query results
- Bulk modify query items
- Copy query URL
- Define a clause
- Delete a query
- Direct-links query
- Edit a query
- Email a query
- Export a query to Excel
- Favorite a query
- Filter a query
- Flat-list query
- Group a clause
- Group a chart by tags
- Open a query
- Query across projects
- Query based on tags
- Rename a query
- Run a query
- Save a query
- Set query permissions
- Tree query
- Triage query items
- View a query
- Understand link types
- Ungroup a clause
- Work Item Query Language (WIQL)

Operators and macros supported for each data type

The following table indicates the operators and macros available for the different field data types. Each field is associated with a data type. You can find the data type listed in the descriptions of each field, which you can look up using the Work item field index. Operators available for defining a query clause depend on the data type of the field that you select. For more detailed descriptions of data types, operators, and macros, see Query fields, operators, and macros.

> **NOTE**
> The following macros are only supported from the web portal: **@CurrentIteration**, **@CurrentIteration +/- n**, **@Follows**, **@MyRecentActivity**, **@RecentMentions**, **@RecentProjectActivity**, and **@TeamAreas**. Queries that contain these macros won't work when opened in Visual Studio/Team Explorer, Microsoft Excel, or Microsoft Project.

Data type

Description

Supported operators and macros

Boolean [1]

Supports a True/False value. Query samples: Query by assignment or workflow changes.

= , <> , =[Field] , <>[Field]

DateTime

A date field in which you can specify a variable, such as **@Today** or **@Today-1**, or a value, such as 1/1/2012. Enter dates in the Date Pattern you set for your personal profile. (See Set personal preferences for details.)

For query examples, see Query by date or@CurrentIteration.

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever **Macros**: **@Today**, valid with any **DateTime** field

Additional macros supported on Azure DevOps 2019 Update 1 and later versions:: **@StartOfDay**, **@StartOfWeek**, **@StartOfMonth**, and **@StartOfYear**, valid with any **DateTime** field

Double

Also referred to as **Decimal** and includes **picklistDouble**[2]. A real number, such as 0.2 or 3.5.

Query examples: Query by numeric fields.

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever

GUID

A character string that represents a unique ID.

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever

History

Custom formatted field used to track historical information and only assigned to the **History** field.

Query examples: History and auditing.

Contains Words, Does Not Contain Words

HTML

Text strings that support formatted descriptions, such as the **Description** or **Repro Steps** fields. These fields are automatically indexed for full-text search when full-text search is available. Query samples: Query by titles, IDs, and rich-text fields.

Contains Words, Does Not Contain Words, Is Empty[3], Is Not Empty[3]

Identity

A String field that is used to hold a user identity.Query samples: Query by assignment or workflow changes.

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever **Macros**: **@me** valid for all Identity fields

Integer

Also includes **picklistInteger**[2]. A 32-bit integer that is signed, such as 0, 1, 2, 34.

Query samples: Query by numeric fields

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever

Macros: **@Follows**[4],
@MyRecentActivity[5],
@RecentMentions[5],
@RecentProjectActivity, valid when used with the **ID** field

PlainText

Multi-line text strings that support long descriptions and are automatically indexed for full-text search, when full-text search is available.
Query examples: Query by titles, IDs, and rich-text fields.

Contains Words, Does Not Contain Words, Is Empty[3], Is Not Empty[3]

String

Also includes **picklistString**[2]. Short single-line text that can contain up to 255 Unicode characters. String fields support the **Title** field, picklists (drop-down menus), user accounts, **Tags**, and other fields.

Query examples: Query by titles, IDs, and rich-text fields and Query by picklist value.

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever

Macros: **[Any]**, valid with the **Work Item Type** field
@Project[6], valid with the **Team Project** field

TreePath

Field type that supports the **Area Path** and **Iteration Path** fields. You define the tree structure for a project —area paths and iteration paths.

Query examples: Query by area or iteration path and Query by date or current iteration.

Under, Not Under

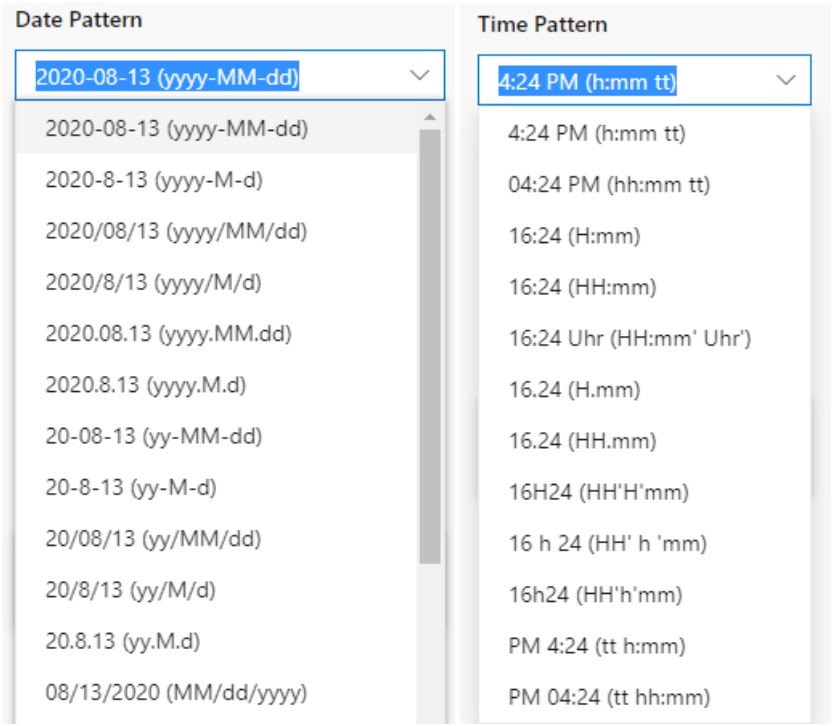
Macros: **@TeamAreas**[7], valid with **Area Path** field
@CurrentIteration[8] and **@CurrentIteration+/- n**[9] valid with the **Iteration Path** field

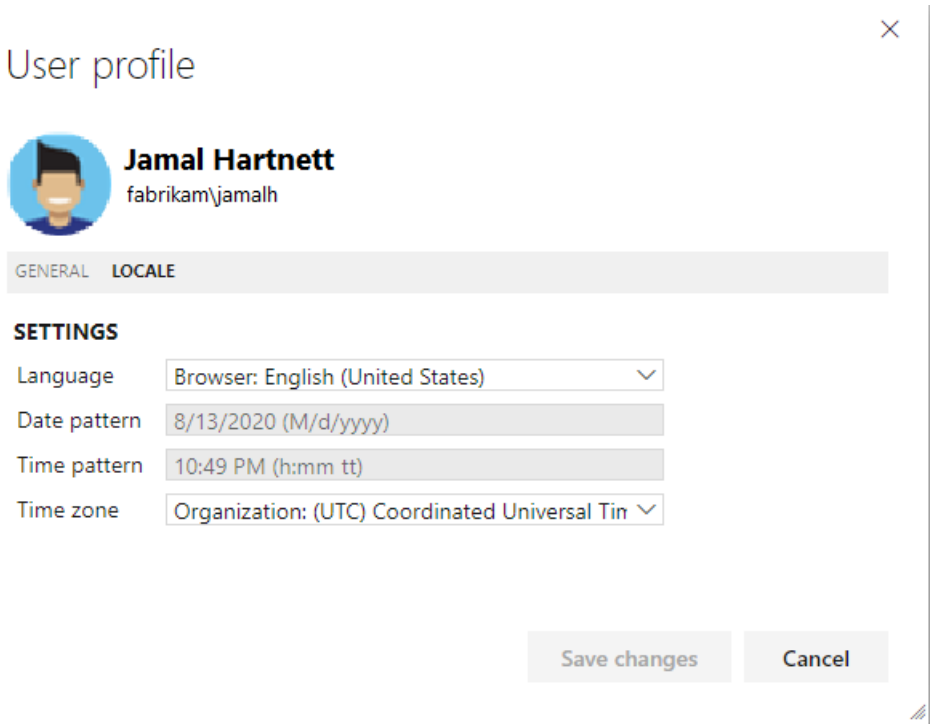
NOTE

1. The **Boolean** data type field is supported for TFS 2017 and later versions.
2. The **picklist...** data types are only assigned to custom fields defined for an inherited process. The Inherited process model is only supported for Azure DevOps Services and Azure DevOps Server 2019.
3. The **Is Empty** and **Is Not Empty** operators are supported for Azure DevOps Server 2019 RC2 and later versions.
4. The **@Follows** macro is supported for TFS 2017 and later versions.
5. The **@MyRecentActivity**, **@RecentMentions,** and **@RecentProjectActivity** macros are supported for Azure Boards and TFS 2018.2 and later versions.
6. The **@Project** macro is supported for TFS 2015.1 and later versions. The system automatically defaults to filtering based on the current project. To learn more, see Query across projects.
7. The **@TeamAreas** macro is supported for Azure Boards and Azure DevOps Server 2019 and later versions.
8. The **@CurrentIteration** macro is supported for TFS 2015 and later versions, and only when run from the web portal.
9. The **@CurrentIteration +/- n** macro is supported for Azure Boards and Azure DevOps Server 2019 and later versions, and only when run from the web portal.

Date and time pattern

The date and time pattern you enter for **DateTime** fields should match that which you select through your profile. To view or change your selection, see Set user preferences, Time and Locale.





Example queries for select fields

The following table lists common query fields and their data type for which sample queries are provided. To determine the data type of a field, see Work item fields and attributes, List field attributes.

A

- Acceptance Criteria (HTML)
- Activated By (Identity)
- Activated Date (DateTime)
- Activity (String)
- Area Path (TreePath)
- Assigned To (Identity)
- Attached File Count (Integer)
- Automated Test Name (String)
- Automated Test Type (String)

B

- Blocked (String)
- Board Column (String)
- Board Column Done (Boolean)
- Board Lane (String)
- Business Value (String)

C

- Changed By (Identity)
- Changed Date (DateTime)
- Closed By (Identity)
- Closed Date (DateTime)
- Comment Count[1] (Integer)
- Committed (String)
- Completed Work (Decimal)

- Created By (Identity)
- Created Date (DateTime)

D-E-F

- Discipline (String)
- Description (HTML)
- Due Date (DateTime)
- Effort (Decimal)
- External Link Count (Integer)
- Finish Date (DateTime)
- Found In Build (String)

H-P

- History (History)
- Hyperlink Count (Integer)
- ID (Integer)
- Integrated in Build (String)
- Iteration Path (TreePath)
- Link Comment (Integer)
- Node Name (String)
- Original Estimate (Decimal)
- Parameters (HTML)
- Priority (Integer)

R

- Reason (String)
- Related Link Count (Integer)
- Remaining Work (Decimal)
- Repro Steps (HTML)
- Resolved By (Identity)
- Resolved Date (DateTime)
- Resolved Reason (String)
- Rev (Integer)
- Revised Date (DateTime)

S

- Severity (String)
- Size (Decimal)
- Stack Rank (Decimal)
- Start Date (DateTime)
- State (String)
- State Change Date (DateTime)
- Steps (HTML)
- Steps to Reproduce (HTML)
- Story Points (Decimal)
- System Info (HTML)

T

- Tags (String)

- Target Date (DateTime)
- Task Type (String)
- Team Project (String)
- Test Suite Type (String)
- Title (System)
- Triage (String)

V-W

- Value Area (String)
- Work Item Type (String)

> **NOTE**
>
> 1. The **Comment Count** field is supported for TFS 2017 and later versions.

Related articles

- Query by field value comparisons
- Guidance to create high-performing queries
- Use categories to group work item types
- Define a managed query
- Work item field index

Run a semantic work item search in Azure Boards and Azure DevOps

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

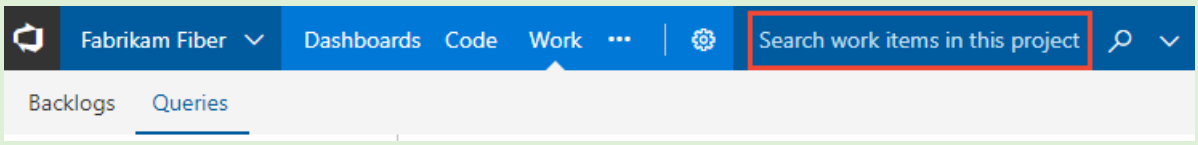
You can find work items by using shortcut filters or by specifying keywords or phrases. You can also use specific fields/field values, assignment or date modifications, or using Equals, Contains, and Not operators. Searching isn't case-sensitive. Use semantic searches when you want to do the following tasks:

- Find a specific work item using its ID or a keyword
- Find one or more work items across all projects in a fast, flexible manner
- Run a full text search across all work item fields
- Review work items assigned to a specific team member
- Search against specific work item fields to quickly narrow down a list of work items
- Determine what key words will support a managed search

You can run a powerful semantic search from the web portal for Azure DevOps Services or for on-premises deployments when the server instance has been configured with the work item search extension.

TIP

If semantic search has been configured, you'll notice that the search box moves into the blue bar as shown in the following image.



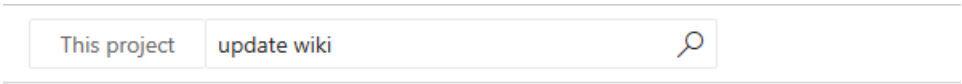
Start a semantic search for work items

With semantic search you can search:

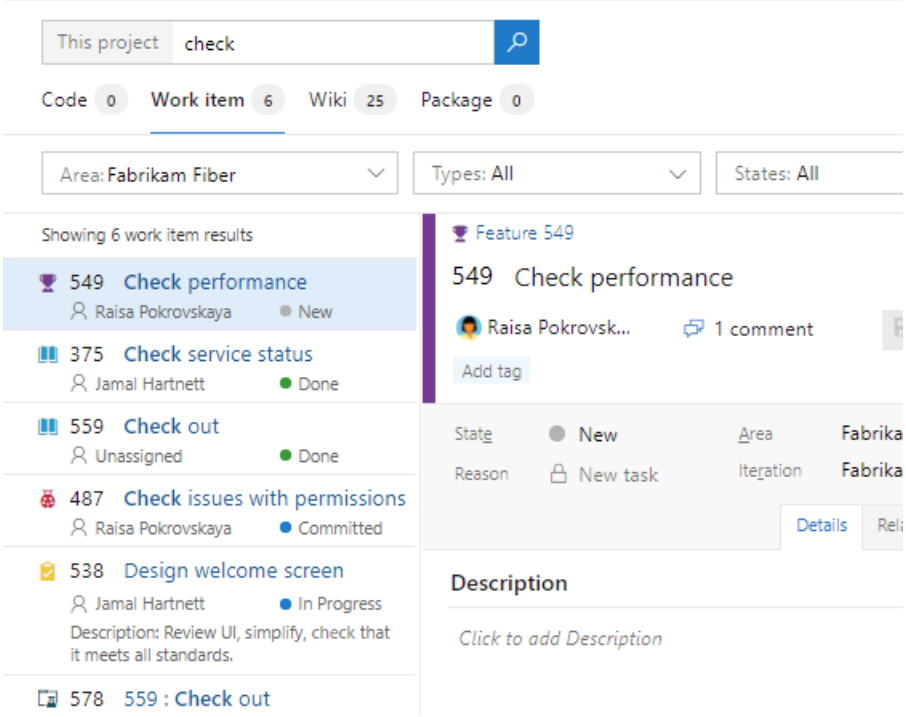
- Across one or more projects
- Across all work item fields using free text
- Against specific work item fields

Free text search easily searches across all work item fields, including custom fields, which enables more natural searches. Search results are displayed in a snippet view where the search matches found are highlighted. Semantic search also integrates with work item tracking, providing familiar controls to view, edit, comment, and share information within a work item form.

1. Choose any **Boards** page, enter a keyword or phrase in the search box, and press *Enter* or choose the start search icon.



2. Search results are displayed in a snippet view where the matches found are shown in bold.



This search is a full text search that uses simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" will also find instances of the word "updated" and "update". Searches *aren't* case-sensitive.

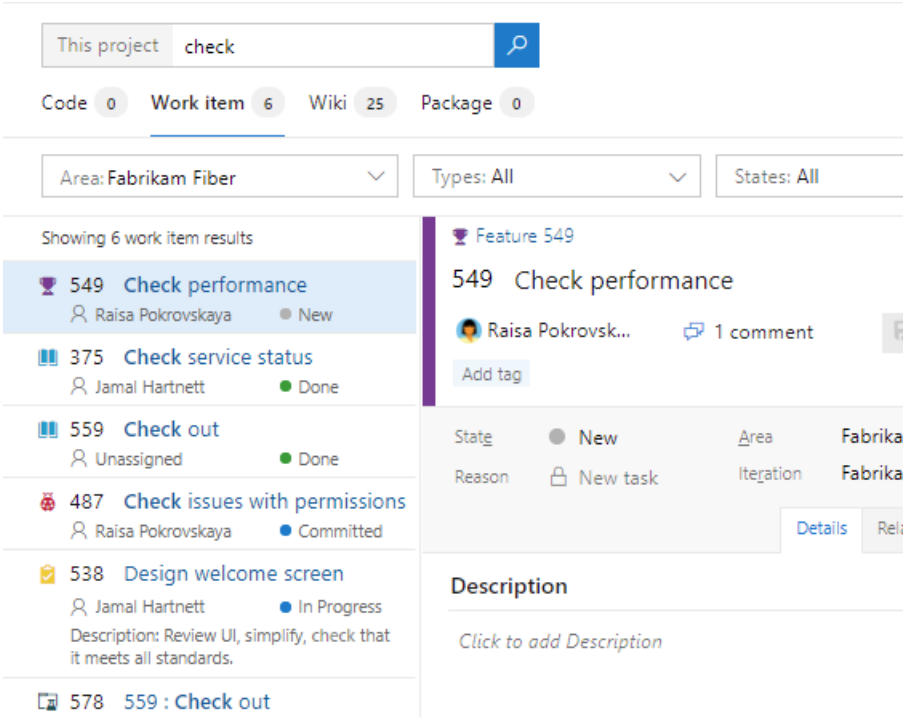
3. Select a snippet of a work item to display it in the right window.

Open the search results in a new browser tab from a search box by pressing *Ctrl* + *Enter* or by holding *Ctrl* and clicking the icon. In Google Chrome, press *Ctrl* + *Shift* + *Enter* to switch the focus to the new browser tab.

1. In the search box, check that the text says *Search work items*. If it doesn't, use the selector to select it.



2. Enter a search string in the text box, and press *Enter* (or choose the icon) to start your search.
3. Search results are displayed in a snippet view where the matches found are shown in bold.



This search is a full text search that uses simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" will also find instances of the word "updated" and "update". Searches *aren't* case-sensitive.

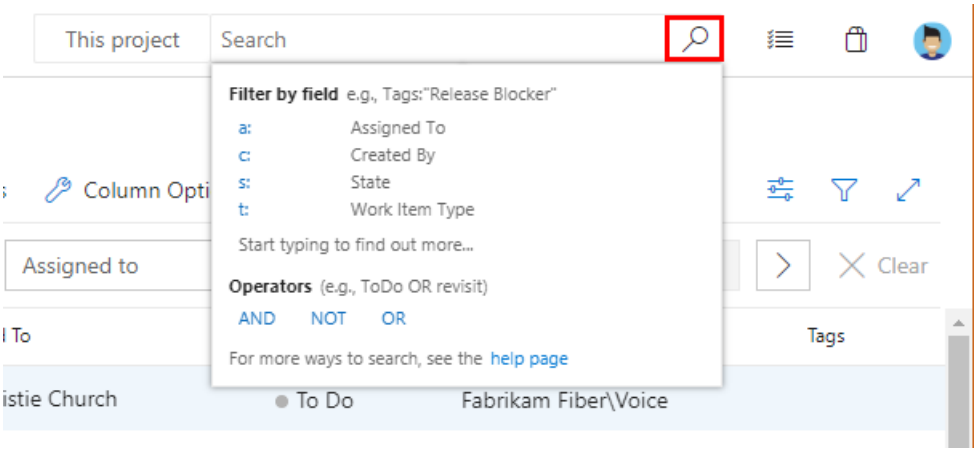
4. Select a snippet of a work item to display it in the right window.

   Open the search results in a new browser tab from a search box by pressing *Ctrl* + *Enter* or by holding *Ctrl* and clicking the icon. In Google Chrome, press *Ctrl* + *Shift* + *Enter* to switch the focus to the new browser tab.

Choose **New navigation** for guidance. **Previous navigation** isn't supported for Azure DevOps Server 2019.

Fine-tune semantic search results

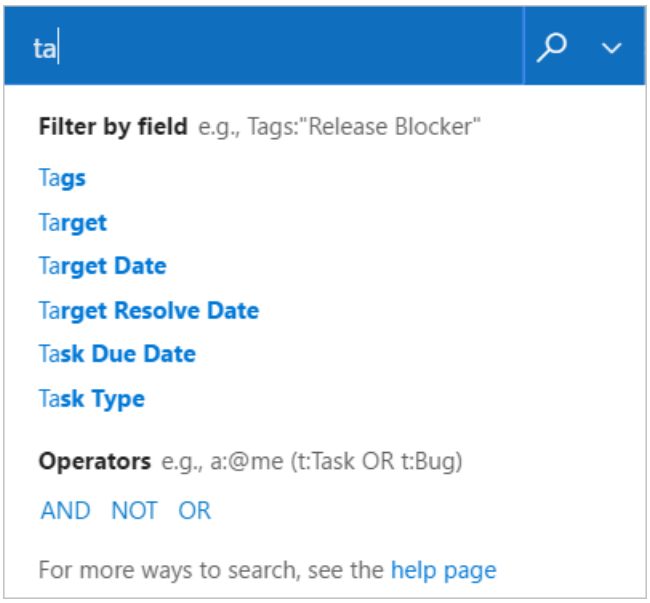
1. Fine-tune your search by specifying the fields to search. Enter `a:` and a user name to search for all items assigned to that user.



   The quick filters you can use are:

   - `a:` for **Assigned to:**
   - `c:` for **Created by:**
   - `s:` for **State**
   - `t:` for **Work item type**

2. Start typing the name of a field in your work items; for example, type `ta`.

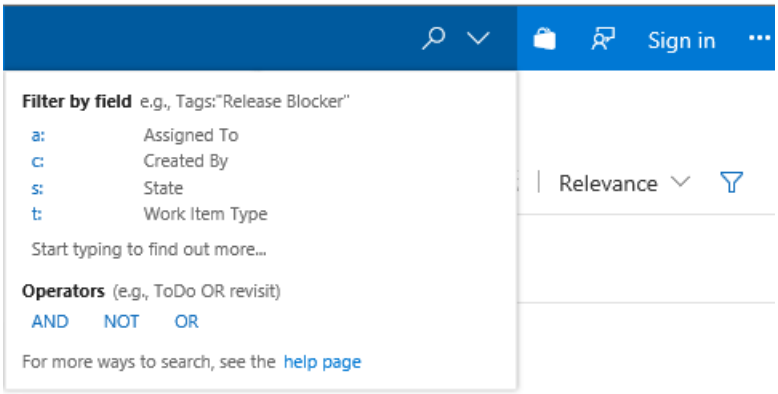


The dropdown list shows work item field name suggestions that match user input and help the user to complete the search faster. For example, a search such as `tags:Critical` finds all work items tagged "Critical".

3. Add more filters to further narrow your search, and use Boolean operators to combine terms if necessary. For example, `a: Chris t: Bug s: Active` finds all active bugs assigned to a user named "Chris".

4. Narrow your search to specific types and states, by using the drop-down selector lists at the top of the results page.

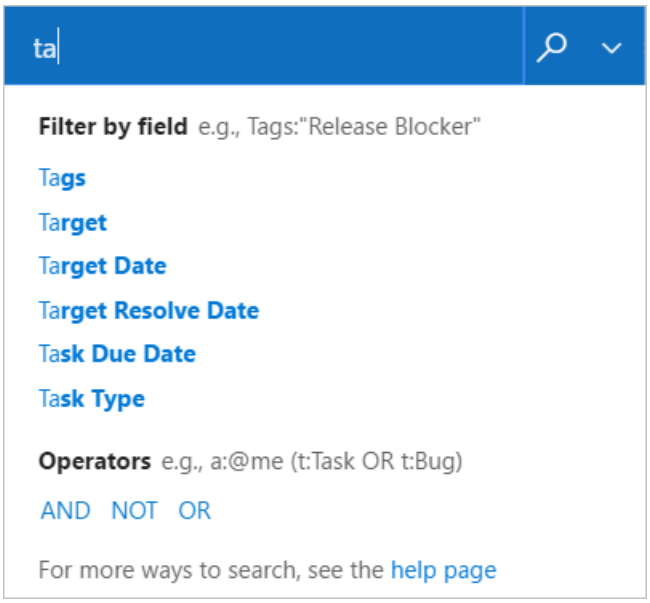
1. Fine-tune your search by specifying the fields to search. Enter `a:` and a user name to search for all items assigned to that user.



The quick filters you can use are:

- `a:` for **Assigned to:**
- `c:` for **Created by:**
- `s:` for **State**
- `t:` for **Work item type**

2. Start typing the name of a field in your work items; for example, type `ta`.



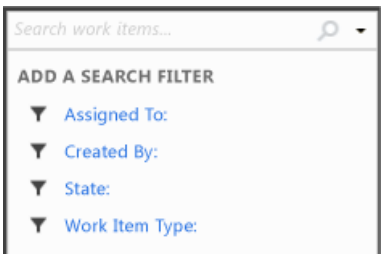
The dropdown list shows work item field name suggestions that match user input and help the user to complete the search faster. For example, a search such as `tags:Critical` finds all work items tagged "Critical".

3. Add more filters to further narrow your search, and use Boolean operators to combine terms if necessary. For example, `a: Chris t: Bug s: Active` finds all active bugs assigned to a user named "Chris".

4. Narrow your search to specific types and states, by using the drop-down selector lists at the top of the results page.

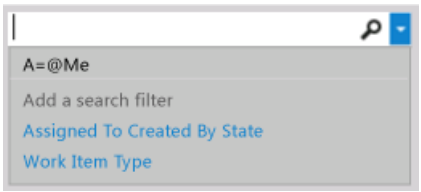
From the web portal

Improvised search isn't available from Azure DevOps Services. Only semantic search.

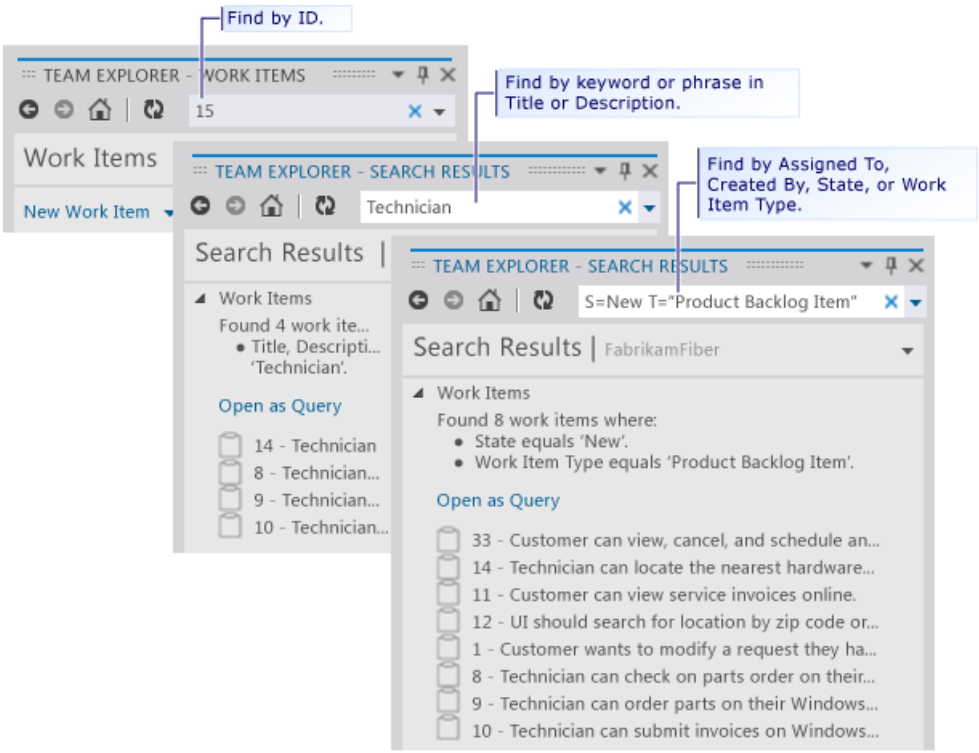


From Visual Studio Team Explorer

Open the ▾ context menu and select an option.



Finding work items using the search box (Team Explorer)



You can combine shortcuts and use search operators within the search box.

Use the **Clear** button to remove content from the search box. To switch your context to the search box from within Visual Studio, enter **Ctrl+'**.

Find items based on keywords or phrases

Keywords or phrases that you type into the search box return a list of work items that contain those keywords or phrases in the **Description**, **Repro Steps**, or **Title** fields. Enclose each phrase in quotation marks.

In the **Search work items** box, type a keyword or phrase that appears in the **Title**, **Description**, or **Repro Steps** fields for the work items of interest.

Enclose multiple words in quotation marks.

For example, to find work items with the specified keywords in the **Title** or **Description** fields:

- For the keyword "duplication", enter **duplication**.
- For the phrase "Getting Started", enter **"Getting Started"**.
- For the phrase "Getting Started" or the keyword "feature", enter **feature "Getting Started"**.

FILTER FOR ITEMS THAT CONTAIN THESE KEYWORDS OR PHRASES:	TYPE THE FOLLOWING STRING:
Duplication	`duplication`
Getting Started	`"Getting Started"`
Feature and Getting Started	`feature "Getting Started"`

You can run partial or exact match queries on a keyword or a phrase contained within any text field. Or, you can run a full-text search query by filtering on keywords and phrases contained within the full-text search index. Team Foundation automatically indexes all long-text fields with a data type of **PlainText** and **HTML** and the **Title** field for full-text search.

Find items based on specific fields and field values

To find work items based on a keyword or phrase contained within other text string fields, specify either the friendly name or the reference name of the field. Enclose each phrase in quotation marks. You can determine the friendly name of a field by hovering over the field within a work item form. To determine the reference name of commonly used fields or to find a field that isn't listed on the form, see Work item field index.

FILTER FOR ITEMS THAT MEET THIS CRITERIA:	TYPE THE FOLLOWING STRING:
Contains one attached file.	`System.AttachedFileCount=1`

Cut user stories.	`T:Story Reason=Cut`
	Or
	`T="User Story" System.Reason=Cut`

Resolved by Peter.	`"Resolved By":Peter`
	Or
	`Microsoft.VSTS.Common.ResolvedBy:Peter`

Modified today.	`"Changed Date"=@Today`
Created yesterday as a test activity.	`"Created Date"=@Today-1 Activity=Test`

NOTE

Some fields, such as **History** and **Description**, do not support partial word text searches. For example, if the **History** field contains the phrase `reproducible behavior` and you search for `History:repro` the work item will not be found. However, if you search for the complete string `History:reproducible` the work item will be found.

Use @Me or @Today macros

The **@Me** macro expands to the full name of the current user in any work item search. The **@Me** macro is especially useful for creating a search that you can share with other users, and it can simplify your work by reducing the number of characters you must type to specify your own user name. For a description of all macros, see Query fields, operators, and macros, Query macros or variables.

Filter for

Type the following string

Currently assigned to you

`A=@Me`

Created by you

`C=@Me`

Resolved yesterday

`Resolved Date=@Today-1`

Modified seven days ago

`System.ChangedDate=@Today-7`

Created yesterday under the Phone Saver team

`Created Date=@Today-1 And Area Path=FabrikamFiber\Phone Saver`

Use Equals, Contains, and Not operators

Use the following search operators to specify search criteria:

= (EQUALS) to search for exact matches of text.
: (CONTAINS) to search for partial matches of text.
- (NOT) to exclude work items that contain certain text. The **NOT** operator can only be used with field names.

The following examples show how to use operators when you create a search string.

FILTER FOR ITEMS THAT MEET THIS CRITERIA:	TYPE THE FOLLOWING STRING:
Assigned to Peter and not Active.	`A:Peter -S=Active`
In which the Activity field wasn't `Development`.	`- Activity=Development`
Resolved by Peter.	`"Resolved By":Peter`
Contain the keyword `triage` in the title or description, aren't assigned to you, and aren't closed.	`triage -A=@me -S=Closed`
Active bugs that are assigned to you that don't contain the keyword `bugbash` in the title.	`S=Active T=bug A=@Me -Title:bugbash`

Related articles

- About managed queries
- Define a query
- Query fields, operators, and macros
- Work item field index

Q & A

Q: Does the search box support less than/greater than operators?

A: No. The search box doesn't recognize comparison operators such as greater than (>) or less than (<). It translates queries with these operators into a search phrase.

View, run, or email a work item query

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

To find work items assigned to you or your team, run a query. Many work item queries are predefined with your process. Members of your team may have created shared queries that you can view and run. Often, it's easier to define a new query by building on the query definition that's already available to you.

Prerequisites

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

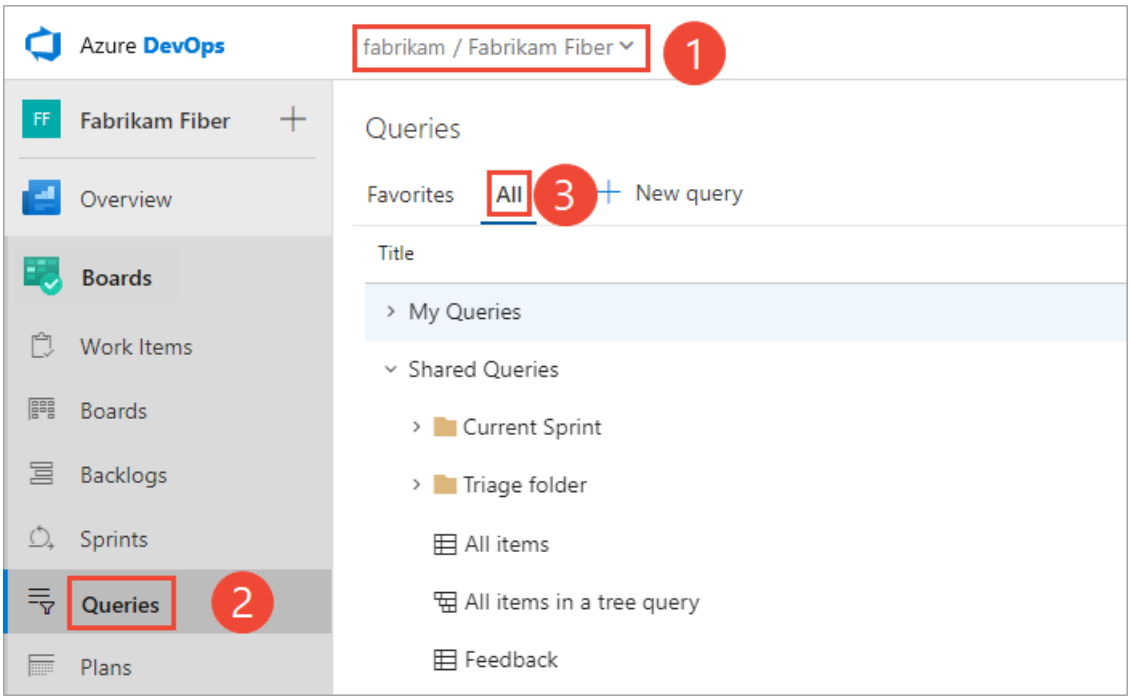
> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

Open Queries

- Browser
- Visual Studio

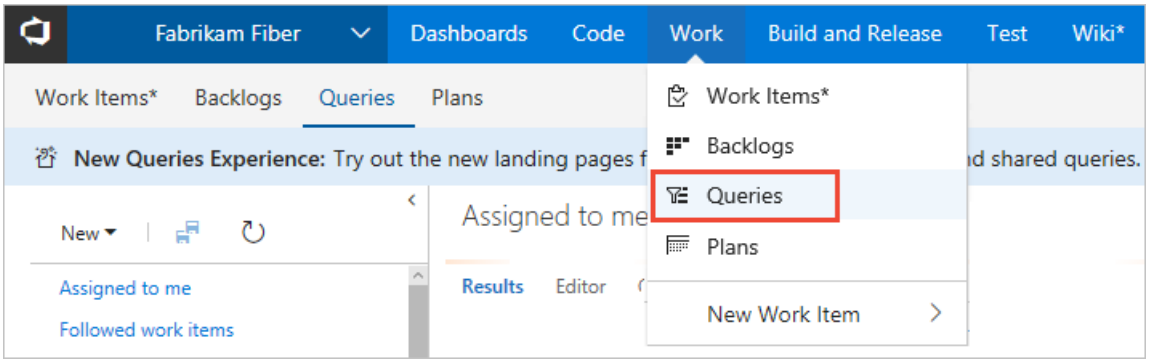
From your web browser, (1) check that you have selected the right project, (2) choose **Boards > Queries**, and then (3) choose **All**.



If it is your first time opening **Queries**, the page opens to **Favorites**. This page lists those queries that you have indicated are a favorite. Otherwise, you can choose **All** to view all queries you've defined and shared queries defined for the project.

> **TIP**
> Queries you or your team have chosen as favorites show up on the **Favorites** page. Favorite queries along with other objects also appear on your **Project** page. To learn more, see Set personal or team favorites.

From your web browser, open **Boards > Queries**.



Run a query in Azure Boards

To run any query, expand a folder and choose the title of the query. The view opens to display the query **Results**.

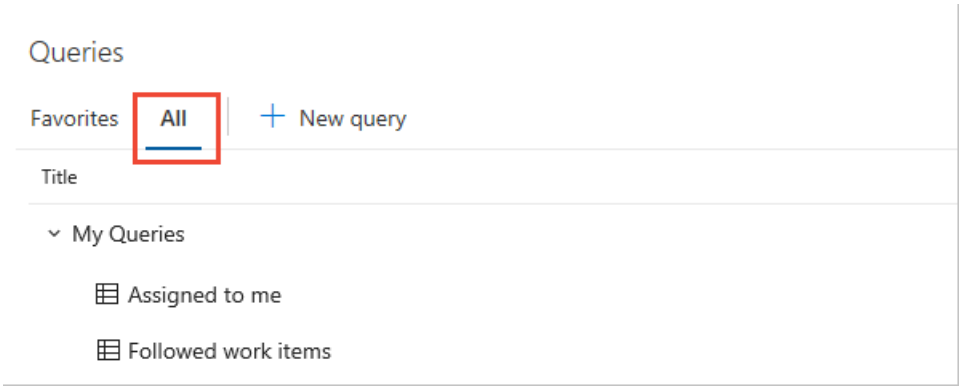
You can also run a query by using the Azure DevOps command line interface.

- Browser
- Visual Studio

TIP

The **Queries** page, as with other web portal pages, remembers the view you last navigated to and returns you to that view.

1. Choose **All** to open the page where you can view all queries you've defined or that are shared within your project.



2. Choose **My Queries** as needed.
   - To view all work items assigned to you, choose **Assigned to me**. This query uses the **@Me** macro to list all work items assigned to you.
   - To view all work items you're following, choose **Followed work items**. This query uses the **@Follows** macro (ID in @Follows) to list all work items you've chosen to follow. To learn more, see Follow a work item or pull request.
3. Choose **Shared Queries** to expand the folder and access queries saved as shared queries.
4. Choose a folder within a breadcrumb to open a query folder.



Run a query from the command line

You can run a query in the CLI with the az boards query command. To get started, see Get started with Azure DevOps CLI.

```
az boards query [--id]
                [--org]
                [--path]
                [--project]
                [--wiql]
```

Parameters

- **id**: The ID of an existing query. Required unless--path or--wiql is specified.
- **wiql**: The query in Work Item Query Language format. Ignored if--id or--path is specified.
- **path**: The path of an existing query. Ignored if--id is specified.
- **org**: Azure DevOps organization URL. You can configure the default organization using `az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using `git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.
- **project**: Name or ID of the project. You can configure the default project using `az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up using

`git config`.

Example

The following command runs a query with the specified ID and shows the result in table format.

```
az boards query --id 6c286d74-26a5-4cce-bfcf-bf9123495bfe  --output table

Priority    Node Name         Work Item Type    Title                             Remaining Work
----------  ----------------  ----------------  --------------------------------  ----------------
1           Voice             Bug               Apply fix elsewhere as needed
2           CMMI              Bug               Slow response on form
1           Fiber             Bug               Check issues with permissions     0
2           Fiber             Bug               Voicemail hang issue              0
2           FabrikamBB        Bug               Research slow response time
1           FabrikamBB        Bug               Fix performance issues            0
```

The following command runs a query with the specified WIQL and shows the result in table format.

```
az boards query --wiql "SELECT [Microsoft.VSTS.Common.Priority], [System.NodeName], [System.WorkItemType],
[System.Title], [Microsoft.VSTS.Scheduling.RemainingWork], [System.AssignedTo], [System.State],
[System.Tags], [System.AreaPath] FROM workitems WHERE [System.WorkItemType] = 'Bug' AND [System.AreaPath] =
'Fabrikam Fiber' ORDER BY [System.WorkItemType]" --output table

Priority    Node Name       Work Item Type    Title                 Remaining Work
----------  --------------  ----------------  ----------------      ----------------
2           Fabrikam Fiber  Bug               Slow response on form
2           Fabrikam Fiber  Bug               Check permissions
2           Fabrikam Fiber  Bug               Fix performance issue
2           Fabrikam Fiber  Bug               Secure Sign-in
```

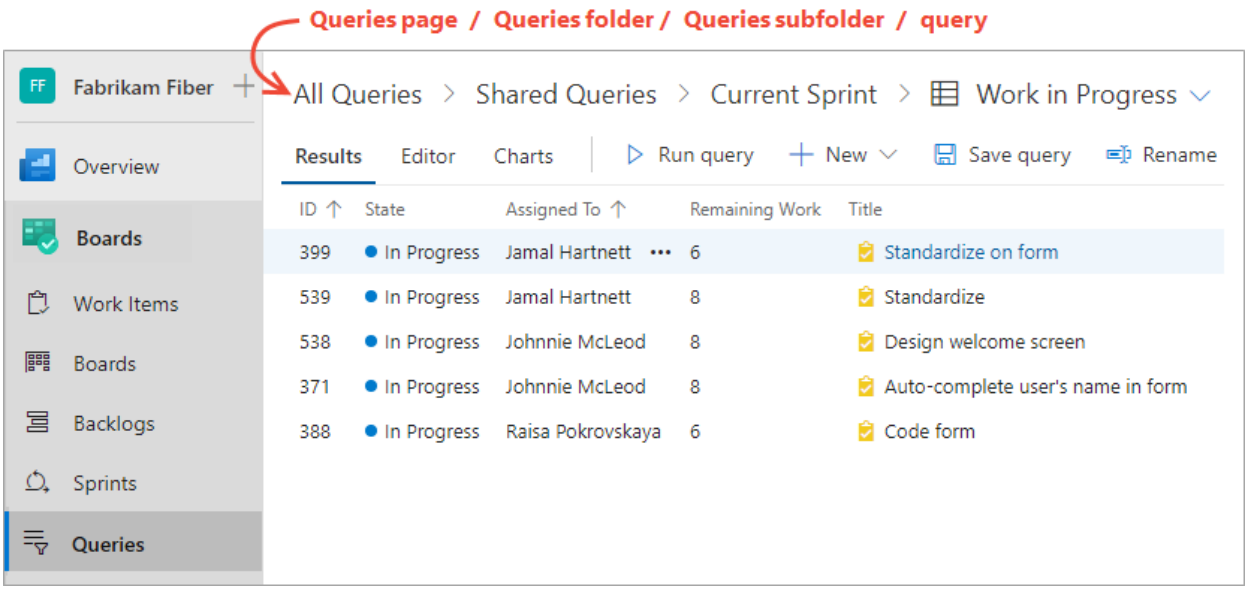
Query directory, query folders, and breadcrumbs

Query folders

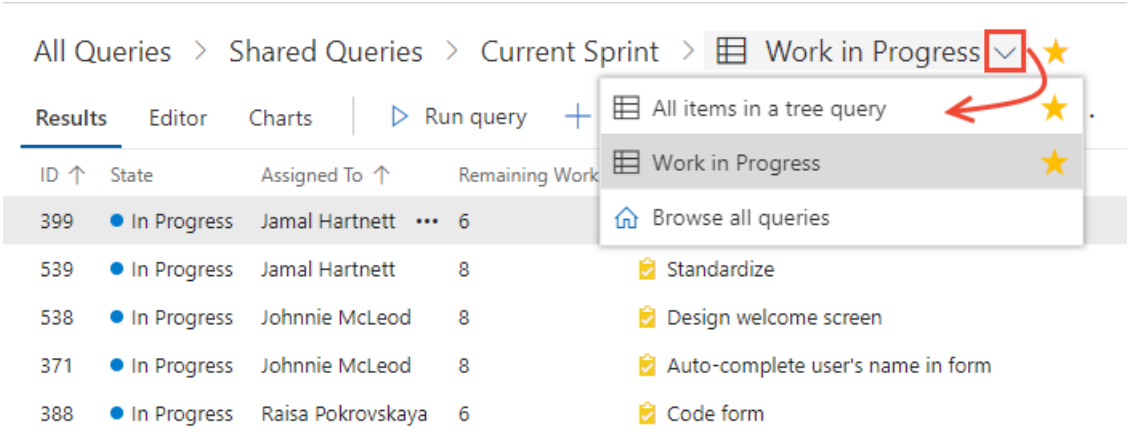
> **NOTE**
> You can't add folders to **My Favorites** or **Team Favorites**.

- Browser
- Visual Studio

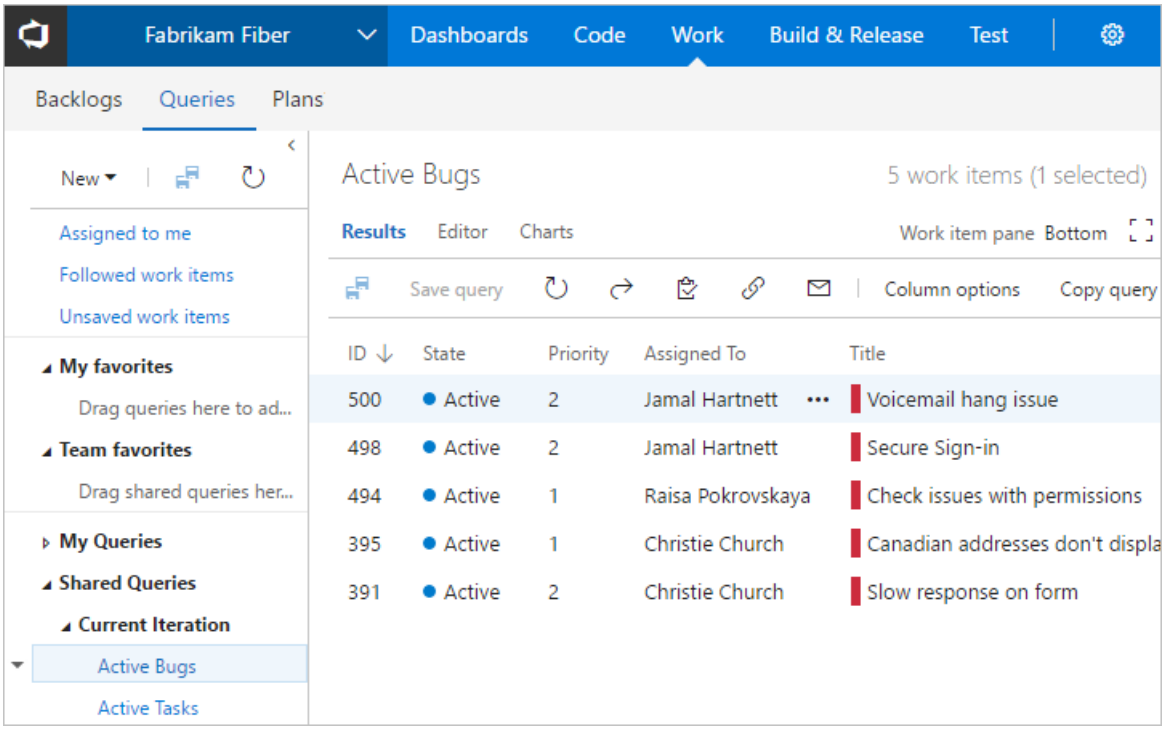
The **Queries** page contains a directory-focused view that you can filter to find specific queries of interest. When working in the **Queries** pages, you can navigate to a subfolder, folder, or page.



Also, you can choose a query that you've favorited from the selector menu, Or, you can choose to browse all queries, which returns you to the **All Queries** page.



The **Queries** page displays the folder structure in the left pane. You can expand and collapse folders, rename folders, and drag and drop queries from one folder to another. To learn more, see Manage and organize queries.



For more information, see Query FAQs, Navigate, and Folders.

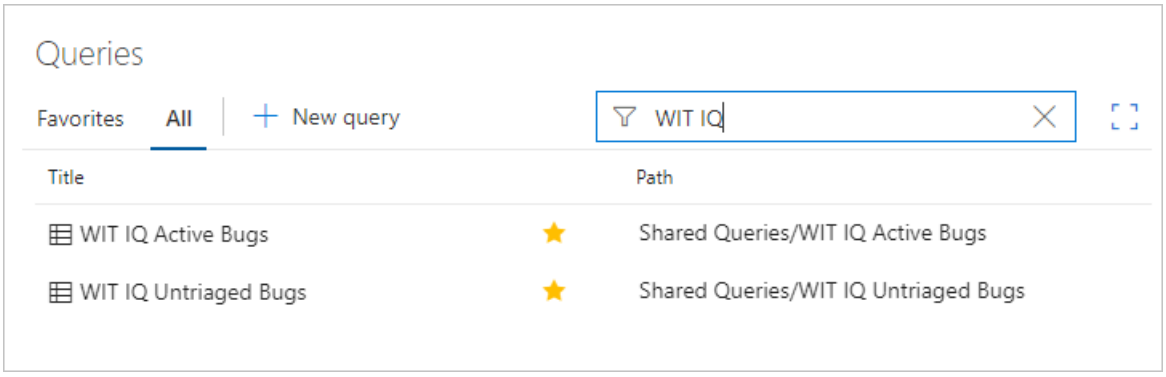
All and Favorites supported tasks

You can do most tasks for viewing and running queries from each of the queries list pages as indicated in the following table. Only queries you save under **My Queries** and have favorited show up under **My Favorites**. Only queries saved under **Shared Queries** can be favorited by a team.

TASK	FAVORITES (BROWSER)	ALL (BROWSER)	WORK ITEMS (TEAM EXPLORER)
View all favorited queries, yours or a team you belong to	✔		✔
View all your queries or shared queries for the current project		✔	✔
Run a query, open the context menu for a query	✔	✔	✔
Expand or collapse container folders or query folders	✔	✔	✔
Filter the list of queries	✔	✔	
Favorite a query (for web portal, choose ☆)		✔	
Unfavorite a query (for web portal, choose ★)	✔	✔	✔
Add a new query: Choose	✔	✔	✔

Filter the list of queries

Enter a keyword into the filter box to filter the set of queries displayed on either the **Favorites** or **All** pages. To learn more about filtering, see Filter backlogs, boards, queries, and plans.



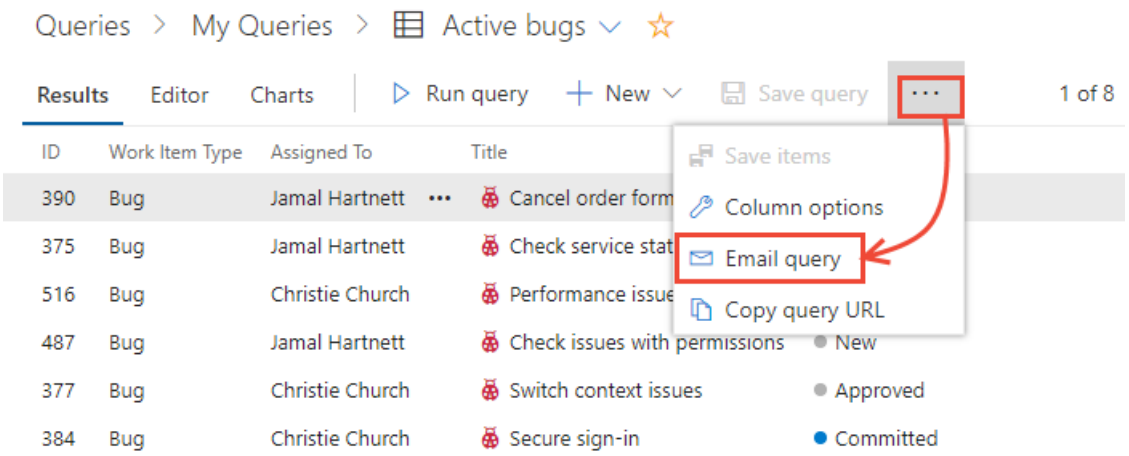
For more information, see Query FAQs, Navigate, and Folders.

Email query items or share a query URL

- Browser
- Visual Studio

From the **Query Editor** or **Results** view, you can email a formatted list of query items or copy the query URL.

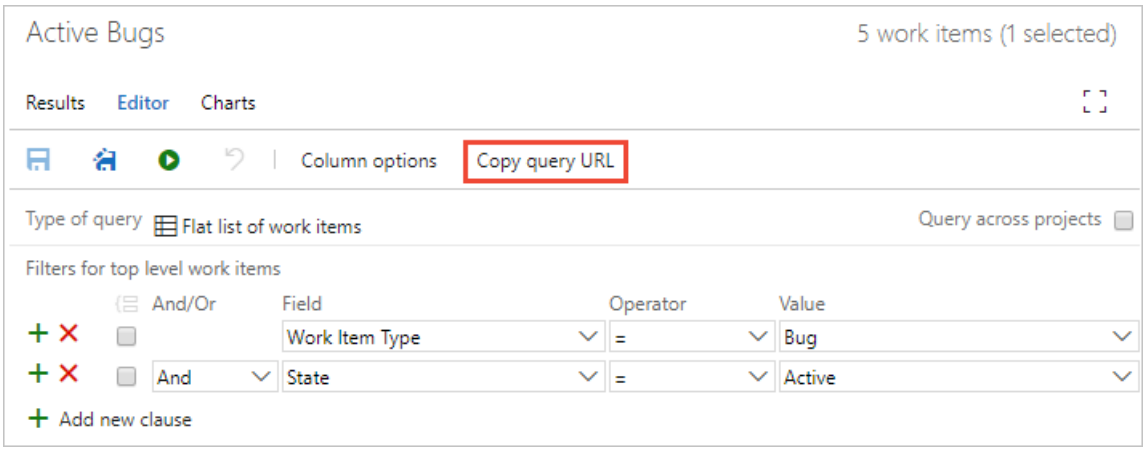
Choose the ··· actions icon to open the menu and select from the options listed, **Email query** or **Copy query URL**.



You can only send the email to individual address for a project member that is recognized by the system. Adding a team group or security group to the to line isn't supported. If you add an email account that the system doesn't recognize, you receive a message that one or more recipients of your email don't have permissions to read the mailed work items.

> **NOTE**
> To email a formatted list to people who aren't project members, you'll need to use the **Copy as HTML** option described in Copy a list of work items. For on-premises Azure DevOps, all email actions require an SMTP server to be configured. If you don't have an SMTP server configured, you can work around this by using **Copy as HTML**.

Choose **Copy query URL**. To email query items, see Copy a list of work items.



NOTE

With **Email query**, the system will email the formatted list to those teammates you select. To email a formatted list to people not part of the project, you'll need to use the **Copy as HTML** option described in Copy a list of work items. All email actions require an SMTP server to be configured. If you don't have an SMTP server configured, you can work around this by using **Copy as HTML**.

Try this next

Define a query

Related articles

- Manage queries and query folders
- Interactively filter backlogs, boards, queries, and plans
- Change column options
- Set personal or team favorites
- Keyboard shortcuts

Define a work item query in Azure Boards

10/4/2022 • 14 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

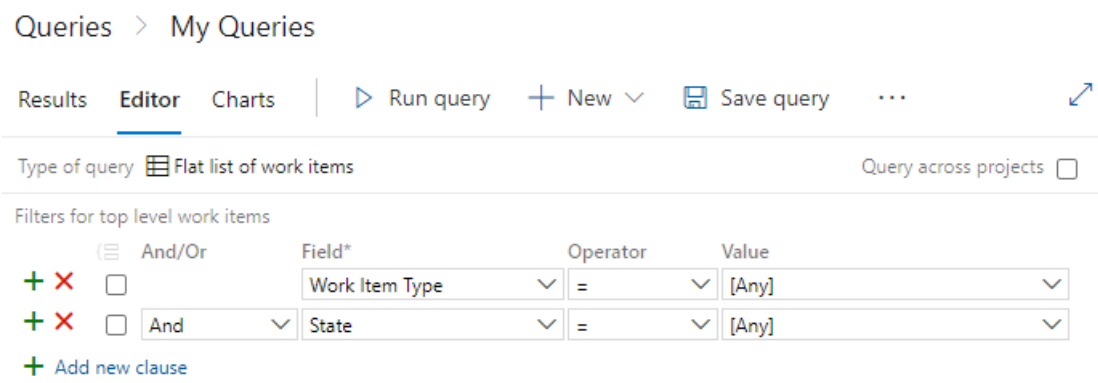
Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Work item queries generate lists of work items based on the filter criteria you provide. You can then save and share these managed queries with others. In contrast, semantic searches list work items, but can't be saved or shared.

Create queries from the web portal or from a supported client, such as Visual Studio Team Explorer and Team Explorer Everywhere. You can also define and import a work item query using WIQL syntax and a .wiq file. To support bulk updates or additions, import or export queries using Excel or .csv files.

Create queries from the web portal or from a supported client, such as Visual Studio Team Explorer and Team Explorer Everywhere. You can also define and import a work item query using WIQL syntax and a .wiq file. To support bulk updates or additions, import or export queries using Excel.

- Browser
- Visual Studio



If you find that your queries take too long to return results, review the Guidance to create high-performing queries.

In this article you'll learn:

- How to add or create a query
- How to query across projects
- How to group and ungroup query clauses
- How to create a tree of work items or a direct-links query

For quick access to all query tasks, supported operators—such as, `Contains`, `In`, `In Group`, and `<>` (not operator) — based on field data type, and query examples, see Query quick reference.

Choose a query filter

From the Query Editor, exercise the following filter functions. Choose the filter to jump to an article with sample queries. Along with the query filters, you can interactively apply filters to query results.

NOTE

Managed queries don't support proximity searches, however semantic searches do. In addition, semantic searches supports both `*` and `?` as wildcard characters and you can use more than one wildcard character to match more than one character. To learn more, see Functional work item search.

Filter features

Macros

- Compare fields
- Key words
- Linked work items
- Logical groupings
- Query macros
- Tags
- Was Ever
- Was Ever (Board Column)
- Wildcard

- Compare fields
- Key words
- Linked work items
- Logical groupings
- Query macros
- Tags
- Was Ever
- Wildcard

- Blank or empty fields
- Boolean searches
- Identity searches
- History and Discussion
- Kanban board fields
- In and Not In Group searches
- Search across projects

- Boolean searches
- History and Discussion
- In and Not In Group searches
- Search across projects

- [Any]
- @Me
- @Today
- @CurrentIteration, @CurrentIteration +/-n
- @Follows
- @MyRecentActivity, @RecentMentions, @RecentProjectActivity
- @StartOfDay, @StartOfMonth, @StartOfWeek, @StartOfYear

- @TeamAreas

- [Any]
- @Me
- @Today
- @CurrentIteration
- @Follows
- @MyRecentActivity, @RecentMentions, @RecentProjectActivity

Along with the filters you use from the Query Editor, you can interactively filter a query result using the **Filter** function. To learn how, see Interactively filter backlogs, boards, queries, and plans.

Prerequisites

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

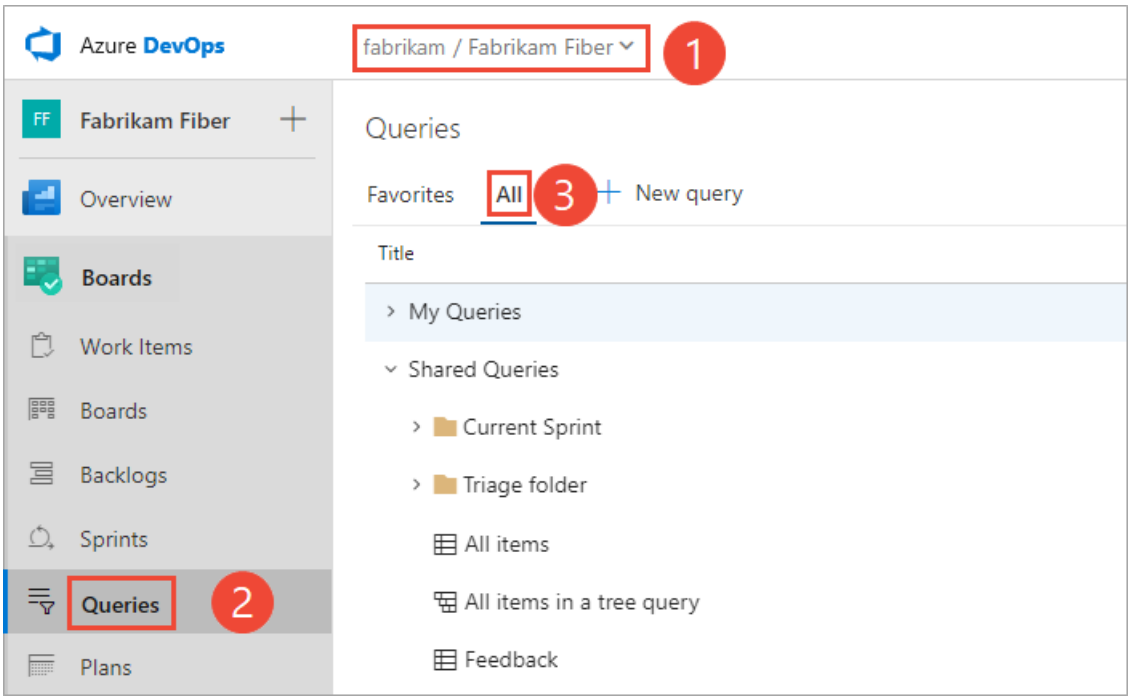
> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

Open Queries

- Browser
- Visual Studio

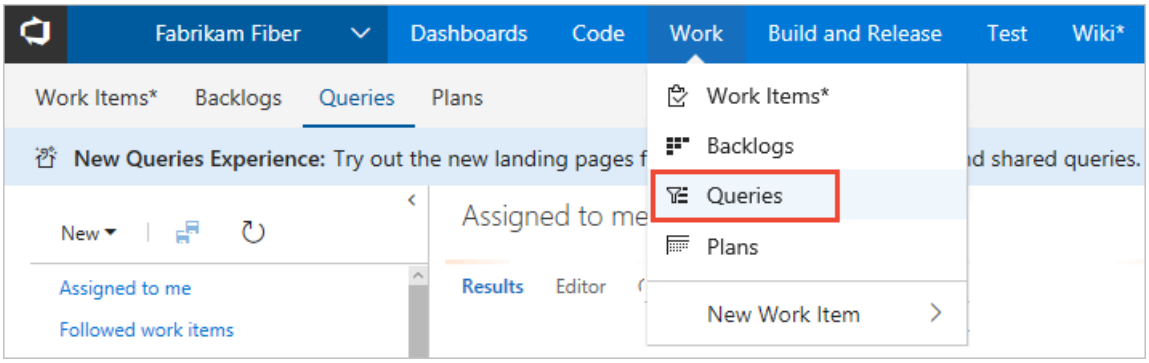
From your web browser, (1) check that you have selected the right project, (2) choose **Boards > Queries**, and then (3) choose **All**.



If it is your first time opening **Queries**, the page opens to **Favorites**. This page lists those queries that you have indicated are a favorite. Otherwise, you can choose **All** to view all queries you've defined and shared queries defined for the project.

> **TIP**
> Queries you or your team have chosen as favorites show up on the **Favorites** page. Favorite queries along with other objects also appear on your **Project** page. To learn more, see Set personal or team favorites.

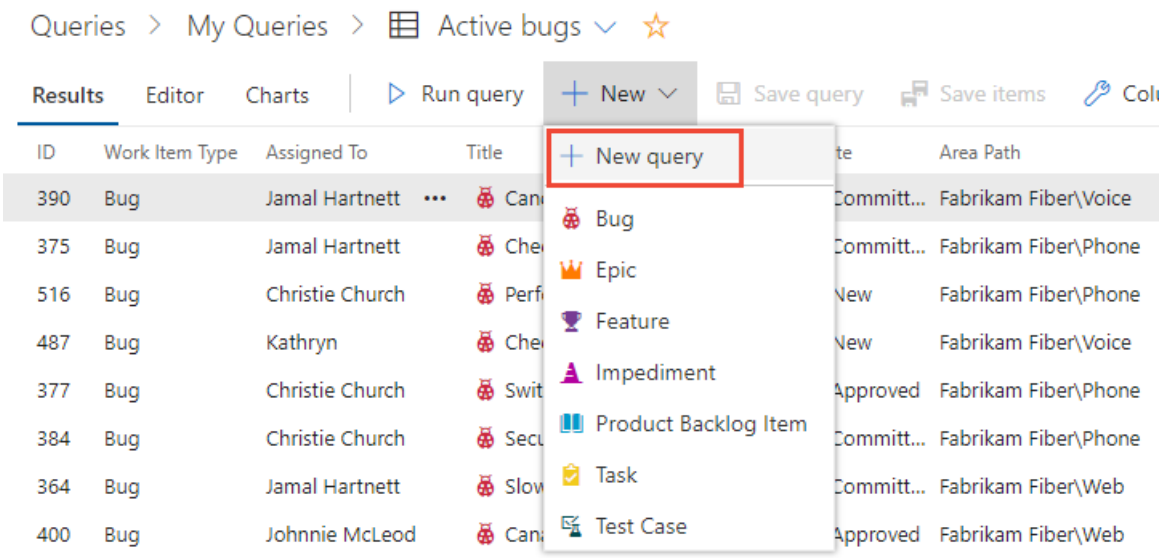
From your web browser, open **Boards > Queries**.



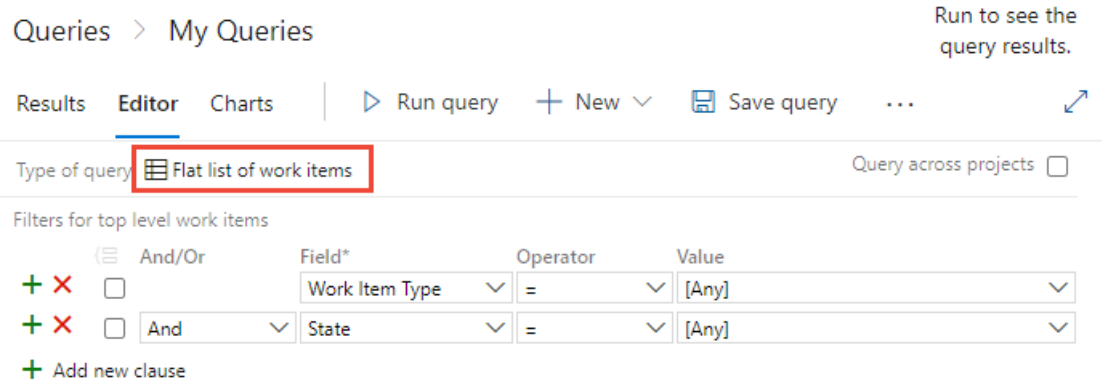
Define a flat-list query

You can start a fresh, new query from the **Queries** tab in the web portal or the **Work Items** tab in Team Explorer.

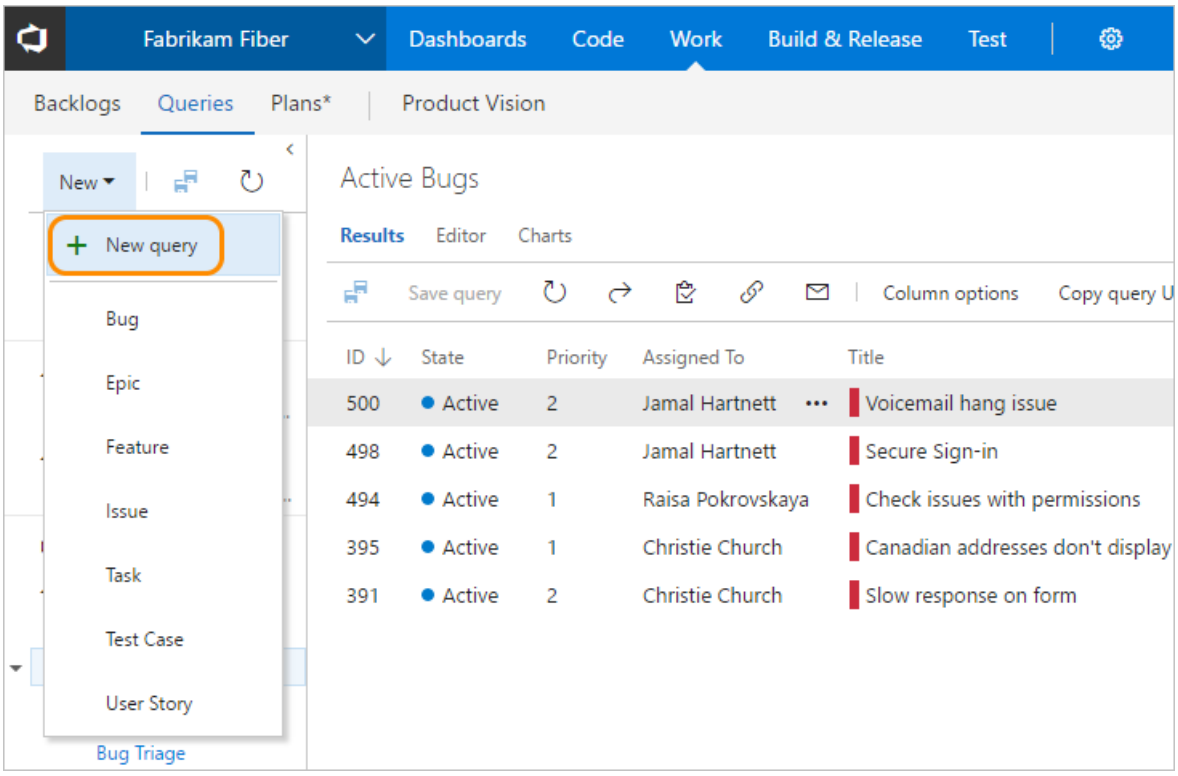
- Browser
- Visual Studio



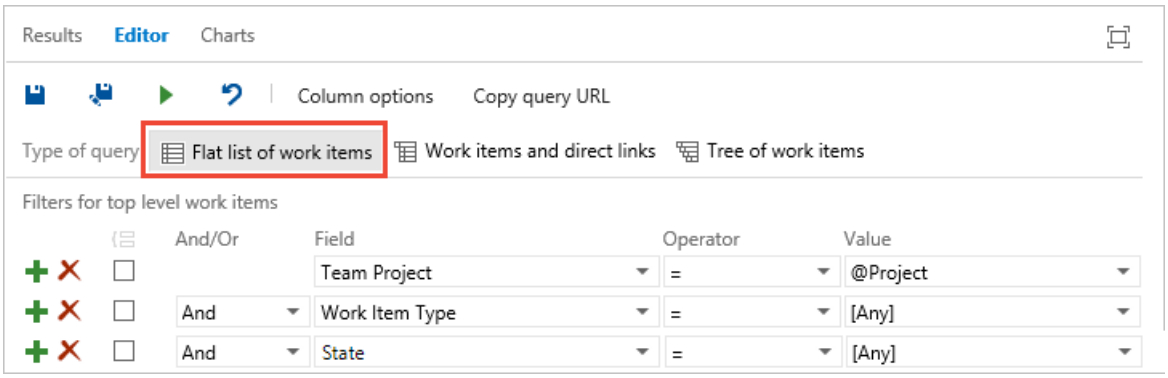
The Query Editor displays with the following default settings: **Flat list of work items**, **Work Item Type=[Any]**, and **State=[Any]**.



You can modify the **Values** and add or remove clauses. Or, change the **Type of query** to Work items and direct links or to a Tree of work items.



The Query Editor displays with the following default settings: **Flat list of work items**, **Team Project=@Project** (the current project), **Work Item Type=[Any]**, and **State=[Any]**.



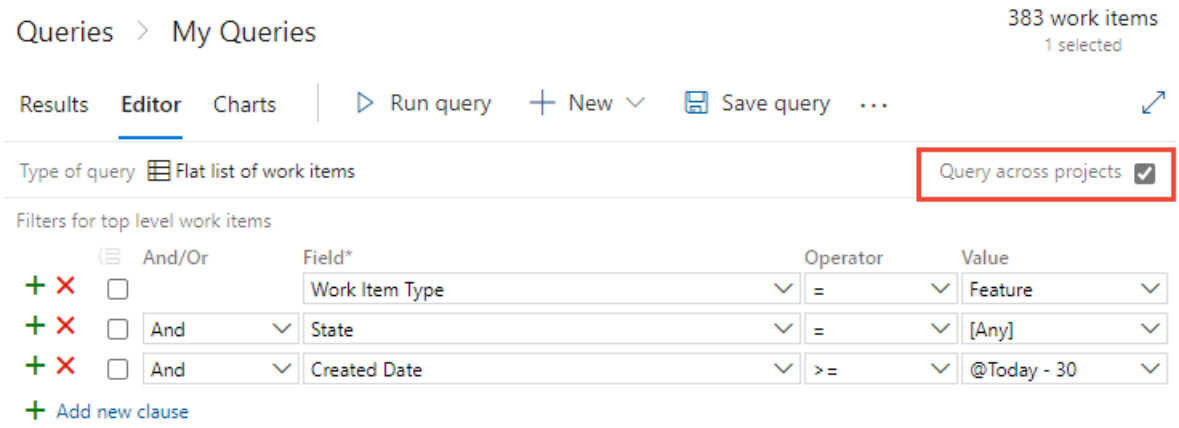
You can modify the **Values** and add or remove clauses. Or, change the **Type of query** to Work items and direct links or to a Tree of work items.

Query across or within projects

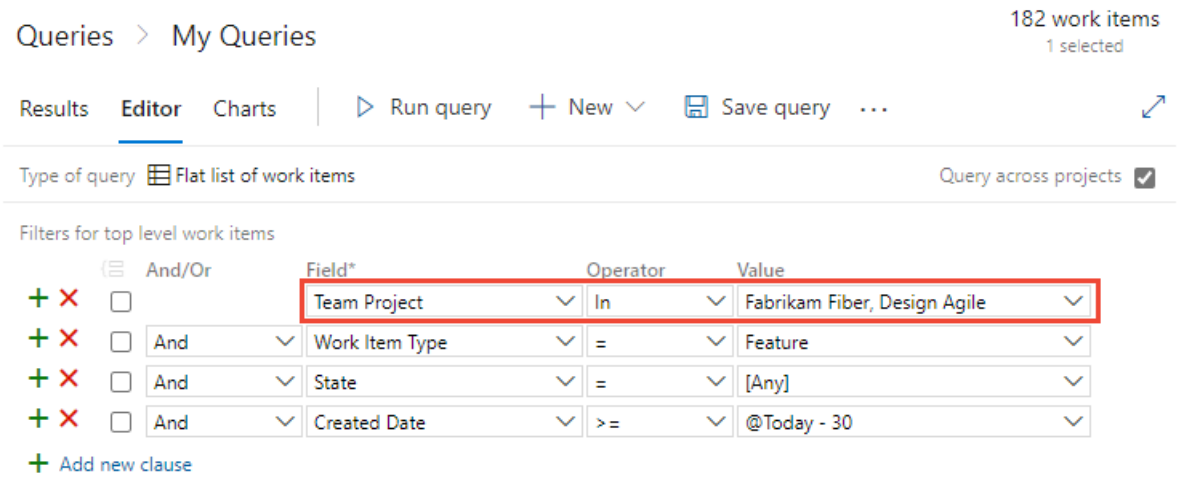
New queries scope to the current project by default. However, you can create queries to find work items defined within the organization or project collection. All queries that you save, however, are saved under a specific project.

- Browser
- Visual Studio

To list work items defined in two or more projects, checkmark **Query across projects**. For example, the following query finds all features created in all projects within the last 30 days.



With the **Query across projects** checked, you can add the **Team Project** field to filter to a select number of projects.



> **NOTE**
> Separate multiple project names with the list separator that corresponds to the regional settings defined for your client computer, for example, a comma (,).

The **Team Project** field is available only after you check **Query across projects**. Further, when **Query across projects** is unchecked, only those fields from those work item types, as defined in the current project, appear in the **Field** drop-down menu. When **Query across projects** is checked, all fields from all work item types defined in all projects in the collection appear in the **Field** drop-down menu.

Define a clause

You create a query by defining one or more clauses. Each clause defines a filter criteria for a single field.

Sample query clause

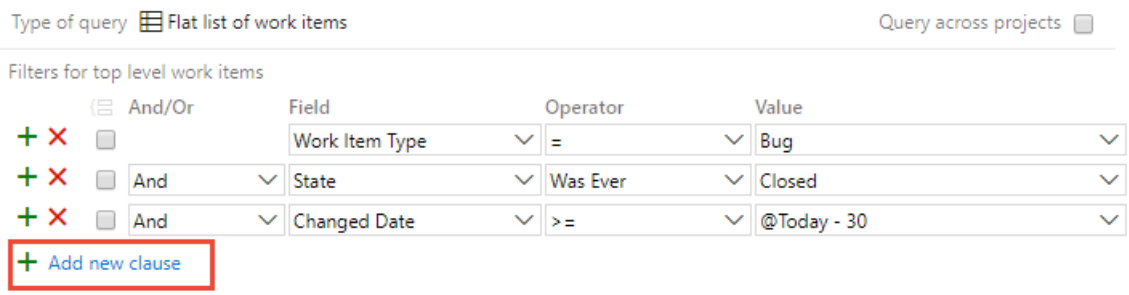
AND/OR	FIELD	OPERATOR	VALUE
And	Assigned To	=	@Me

For a list of available operators based on the field data type, see Query index quick reference.

All clauses you add are added as an **And** statement. Choose **Or** to change the grouping. You group clauses to ensure that the clause statements are run in the sequence required.

- Browser
- Visual Studio

Choose **Add new clause** to add another clause at the end of the query, and then choose the **Field**, **Operator**, and **Value** for that clause.



For example, search for all work items assigned to you by specifying the **Assigned To** field, the equals (=) operator, and the **@Me** macro, which represents your user identity.

> **TIP**
> To view the WIQL syntax for a query, and how parenthesis are used to group clauses, install the Marketplace Wiql Editor. This extension supports viewing the WIQL syntax and exporting it to a WIQL file for use in REST API calls. To learn more, see Syntax for the Work Item Query Language (WIQL).

Checklist for how to define a query clause

1. In the first empty row, under the **Field** column heading, choose the down arrow to display the list of available fields, and choose an item in the list.

   For more information, see Query Fields and Values.

2. In the same row, under the **Operator** column heading, choose the down arrow to display the list of available operators, and choose an item in the list.

   For more information, see Operators.

3. In the same row, under the **Value** column heading, either type a value, or choose the down arrow, and choose an item in the list.

   For more information about how to use a macro or variable to specify the current project, user, date, or other selection, see Variables.

4. To add a clause, choose **Add new clause**.

   You can add a clause to the end of the query, or perform the following tasks with the corresponding icons:

   - **Insert new filter line**
   - **Remove this filter line**
   - **Group selected clauses**
   - **Ungroup clauses**

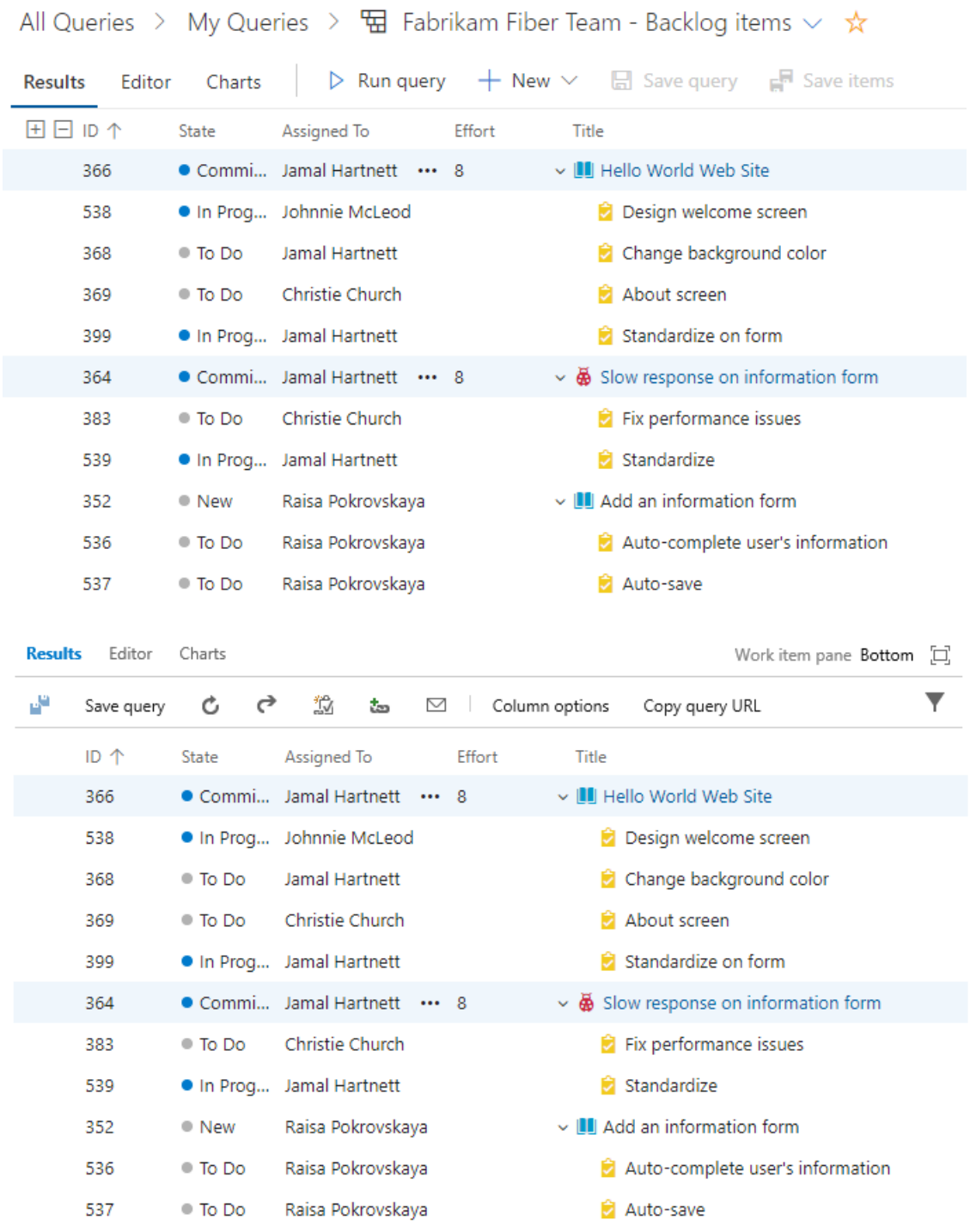
Use a work item tree to view hierarchies

Use the **Tree of Work Items** query to view a multi-tiered, nested list of work items. For example, you can view all backlog items and their linked tasks. To focus on different parts of the tree, choose **Expand all** or

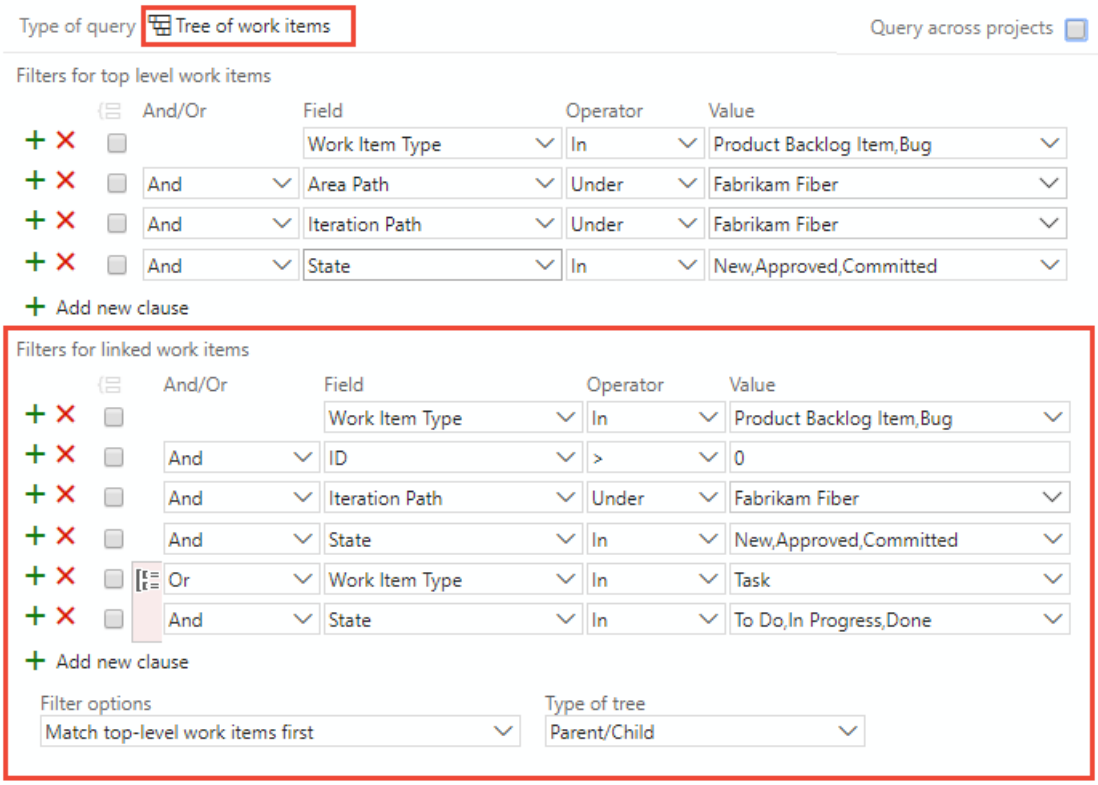
Collapse all.

> **NOTE**
> You can't construct a query that shows a hierarchical view of Test Plans, Test Suites, and Test Cases. These items aren't linked together using parent-child link types. However, you can create a direct links query that lists test-related work items. Also, you can, view the hierarchy through the Test Plans page.

- Browser
- Visual Studio



Define the filter criteria for both parent and child work items. To find linked children, select **Match top-level work items first**. To find linked parents, select **Match linked work items first**.

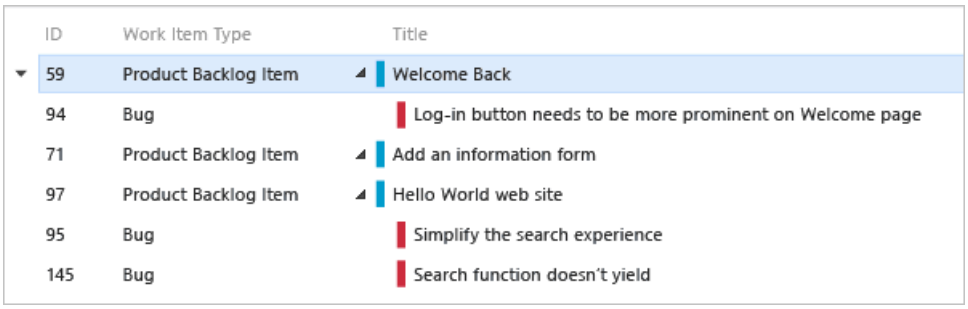


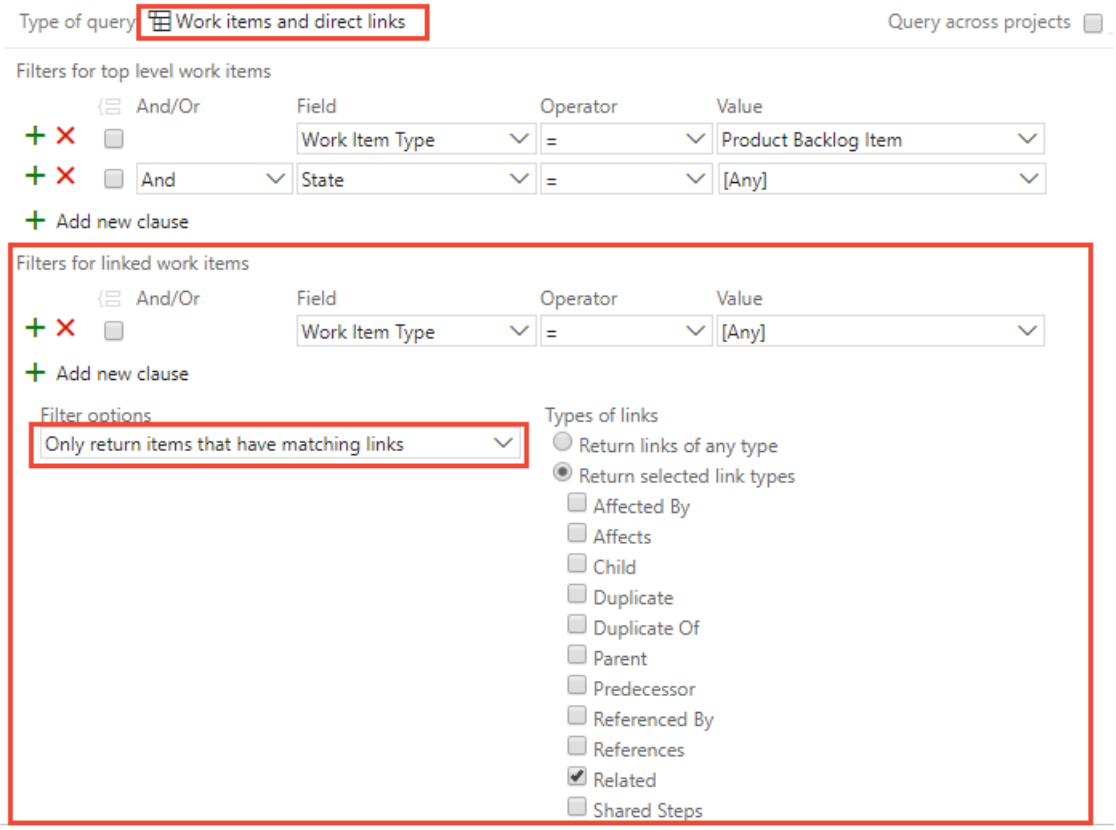
Use direct links to view dependencies

Use the **Work items and Direct links** query to track work items that depend on other tracked work, such as tasks, bugs, issues, or features. For example, you can view backlog items that depend on other items being implemented or a bug being fixed.

Use the direct links query to track dependencies across teams. The query also helps you manage commitments your team makes. Choose the filter criteria for the top and linked work items. And, select the types of links to filter the dependencies.

- Browser
- Visual Studio





Filter your first-tier list of work items by choosing one of these options:

- **Only return items that have matching links**: First-tier work items return, but only if they have links to work items specified by the linked work items filter criteria.
- **Return all top level items**: All first-tier work items return despite the linked work items filter criteria. Second-tier work items that are linked to the first tier return if they match the linked work items filter criteria.
- **Only return items that do not have matching links**: First-tier work items are returned, but only if they don't have links to work items specified by the linked work items filter criteria.

To learn more about each link type, see Linking, traceability, and managing dependencies.

And/Or logical expression

You specify **And** or **Or** to create logical expressions of your query clauses. Specify **And** to find work items that meet the criteria in both the current clause and the previous clause. Specify **Or** to find work items that meet the criterion in either the current clause or the previous clause.

Add one new clause for each work item field to refine your search criteria. Add clauses to return only the set of work items you want. If you don't receive the results you expect from your query, refine it. You can add, remove, group, or ungroup query clauses to improve your query results.

Group query clauses to operate as a single unit separate from the rest of the query. Grouping clauses is similar to putting parentheses around an expression in a mathematical equation or logic statement. When you group clauses, the **AND** or **OR** for the first clause in the group applies to the whole group.

Group clauses

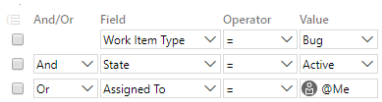
Grouped clauses operate as a single unit separate from the rest of the query. Grouping clauses is similar to putting parentheses around a mathematical equation or logic expression. The **And** or **Or** operator for the first

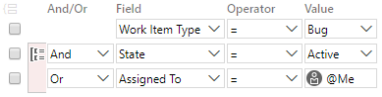
clause in the group applies to the whole group.

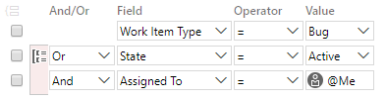
As the following examples show, the grouped clauses are translated to the corresponding logical expression.

> **TIP**
> To view the WIQL syntax for a query, install the WIQL query editor extension which will allow you to see the WIQL version of any Query UI entry. This extension allows you to see just how AND/OR grouped clauses are treated.

QUERY	GROUPED CLAUSES	LOGICAL EXPRESSION

1		`FROM workitems`
		`WHERE`
		`  [System.TeamProject] = @project`
		`  AND (`
		`  [System.WorkItemType] = 'Bug'`
		`  AND [System.State] = 'Active'`
		`  OR [System.AssignedTo] = @me`
		`  )`

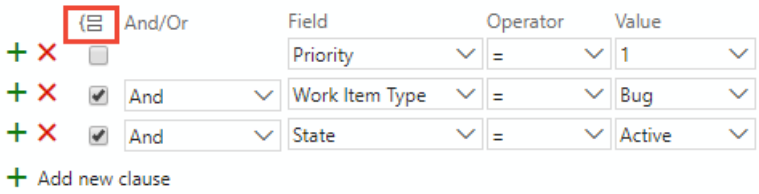
2		`FROM workitems`
		`WHERE`
		`  [System.TeamProject] = @project`
		`  AND [System.WorkItemType] = 'Bug'`
		`  AND (`
		`  [System.State] = 'Active'`
		`  OR [System.AssignedTo] = @me`
		`  )`

3		`FROM workitems`
		`WHERE`
		`  [System.TeamProject] = @project`
		`  AND (`
		`  [System.WorkItemType] = 'Bug'`
		`  OR (`
		`  [System.State] = 'Active'`
		`  AND [System.AssignedTo] = @me`
		`  )`
		`  )`

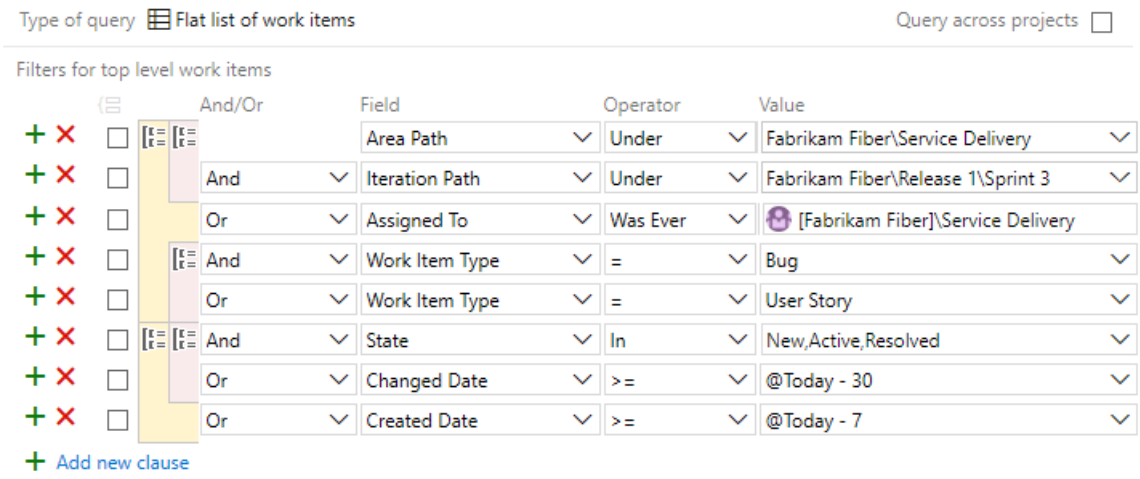
These queries return work items that are type Bug and meet the following logical expressions:

- Query 1: `AND State=Active OR Assigned to @Me`
- Query 2: `AND (State=Active OR Assigned to @Me)`
- Query 3: `OR (State=Active AND Assigned to @Me)`

To group one or more clauses, select them and then choose the group clauses icon.



You can also group several grouped clauses. Check the boxes of each clause that's already been grouped. Then, choose the group clauses icon.



If your query results don't return expected results, follow these steps:

- Make sure that each clause is defined as you intended.
- Verify And/Or assignments to each clause. If your results contain more work items than expected, often an Or clause is present instead of an And clause.
- Determine if you need to group or change the grouping of the query clauses and the And/Or assignments of each grouped clause.
- Add more query clauses to refine your query filter criteria.
- Review the options available to specify fields, operators, and values.

Ungroup a clause

- Browser
- Visual Studio

To ungroup a clause, choose the ungroup clauses icon for the grouped clause.

Related articles

That's the basics about defining queries. For an index of query examples, see Query quick reference.

- Query FAQs
- Chart a flat-list query
- Change column options
- Work item field index
- Keyboard shortcuts

Manage and organize queries in Azure Boards and Azure DevOps

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Organize your personal or shared queries by adding a query folder. You can then add queries to or move existing queries into those folders. You can create queries and query folders from the web portal or from a supported client, such as Visual Studio Team Explorer and Team Explorer Everywhere, a plug-in for Eclipse.

> **NOTE**
> To create and manage queries in Visual Studio 2019, you need to Set the Work Items experience to the legacy option. Also, you can perform bulk drag-and-drop of queries into query folders from Visual Studio but not from the web portal.

Prerequisites

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

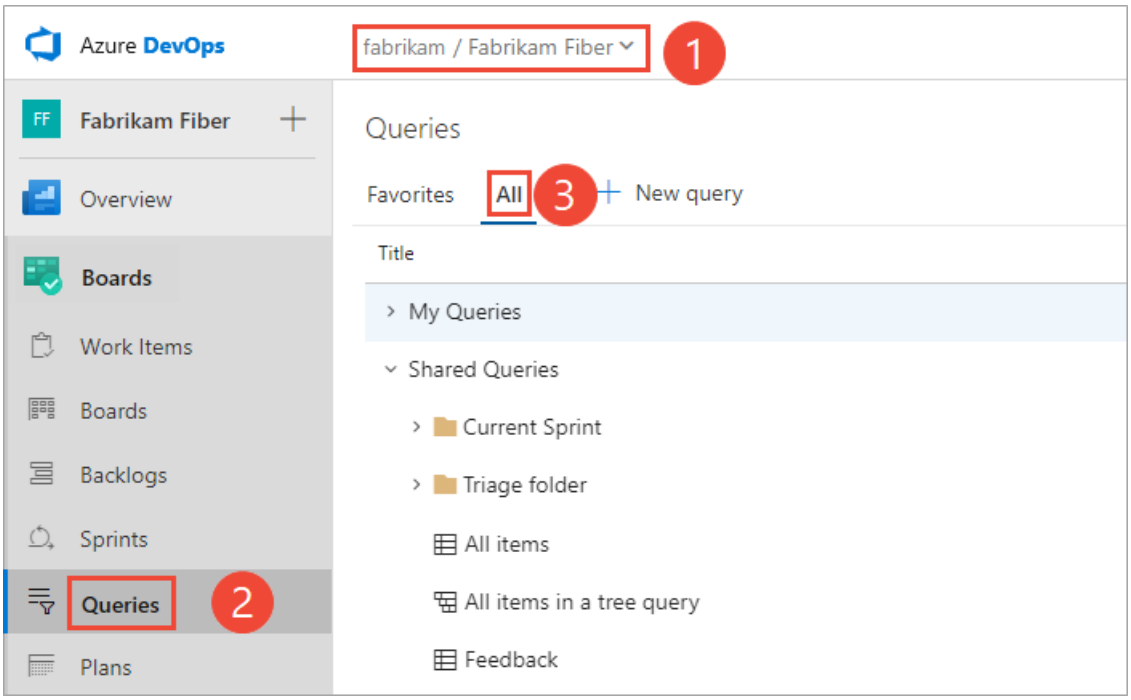
> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

Open a query

- Browser
- Visual Studio

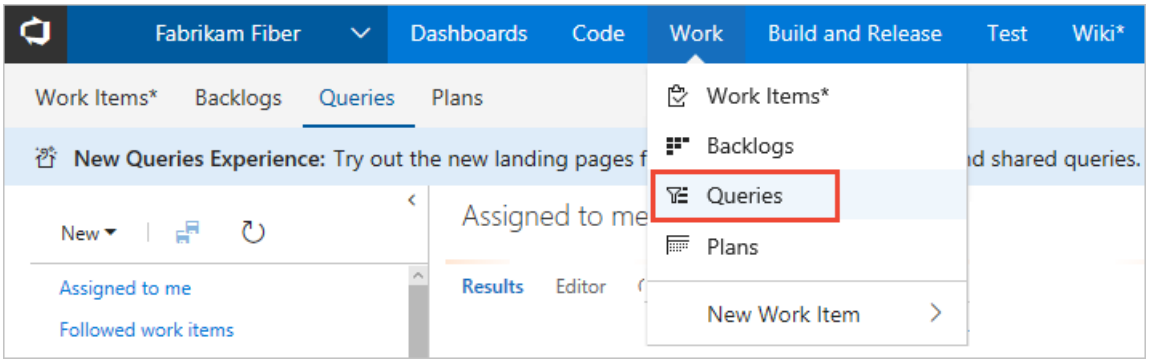
From your web browser, (1) check that you have selected the right project, (2) choose **Boards > Queries**, and then (3) choose **All**.



If it is your first time opening **Queries**, the page opens to **Favorites**. This page lists those queries that you have indicated are a favorite. Otherwise, you can choose **All** to view all queries you've defined and shared queries defined for the project.

> **TIP**
> Queries you or your team have chosen as favorites show up on the **Favorites** page. Favorite queries along with other objects also appear on your **Project** page. To learn more, see Set personal or team favorites.

From your web browser, open **Boards > Queries**.

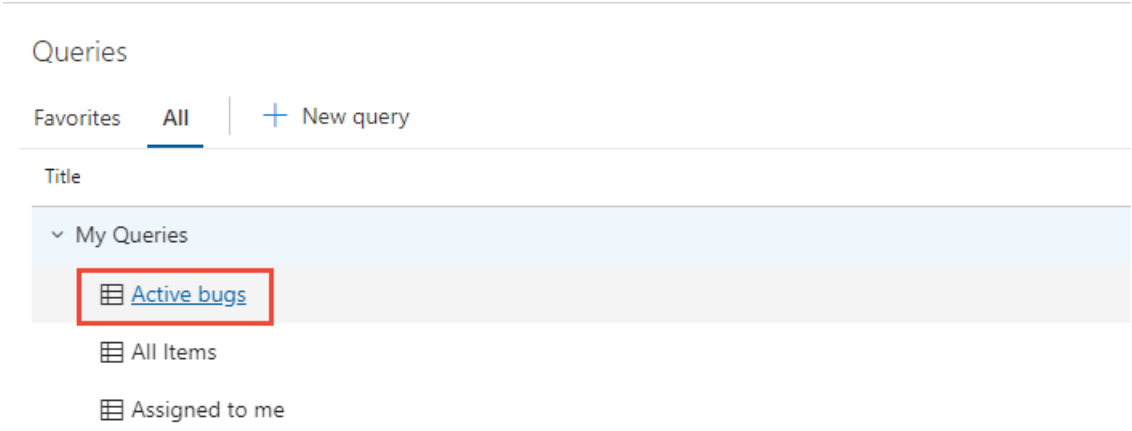


Run, edit, and save a query

The easiest way to define a query is to start with an existing shared query. The following example shows how to find all closed bugs by modifying the *Active Bugs* shared query provided with the Agile process template. Examples are based on the user interface provided through the web portal.

- Browser
- Visual Studio

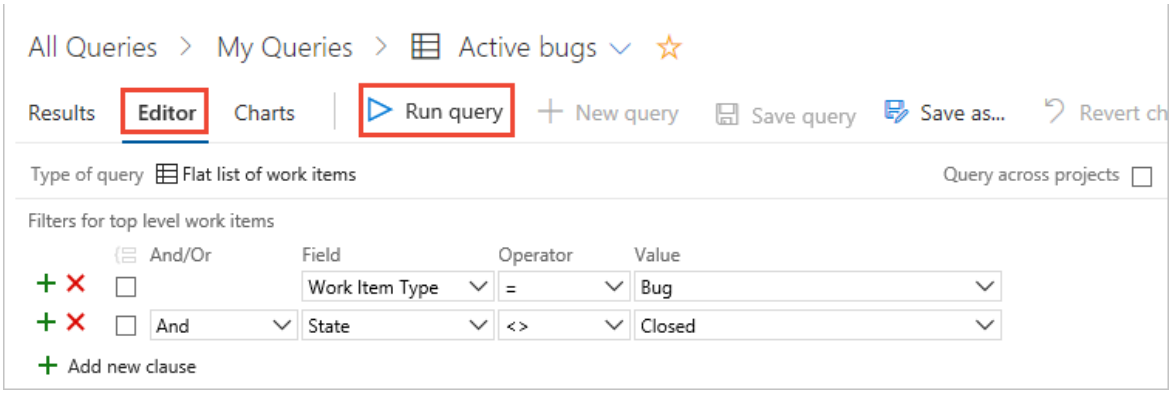
1. Open a shared query. For example, from the web portal, open the *Active Bugs* or similar flat list query.



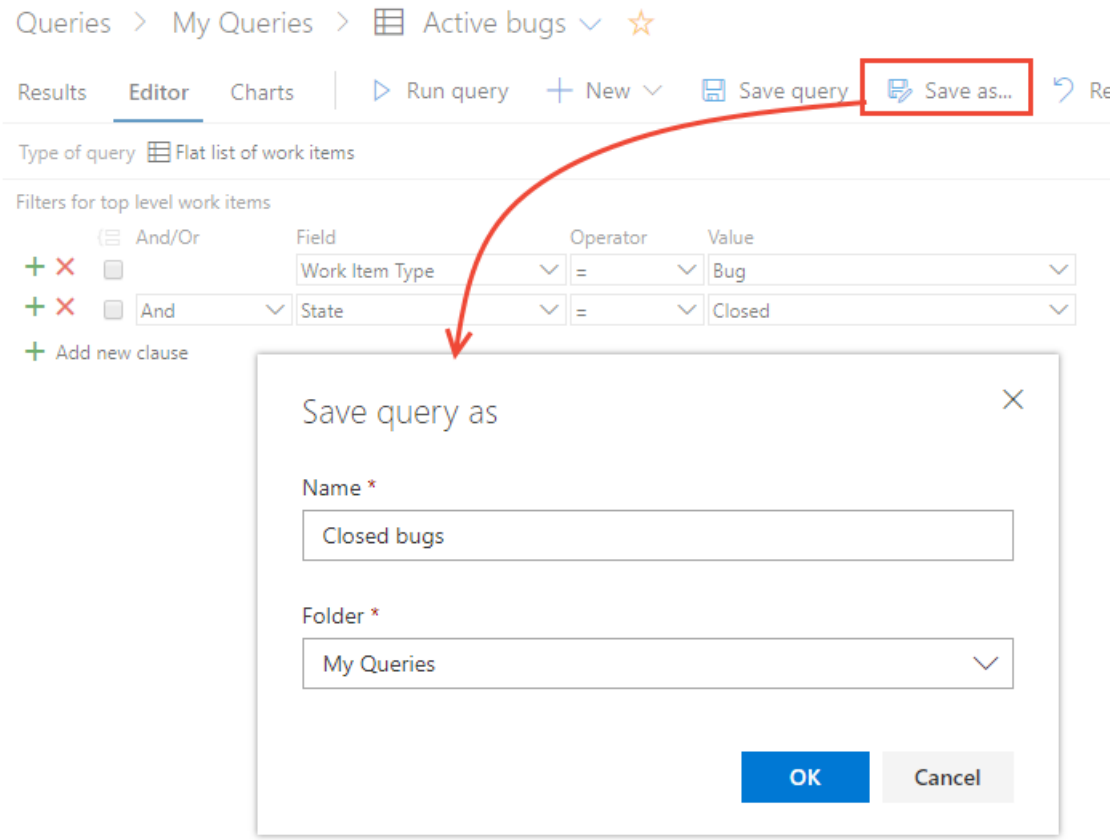
TIP

If you're working in Visual Studio Team Explorer, open the **Work** page to access your queries and shared queries. If Team Explorer isn't visible, choose **View> Team Explorer** from the top level menu.

2. Edit the query to find closed bugs and then run the query. Use + to insert a clause above the current clause. Use ✕ to delete a clause. Queries are automatically scoped to the current project. To find work items defined in several projects, see Query across projects.

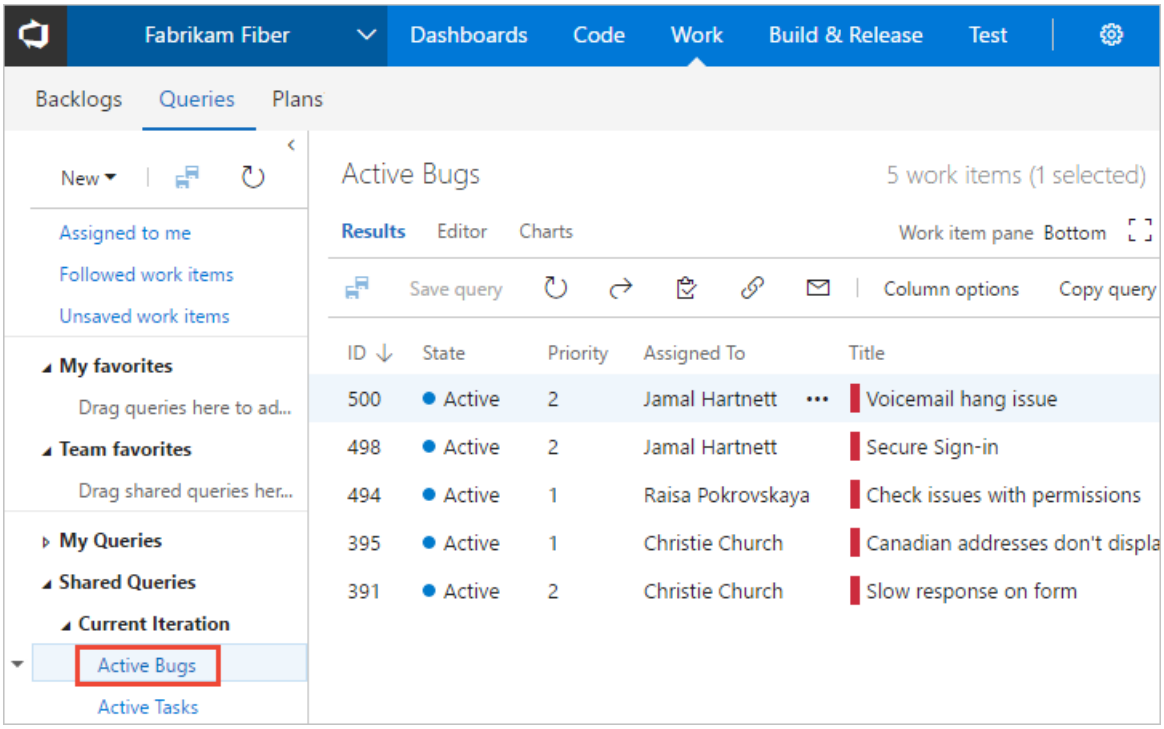


3. Save the query to your **My Queries** folder.



To save a query to the **Shared Queries** folder, you need to be a member of the Project Administrators group, or have your **Contribute** permissions on the folder set to **Allow**. To learn more, see Set query permissions.

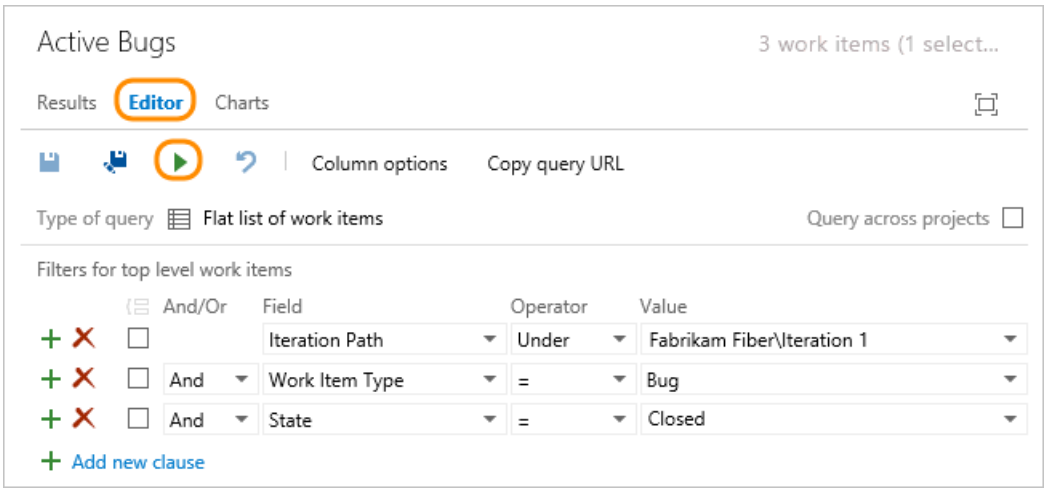
1. Open a shared query. For example, from the web portal, open the *Active Bugs* or similar flat list query.



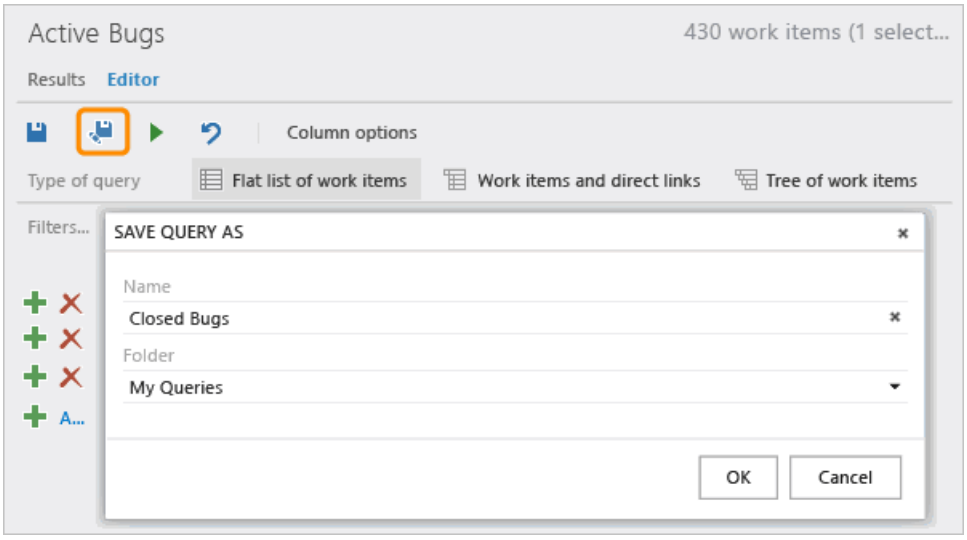
TIP

If you're working in Visual Studio Team Explorer, open the **Work** page to access your queries and shared queries. If Team Explorer isn't visible, choose **View > Team Explorer** from the top level menu.

2. Edit the query to find closed bugs and then run the query. Use + to insert a clause above the current clause. Use ✕ to delete a clause. Queries are automatically scoped to the current project. To find work items defined in several projects, see Query across projects.



3. Save the query to your **My Queries** folder.

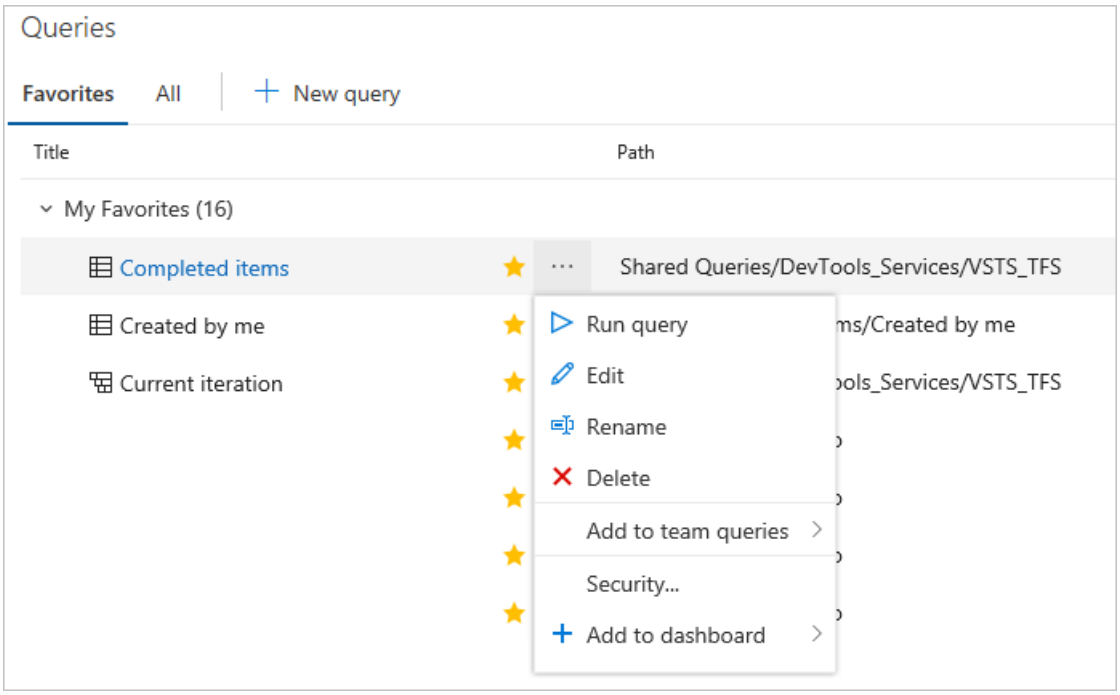


To save a query to the **Shared Queries** folder, you need to be a member of the Project Administrators group, or have your **Contribute** permissions on the folder set to **Allow**. To learn more, see Set query permissions.

Rename or delete a query

- Browser
- Visual Studio

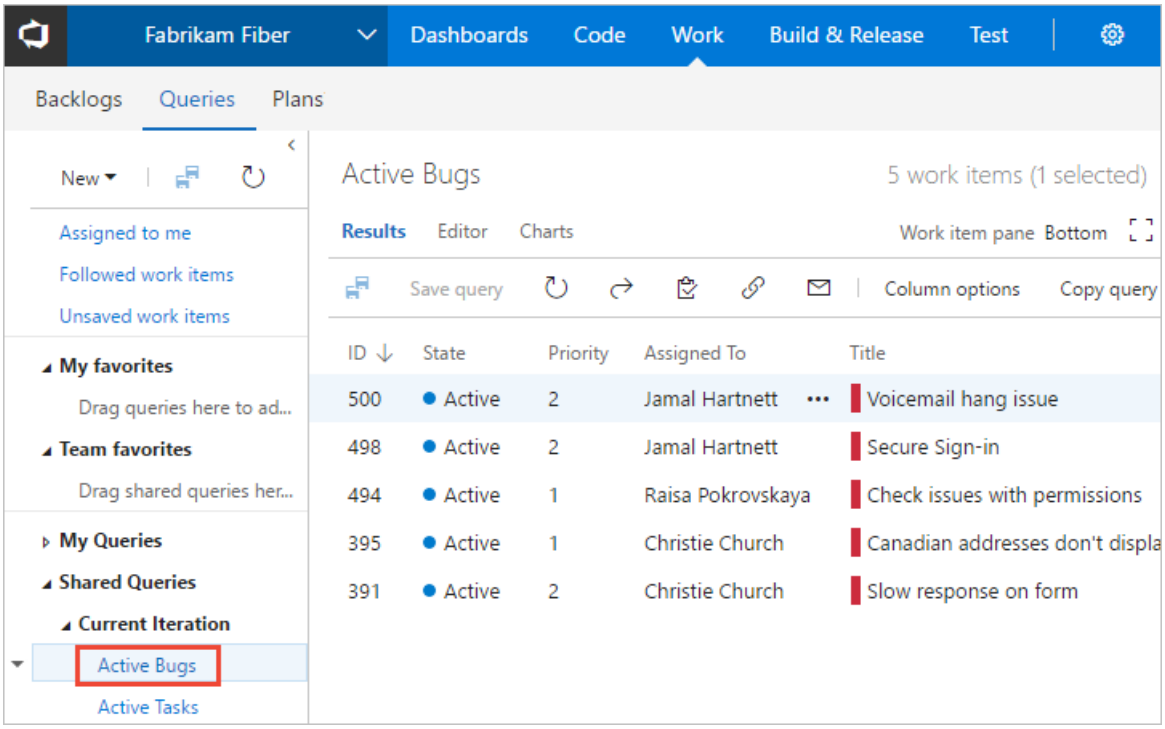
From either the **Favorites** or **All** page, choose the ··· actions icon of a query to run, edit, rename, or delete the query.



For shared queries, you can also choose to do one of these tasks:

- **Add to team queries**: Select the team to add the query as a team favorite
- **Security...**: to set permissions for the query. To learn more, see Set query permissions.
- **Add to dashboard**: Adds a Query tile widget to the team dashboard you select. To learn more, see Add widgets to a dashboard.

Choose the ▾ context menu icon of a query to edit, rename, or delete the query.



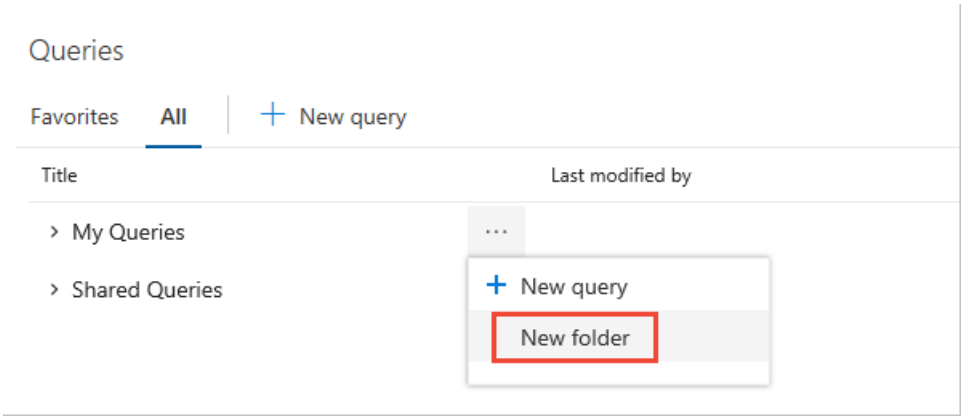
Add a query folder and move items into a folder

> **TIP**
>
> You need **Delete** permissions to rename or move a shared query or folder, and **Contribute** permissions for the folder where you move the query to. To view or set permissions, see Set permissions on queries and query folders.

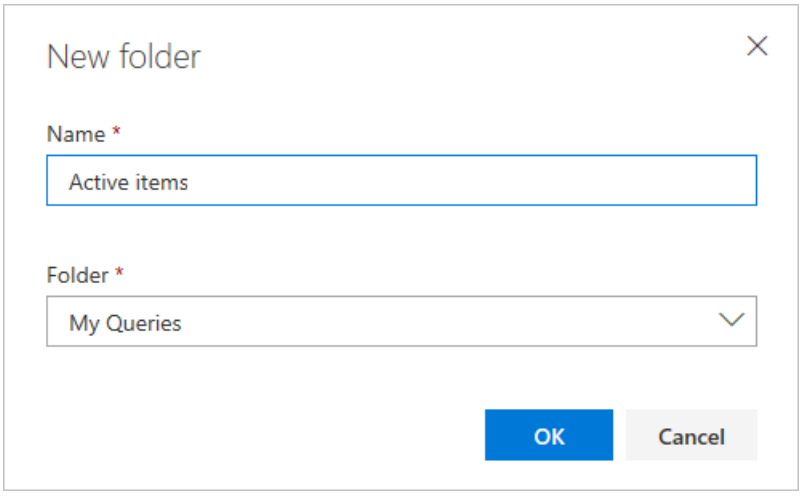
- Browser
- Visual Studio

You add query folders from the **Boards > Queries > All** page.

1. Choose **All**. Expand **My Queries** or Shared Queries depending on where you want to add a query folder.
2. To add a folder, choose the ··· actions icon for an existing folder or the top container folder, and choose **New folder**.

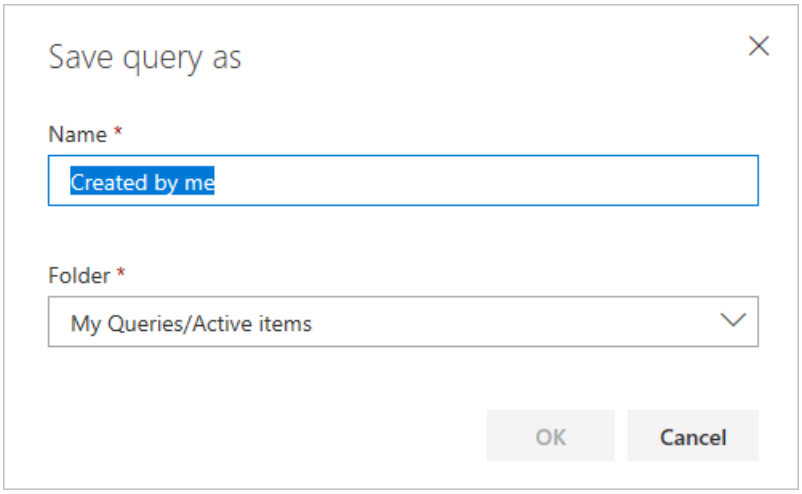


3. Enter the name for the folder in the New folder dialog. If you want to change the location of the folder, select it from the Folder drop down menu.



4. To move items into a folder, drag-and-drop a query onto the folder. From the web portal, you can only drag a single query from outside a folder into a folder.

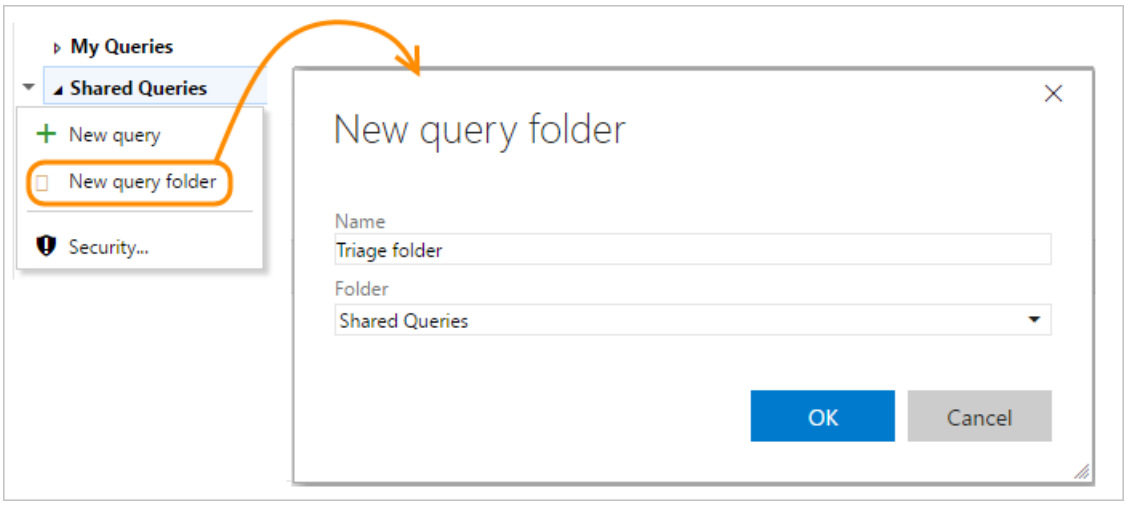
   Optionally, you can choose **More commands** ··· for an existing query, choose **Edit**, and then choose **Save As**. In the Save query as dialog, choose the folder you want to save the query in.



You add query folders from the **Boards > Queries** page.

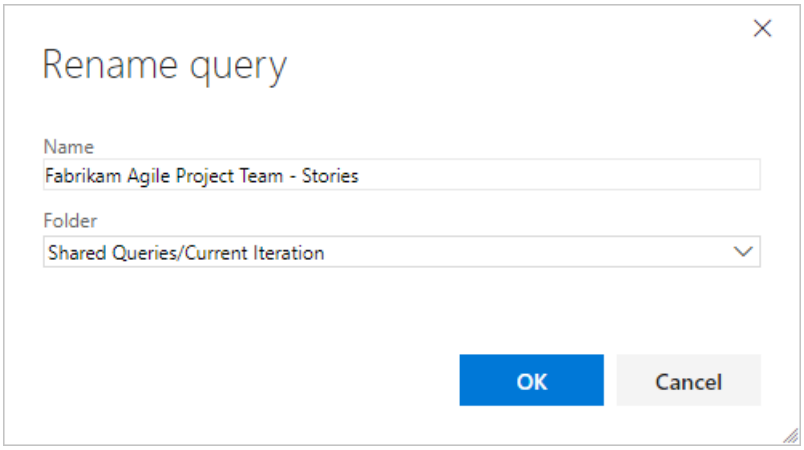
1. To add a folder, choose the ▾ context menu for an existing folder or the top container folder and select **New query folder**.

   Enter the name for the folder in the New query folder dialog.



2. To move items into a folder, drag-and-drop a query onto the folder. From the web portal, you can only drag a single query from outside a folder into a folder.

   Optionally, you can choose the ▾ context icon for an existing query and choose **Rename**. In the Rename query dialog, select the folder you want to save the query in.



Add a query to the dashboard or share it with your team

To add a query to a dashboard, open the ··· actions icon (or ▾ context icon) menu for the query and add it to a specific dashboard or as a team favorite.

Share queries with your team by adding them to a folder under the **Shared Queries** container. To save a query to a **Shared Queries** folder, get added to the **Project Collection Administrators** group or have your permissions set for a folder under Shared Queries.

You can only add shared queries to dashboards or as team favorites, and only if you have team administrator or project administrator permissions.

Related articles

- Query FAQs
- Set query permissions

- Keyboard shortcuts
- Change project-level permissions

Track progress with status and trend query-based charts

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

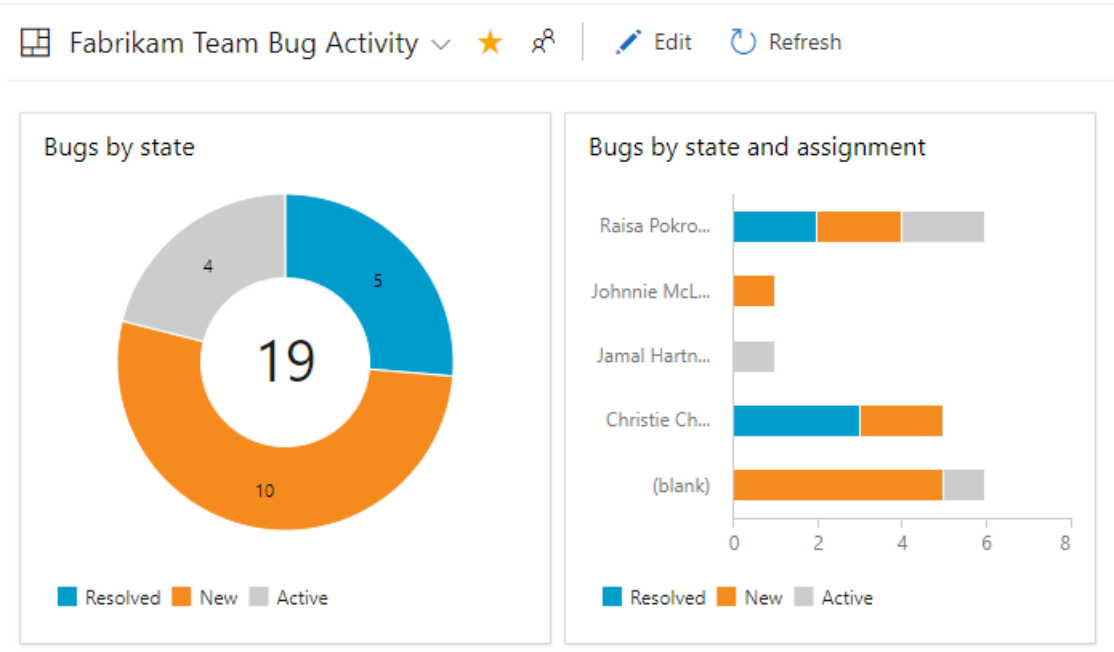
You can quickly view the status of work in progress by charting the results of a flat-list query. Different chart views such as pie, column, pivot, or trend are supported. Charts support viewing a count of work items or a sum of values for select numeric fields, such as Story Points, Effort, or Remaining Work. Group work by State, Assigned To, or other system defined or custom field.

In this article you'll learn how to carry out the following tasks:

- Construct a flat-list query to support your chart
- Create and share your query-based chart
- Create a status pie, column, bar, or pivot chart
- Create a trend chart
- Add a chart to a dashboard

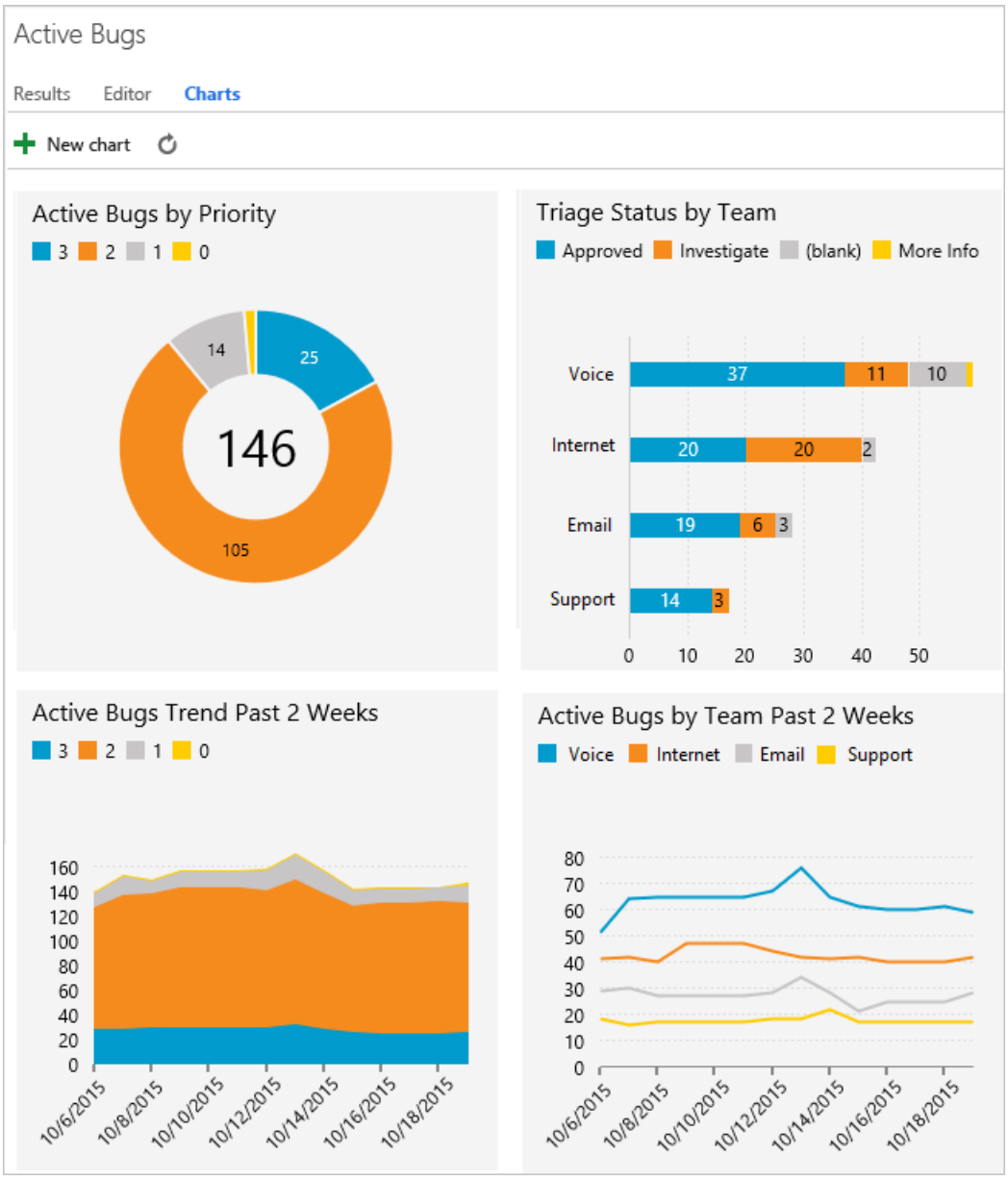
> **NOTE**
>
> This article describes how to configure work tracking query charts. To add existing query charts to dashboards, see Add charts to a dashboard. For information on configuring a **Chart for Work Items** widget, see Configure a chart for work items widget.
>
> For an overview of all work tracking charts and in-context reports, see About dashboards, charts, reports, & widgets.

For example, the following image illustrates two different charts created from the same flat-list query. The pie chart groups the 19 bugs by state, and the bar chart groups the bugs by assignment and their current status.



For example, the following image illustrates four different charts created from the same flat-list query. The pie chart groups the 146 active bugs by priority, and the bar chart groups the bugs by team and their triage status.

The last two chart show two different trend views of the active bugs over the last two weeks.



Prerequisites

Prerequisites to meet include having **Basic** access or higher and to have created a flat-list query. Only flat-list queries support charts.

If you want to add the chart to a dashboard, then you need to save the query under the **Shared Queries** folder, and create the dashboard where you want to add the chart.

- To create a query chart, you must have **Basic** access or higher. Users with **Stakeholder** access can't view or create charts from the **Queries** page, however, they can view charts added to a team dashboard. For details, see Stakeholder access quick reference.
- To add a chart to a dashboard, you must save the query to a **Shared Queries** folder. To do that, you must be granted permissions to save queries under a folder. To get permissions granted, see Set permissions on queries and query folders.
- To add a query chart to a team dashboard, you must be a member of the team or be a member of the **Project Administrators** security group.
- To add a query chart to a project dashboard, you must have created the dashboard or be granted permissions to edit the dashboard, or be a member of the **Project Administrators** security group.
- To view a query chart added to a dashboard, you must have **Read** permissions to the underlying query. If

that permission has been denied, then the widget will display with a *Widget failed to load* message.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query chart features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- To create a query chart, you must have **Basic** access or higher. Users with **Stakeholder** access can't view or create charts from the **Queries** page, however, they can view charts added to a team dashboard. For details, see Stakeholder access quick reference.
- To add a chart to a dashboard, you must save the query to a **Shared Queries** folder. To do that, you must be granted permissions to save queries under a folder. To get permissions granted, see Set permissions on queries and query folders.
- To add a query chart to a team dashboard, you must be a member of the team or be a member of the **Project Administrators** security group.
- To view a query chart added to a dashboard, you must have **Read** permissions to the underlying query. If that permission has been denied, then the widget will display with a *Widget failed to load* message.

To learn more about dashboard permissions, see Set dashboard permissions.

Create a flat-list query

When creating a query to support your chart, follow these guidelines.

- Always select the **Flat list of work items** query type. Other query types aren't supported for charting. For more information, see Define a query, Define a flat-list query.
- Add those fields to either a query clause or the column options that you want to use within your chart. You can group charts by any field except date-time, free-form text, and tag fields. For example:
  - To group by Status, include the **State** field
  - To group by work assignments, include the **Assigned To** field
  - To group by sprints or iterations, include the **Iteration Path**
  - To group by team, include the **Node Name** field that displays the leaf node of the Area Path
  - To group by a custom field, include it.
- To sum a numeric column, include the corresponding field in your query clause or column options. For more examples of charts created from numeric fields, see Query by a numeric field.
- If you plan to add your query to a dashboard, save your query as a **Shared query**.
- You can't group charts by the following field data types:
  - ID
  - Date-time, such as Created Date, Changed Date
  - Plain text, such as Title
  - Rich-text, such as Description, Repro Steps
  - Tags (You can filter a query using tags, however you can't use tags to configure your chart).

> **NOTE**
> You can't group a query-based chart by tags, however, you can group a **Chart for Work Items** widget by tags that you add to a dashboard as described in Configure a chart for work items widget.

- You can't group charts by the following field data types:
  - ID
  - Date-time, such as Created Date, Changed Date

- Plain text, such as Title
- Rich-text, such as Description, Repro Steps
- Tags (You can filter a query using tags, however you can't use tags to configure your chart).

Display of areas and iterations

When you select **Area Path** or **Iteration Path**, only the leaf node appears in the chart. The leaf node is the last node of the full path. For example, `Phone` is the leaf node of `FabrikamFiber/Fabrikam Website/Phone`. If your query contains a mixed level of leaf nodes, your chart might not reflect expected results.

Select the **Node Name** field, the area path leaf node, to see if that improves your results.

Charts display in browsers that support Scalable Vector Graphics (SVG). Supported browsers include Microsoft Edge, Internet Explorer 9 and later versions, Chrome, Firefox and Safari on Mac. Charts aren't optimized for mobile or touch displays.

> **NOTE**
> Internet Explorer is no longer supported for Azure DevOps Services, nor for Azure DevOps Server 2020.1.

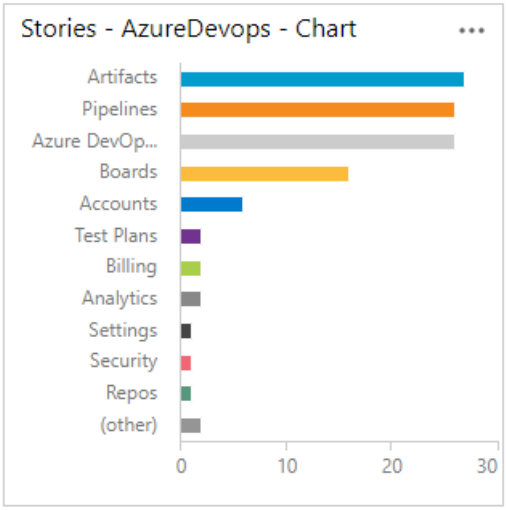
Sort by Value or Label

Most charts allow you to choose how you want to sort the data. You can sort by **Value** or **Label** and select **Ascending** or **Descending**.

- **Value**: Sorts data by the numeric value
- **Label**: Sorts by the label selected for grouping the data.

Limited display of series

When a chart contains more than eight or 12 items within the data series, values in the 9 or 13-plus items are consolidated into a set labeled "other"? However, if you increase the chart size through the configurable widget on a dashboard you may increase the series limit.



When a chart contains more than seven items within the data series, values in the eight-plus items are consolidated into a set labeled "other"?

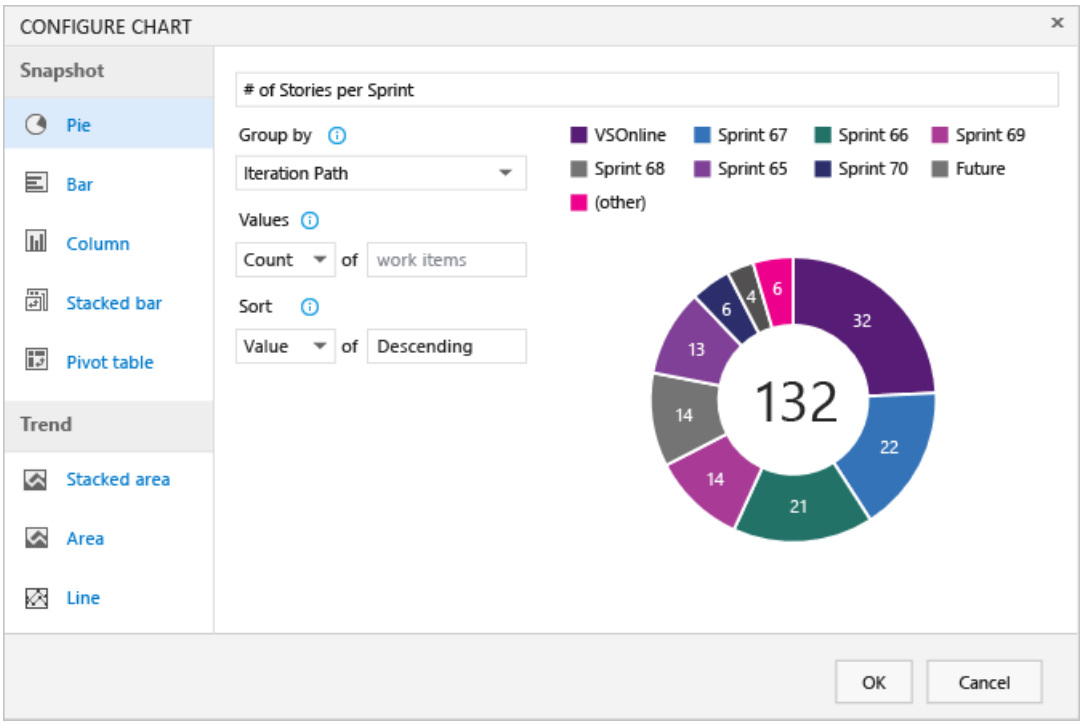
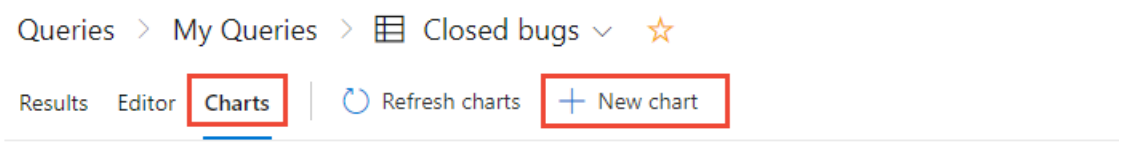


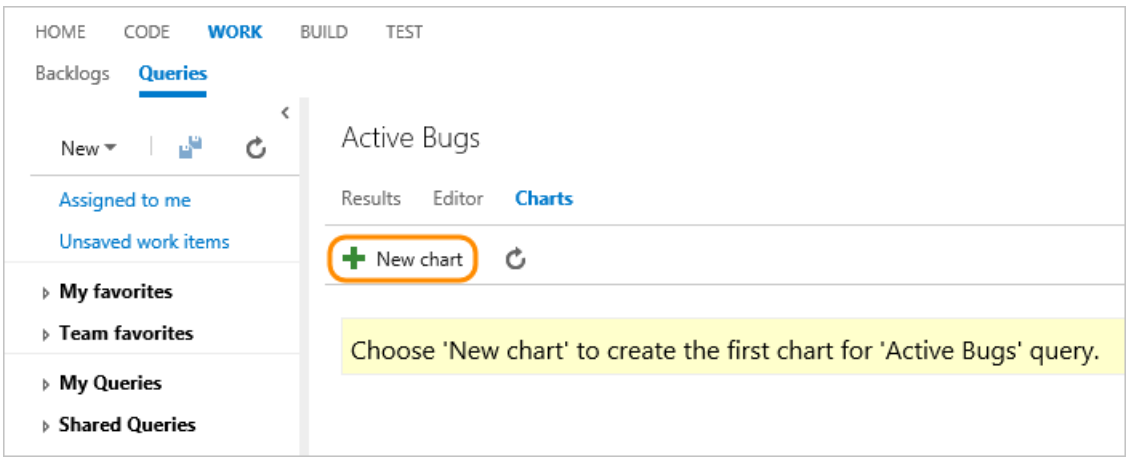
Chart availability

- Charts saved under **Shared Queries** are viewable by all team members, except members with **Stakeholder** access, and can be added to dashboards.
- Charts that you create for queries under your **My Queries** folder are visible only to you.
- You can copy and email the URL of any chart page to share it with a project member.
- To create similar charts for tests, see Track test status.

Create a query-based chart

1. From **Queries**, open the chart editor for a flat list query. You must belong to the Contributors group to create charts.

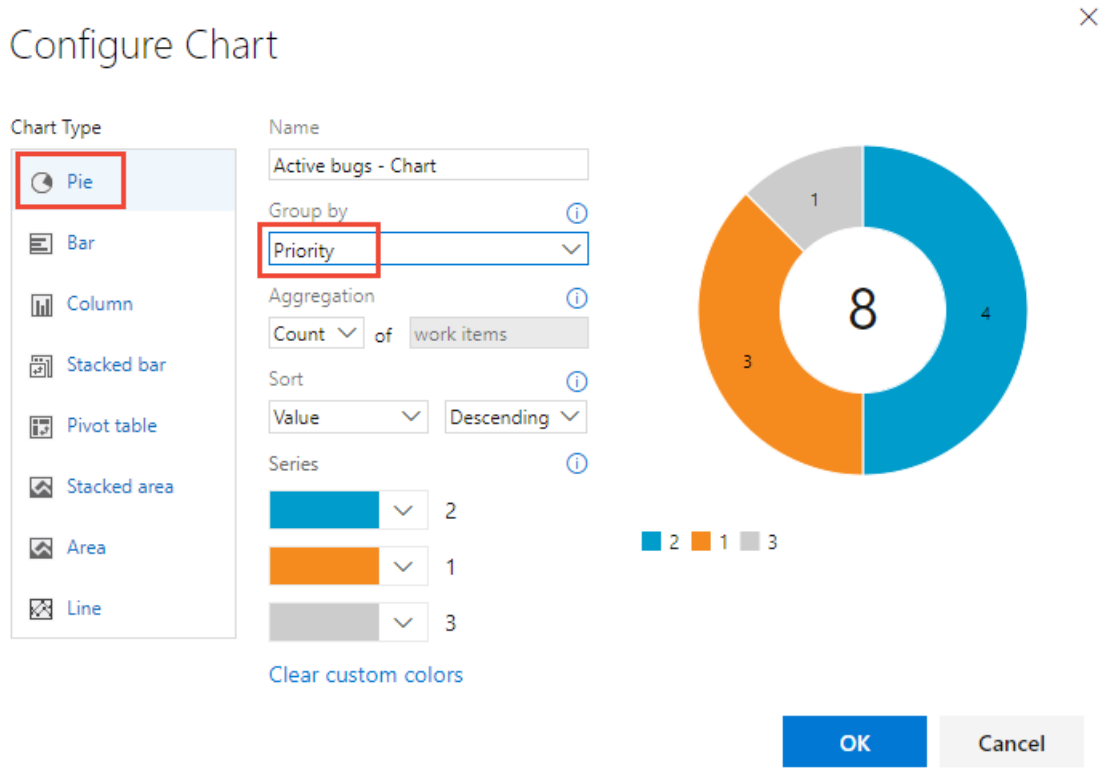


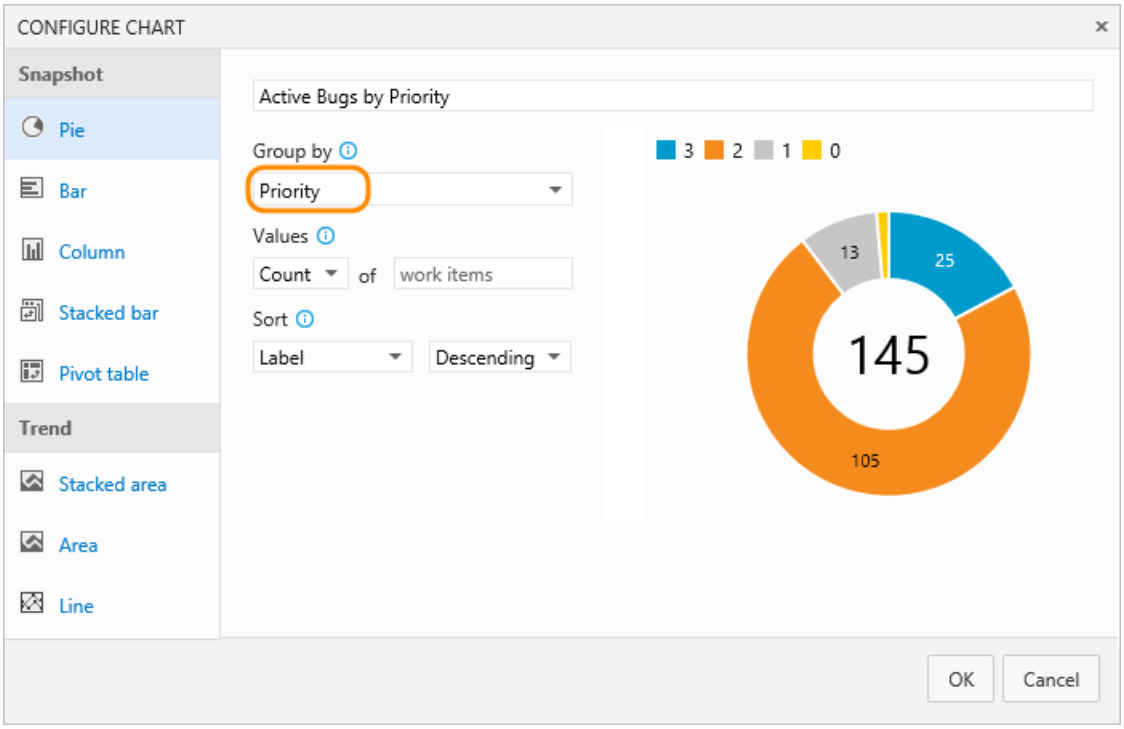


If you have **Stakeholder** access, the **Charts** and **New Chart** links won't appear.

2. Select the chart type and field for grouping values. When you use pie, bar, and column charts, select a

single field to view a count of work items.





If you don't see the field you want in the **Group by** drop-down list, add the field as a column to the query and save the query. Also, the **Aggregation** options depend on the fields used in the query or those selected from the **Column Options**.

If you receive an error message when you close the chart editor, you need to request Basic access.

3. To sort the results, select **Value** or **Label** as the sort option and then **Ascending** or **Descending**.

   To change a color, select a color from the Series set of color pickers.



To change a color, select a color on the chart and pick a new color from the color picker.

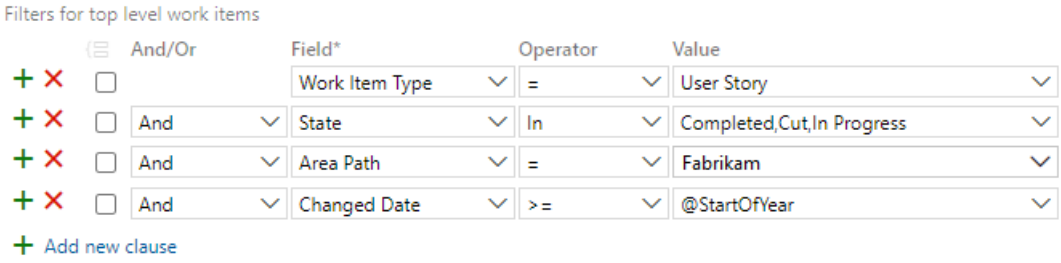
Charts automatically update when you edit the query or refresh the query results.

Add a Pie chart

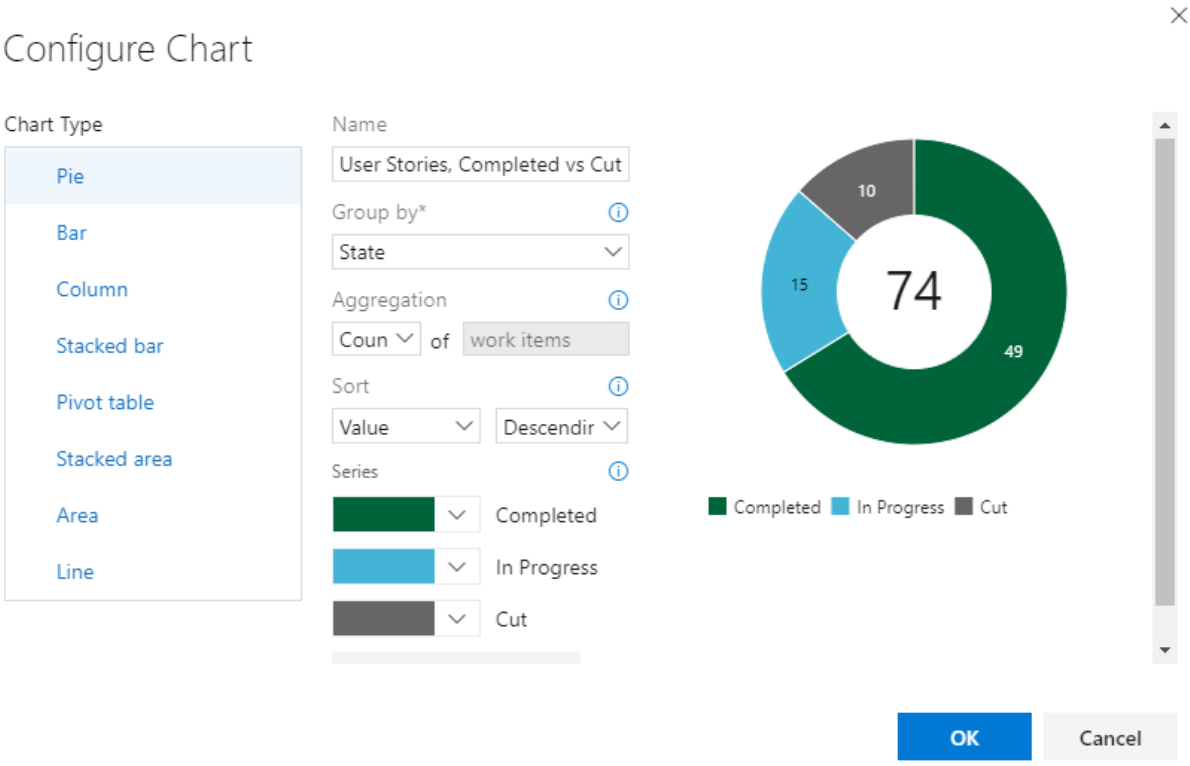
Use a pie chart to show group percentages with six or fewer categories. Good examples of pie charts are:

- Active Bugs Status, group by State
- User Story Status, group by State
- User Story Progress, group by Completed, In Progress, or Cut

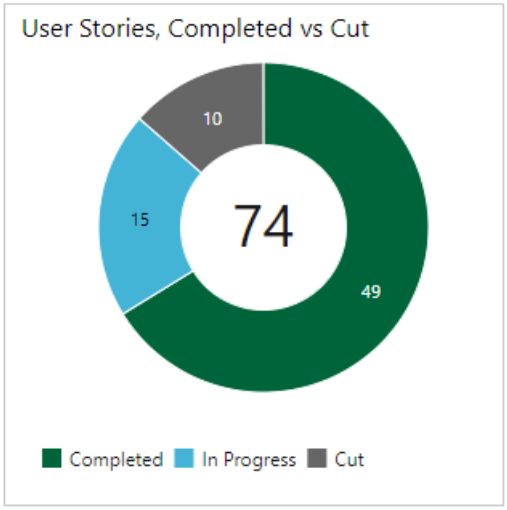
For example, the following query filters User Stories based on the State for Cut, In Progress, and Completed since the start of the year.



The pie chart configuration is as shown in the following image.

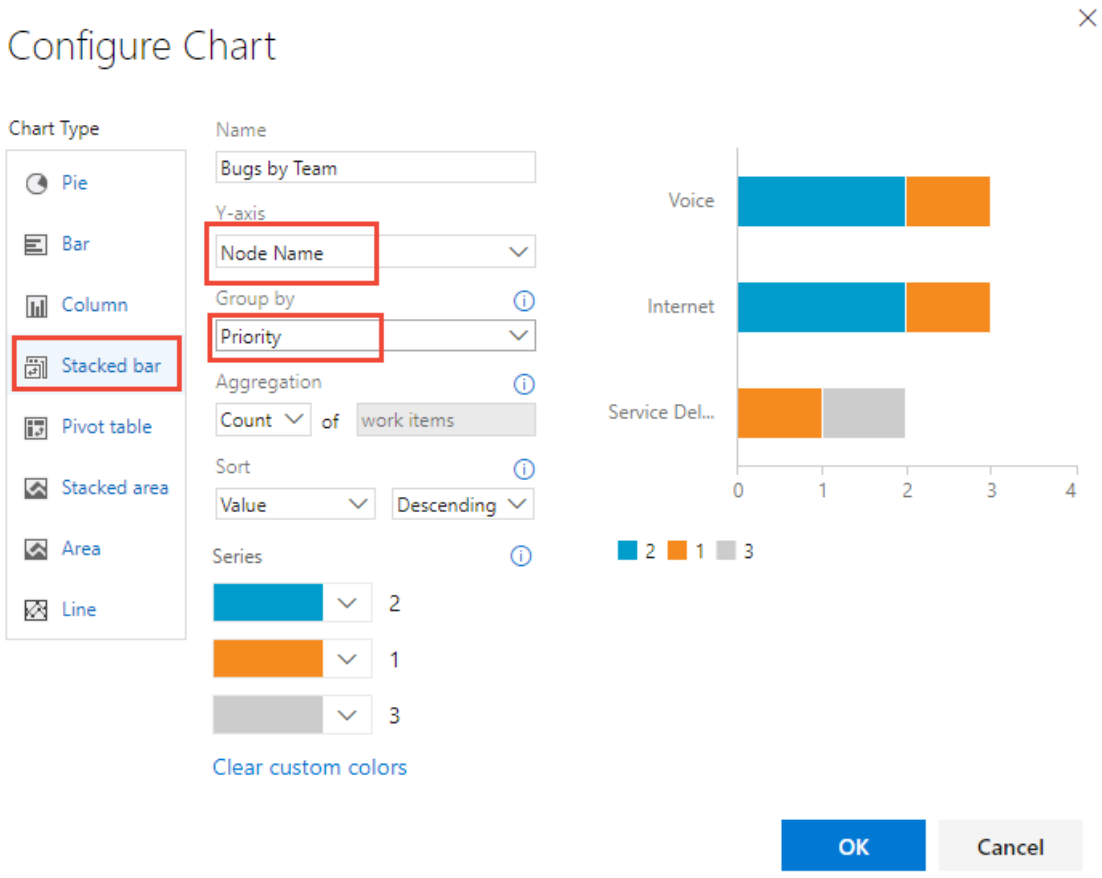


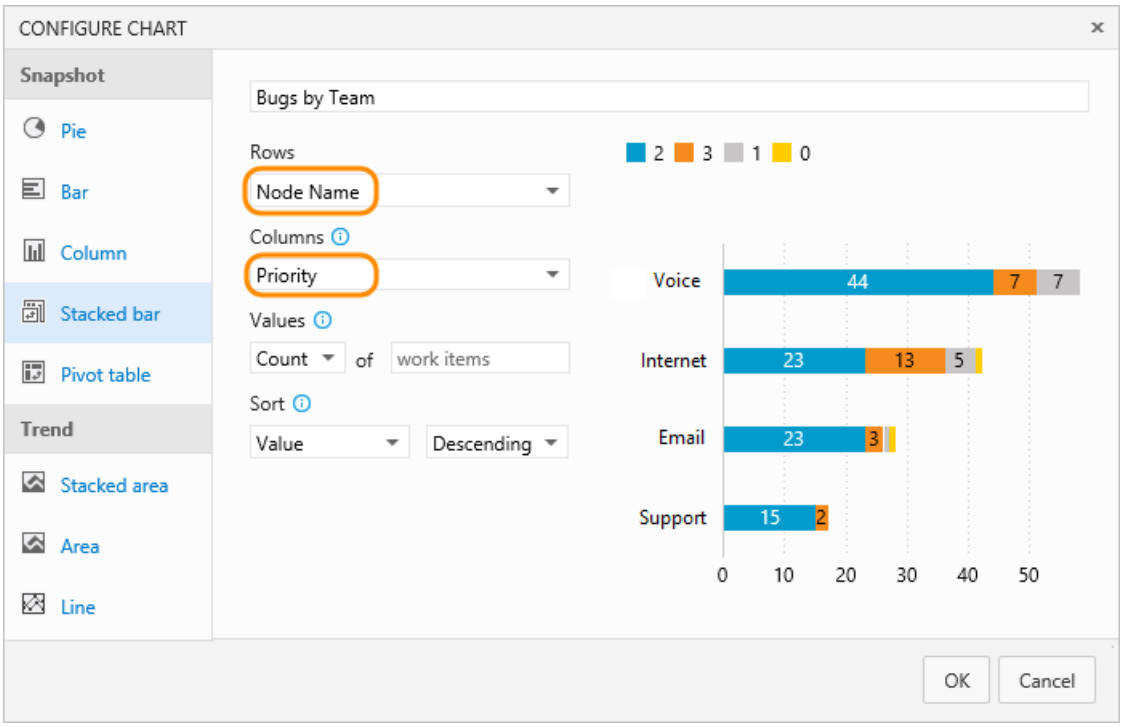
The combined query and chart configuration yield the following pie chart.



Add a Stacked bar chart

A stacked bar chart lets you track progress against two field values. Node Name will display the last leaf within an area path. Use this when you want to show data across teams, and each node corresponds to a team.

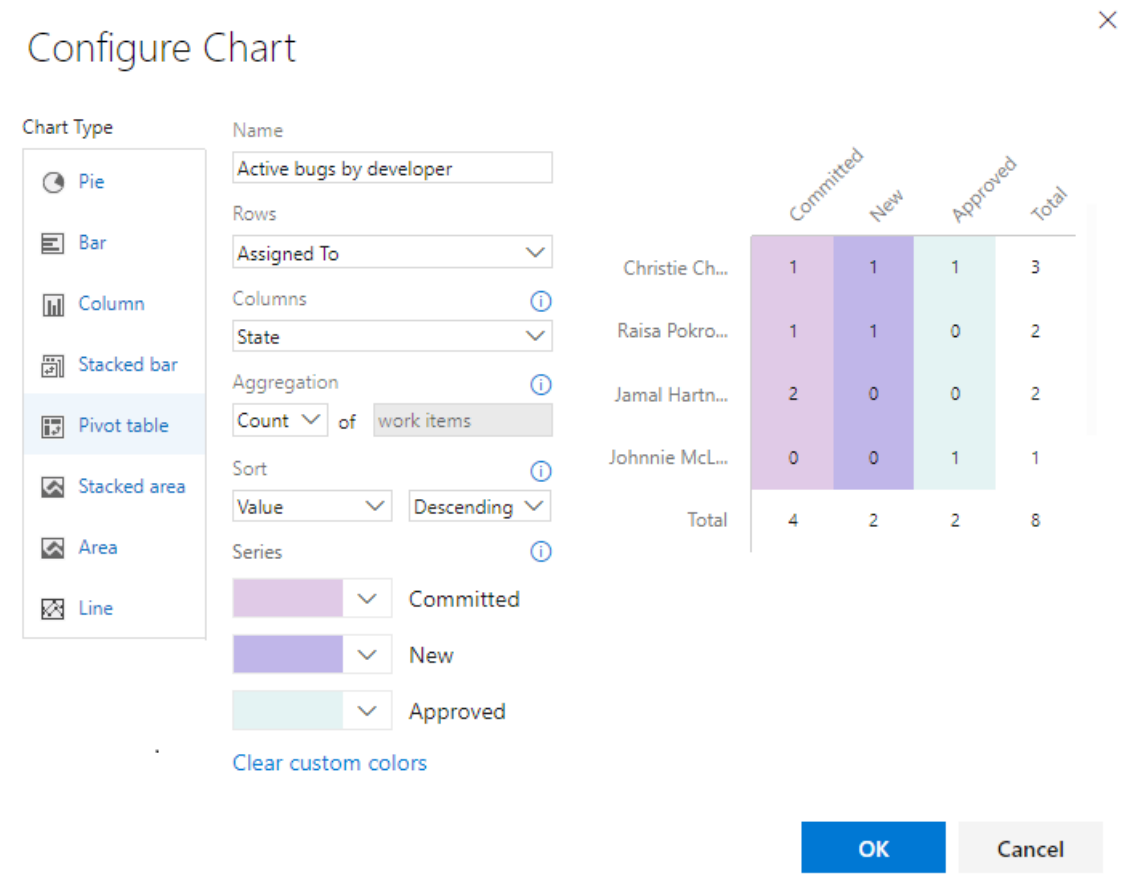




Add a Pivot table

The Pivot table displays a table of configurable rows and columns, with columns showing a count of work items or sum of a numeric field. Choose a Pivot table when you want to compare across areas the work being performed.

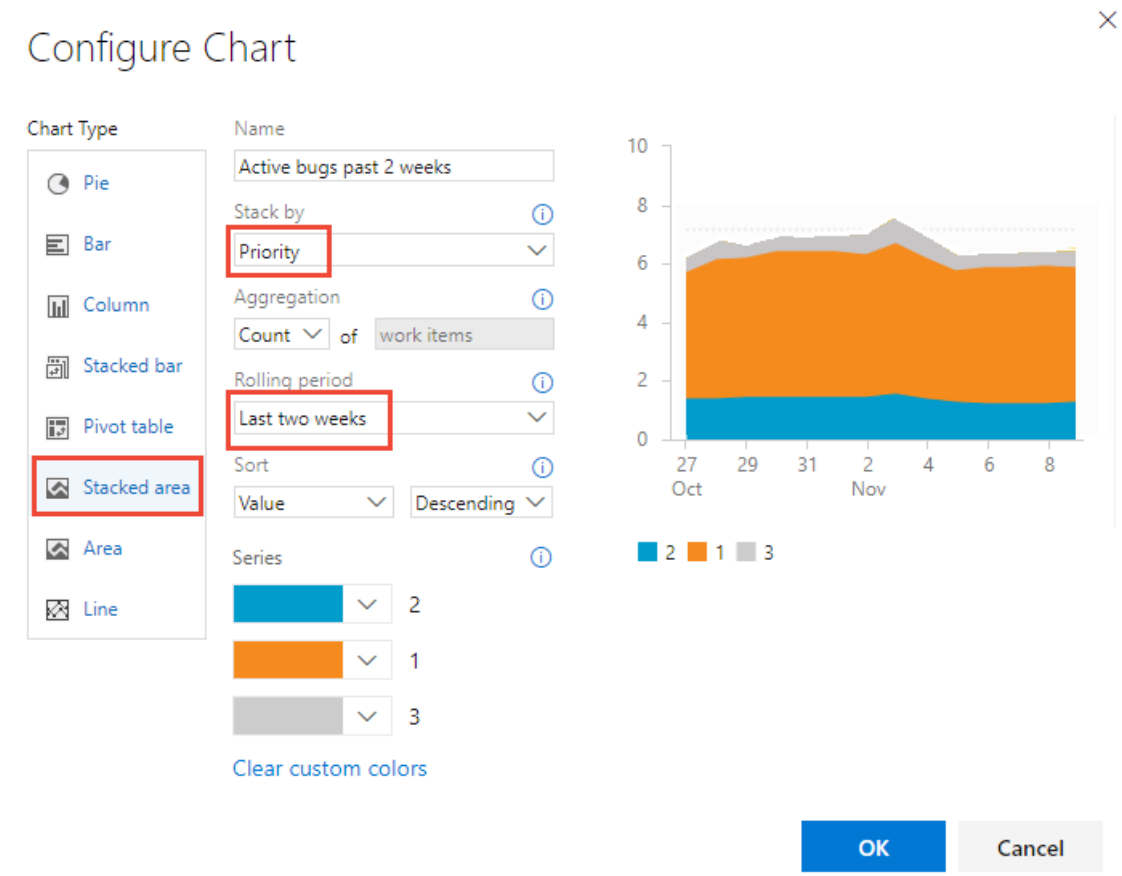
The following image shows an example of active bugs assigned to developers and their current state.

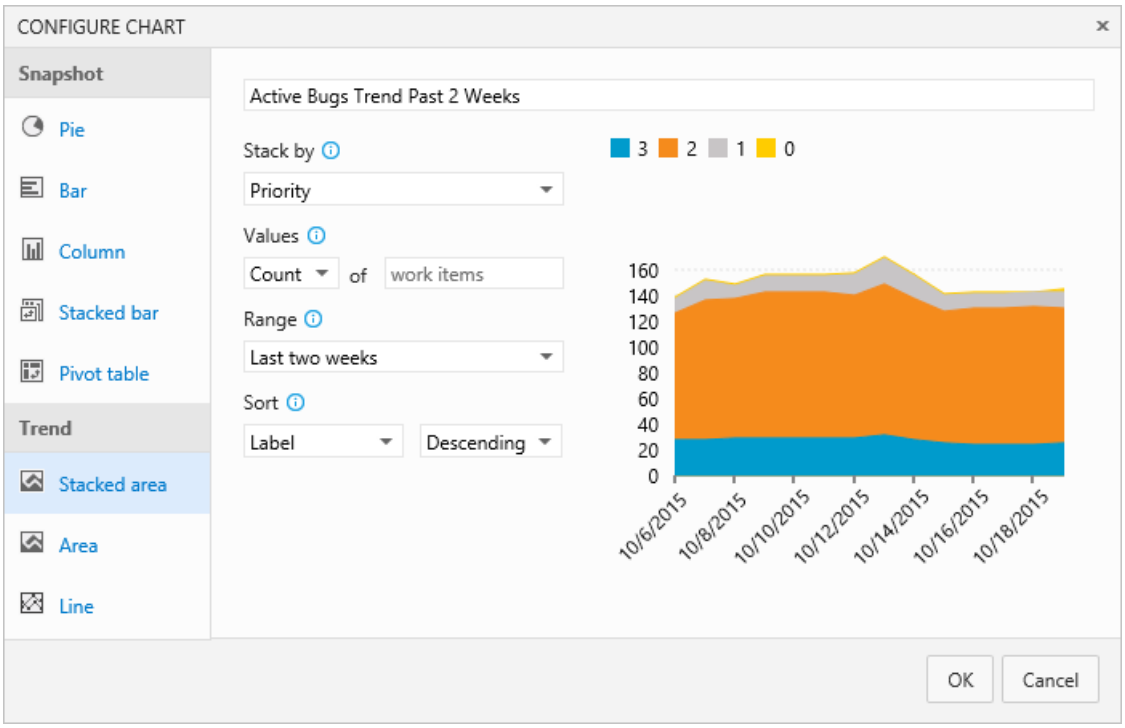


Add a Trend chart

Trend charts let you view progress over time. You can select a rolling period ranging from the last week to the

last year.



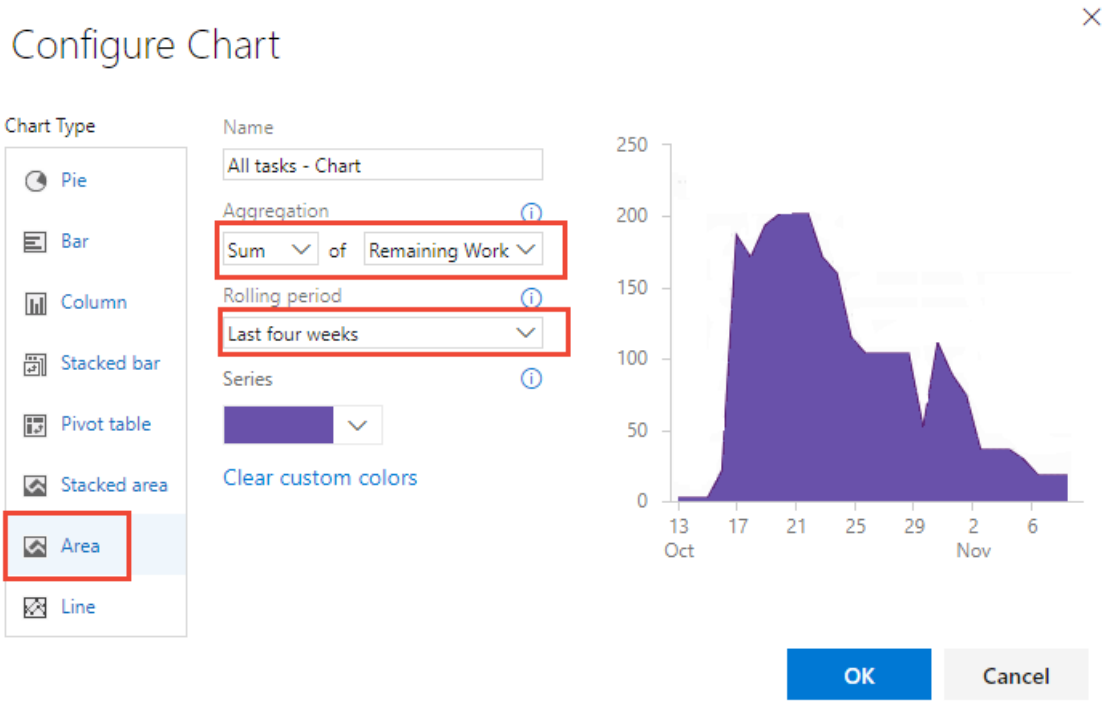


Trend data is extracted from the work tracking data store. Like most data stores, the schema of the relational database is designed and optimized for the online transactional processing of data. As the tool or plug-in performs an activity, it writes the latest information to the operational store. Therefore, data in the operational store is constantly changing and being updated, and all data is current.

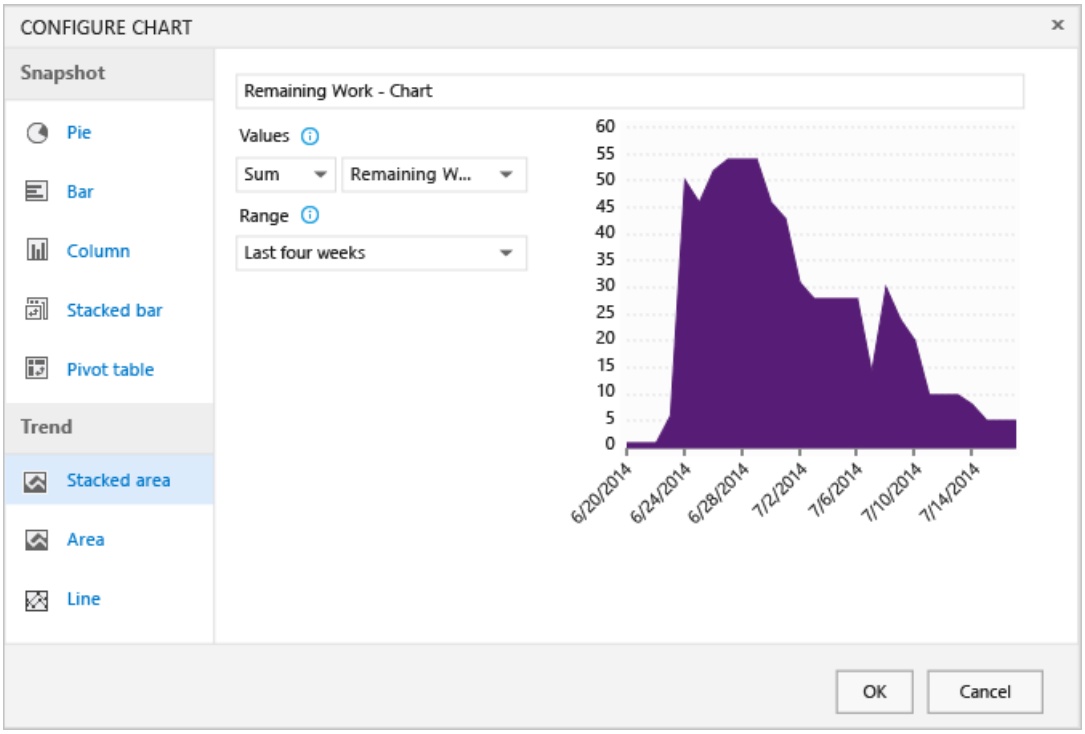
Add a Burndown chart

Burndown charts are useful for determining how quickly work is progressing based on a numeric field value, such as Story Points, Effort, or Remaining Work, or on a count of work items.

To create a burndown chart, make sure to add the numeric field you want to your query. To view a burndown chart of tasks, select the **Sum** operator for **Remaining Work**.



In addition to query-based burndown charts, you can Configure a Burndown or Burnup widget.



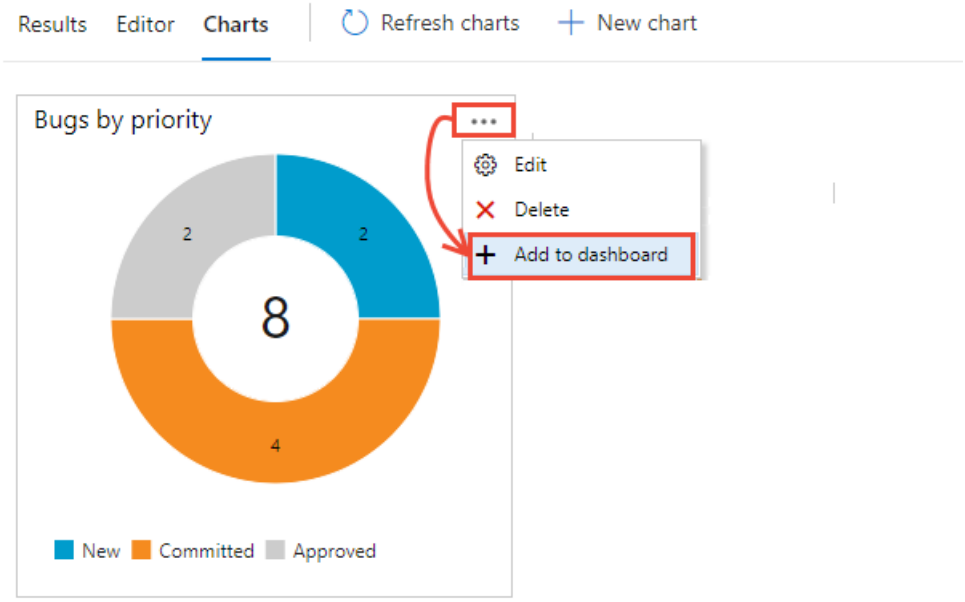
Add chart to a dashboard

A chart added to a dashboard is added through the addition of a **Chart for Work Items** widget. You can add the chart to a dashboard as shown in the following procedure, or by adding the **Chart for Work Items** widget directly. To learn more, see Configure a chart for work items widget.

TIP

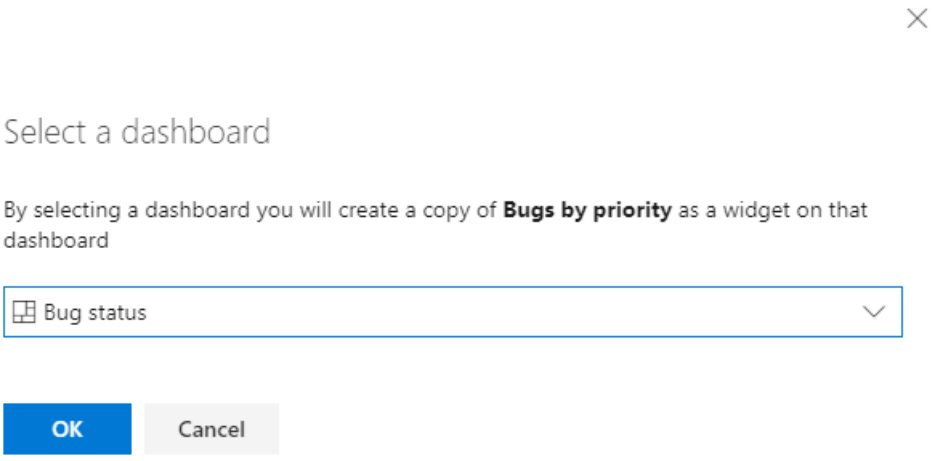
All query charts are limited in size. However, charts added to a dashboard can be re-sized and re-configured by opening the **Chart for Work Items** widget used to display them.

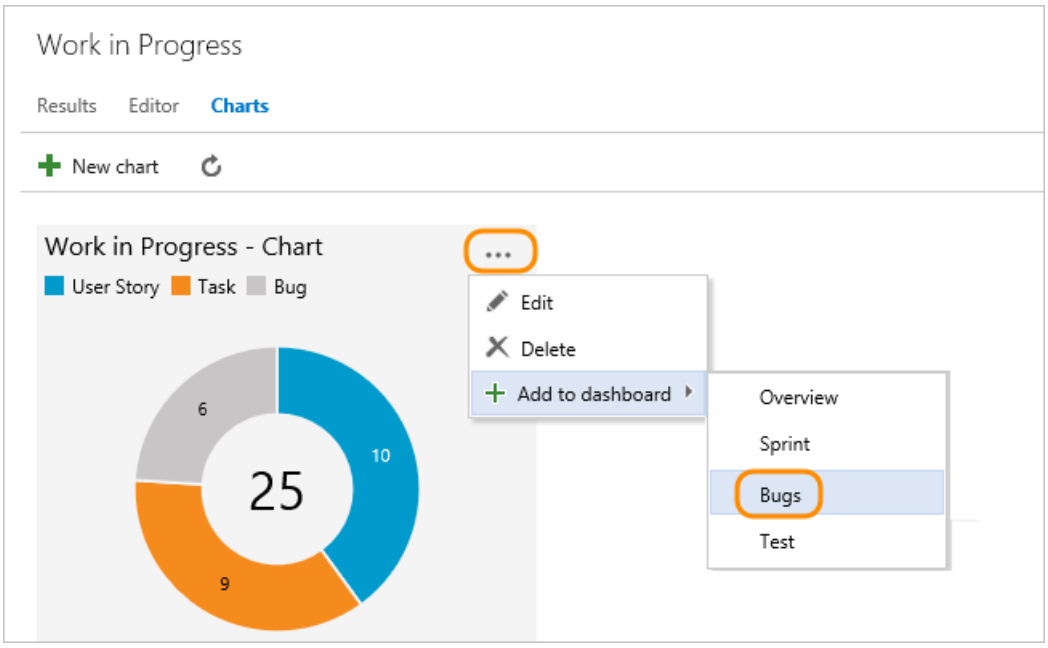
- Select the ··· actions icon for the chart you want to add, and select **Add to dashboard**.



The **Add to dashboard** menu option is only available for queries that have been saved to a **Shared Queries** folder.

In the dialog that opens, select the dashboard to add the chart to.





To add other types of charts, such as test results and build summary charts, see Add widgets and chart to a dashboard.

Query-based charts versus Excel-generated PivotCharts

Query-based charts generate data from the work item tracking data store and therefore displays the most recent data. Excel PivotCharts access data published to the Analysis Services cube, which is refreshed every two hours by default. Excel charts require your project's project collection is configured with SQL Server Reporting Services and Analysis Services.

Related articles

- Configure a chart for work items widget
- FAQs on Azure DevOps dashboards, charts, and reports
- Cumulative flow diagram
- Team velocity
- View/configure sprint burndown
- Track test status
- Add widgets and chart to a dashboard
- Widget catalog

Triage work items with a work item query in Azure Boards and Azure DevOps

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2019 | Visual Studio 2015 | Visual Studio 2013

Using a work item query you can quickly review and update work items. Teams often use the triage mode for a query to complete the following tasks:

- Set the priority of a bug or work item
- Assign a work item to a sprint or team member
- Add details to the description, acceptance criteria, or repo steps
- Link-related work items
- Update the status of work items

In this article you'll learn how to:

- Use triage query mode to update a list of work items
- Bulk save work items that you've updated

Prerequisites

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query features just like users with **Basic** access. For details, see Stakeholder access quick reference.

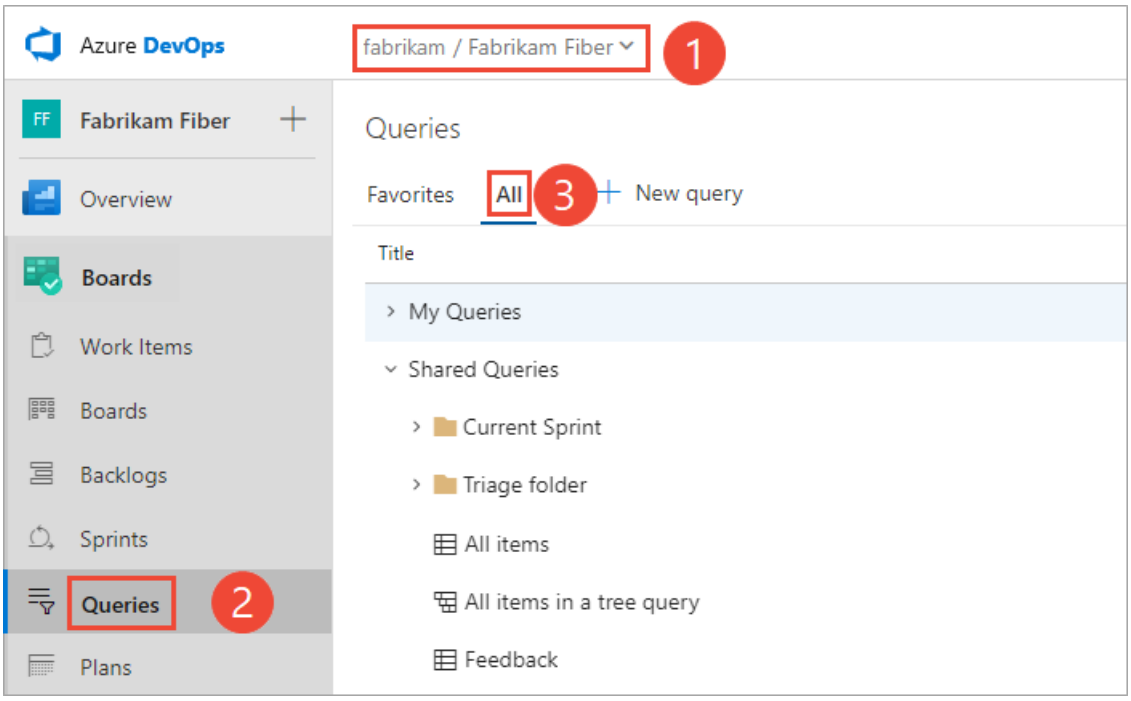
- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

Open Queries

- Browser
- Visual Studio

From your web browser, (1) check that you have selected the right project, (2) choose **Boards > Queries**, and

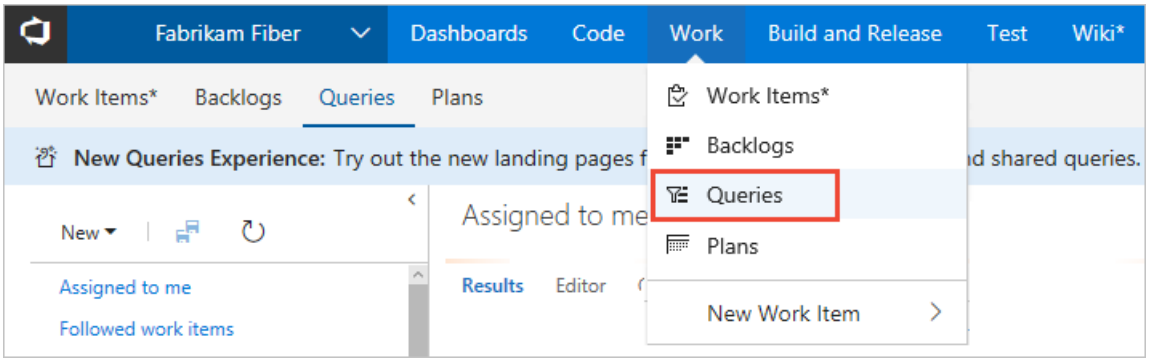
then (3) choose **All**.



If it is your first time opening **Queries**, the page opens to **Favorites**. This page lists those queries that you have indicated are a favorite. Otherwise, you can choose **All** to view all queries you've defined and shared queries defined for the project.

> **TIP**
> Queries you or your team have chosen as favorites show up on the **Favorites** page. Favorite queries along with other objects also appear on your **Project** page. To learn more, see Set personal or team favorites.

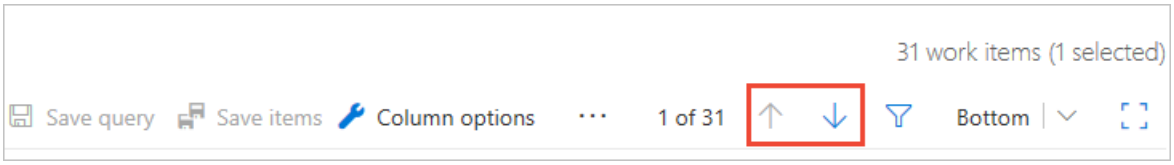
From your web browser, open **Boards > Queries**.



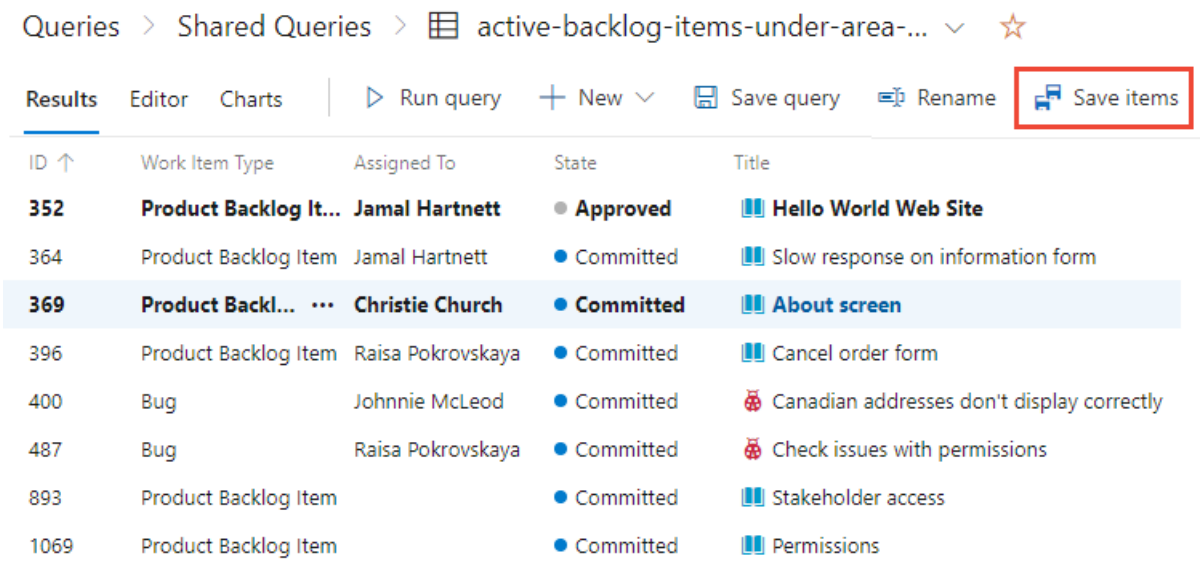
Use triage buttons and save work items

- Browser
- Visual Studio

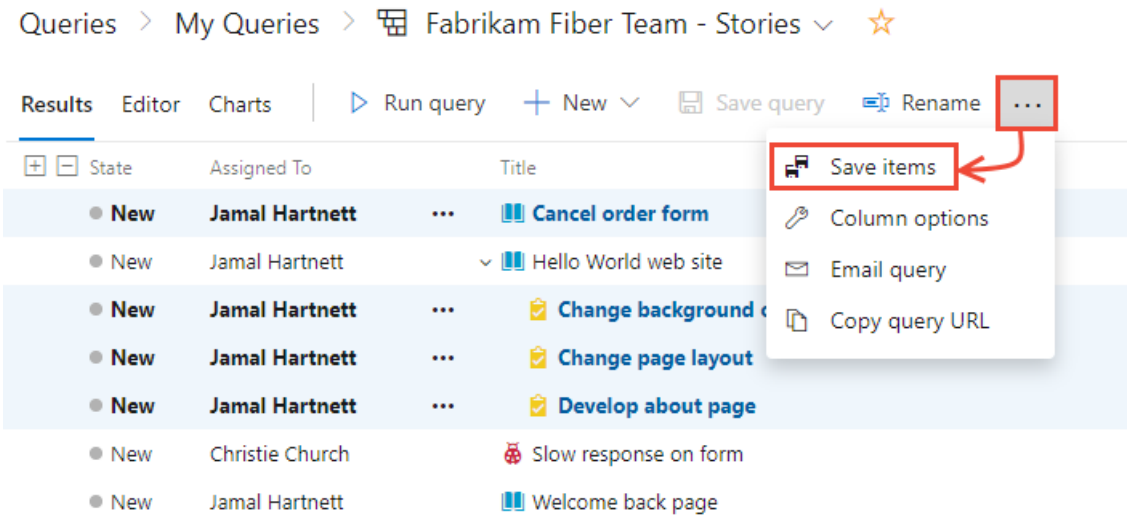
The buttons to move up or down within the query results list are outside the work item form. Choose **Bottom** to cycle through the choices for where the work item form appears: **Bottom**, **Right**, or **Off**.



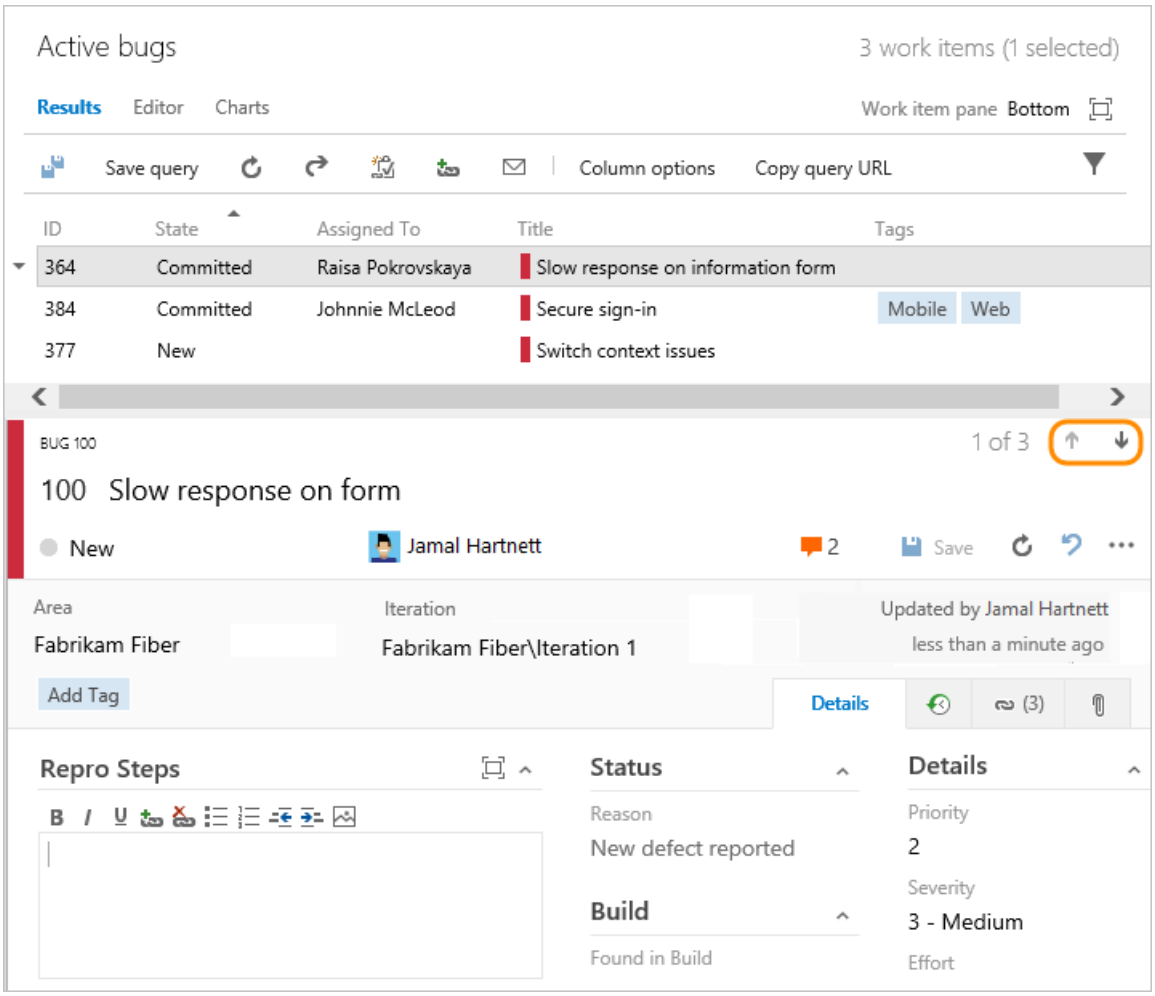
You can save each work item as you change it. Or, you can update multiple work items and save them all at once with **Save Items**.



If you don't see **Save Items**, choose the **More commands** and select the **Save Items** option.



The buttons to move up or down within the query results list are inside the work item form. Choose **Bottom** to cycle through the choices for where the work item form appears: **Bottom**, **Right**, or **Off**.

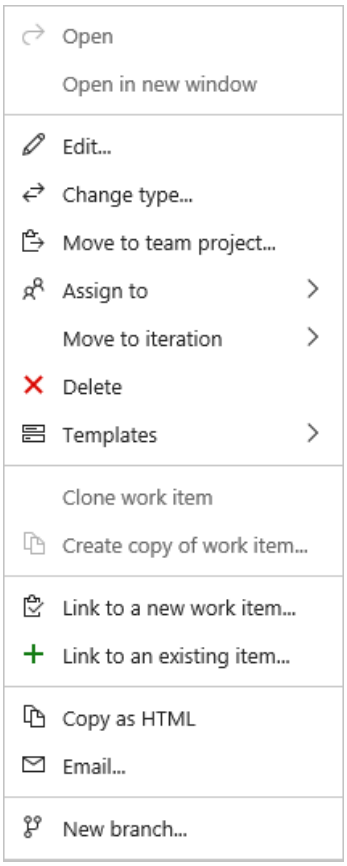


You can save each work item as you change it. Or, you can update multiple work items and save them all at once with **Save Items**.

Bulk update and save changes to work items

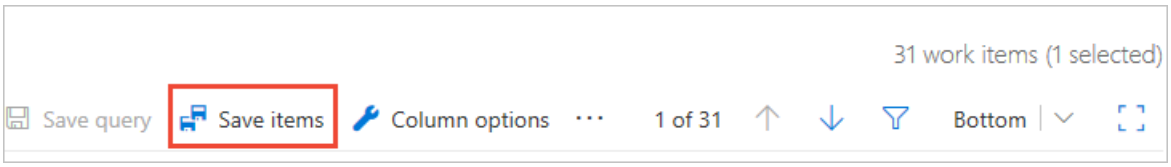
From the web portal, you can update several work items through the triage process, and then do a bulk save when you're finished.

Multi-select the work items you want to update, choose **More commands**, and then select the option you want from the menu of options.

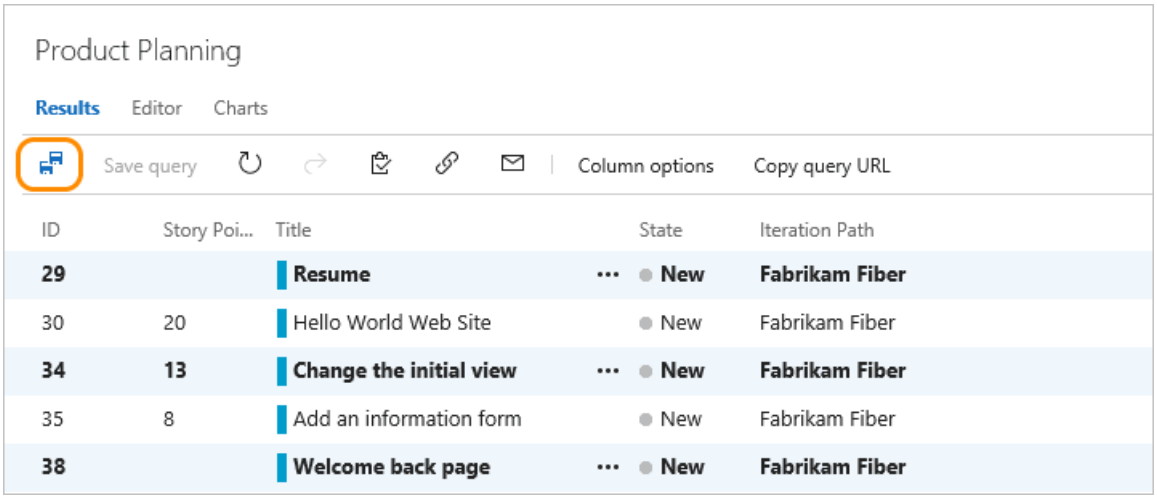


Work items shown in bold text indicate that local changes haven't yet been saved to the data store.

Choose **Save items** to save all work items you've modified.



Choose the double-save icon to save all work items you've modified.



Related articles

- Best tool to add, update, and link work items
- Manage bugs
- Create a query

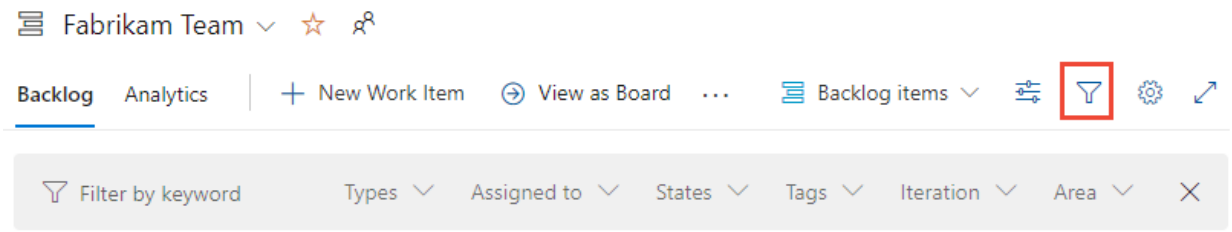
Interactively filter backlogs, boards, queries, and plans in Azure Boards

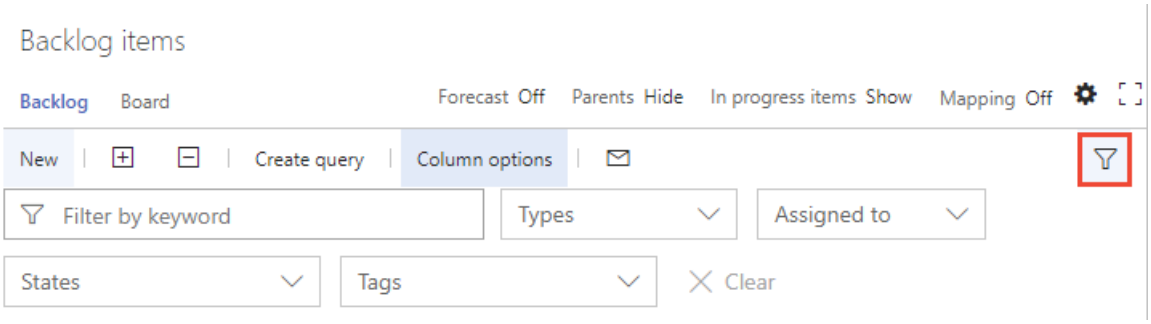
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

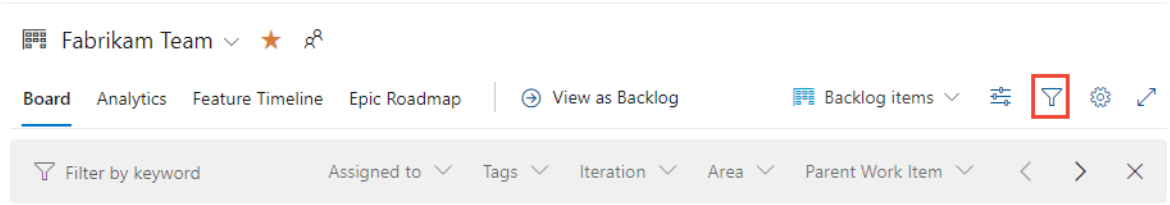
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

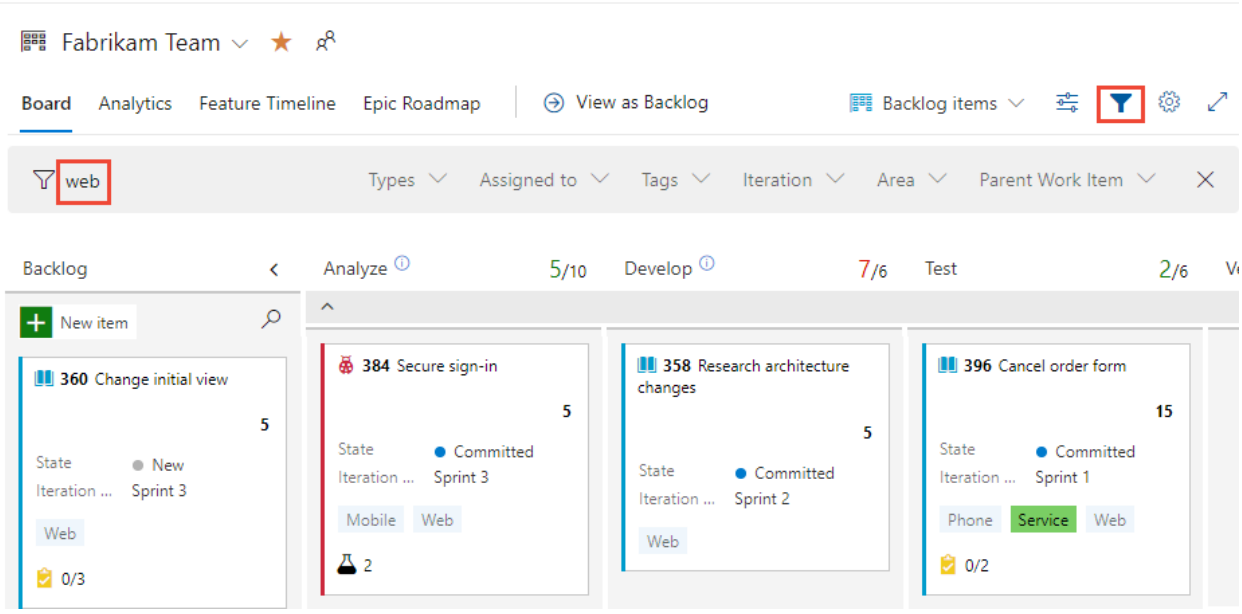
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

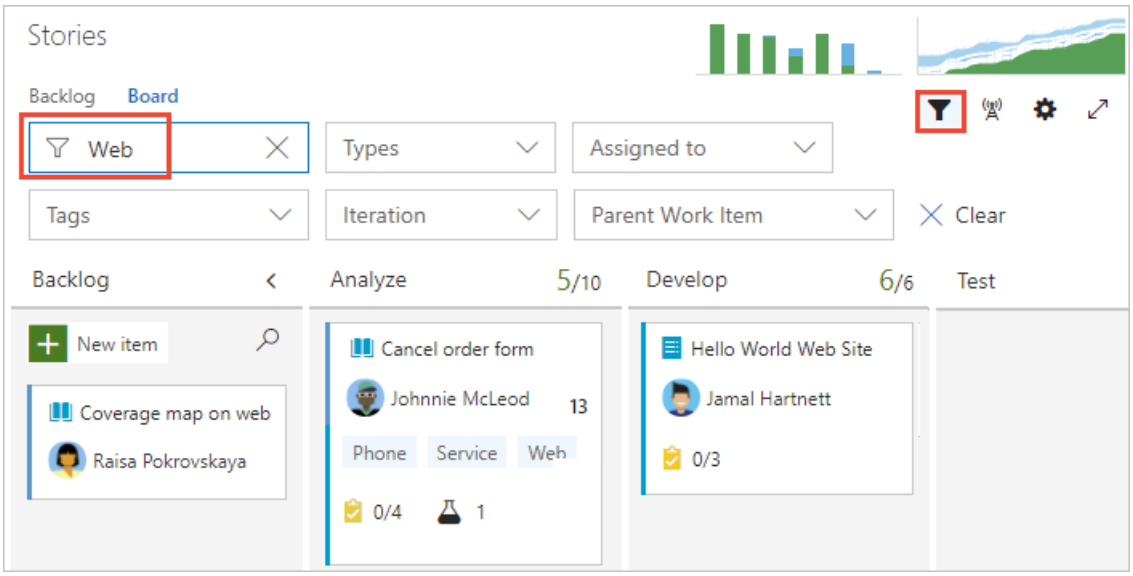
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

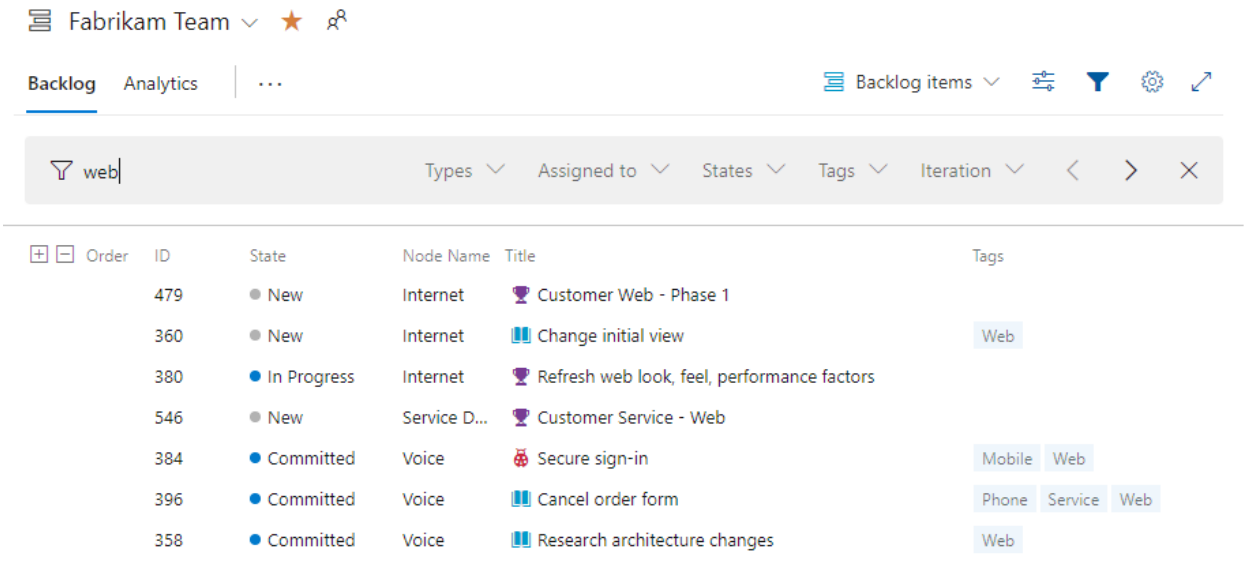


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

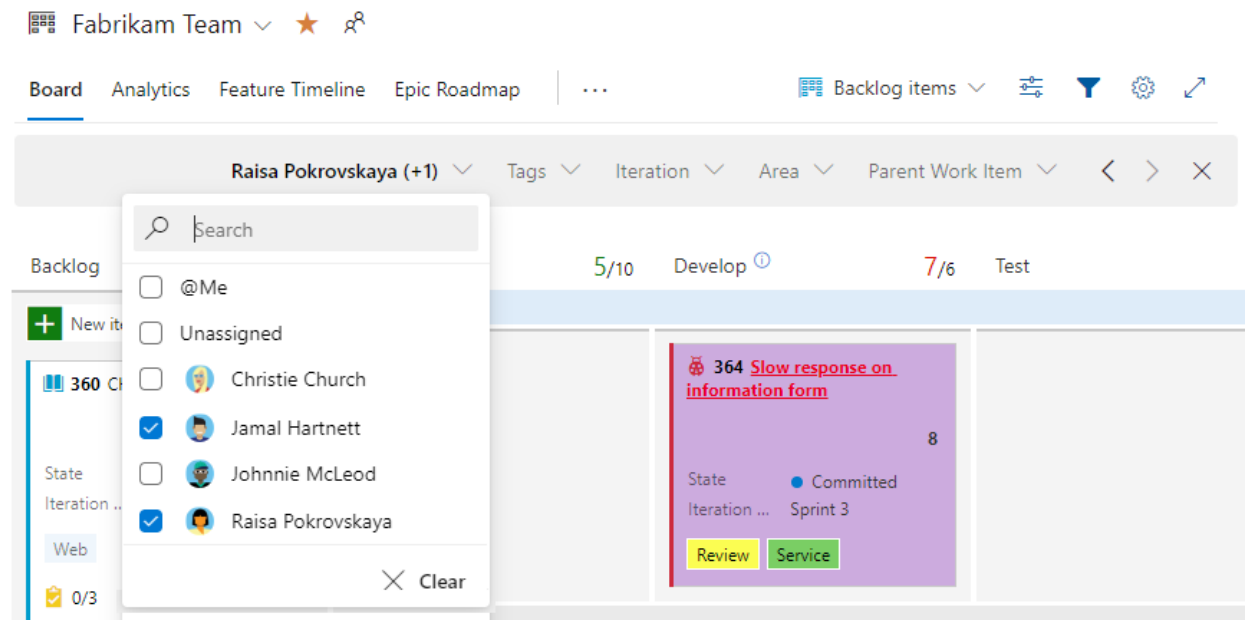
Features board

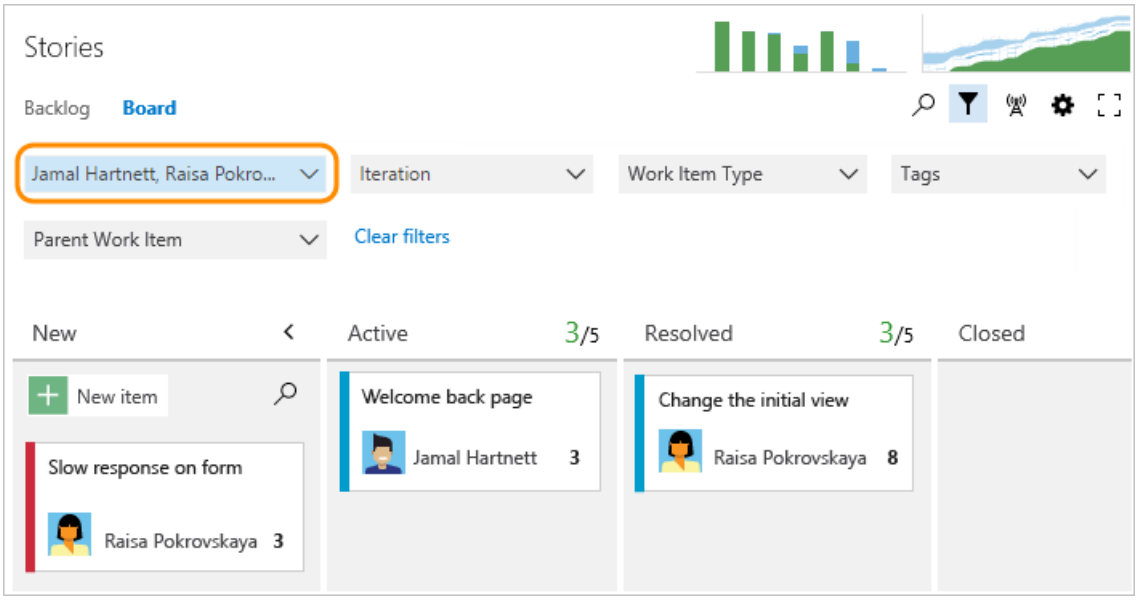
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

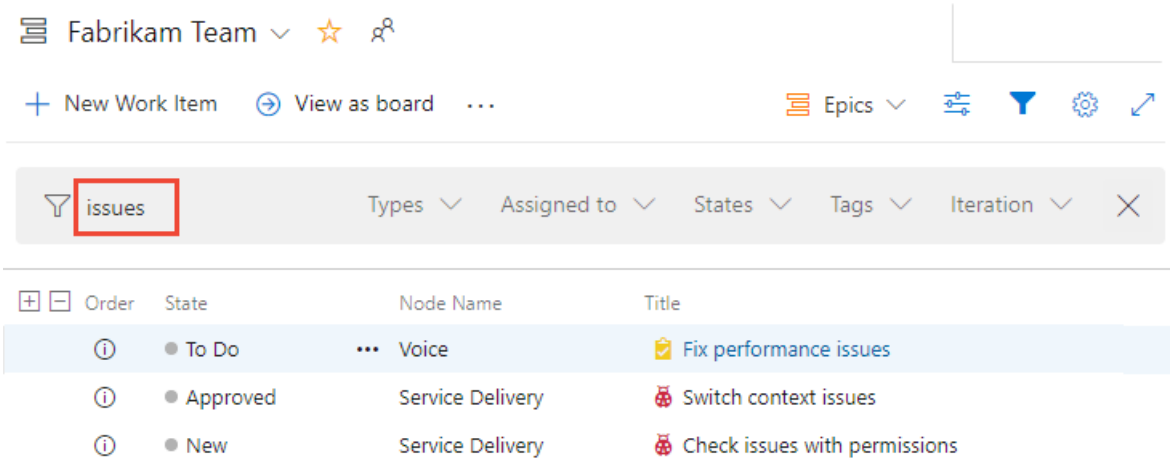
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

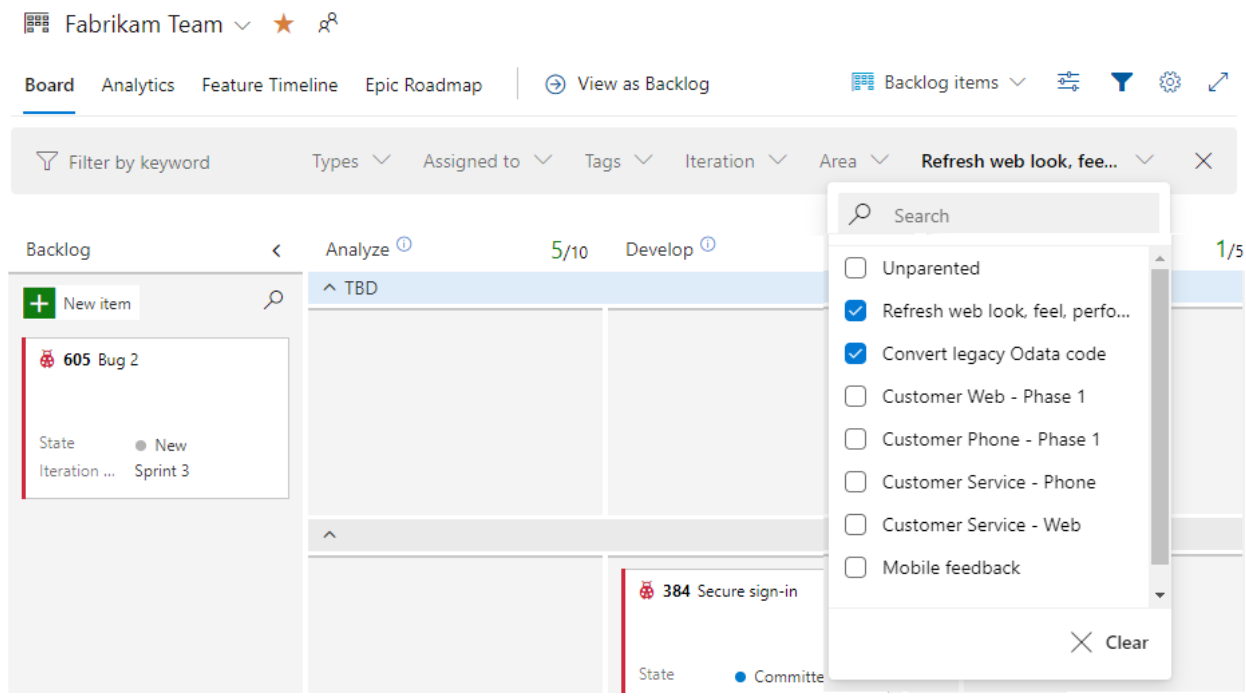
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

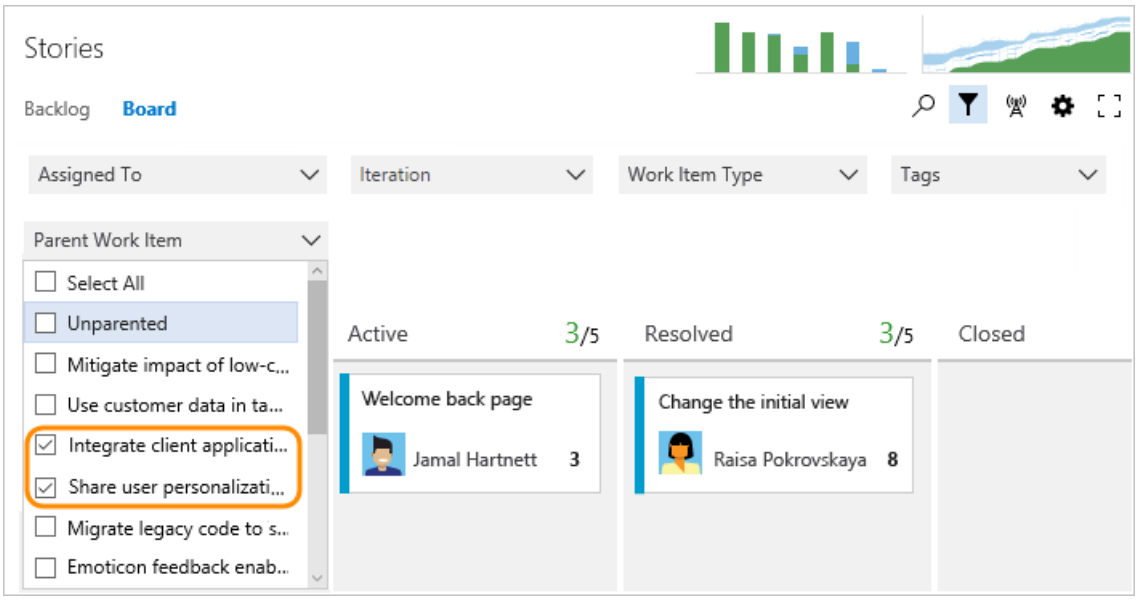
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



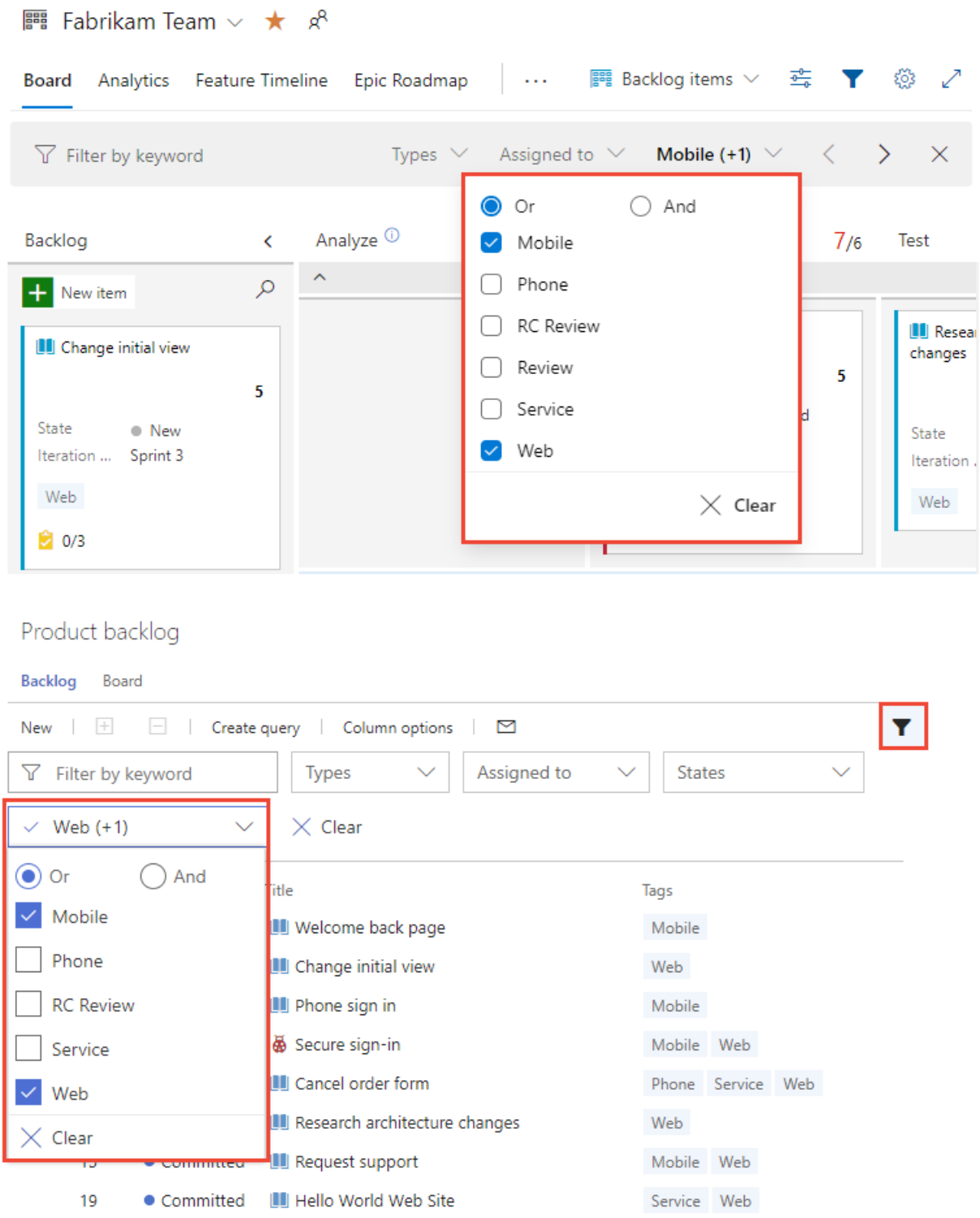


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

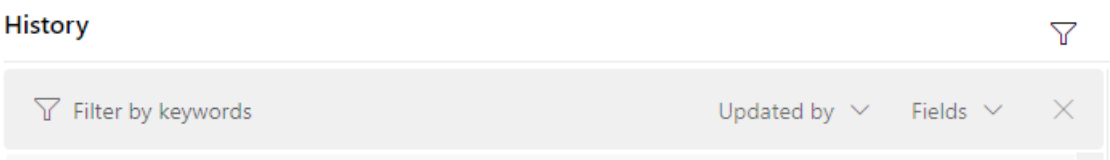


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Guidance to create high-performing queries in Azure Boards

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

While you can easily create work item queries, to create high-performing queries requires a deeper understanding. By improving your query performance, you improve your individual productivity, dashboard performance, and resource rate limits.

> **NOTE**
> Reference to service or resource rate limits only applies to queries run against Azure DevOps Services. To learn more, see Service limits and rate limits.

This article provides general guidelines on how to write a high-performing query. These guidelines apply to the following queries you create:

- Web portal queries
- Work Item Query Language (WIQL) queries
- az boards query command line
- REST API queries

- Web portal queries
- Work Item Query Language (WIQL) queries
- REST API queries

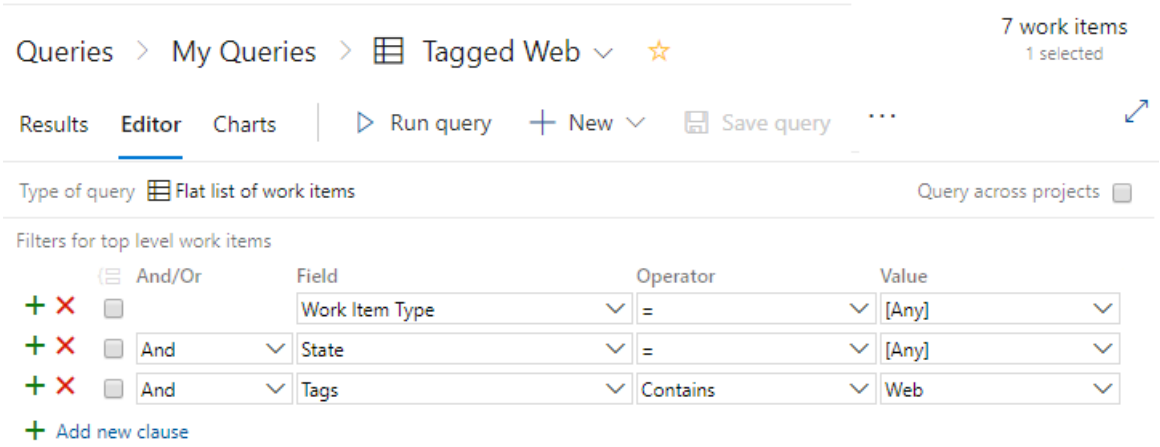
Create focused, selective queries

The first guideline for a high-performing query is to define a highly selective query. To do this, apply all filters that are necessary for your query. The more selective the query is, the smaller the result set is. The smaller the result set is, the more targeted and selective your query is.

Use tags to categorize work items

Use work item tags to categorize your work items instead of a custom field. Queries that filter on tags usually perform faster over those queries that filter on string matches.

Unlike custom field matches or partial matches, a query with a `Tags Contains` operation doesn't require a complete scan of all work item tables.



Use `Contains words` for string matches

To filter on a string match, use the `Contains Words` instead of the `Contains` operator. The `Contains Words` operator runs a full-text search on the specified field, which tends to complete more quickly.

The `Contains` operator runs a table scan, which is a slower operation than that used with the `Contains Words` operator. It also consumes more CPU cycles. These CPU cycles can cause you to encounter rate limitations. For more information, see Service limits and rate limits and Rate limits.

Specify small groups with the `In Group` operator

The `In Group` operator filters work items based on matches within a group of values. The group of values correspond to those contained within a team, security group, or work tracking category. For example, you can create a query to find all work items that are assigned to any member of a team. Or, find all work items that belong to the requirements category (`Microsoft.RequirementCategory`).

When you filter on a group that contains a large number of members, your result set tends to be larger and non-selective. Also, if a group corresponds to a large Azure Active Directory (Azure AD) group, the query generates a fairly large cost to resolve that group from Azure AD.

Avoid use of negated operators

Negated operators—such as `<>, Not In, Not Under, Not In Group` —tend to generate non-selective queries and large result sets.

Only use negated operators when absolutely necessary. Always try to find alternatives first. For example, if **Field1** has values *A*, *B*, *C*, *D*, specify the `Field1 In A, B, C` clause, instead of the negated `Field1 <> D` clause.

Avoid string comparisons

Queries that contain string comparisons generate table scans that are inherently inefficient. Instead, we recommend you use tags or a specific custom field as alternatives, particularly when a query performs poorly.

Limit `Or` operators

Try to limit the number of `Or` operators defined in your query. Queries run better when fewer `Or` operators are used. Too many `Or` operators can make your query non-selective. If your query runs slowly, reorder the `Or` operator clause towards the top of the query clauses.

Save your query

Save your query to improve query performance. Due to internal optimizations, saved queries tend to perform better over unsaved queries. Always save your query when you plan to reuse it. Even WIQL queries run through a REST API, save the WIQL through the web portal to make your REST API calls less prone to future performance regressions.

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Related articles

- Service limits and rate limits
- Create managed queries
- Query fields, operators & macros
- WIQL syntax
- Query quick reference
- az boards query command

Set permissions on queries and query folders in Azure Boards and Azure DevOps

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As with most project objects, you can control access by setting permissions. With queries, you can configure users and groups to create, delete, view, and manage permissions of shared queries and shared query folders.

All users, except those users assigned to the Readers group, can create and edit their own queries and save them under **My Queries**. Only the signed in user can view queries saved under their **My Queries** space.

By default, only members of the Project Administrators group can create and edit queries and folders under **Shared Queries**, or change the permissions for a query or folder.

By creating folders under Shared Queries, you can grant permissions to users for each folder. For example, if you have several teams contributing to a project, then you might want to create a folder under Shared Queries for each team to manage their own set of shared queries.

Prerequisites

- To create or edit a shared query or manage permissions, you must be a member of the **Project Administrators** groups with **Basic** or higher access level. Or, you must have your **Contribute** permission set to **Allow** for the shared query folder. To get added to this group, see Change project-level permissions
- Or, to create a query or folder under a shared query folder, you must have the **Contribute** permission set explicitly to **Allow** for the query folder and be granted **Basic** or higher access level.
- Or, to change permissions of a query or query folder, you must have the **Manage Permissions** permission set explicitly to **Allow** for the query folder and be granted **Basic** or higher access level.

Users with Stakeholder access can't create or save queries in a Shared folder. To learn more about access levels, see Stakeholder access quick reference.

> **TIP**
> Consider creating a query folder for each team and give the team administrators or the team group query permissions to manage their folder.

Default query permissions

A ✔ in the following table indicates that the corresponding security group has permission to exercise the task by default.

Task

Readers

Contributors

Project admins

View and run managed queries, view query charts

✔

✔

✔

Create and save managed My queries, query charts

✔

✔

Create, delete, and save Shared queries, charts, folders

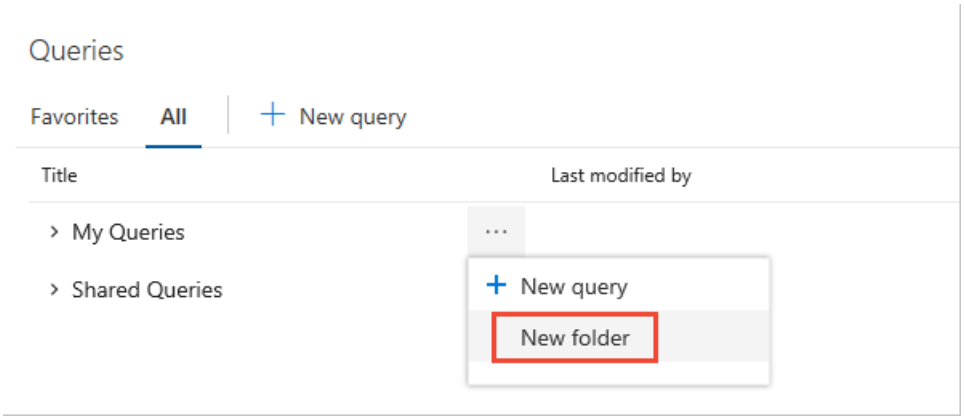
✔

Set permissions on a new query folder

You set permissions from the web portal. To open **Queries**, see View, run, or email a query.

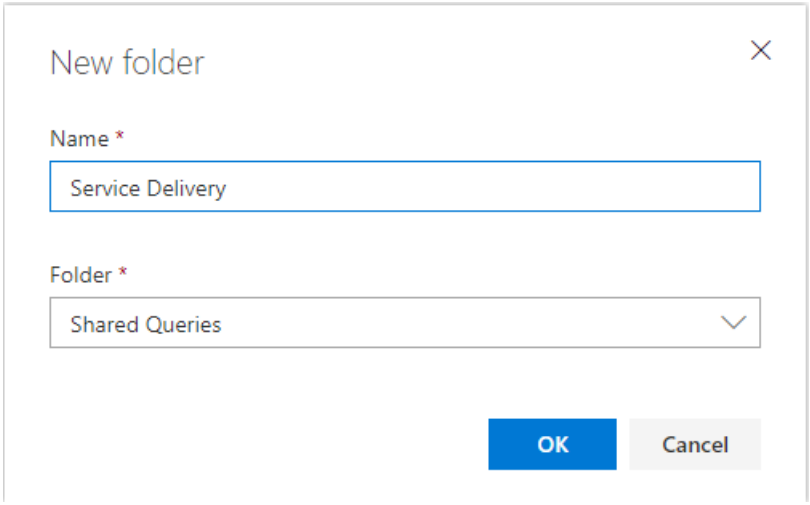
> **TIP**
> You need **Delete** permissions to rename or move a shared query or folder, and **Contribute** permissions for the folder where you move the query to.

1. Choose **All**. Expand **Shared Queries**.
2. To add a folder, choose the ··· actions icon for an existing folder or the top container folder, and choose **New folder**.



3. Enter the name for the folder. If you want to change the location of the folder, select **Rename** from the folder drop-down menu.

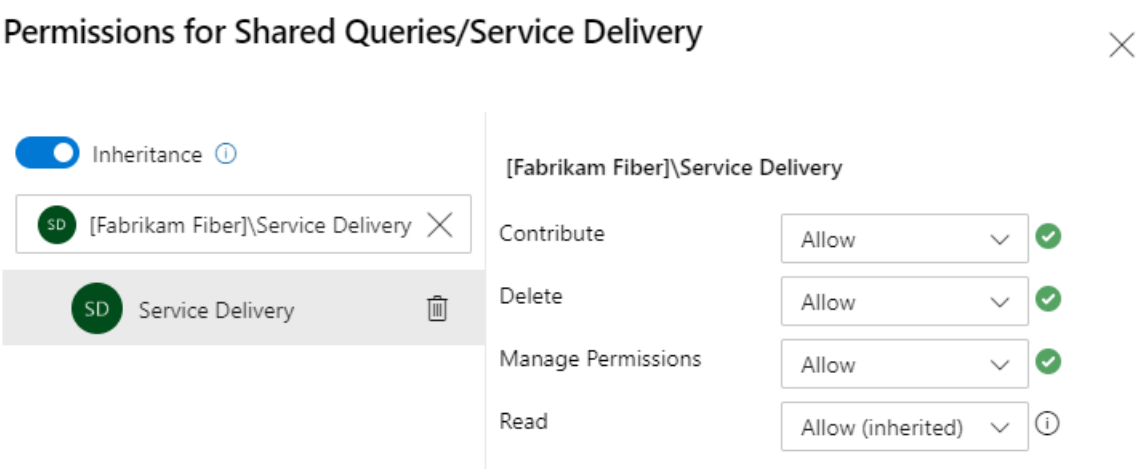
   Here we name the folder *Service Delivery* with the intention that it will be used by the Service Delivery team.



4. To set permissions for the folder you just added, choose the ··· actions icon and select **Security**.

5. Change the permissions so that the team member or group can contribute and manage permissions for the folder. Enter the name of a user or group within the search box.

   Here we add the Service Delivery team and grant them permissions to create and manage permissions to all queries and folders under the Service Delivery folder.

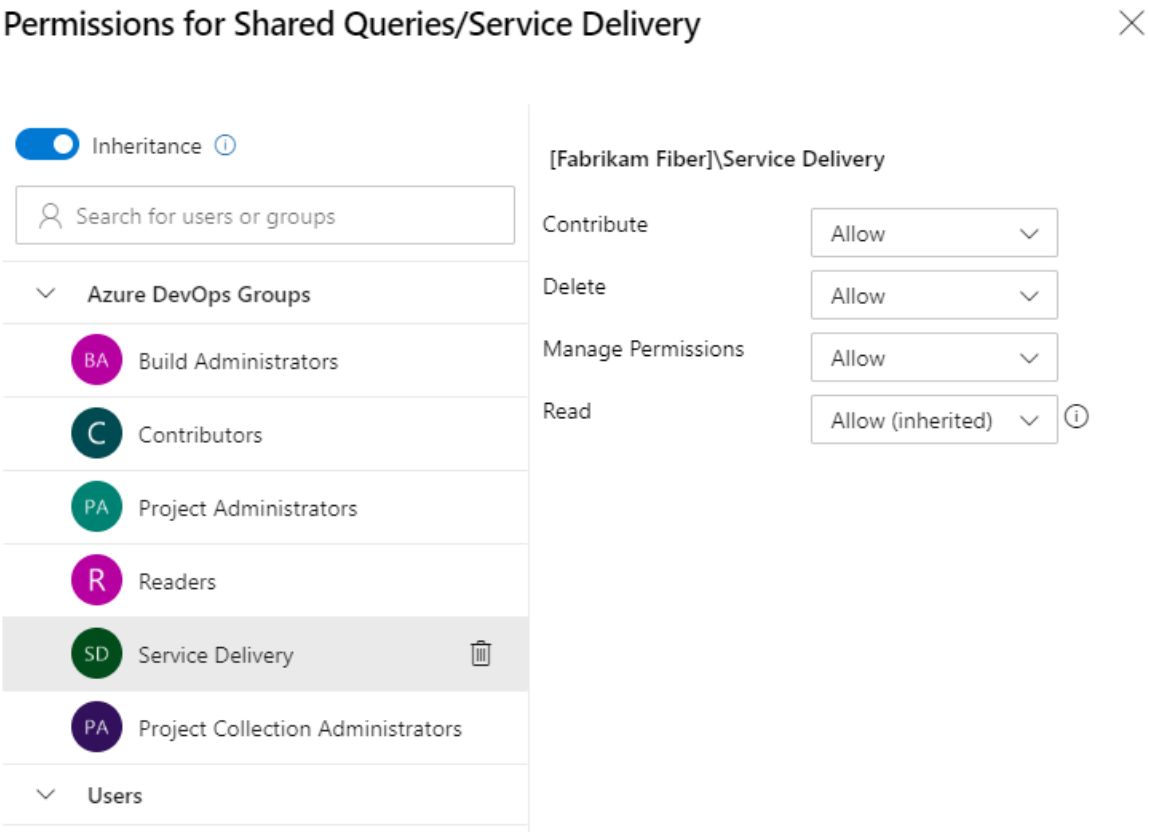


   Contribute allows team members to create and edit queries and folders under the folder where the permissions were granted. And, **Manage Permissions** allows team members to manage the permission settings on queries and subfolders.

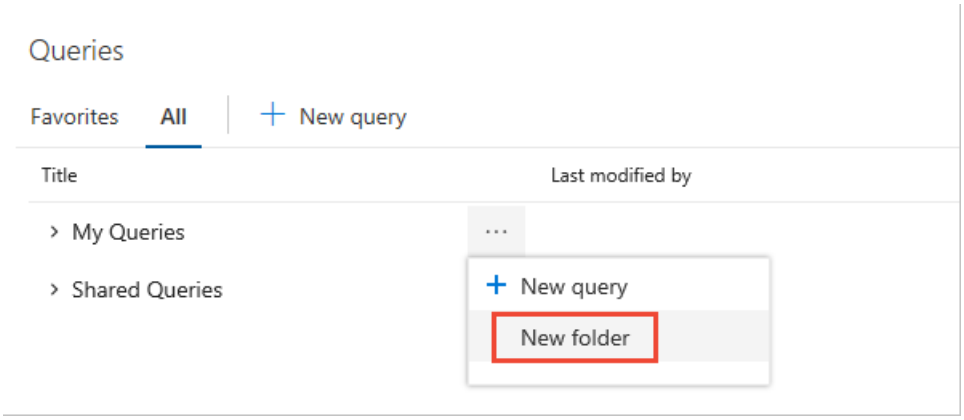
6. (Optional) Turn off inheritance. Default is **On**. By turning off inheritance for a folder, you disallow inheritance of permissions that exist up the chain of query folders. To learn more, see Permissions, Inheritance.

7. Close the dialog when done.

8. Reopen the Security dialog and choose Service Delivery to verify that the permissions are set.

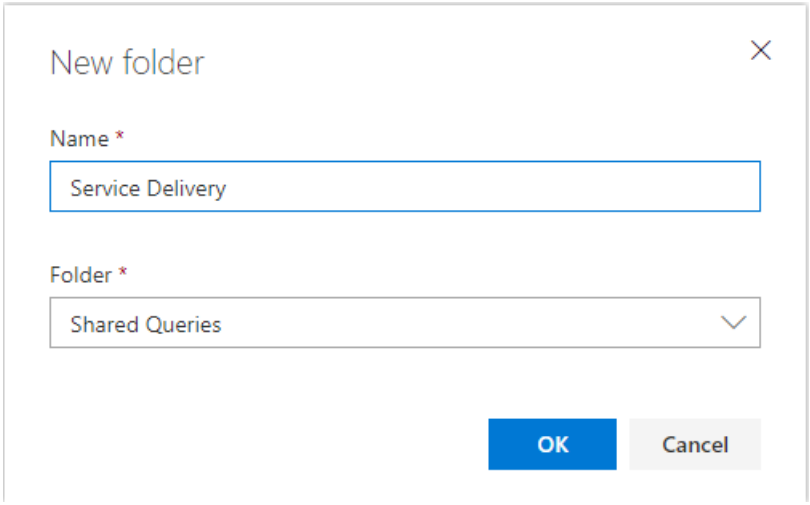


1. Choose **All**. Expand **Shared Queries**.
2. To add a folder, choose the ··· actions icon for an existing folder or the top container folder, and choose **New folder**.



3. Enter the name for the folder. If you want to change the location of the folder, select it from the Folder drop down menu.

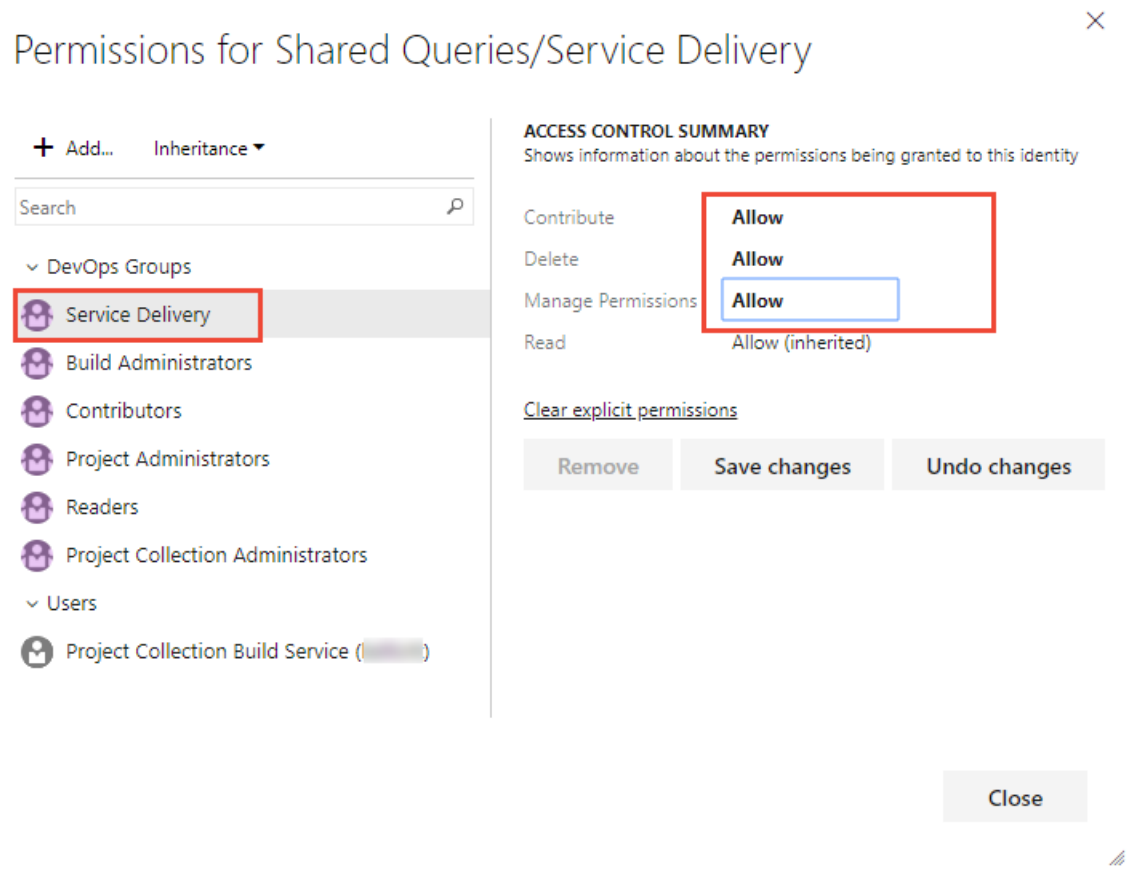
   Here we name the folder *Service Delivery* with the intention that it will be used by the Service Delivery team.



4. To set permissions for the folder you just added, choose the ··· actions icon and select **Security**.

5. Change the permissions so that the team member or group can contribute and manage permissions for the folder. Choose the **Add...** menu to add a user identity or group.

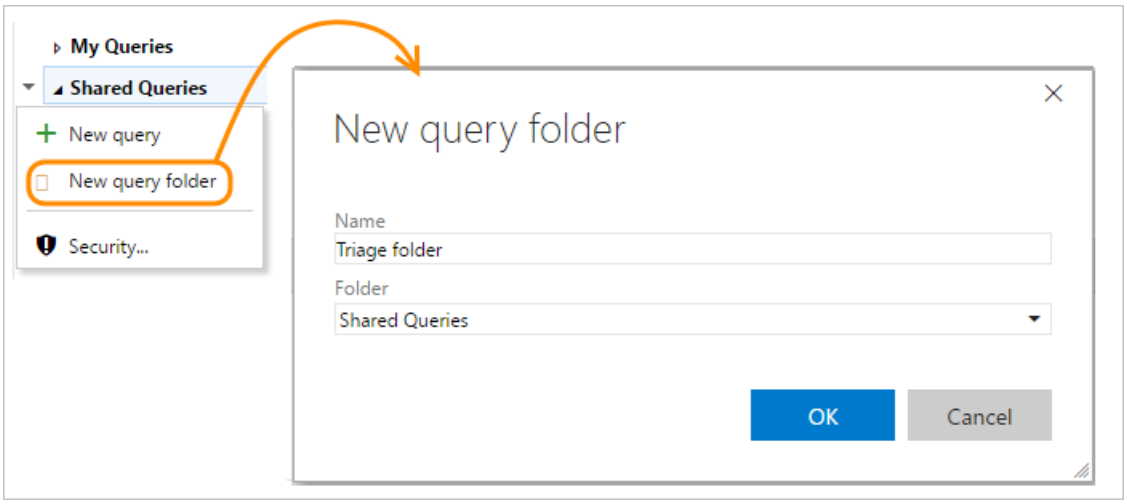
   Here we add the Service Delivery team and grant them permissions to create and manage permissions to all queries and folders under the Service Delivery folder.



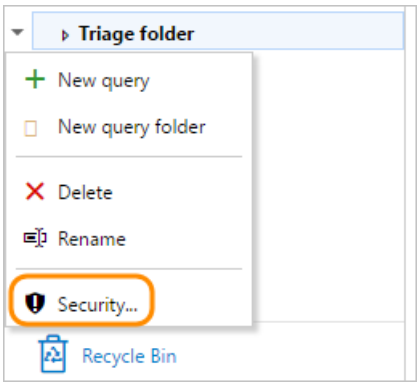
   Contribute allows team members to create and edit queries and folders under the folder where the permissions were granted. And, **Manage Permissions** allows team members to manage the permission settings on queries and subfolders.

6. (Optional) Turn off inheritance. Default is **On**. By turning off inheritance for a folder, you disallow inheritance of permissions that exist up the chain of query folders. To learn more, see Permissions, Inheritance.

1. Add a query folder under **Shared queries** or a subfolder. Choose the ▾ context menu icon for the folder and choose **New query folder**.

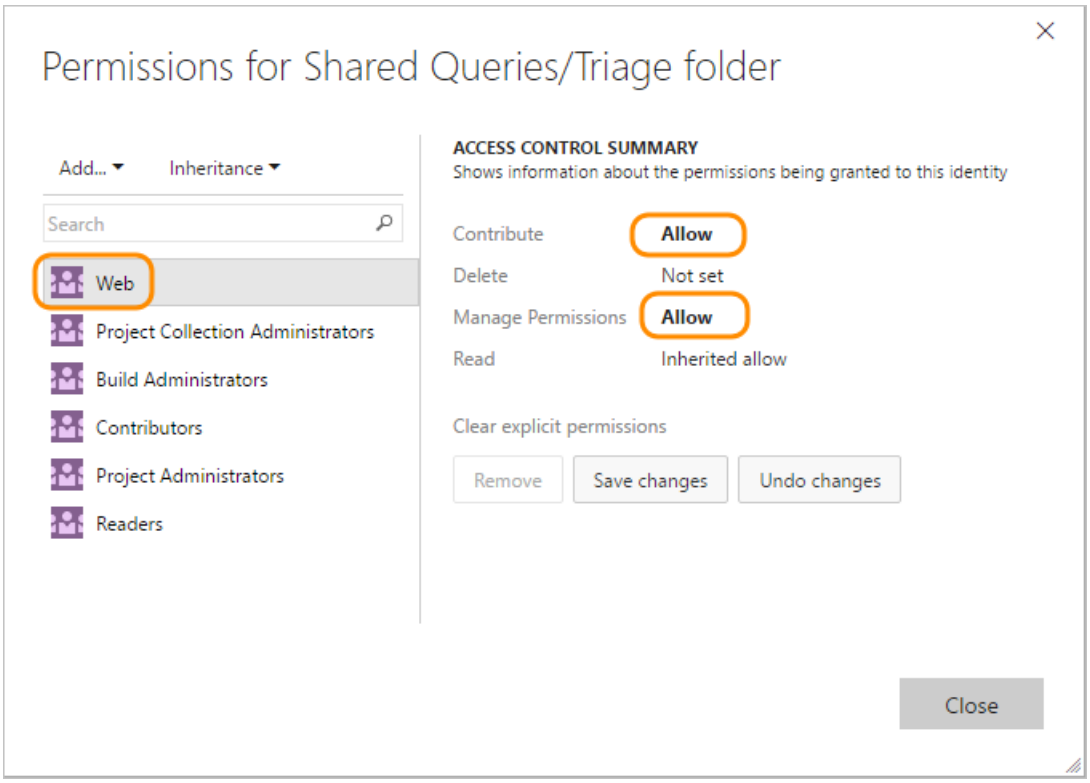


2. To set permissions for the folder, choose the ▾ context menu icon for the folder you just added and choose **Security**.



3. Change the permissions so that the team member or group can contribute and manage permissions for the folder.

   Here we add the Web team and grant them permissions to create and manage permissions to all queries and folders under the Triage folder.



   Choose the **Add...** menu to add a user identity or group.

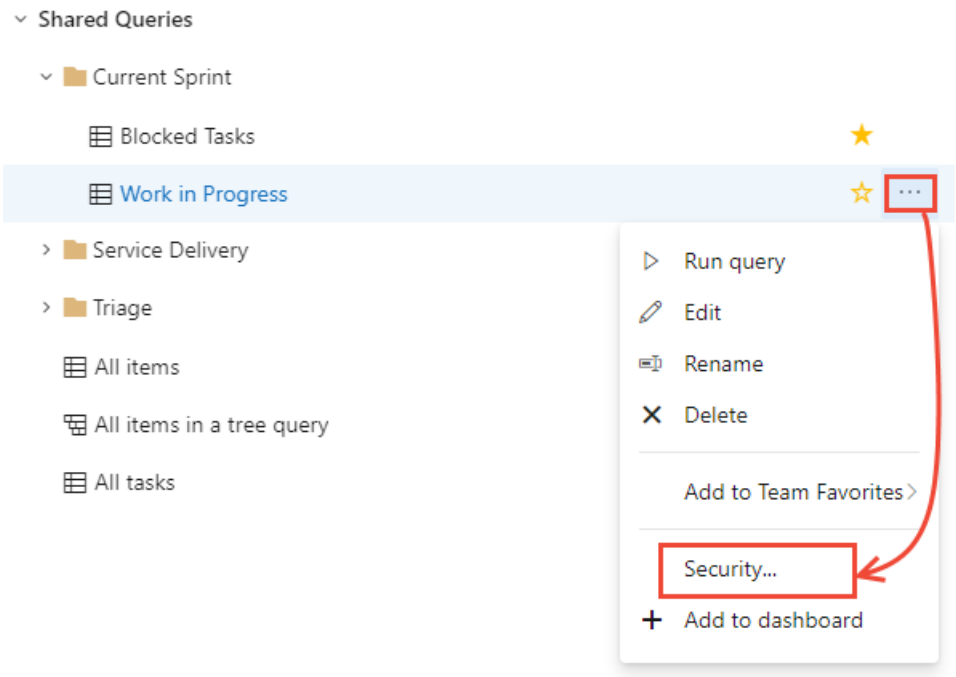
Contribute allows team members to create and edit queries and folders under the folder where the permissions were granted. And, **Manage Permissions** allows team members to manage the permission settings on queries and subfolders.

4. (Optional) Turn off inheritance. Default is On. By turning off inheritance for a folder, you disallow inheritance of permissions that exist up the chain of query folders. To learn more, see Permissions, Inheritance.

Set permissions on a shared query

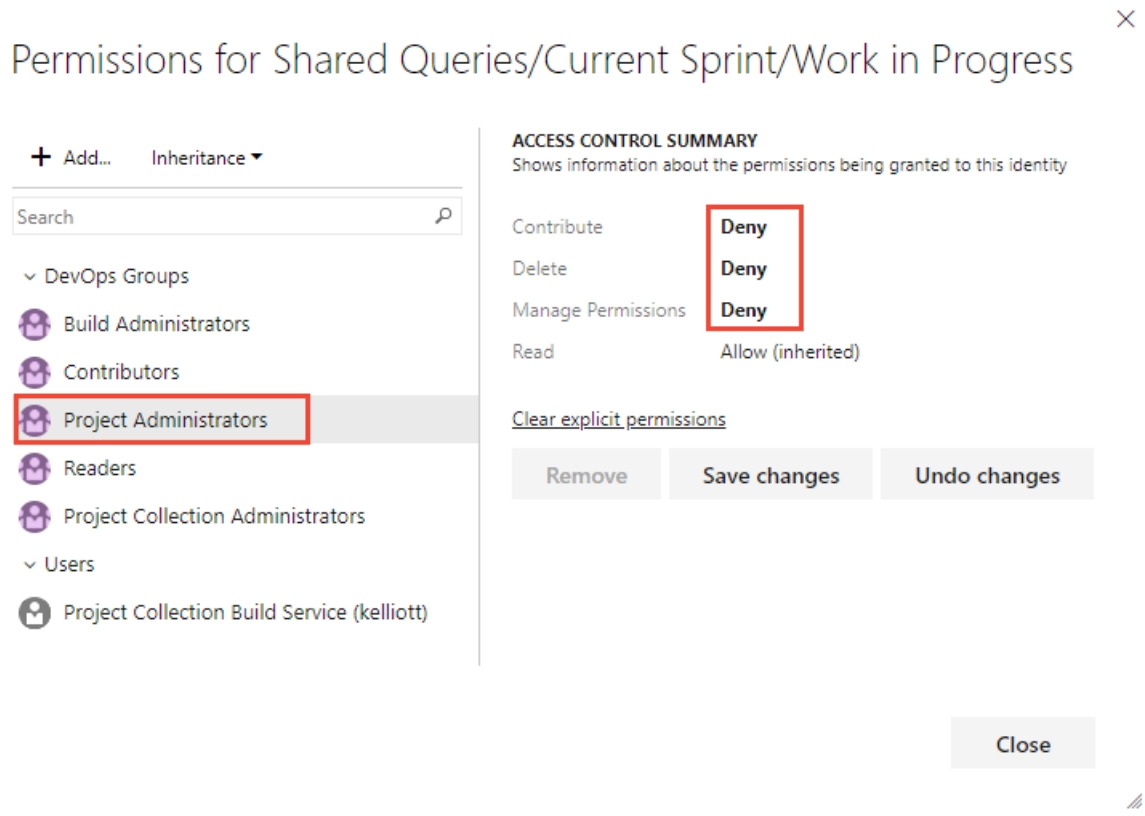
To keep anyone else from modifying a shared query that you create, you may want to set permissions on a specific query. You can set permissions by opening the permissions dialog for the specific query.

1. Choose the ··· actions icon and select **Security**.

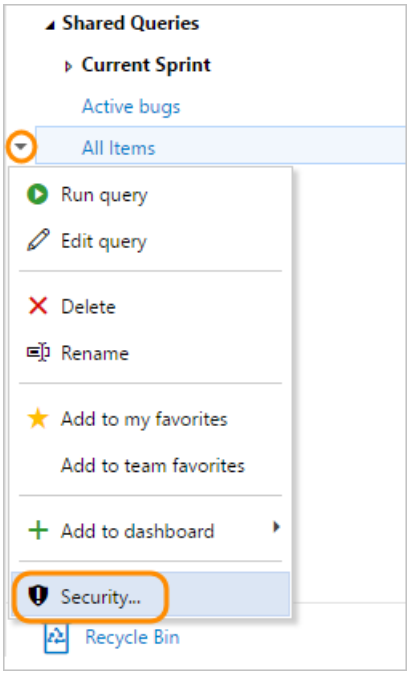


2. Change the permissions so that the team member or group can't edit, delete, or change permissions for the query.

   Here we deny permissions for project admins.

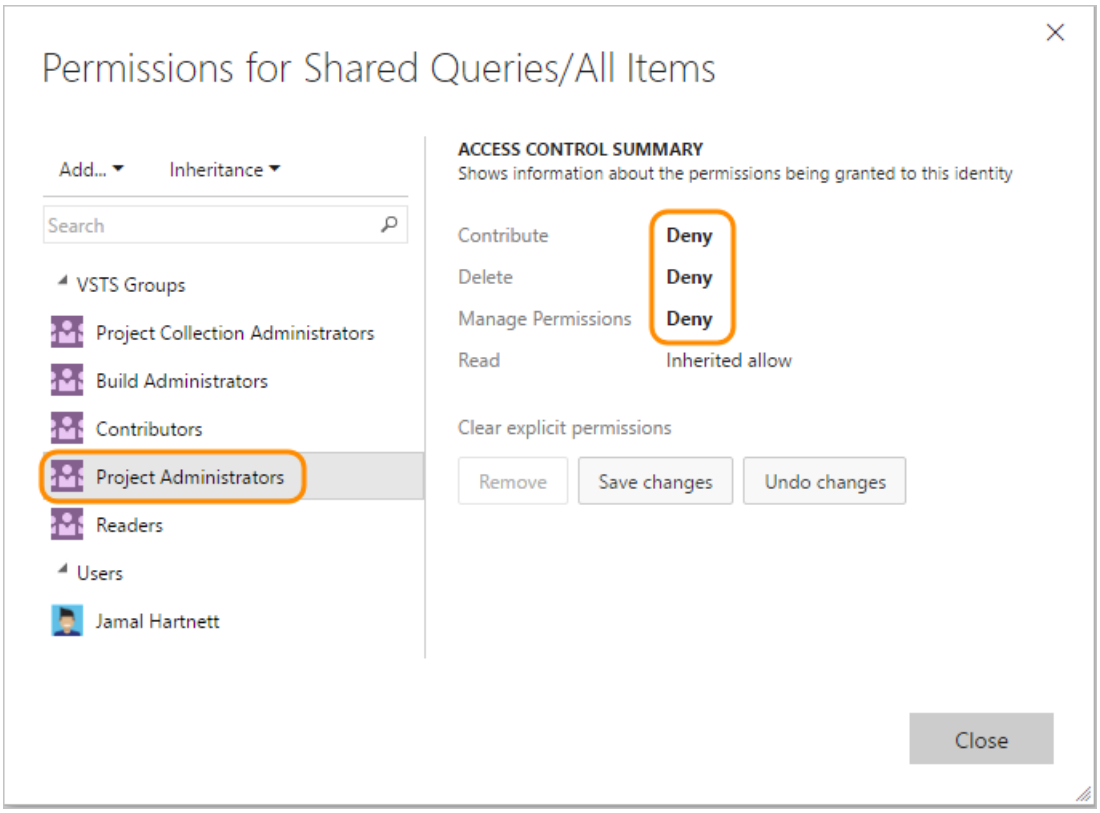


1. Choose the ▾ context menu icon and select **Security**.



2. Change the permissions so that the team member or group can't edit, delete, or change permissions for the query.

   Here we deny permissions for project admins.



Related articles

With queries, you cannot only list work items, you can create status and trend charts and add them to dashboards. You can learn more about permissions and working with queries from these resources:

- Manage queries and query folders
- Permissions and access for work tracking
- Add a chart to a dashboard
- Dashboards

Manage columns in a work item list in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Visual Studio 2022 | Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015 | Visual Studio 2013

Each column corresponds to a work item field. You can add and remove columns within work item lists to show the fields of interest to you. Or, you can drag a column to a new position. Your settings persist for each page you customize and are only valid for your views.

Specifically, you can do the following actions from the following list views:

Action	Backlogs	Sprint backlogs	Queries	Work items
Add or remove a column field	Yes	Yes	Yes	Yes
Add or remove the Parent field	Yes	Yes	Yes	Yes
Add or remove a rollup column	Yes	No	No	No
Sort on a column	No	No		

Yes

Yes

> **TIP**
> Unlike a query result, you can't sort a backlog by a column. However, you can use the **Create Query** link on each backlog to create a query that you can sort on any field column you choose from the **Sorting** tab of the Column options dialog. While you may be able to add a field to sort on, not all fields are supported. For example, selection of the **Parent**, **History**, **Description**, or other rich-text field will result in the display of an error message as you can't sort on these fields.

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. You can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project. If your project uses the Inherited process model, you can view the list of fields defined for your collection from **Organization Settings > Process > Fields**

You can add most fields listed in the Work item field index. All fields defined within the project collection or organization are available for selection, even those fields that aren't used for your particular project.

About persistence of column options

Once you set the column options for a specific view, your settings persist until you change them. The following notes apply to specific views.

- Column options you set for a backlog apply only to the active team and backlog. Options you set for the product backlog differ from the options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query.
- Column options you set for one of the supported Work items views persists for a specific view only, such as **Assigned to me**, **Following**, **Mentioned**, and so on.

- Column options you set for a backlog apply only to the active team and backlog. Options set for the product backlog differ from those options you set for a portfolio backlog.
- Column options you set for a Sprint backlog persist for all sprints you select until you change them.
- Column options you set for a query persist when you save the query. For Shared queries, your settings can get overwritten when someone else saves different column option settings.

> **NOTE**
> You can't set column options for other members of your team, nor can you set default column options.

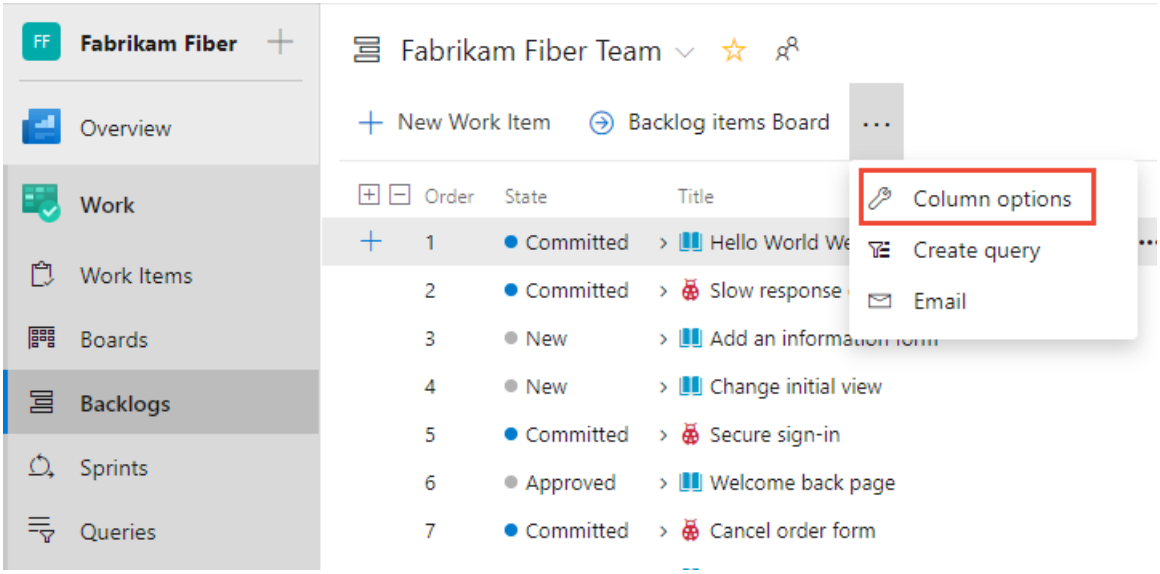
> **NOTE**
> You can't set column options for other members of your team. Also, for projects that use the Inheritance process model, you can't set default column options. For projects that use the On-premises XML process model, you can set the default column options for product, portfolio, and sprint backlogs. To learn how, see Process configuration XML element reference.

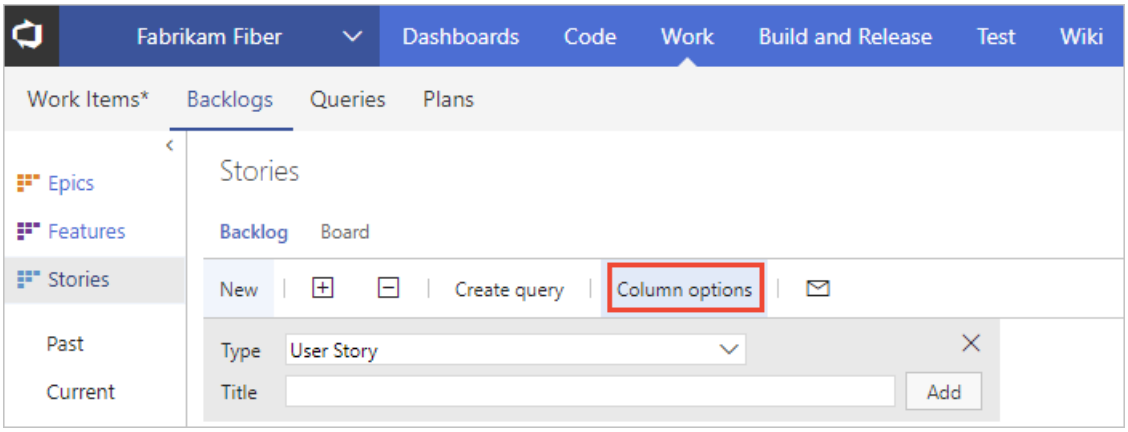
NOTE

You can't set column options for other members of your team. For projects that use the On-premises XML process model, you can set the default column options for product and portfolio backlogs. To learn how, see Process configuration XML element reference.

Open the Column options dialog

Start by opening the **Column Options** dialog. If you don't see the option, choose the ... and choose from the options provided.





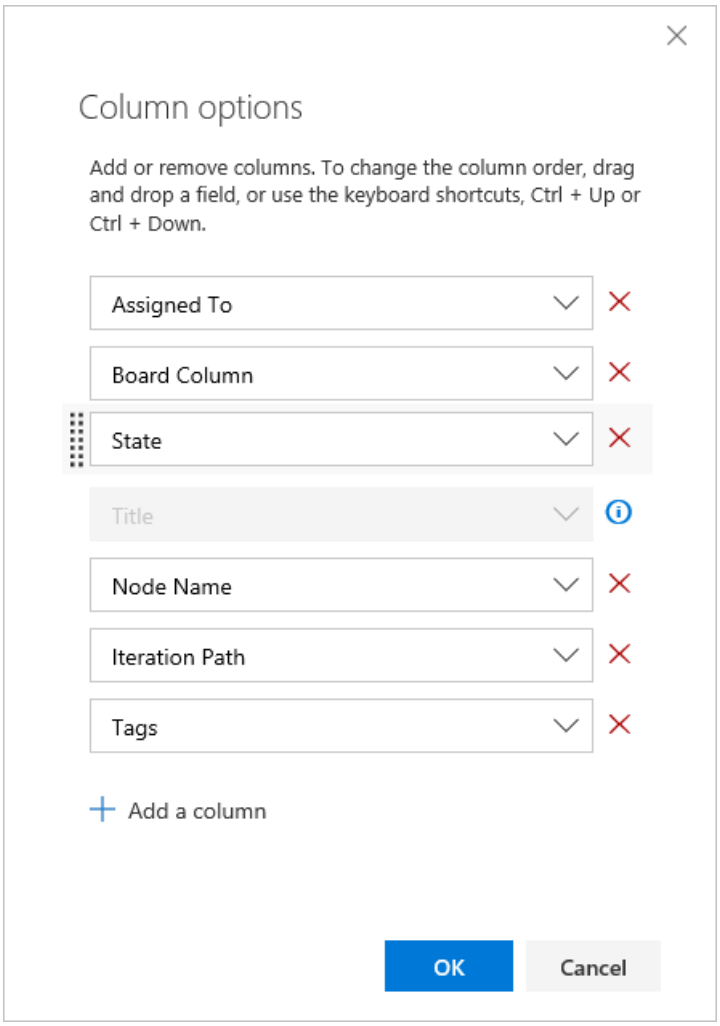
Add or remove columns

- Browser
- Visual Studio

In the Column options dialog, choose **Add a column** to add a field that isn't shown. To change the order of the fields, drag-and-drop the field where you want it within the set of selected fields. And, to remove a field, choose the ✕.

NOTE

The following dialog is available from TFS 2018.2 and later versions.

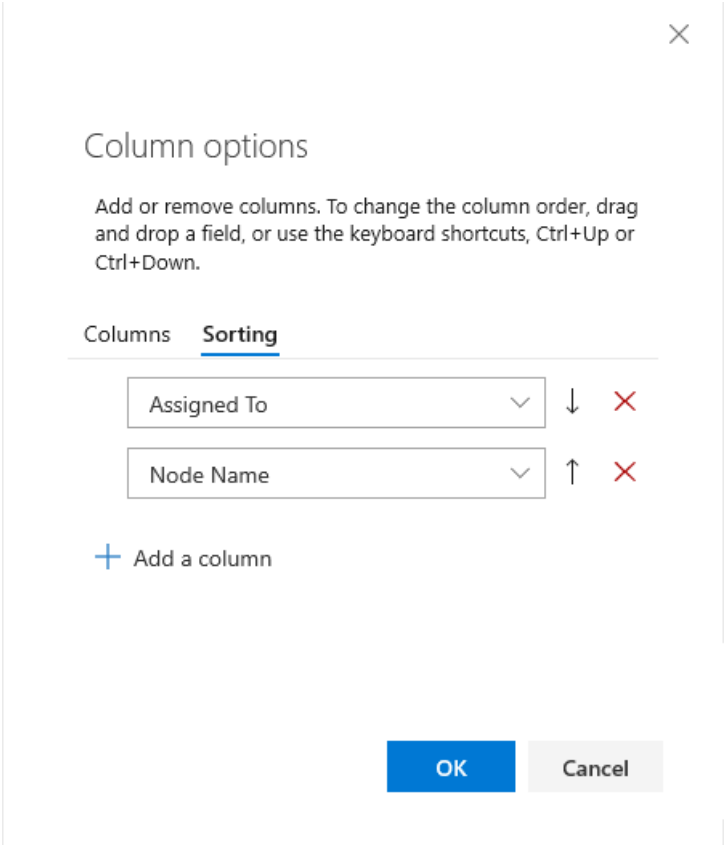


Add or remove rollup columns

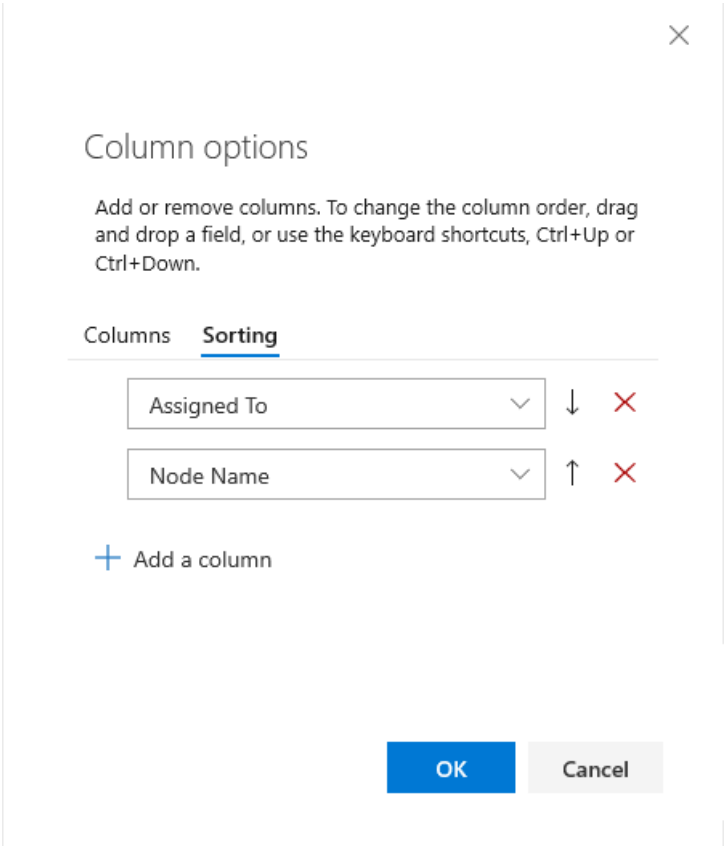
Rollup columns can display progress bars or the sum of numeric fields of child items. You can add them to any product or portfolio backlog. To learn more, see Display rollup progress or totals.

Sort on a column

You can sort query results and Work items views. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



You can sort query results. From the Column options dialog, choose **Sorting**. Add or remove a column field and drag and drop it into the order you want. Choose the up or down arrows to choose whether it sorts in ascending or descending order.



Use keyboard shortcuts to change the column order, column width, or sort options

You can change the column order, column size, or sort options by using the following keyboard commands:

- To change the column order, choose the field and drag it to a new location
- To resize a column, choose the column divider to the right of the field and drag to a new location
- For query results:
  - Add the field as a column in order to sort by that field
  - To sort by a column, hold down the SHIFT key and select on the field
  - To reverse the sort order, SHIFT+click on the field
  - To sort by multiple columns, SHIFT+click on each column in the order you want to sort

For other keyboard shortcuts, enter **?** to display available shortcuts based on the page you're on.

Related articles

- Display rollup progress or totals
- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- View, run, or email a work item query
- Create managed queries
- Customize a sprint Taskboard

- Interactively filter backlogs, boards, queries, and plans
- Work item field index
- Backlogs, boards, and plans
- Create managed queries

Cross-service and enhanced query operations

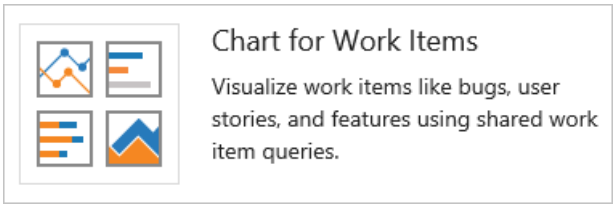
10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

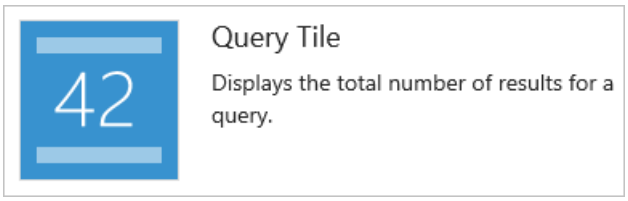
Managed queries are primarily focused on listing and working with work items. However, the query capabilities also support several cross-service operations, some of which require installation of a Marketplace extension.

Query-based dashboard widgets

Along with defining flat-list query charts that you can add to a dashboard, you can configure one of the following query-based widgets.



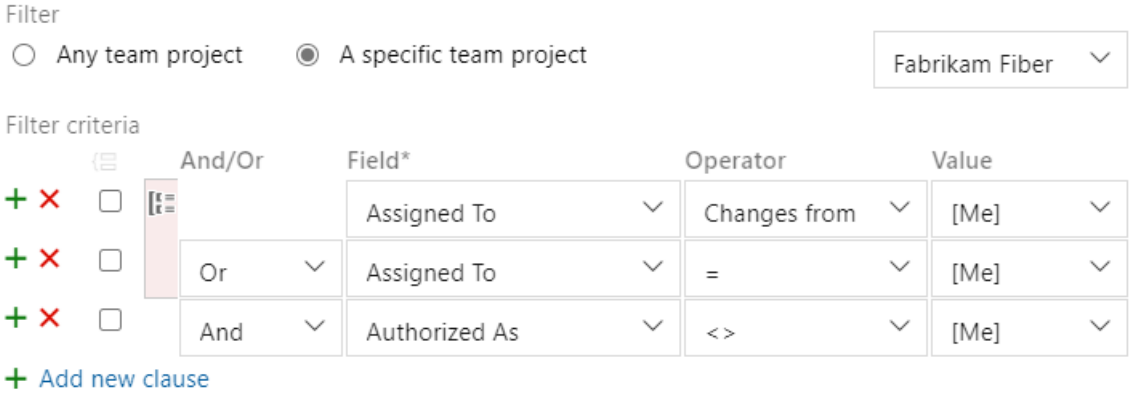




Other query-based widgets are available from the Azure DevOps Marketplace.

Query filters used in notifications

Notifications support query filter criteria so that you can customize when you'll receive notifications. There may be limitations in what fields can be used to support notification filtering.



To learn more, see Manage your personal notifications

WIQL syntax, Wiql editor, and `.wiq` files

Managed queries use the work item query language (WIQL), a language similar to Structured Query Language (SQL). You can gain access to the WIQL syntax behind a managed query when you install the **Wiql Editor** Marketplace extensions.

This extension adds the following to your Azure Boards web portal:

- **Edit query wiql** (Work Item Query Language) to the **More commands** menu on the Query Editor and Query Results pages.
- **WIQL Playground** page to **Boards**.

From these interfaces you can exercise the following query features:

- View the underlying WIQL syntax of a managed query
- Run historical queries using the `ASOF` operator (`ASOF` operator won't be saved in a managed query)
- Specify a `MODE` clause for linked work items to further refine the query results (`MODE` clause won't be saved in a managed query)
- Move, copy, or paste query clauses
- Import and export queries as `.wiq` files from one project, organization, or collection to another
- Export WIQL for use in REST API calls.

You may find that it's easier to read queries with lots of clauses or heavily nested clauses from the WIQL format. For exceptionally large queries, the load times from a WIQL query may yield better load times.

WIQL syntax

WIQL has five parts shown in the following syntax snippet. The WIQL syntax is not case-sensitive.

```
SELECT [State], [Title]
FROM WorkItems
WHERE [Work Item Type] = 'User Story'
ORDER BY [State] Asc, [Changed Date] Desc
ASOF '6/15/2010'
```

To learn more about WIQL, see Syntax for the Work Item Query Language (WIQL).

> **NOTE**
> For queries made against Azure DevOps, the WIQL length must not exceed 32K characters. The system won't allow you to create or run queries that exceed that length.

`.wiq` file structure

The following syntax shows the structure of the `.wiq` file. For details on each `WorkItemQuery` syntax element, see Define a work item query to add to a process template

```
<WorkItemQuery Version="1">
      <TeamFoundationServer>collectionURL </TeamFoundationServer>
      <TeamProject>TeamProjectName </TeamProject>
      <Wiql>
      WorkItemQueryLanguage
      </Wiql>
</WorkItemQuery>
```

Extensions and manage queries

The following Azure DevOps Marketplace extensions work with managed queries to provide more functionality.

> **NOTE**
> Most Marketplace extensions are not supported features of Azure Boards and therefore not supported by the product team. For questions, suggestions, or issues you have when using these extensions, visit their corresponding extension page.

- Query Based Boards supports viewing a flat-list query of work items as a Kanban board. The query can contain different work item types and work items defined in different projects.
- **Query Tile PRO**: Adds the **Query Tile PRO** widget to the widget catalog for dashboards. This widget provides support for all query types (not just flat list queries) and provides more options to configure calculated values on the widget.
- **Wiql to OData**: Adds the **Translate to OData** option to the **More commands** menu on the Query Editor and Query Results pages. You can then use this query or augment it to retrieve the work items from the Analytics service. To learn more, see Query your work tracking data using OData Analytics.
- **Open in Power BI**: Adds the **Open in Power BI** option to the **More commands** menu on the Query Editor and Query Results pages. You can then use Power BI to generate reports based on the Analytics work tracking data. You can add these reports to an Azure DevOps dashboard. To learn more, see Query your work tracking data using OData Analytics.
- **Enhanced Export**: Lets you export work item queries or test plans to document-like formats. To get the output formatted the way you want, you can select different templates to get the form and layout of your choice. You can preview, print or even open the document directly in Office.

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Related articles

- About managed queries
- Syntax for the Work Item Query Language (WIQL)
- Work across projects FAQs

Import or update work items in bulk by using CSV files in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

You can import and export work items in bulk using a CSV formatted file. While you can continue to use Excel for bulk import and updates, you can use the native import/export feature that doesn't require Excel. To learn more about using Excel, see Bulk add or modify work items with Excel.

You can export of work items in bulk using a CSV formatted file. While you continue to use Excel for bulk import and updates, you can use the native export feature from Queries that doesn't require Excel. To learn more about using Excel, see Bulk add or modify work items with Excel.

> **NOTE**
> The export feature is available with Azure DevOps Server 2019 Update 1 and later versions. The import feature is available with Azure DevOps Server 2020 and Azure DevOps Services.

Import new work items

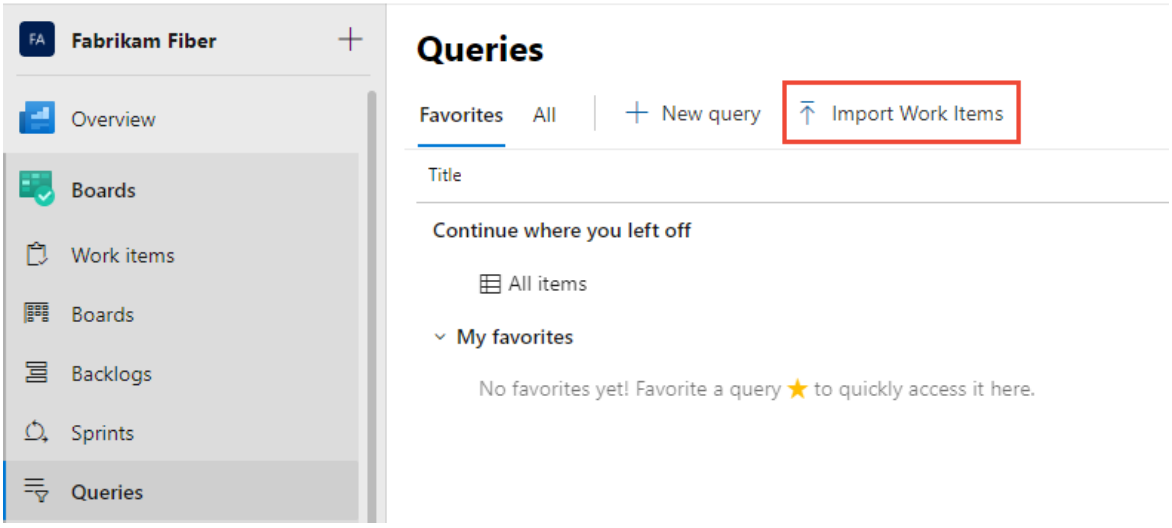
All work items you import are created in a new state. This rule means that you can't specify field values that don't meet the field rules for the new state. For example, when you create a new user story with the Agile process, the State=New and the Reason=New. No other values can be specified.

1. Create a local ***import.csv*** file and open it in Visual Studio Code or Excel.
2. The file must contain the **Work Item Type** and the **Title** fields. You can include other fields as needed. For a list of default fields, see Work item field index.

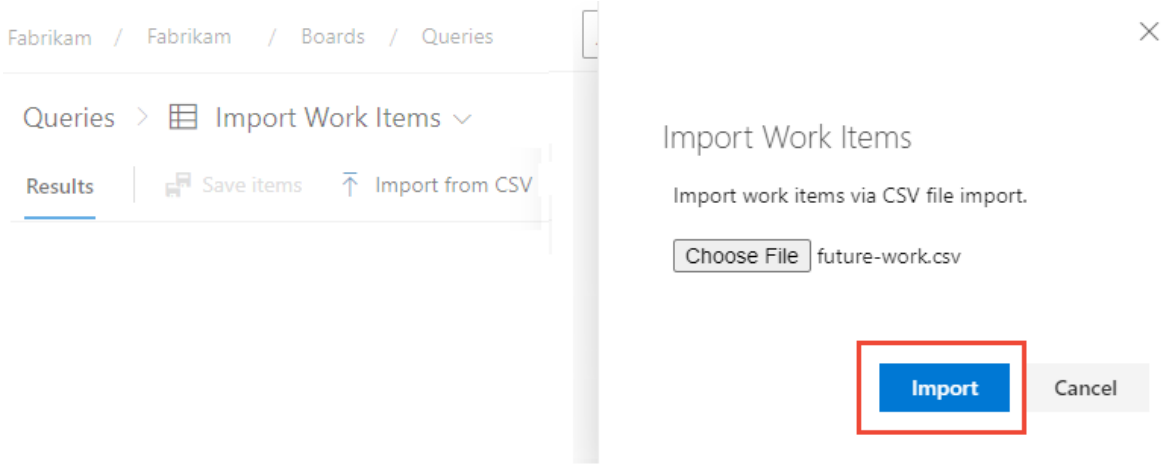
   In the following example, we include the **Priority** field.

   ```
   Work Item Type,Title,Priority
   Issue,Fix issues with code,1
   Issue,Merge testing modules,3
   Issue,Open private preview for select customers,2
   Issue,Enable feature for customer champs,2
   Issue,Remove old test code,2
   ```

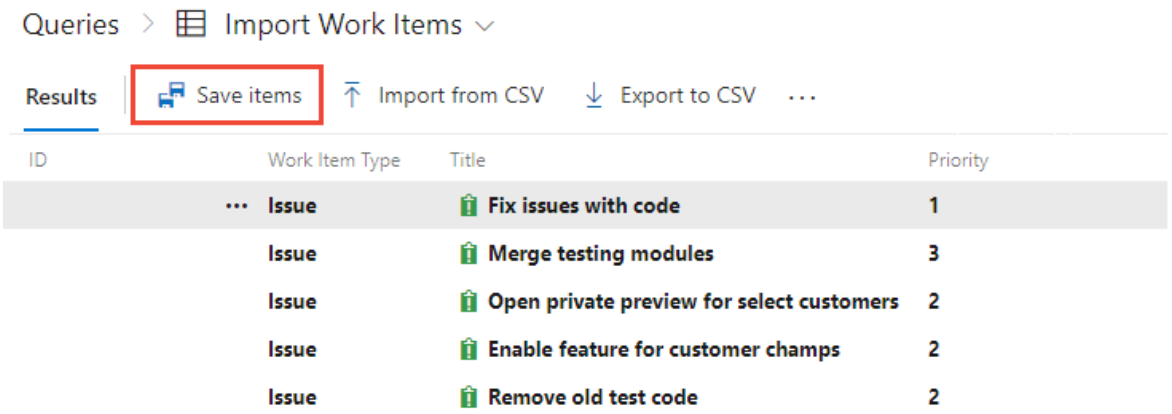
3. From the web portal for your project, open **Boards>Queries** and choose the **Import Work Items** option.



4. Select your CSV file and then choose **Import**.

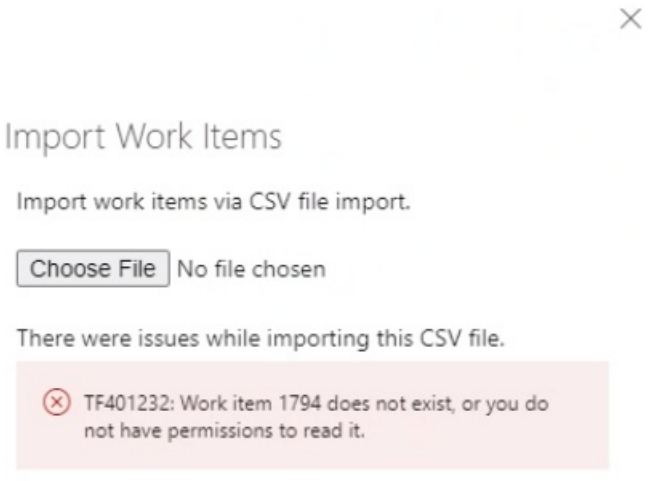


5. The import process loads the imported work items into the queries view in an **unsaved** state. No IDs are assigned. Verify the results are what you want. Then, choose **Save Items** to save the work items.

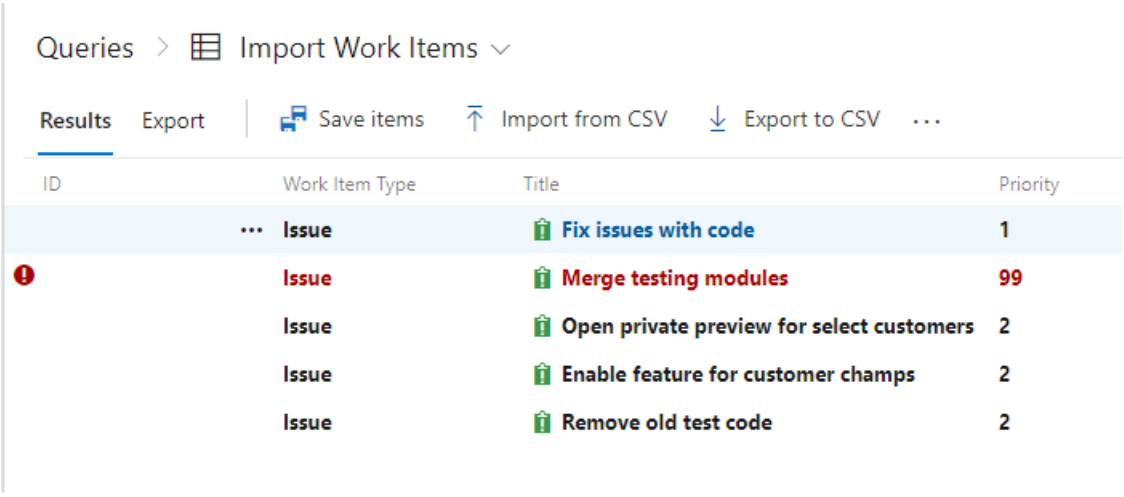


NOTE

Make sure you don't assign IDs to new work items that you're adding. You'll receive an error message similar to the following if you do so.



6. The system highlights those work items with data issues. Resolve the data issues before you save the work items. In this example, an invalid value has been entered into the Priority field. Fix the data by opening the work item directly. Instead, use bulk edit to fix several work items with the same issue.

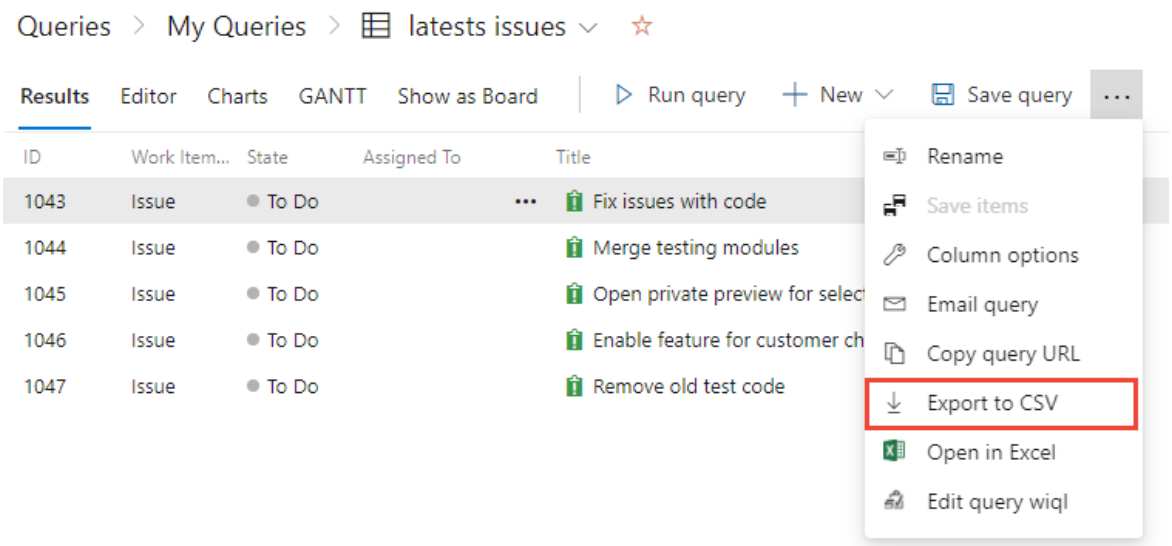


TIP

You can add parent-child links between work items you import by indenting the title columns as shown in the example later in this article, Can I import a CSV file that have parent-child links?. However, you can't specify any other link types when importing or updating work items.

Update existing work items

1. To update work items, create a query that contains all the columns you want to export and possibly edit. Save your query and select **Export to CSV** to save the data.csv file to your local machine.



The exported file should look similar to the following syntax:

```
ID,Work Item Type,State,Assigned To,Title,Tags
"1043","Issue","To Do",,"Fix issues with code",
"1044","Issue","To Do",,"Merge testing modules",
"1045","Issue","To Do",,"Open private preview for select customers",
"1046","Issue","To Do",,"Enable feature for customer champs",
"1047","Issue","To Do",,"Remove old test code",
```

2. Make the edits to your work items. Your CSV file must contain the **ID**, **Work Item Type**, **Title**, and **State** fields. Any other fields you want to include are optional.

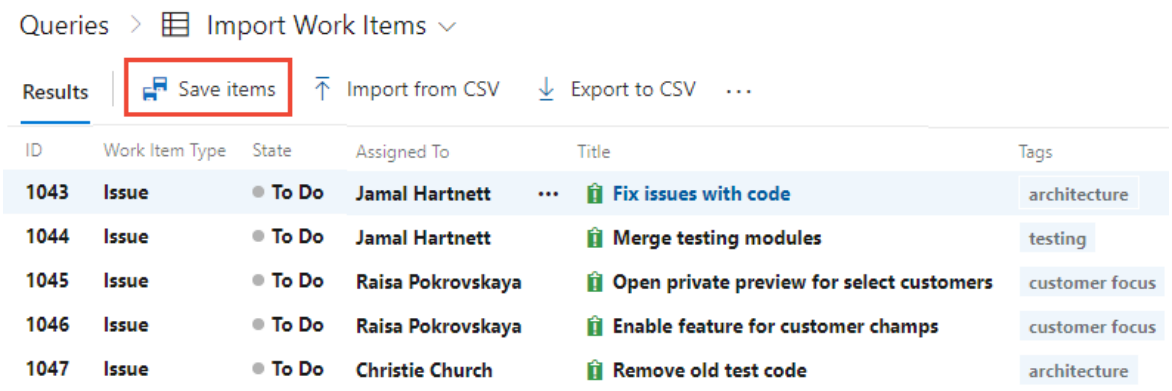
> **NOTE**
>
> When importing identity fields, the name and email must be entered in the following format `"Display Name <email>"`. For example, to assign work to Jamal Hartnett, specify `"Jamal Hartnett <fabrikamfiber4@hotmail.com>"`. If you specify a value that isn't recognized as a valid user to the system, you may encounter problems with the import.

In the following example, we change several values on existing working items.

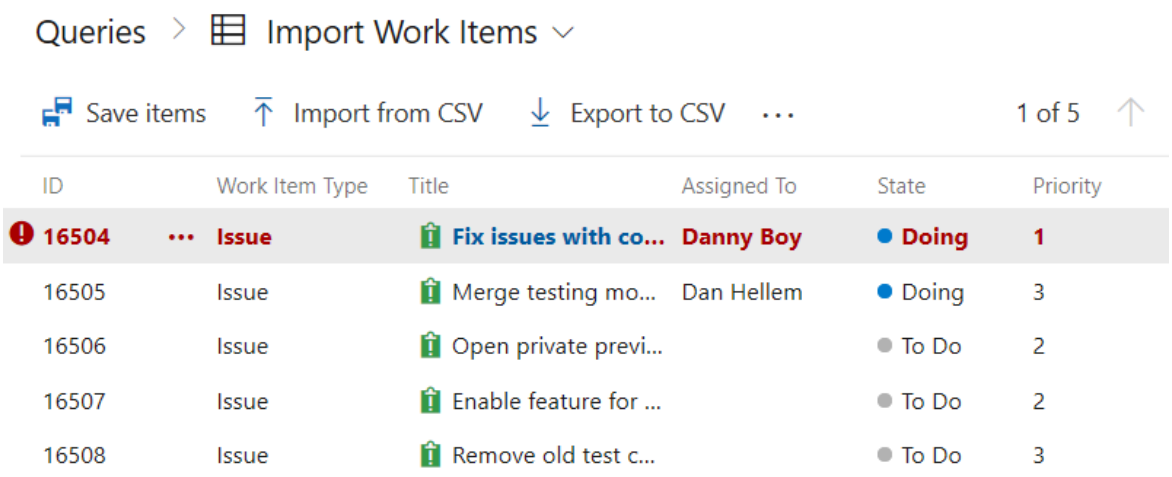
```
ID,Work Item Type,State,Assigned To,Title,Tags
"1043","Issue","To Do","Jamal Hartnett <fabrikamfiber4@hotmail.com>","Fix issues with
code",architecture
"1044","Issue","To Do","Jamal Hartnett <fabrikamfiber4@hotmail.com>","Merge testing modules",testing
"1045","Issue","To Do","Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>","Open private preview for
select customers","customer focus"
"1046","Issue","To Do","Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>","Enable feature for customer
champs","customer focus"
"1047","Issue","To Do","Christie Church <fabrikamfiber1@hotmail.com>","Remove old test
code",architecture
```

3. Save the file and import (see steps 4-6 from the previous import section.)

4. The results list with work items that contain value changes appear highlighted in bold. Choose **Save Items** to apply the changes.



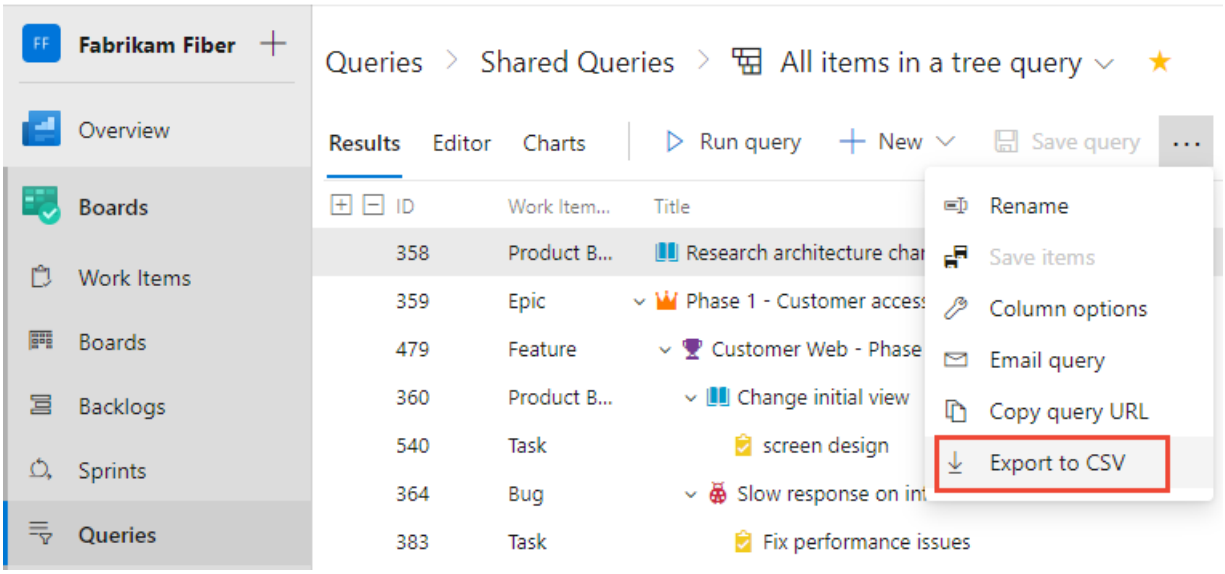
5. Work items with data issues are highlighted in red and need to be resolved before you can save them. In this example, an invalid value appears in the Assigned To field. Fix the data by opening the work item directly. Instead, you can use bulk edit if you have many work items with the same issue.



Export list as a CSV file

From any query, you can export a list of work items as a comma-delimited list. Open the query, choose the ··· actions icon, and choose **Export to CSV**.

> **NOTE**
> Requires Azure DevOps Server 2019 Update 1 or later version.



Export and import work items to a different project

You can use this feature to export work items from one project and import them to another project. However, before importing them to another project, you must remove the work item ID. You come across an error if you attempt to import new work items to a project with an ID specified.

Import or update rich-text fields

You can update or import rich-text fields such as the **Description** or **Acceptance Criteria** fields. Rich-text fields are HTML formatted fields. Replace lines ending in CRLF by surrounding sentences with `<p>... </p>`.

For example, you can import the following work item, which includes three lines of text in the Description field.

```
Work Item Type,Title,Description
"Product Backlog Item","Hello World Web Site - 8","<p><strong> You can include bold text</strong></p>
<p><em> And italic text</em></p><p><u> Underline text</u></p>"
```

Q & A

Can I import new items and update existing items in the same CSV file?

Absolutely! Leave the ID field empty for any new work items. In the following example, the last entry for an Epic doesn't specify an ID.

```
ID,Work Item Type,Title,Assigned To,State,Priority,Tags
"16504","Issue","Fix issues with code",,"To Do","1",
"16505","Issue","Merge testing modules",,"To Do","3",
"16506","Issue","Open private preview for select customers",,"To Do","2",
"16507","Issue","Enable feature for customer champs",,"To Do","2",
"16508","Issue","Remove old test code",,"To Do","2",
,"Epic","Track Telementry for data imports",,"To Do","2",
```

Can I import a CSV file that has parent-child links?

Yes, you can add child work items by indenting title columns. The following example adds three child Issues under the already defined Epic.

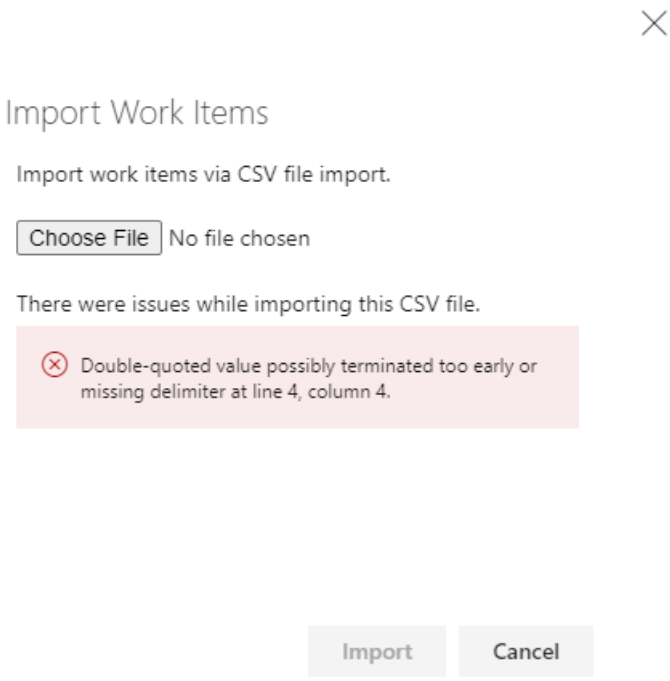
```
ID,Work Item Type,Title 1,Title 2,Assigned To,State,Priority,Tags
"165","Epic","Track Telementry for data imports",,,"To Do","2",
,"Issue",,"Fix issues with code",,"To Do","1",
,"Issue",,"Open private preview for select customers",,"To Do","2",
,"Issue",,"Enable feature for customer champs",,"To Do","2",
```

Here's a better visual in Excel.

	A	B	C	D	E	F	G	H
1	**ID**	**Work Item Type**	**Title 1**	**Title 2**	**Assigned To**	**State**	**Priority**	**Tags**
2	165	Epic	Track Telementry for data imports			To Do	2	
3		Issue		Fix issues with code		To Do	1	
4		Issue		Open private preview for select customers		To Do	2	
5		Issue		Enable feature for customer champs		To Do	2	
6								

How do I know if my imported file has errors?

Any problems with the formatting of your CSV file appear in the Results page of the import view. You can't import the work items until the formatting and syntax is correct.



The work items' results always lists the data errors found for individual work items. Fix each error either from the web portal, or in the CSV file and import again.

Related articles

- Work item field index
- Bulk add or modify work items with Excel
- FAQs: Work in Excel connected to Azure Boards

Add or modify Azure Boards work items in bulk with Microsoft Excel

10/4/2022 • 28 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you need to add or modify many work items, using Microsoft Excel can save you time. Excel supports adding work items, updating existing work items, adding links and attachments to multiple work items, and more. You can also use native Excel features to support other actions, such as summing a column, copy-and-paste rows, fill down data into cells, and more.

In this article you'll learn how to complete the following tasks:

- Choose the type of list or query to support your task
- Use select Excel features when connected to Azure Boards
- Import or update work items, either a flat list or hierarchy tree list
- Publish and refresh your work items
- Convert a flat-list to a tree-list, change your list type or query
- Add work items to your worksheet
- Add and remove work item fields from your worksheet
- Select user accounts for identity or person-named fields
- Link work items, find work items to link to, edit links, and more
- Add attachments to one or more work items
- Open a work item from Excel to add additional information (opens in web portal)
- Edit Area and Iteration Paths (opens in web portal)

For information about connecting to Excel, see Connect Azure Boards to an Office client. For answers to specific questions about the integration of Excel and Azure DevOps, see FAQs: Work in Excel connected to Azure Boards .

> **NOTE**
> If you don't have access to Excel, you can still perform bulk import and update using CSV formatted files. To learn more, see Bulk import or update work items using CSV files.

Prerequisites

> **NOTE**
> macOS isn't supported. Even if you've installed Visual Studio for Mac, connection to Azure DevOps from Excel isn't supported.

- Installed Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Installed Azure DevOps Office Integration 2019 (free).

> **NOTE**
> The only way to get the Azure DevOps Office Integration plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Office Integration installer. The plug-in supports connection to Azure Boards and Azure DevOps Server from Excel.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To view or modify work items, you must have these permissions set to **Allow**: **View work items in this node** and **Edit work items in this node**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To use the Select User feature, you need to install Visual Studio (at least VS 2015.1 or later version or Team Foundation Server Office Integration 2015 Update 2 or later version. You can download the free version of Visual Studio Community. Get this feature to avoid data validation errors by misspelling user names and when you must assign user names from a large group of user accounts.

- Installed Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Installed Azure DevOps Office Integration 2019 (free).

> **NOTE**
> The only way to get the plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Standalone Office Integration installer. The Azure DevOps Office Integration 2019 plug-in supports connection to Azure Boards and Azure DevOps from Excel, Project, and the PowerPoint-based storyboarding tool.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To view or modify work items, you must have these permissions set to **Allow**: **View work items in this node** and **Edit work items in this node**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To use the Select User feature, install Visual Studio (at least VS 2015.1 or later version or Team Foundation Server Office Integration 2015 Update 2 or later version. You can download the free version of Visual Studio Community. Get this feature to avoid data validation errors by misspelling user names and when you must assign user names from a large group of user accounts.

To learn more about compatibility requirements, see Compatibility with Azure DevOps Server.

Choose list and query type

When you work in Excel connected to Azure Boards, you're always working with a query type and a list type. Queries correspond to the queries you create using the Query Editor.

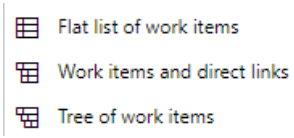
- **Query types**:
  - **None**: Indicates it's an **input list**.
  - **Query title**: Indicates the list of work items is tied to the specified **query**.
- **List types**:
  - **Flat list**: Simple list of work items that shows a single **Title** column. No link management is supported.

- **Tree list**: Hierarchical list of work items that support creating and updating tree topology links, such as Parent-Child links, between work items. These lists include two or more **Title** columns.

You can add, modify, publish, and refresh work items using any query type and list type.

Query types

Azure Boards supports three query types. The icon next to each query indicates the query type. The first two query types, **Flat list of work items** and **Work items and direct links** are imported as flat list queries.

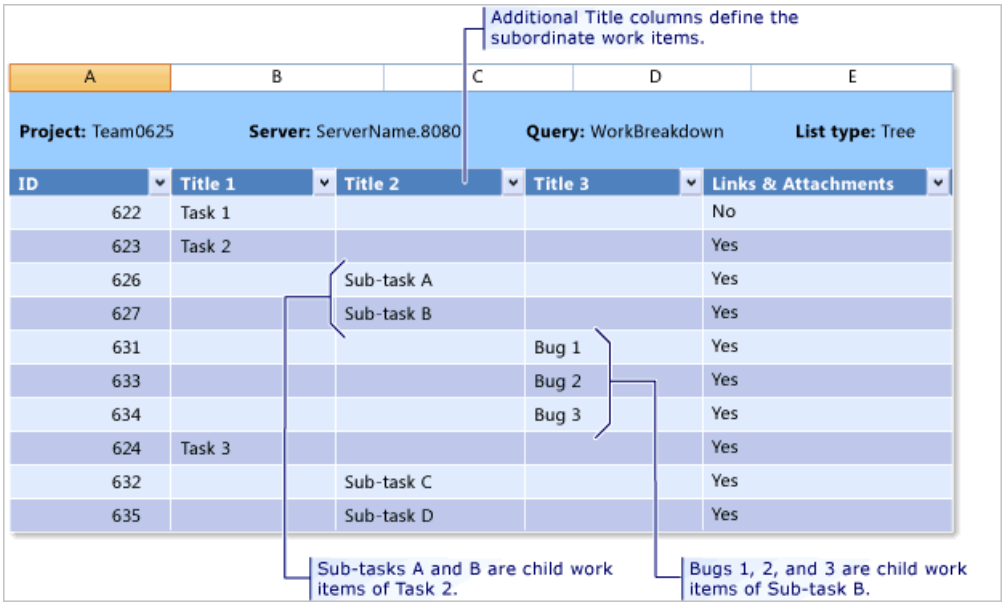


Only the **Tree of work items** queries import as a tree list. Direct links queries are imported as a flat list into Excel as modifying multiple types of links aren't a supported feature in Excel.

Tree lists

You can bulk add a nested list of work items, such as a work breakdown structure or a hierarchical set of user stories and customer experiences. For example, you can add a nested list of tasks, subtasks, and bugs, as shown in the following illustration, or linked tasks to product backlog items.

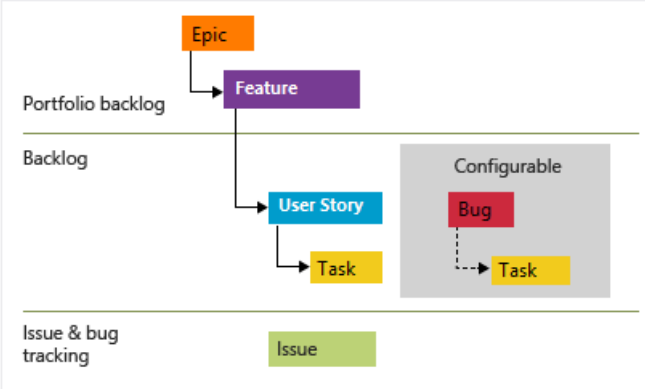
Here's how a three-level nested tree of items appears in Excel.



Parent-child links or other tree topology link types support creating a hierarchical backlog structure. The work item types that participate in the hierarchy differ with different processes and are shown in the following images.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

To import a hierarchical list, see Add or import a hierarchical list of work items later in this article.

My queries versus shared queries

You can open in Excel any query you've defined in Azure Boards. That includes queries defined under My Queries and Shared Queries. However, if you plan to share the workbook with other team members, then you should only work with a Shared Query. Other team members can't use the workbook or worksheet if it's based on a personal query stored under your My Queries folder.

Use list and query types

In general, you Use a flat list to bulk add or modify several types of work items at once, such as backlog items, tasks, bugs, or issues. Use a tree list to bulk add or modify work items and their tree-topology links.

Here's some more guidance:

- Use an **input list, flat list**: To import a list of work items or create new work items, no hierarchy
- Use an **input list, tree list**: To complete top down planning and import hierarchically linked work items
- Use a **query list, tree list**: To view and modify the hierarchy of link relationships of many existing work items.
- Use a **query list, flat list**: To bulk update a list of work items or create new work items, no hierarchy

- Use an input list, flat list: To import a list of work items or create new work items, no hierarchy
- Use an input list, tree list: To complete top down planning and publish parent-child linked work items
- Use a query list, flat list: To create an Excel report based on the query of work items

> **NOTE**
> To create an Excel report, you're project collection must be configured to support Analytics reporting. For more information, see Create Excel reports from a work item query.

- Use a query list, tree list: To view and modify the hierarchy and parent-child link relationships of many existing work items.

Use Excel features

You can use most Excel features when working with a list of work items. For example, you can use the following features:

- Format a cell or apply conditional formatting to a cell or column
- Cut and paste from one cell to other cells
- Cut and paste a single row

- Sum a column or add other formulas
- Fill down cells
- Filter
- Add multiple worksheets to your workbook

Each worksheet in Excel can contain a different input list or query. However, all worksheets within the workbook must connect to the same project within an organization or project collection.

The following features work slightly differently when working with a worksheet connected to Azure Boards.

- Each cell or column of cells corresponds to a work item field. Each field is associated with a data type. Excel won't allow you to enter data into a cell that doesn't meet the data type and requirements for that field.
- You can only insert a single row at a time within the worksheet.
- You can copy and paste multiple rows within the worksheet.
- To move a work item within a hierarchy, cut the entire row and paste it under the work item you want as its parent.
- Use **Outdent** and **Indent** to change the location of a work item within the tree.
- Undo (Ctrl Z) may not work. If you do something that you want to revert, you can refresh the worksheet.

We recommend you publish and refresh your worksheet often to make sure your local work remains in sync with Azure Boards data store.

To learn more about Excel, see Basic Excel tasks .

Sort work items

You can sort work item flat lists using the Excel sort feature.

However, if you're working from a tree list, you don't want to do any type of sort. Doing so changes the tree structure and as such, the links between work items.

If you want to use Excel to manage the order of your work items as they appear in a team backlog, you can do that by using the Stack Rank or Backlog Priority field (Agile or Scrum process). You can set values in these fields, publish your worksheet, and refresh your backlog. Your backlog items should appear reordered based on lowest to highest number. However, the next time the backlog is reordered from the backlog, the values you entered are subject to change.

If you want to maintain a certain order of work items, consider adding a custom field to manage the sort order, and then use that within Excel to sort your flat list of work items. This option, however, won't change the order that appears in your backlog.

Tasks you can and can't do with Excel

You can do the following tasks:

- Add tags and bulk update work items with tags as described in Add work item tags to categorize and filter lists and boards. Add the **Tags** field to your worksheet. Add multiple tags separated by a semicolon (;).
- You can add simple text to a rich-text field, but if you're updating several work items in bulk, you may lose formatting in existing work items.
- You can work offline and then reconnect and publish your changes. For details, see Connect Azure Boards to an Office client, Work offline, and reconnect.

You can't do the following tasks from an Excel worksheet:

- You can't delete work items
- You can't change the work item type of an existing work item
- You can't move work items to another project

- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread
- You can't link to a remote work item.

- You can't delete work items
- You can't change the work item type of an existing work item
- You can't move work items to another project
- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread.

- You can't delete work items
- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread

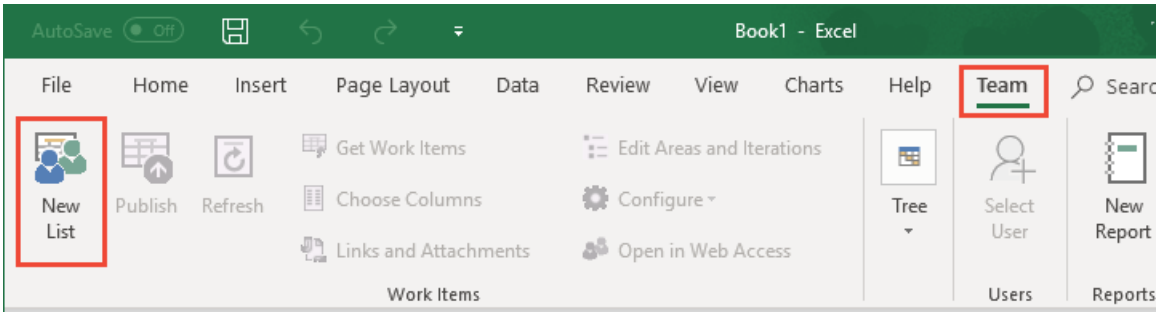
Import work items as a flat list

1. Open Excel and connect to your Azure Boards project. Use one of the four methods provided in Connect Azure DevOps project to Excel.

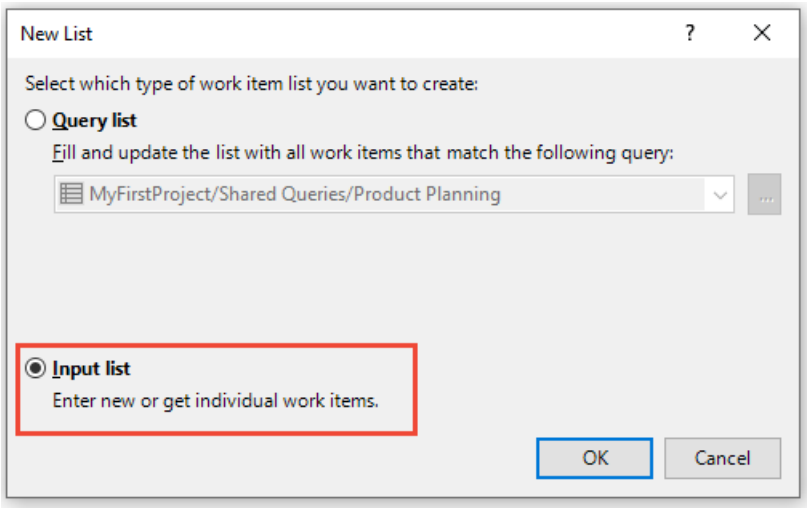
   > **NOTE**
   > When you connect to Azure Boards in the cloud, the **Team Project Collection** is automatically selected as there is only one collection associated with your Azure DevOps Services organization. When you connect to Azure Boards in an on-premises server, you choose the **Team Project Collection** prior to choosing the project.

2. In Excel, start with a blank worksheet. If you don't see the **Team** ribbon (or the **Team** menu if you use Excel 2007), see Azure DevOps Office integration issues.

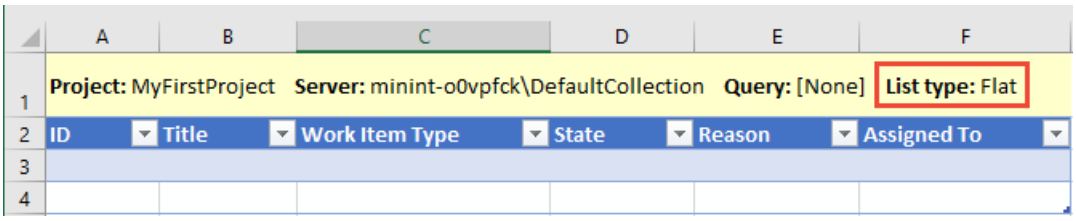
3. Choose **New List** from the Team ribbon.



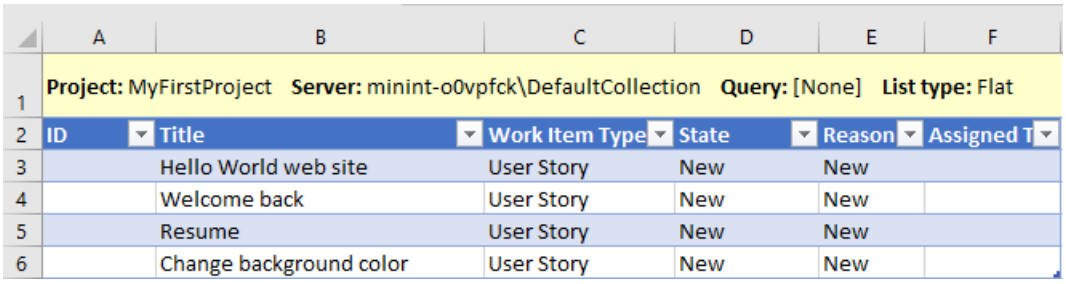
4. From the **New List** dialog, choose **Input list**.



5. Your worksheet is now bound to your project as an input list (Query[None]), flat list.

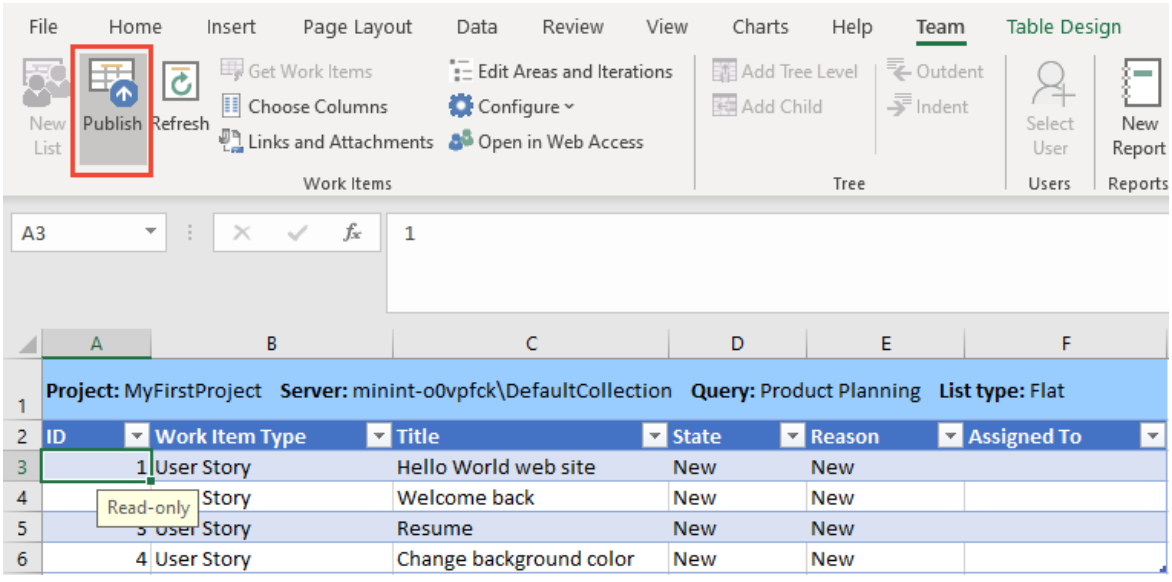


6. Specify the titles of the work items you want to add and their work item type.



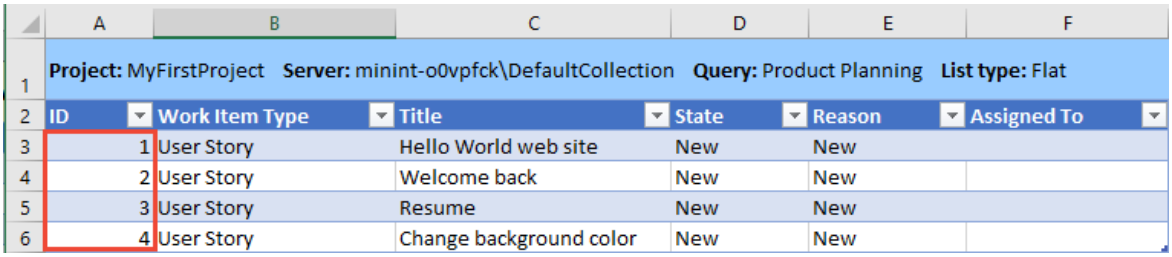
Notice how the **State** and **Reason** fields automatically fill in with default values once your select the work item type.

7. Publish your worksheet.



Make sure your cursor is in a cell that contains data. Otherwise, the **Publish** button might appear disabled.

Notice how IDs are now assigned to your work items.

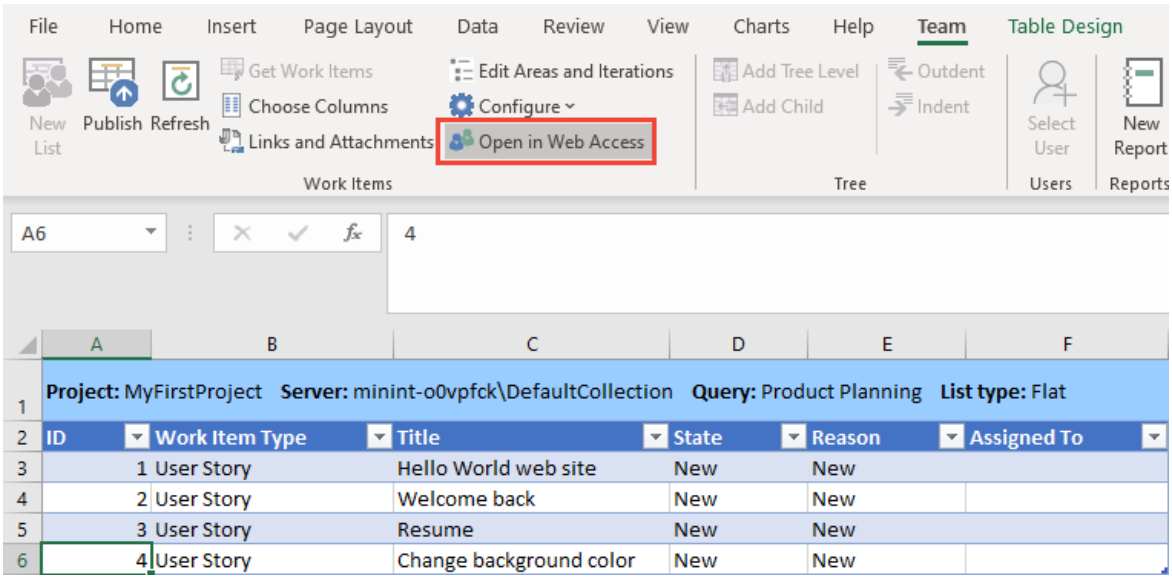


	A	B	C	D	E	F
1	**Project:** MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** Product Planning **List type:** Flat					
2	**ID**	**Work Item Type**	**Title**	**State**	**Reason**	**Assigned To**
3	1	User Story	Hello World web site	New	New	
4	2	User Story	Welcome back	New	New	
5	3	User Story	Resume	New	New	
6	4	User Story	Change background color	New	New	

8. To assign values to other fields, open **Choose Columns**, add the fields, make the assignments, and publish your changes.

> **TIP**
>
> If you're adding work items that you want to appear on a team backlog, make sure that you add and specify the team's Area Path and Iteration Path. If you need to add Area Paths or Iteration Paths, choose the **Edit Areas and Iterations** link. The link opens a web browser to the Project Settings page. To learn more, see Define area paths and assign to a team and Define Iteration Paths and configure team iterations.

9. To open a work item to add more information, Choose the work item you want to open and then choose **Open in Web Access**. Before you do, make sure you publish any changes you've made.



	A	B	C	D	E	F
1	**Project:** MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** Product Planning **List type:** Flat					
2	**ID**	**Work Item Type**	**Title**	**State**	**Reason**	**Assigned To**
3	1	User Story	Hello World web site	New	New	
4	2	User Story	Welcome back	New	New	
5	3	User Story	Resume	New	New	
6	4	User Story	Change background color	New	New	

A web browser opens and displays the work item.

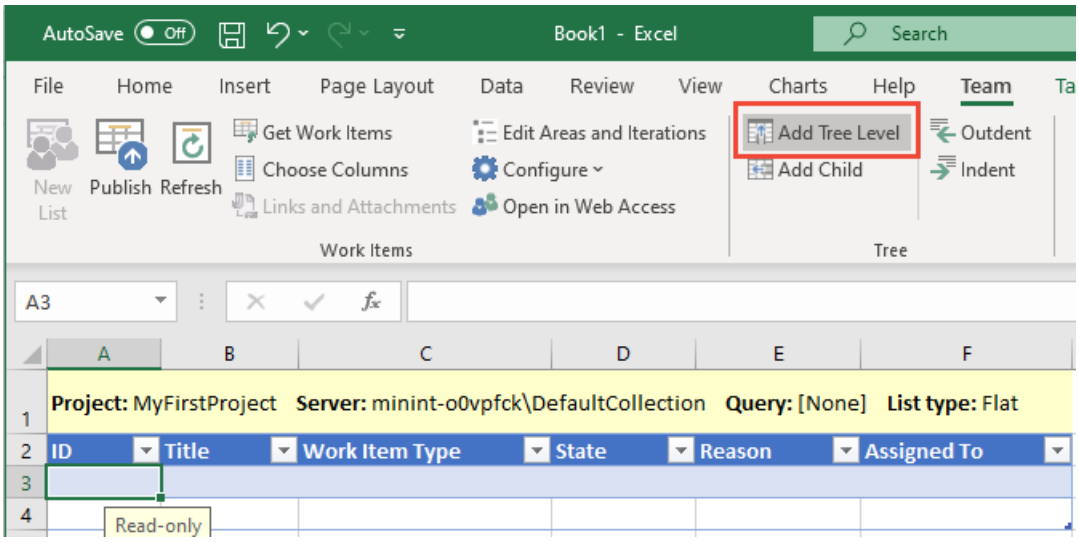
If you make changes to the work item, you should then immediately refresh your worksheet to capture the changes.

Import work items as a tree list

You can add a hierarchy of work items linked using parent-child links or other tree topology link type.

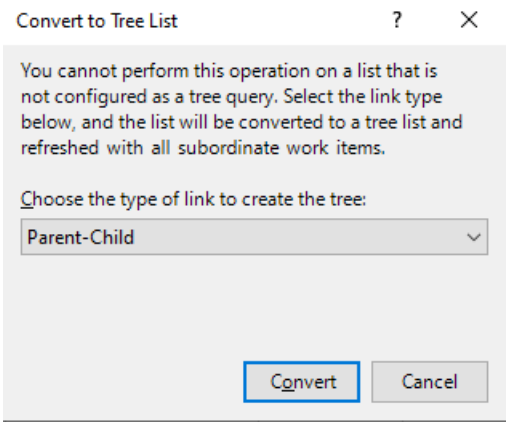
> **IMPORTANT**
>
> Don't sort a tree list. Sorting a tree list can change the hierarchical link relationships.

1. Starting from Step 5 from the previous procedure, convert your flat list, input list into a tree list. Choose a cell within the flat list and then choose **Add Tree Level**.

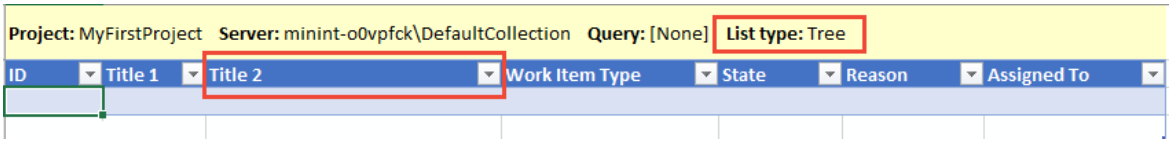


If the **Add Tree Level** is disabled, you're working from a query list. To convert your list to a tree list, you must first reconfigure your list to an input list.

2. Choose the link type to use when adding work items to a hierarchy, and then choose **Convert**. The most usual choice is **Parent-Child**. You can only select from tree topology link types. To learn more, see Link type topologies and restrictions.



Note the List type has changed to **Tree**, and a second **Title** column appears.



3. To add more levels to the hierarchy, choose **Add Tree Level** again. For example, if you want to add a hierarchy of Epics, Features, and User Stories, you'll want to have **Title 1**, **Title 2**, and **Title 3** columns.

   If you want to add tasks, add another tree level to have four title columns. To remove a column, see Remove a tree level.

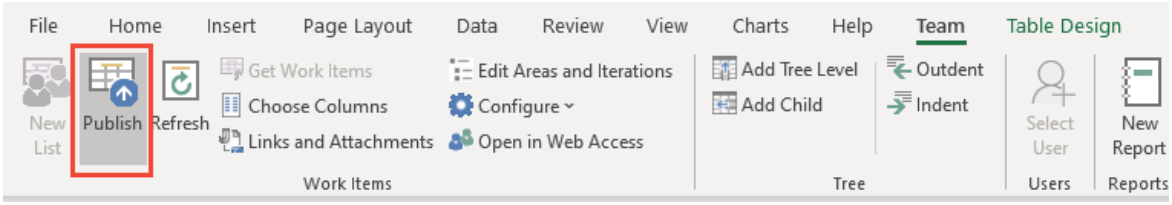
4. Save your Excel file.

5. Enter the **Work Item Type** and **Titles** for the hierarchy you want to import. The **State** fields automatically fill in with default values once you select the work item type.

Project: MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** [None] **List type:** Tree

ID	Work Item Ty	Stat	Title 1	Title 2	Title 3
	Epic	New	Improve User Experience		
	Feature	New		Integrate client application with popular email clients	
	User Story	New			Implement a factory which abstracts the email client
	User Story	New			As a user, I can select a number of support cases and use cases
	User Story	New			Customer request support for GMail
	User Story	New			Customer request support for Yahoo mail
	User Story	New			Customer request support for Roundup
	Feature	New		Integrate client app with IM clients	
	Feature	New		Emoticon feedback enabled in client application	
	User Story	New			As a user, I can select an emoticon and add a short description
	User Story	New			Add animated emoticons
	Epic	New	Framework to port applications to all devices		
	Feature	New		Migrate legacy code to portable frameworks	
	User Story	New			Implement a factory that migrates legacy to portable frameworks
	User Story	New			As a developer, I can analyze a code base to determine compliance with
	Epic	New	Counter the Heartbleed web security bug		
	Feature	New		Scan all software for the Open SLL cryptographic code	
	User Story	New			Scan all code base and identify the affected code
	Feature	New		Update software to resolve the Open SLL cryptographic code	
	User Story	New			Update and retest suite code base affected by the vulnerability
	User Story	New			Update and re-test service code based affected by the vulnerability
	Epic	New	Shared Personalization and state		
	Epic	New	Service interfaces to support REST API		
	Epic	New	Convert legacy Odata service interfactes to REST API		
	Feature	New		Convert all services from using experiemental code	
	Feature	New		Convert all client service calls from using experimental code	

6. Publish your worksheet.

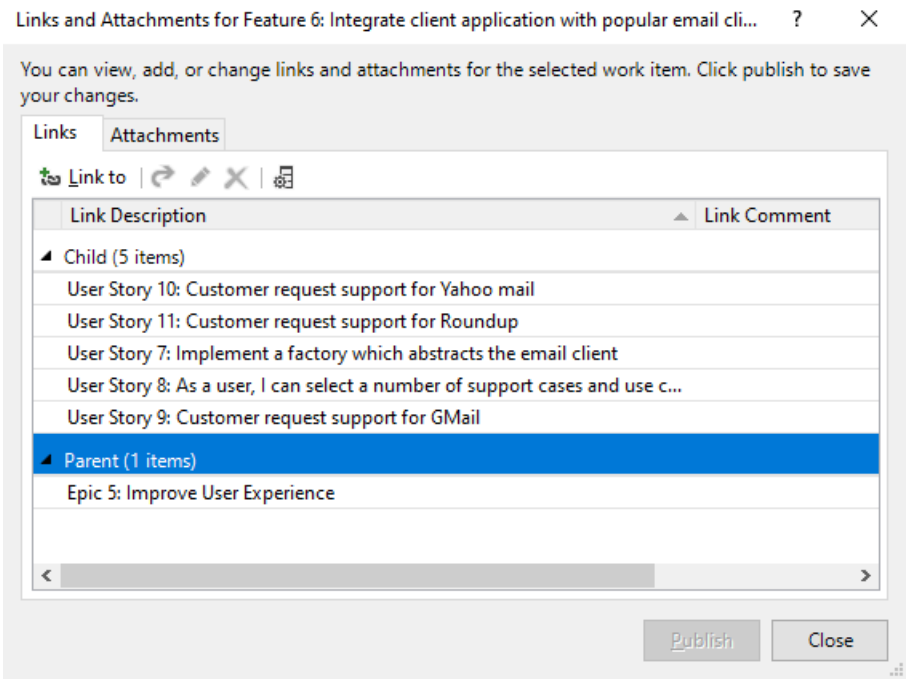


Make sure your cursor is in a cell that contains data. Otherwise, the **Publish** button might appear disabled.

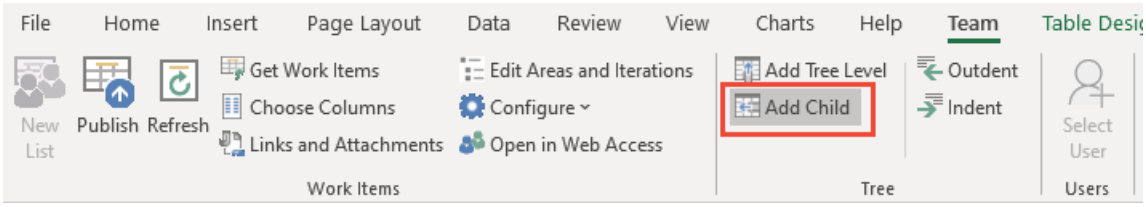
IDs are now assigned to your work items. In the background, the link type you selected is used to link each work item in the hierarchy. Epics are linked to Features, Features are linked to User Stories.

7. To check the links made, choose a work item and choose **Links and Attachments**.

For example, here we show the Child and Parent links created for a Feature that was imported.

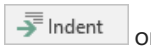


8. To enter a row under a work item where you want to add a child, choose the row and then choose **Add Child**.



9. To assign values to other fields, open Choose Columns, add the fields, make the assignments, and publish your changes.

10. To change the hierarchy, cut and paste the row of a work item to place it under the new parent. Make sure that you select the entire table row. When you publish the change, the old hierarchical links are deleted and the new hierarchical link are created.

    You can use the 

or 

indent/outdent icons to demote or promote a work item within the tree hierarchy. Verify that the column to the left or right of the parent work item's title is a **Title** column. The header at the top of the column should read **Title n**, if it doesn't, add a tree level.

Remove a tree level

1. First, publish changes that you've made to work items before you remove a tree level. Removing a tree level requires a refresh, which overwrites data in the work item list. You'll lose any data you haven't published.

2. Next, delete any content under the tree-level **Title** *number* column you want to remove—the highest numbered column—. This column should be the highest numbered column in the tree.

3. **Refresh** your worksheet. The column containing empty values for the Title is removed.

   You'll receive an error message if you attempt to delete the column.

Useful tips when working with a tree list

- Excel interprets the data in the **Title** columns to determine the pattern of links between work items. When you publish changes, any of the following conditions can result in an error, an invalid link, or a tree link to be created between incorrect work items:
  - A row between the work items is blank within the hierarchy

- The Title of a work item is in the wrong column. Make sure you enter a title for each child work item.
- Within a row, multiple **Title** columns contain data. Enter text in only one of the title columns within each row.
- The list was sorted. Don't sort a tree list. Sorting a tree list can change the hierarchical link relationships. If you do sort a tree list, you can recover from this operation by immediately refreshing.

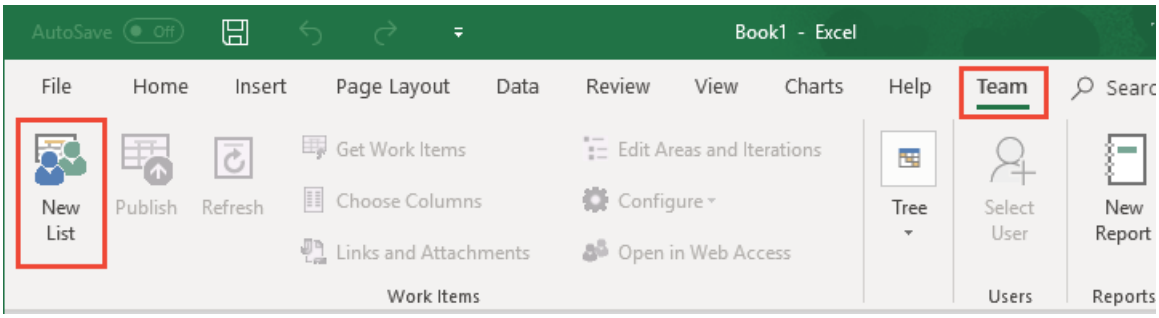
- To resolve an error, see Resolve invalid links.
- A parent-child linked work item can only have one parent. You can't add the same work item task to two backlog items. Instead, you need to define distinct work item tasks.

Update work items in bulk with a query list

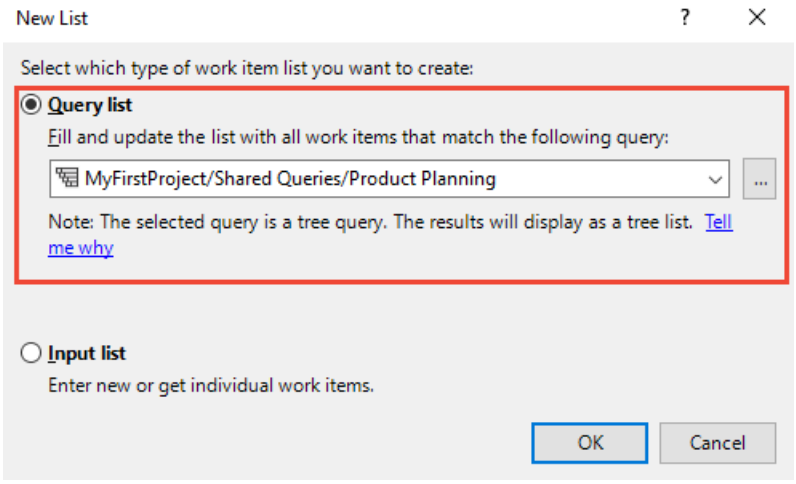
The easiest way to bulk update many work items is to create a query with the work items you want to update, and then open that query in Excel.

> **TIP**
> **Follow these tips to keep your work in sync:**
>
> - When you first open a saved worksheet, use (**Refresh**) to download the latest data from the data store.
> - Enter data for additional fields by adding columns to the worksheet using **Choose Columns**.
> - To avoid data conflicts, publish your additions and modifications often.
> - To prevent loss of data before you publish or refresh, save your workbook periodically.

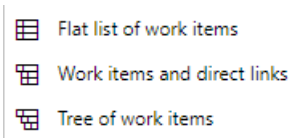
1. From the web portal or Visual Studio, create the work item query that contains the work items you want to update. For details, see Create and save managed queries with the query editor.
2. Open Excel and connect to your Azure Boards project. Use one of the four methods provided in Connect Azure DevOps project to Excel.
3. If you opened your query from the web portal or Visual Studio, you're done. Make any changes you want. Open **Choose Columns**, add fields, make assignments, and publish your changes.
4. If you start from Excel, open a blank worksheet. You can add a worksheet to an existing workbook, as long as you're choosing a query from the same project the workbook is bound to.
5. Choose **New List** from the Team ribbon.



6. From the **New List** dialog, choose **Query list**, and select the query you want from the drop-down menu.



The icon next to each query indicates the query type. The first two query types, **Flat list of work items** and **Work items and direct links** are imported as flat list queries. Only the **Tree of work items** queries import as a tree list.



7. With the work items imported to Excel, make the modifications you want and publish your changes.

   If you're working with a tree list, see also the information provided in Import a hierarchical list of work items.

Enable Tree commands

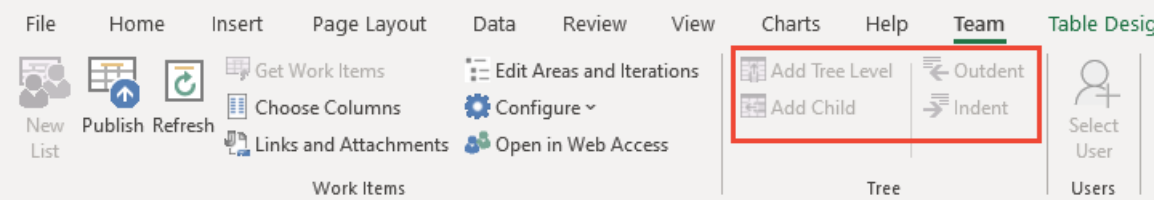
If the **Tree group** commands aren't available, your worksheet is configured as a flat list, query list. Convert the list to either an input list or a list based on a tree query to enable the Tree group commands. To learn how, see the next section on Change your list type or query.

Change your list type or query

You can change the work items listed in your worksheet. Specifically, you can:

- Change your flat list to a tree list
- Change from a query list to an input list
- Change from an input list to a query list
- Change the query your worksheet references

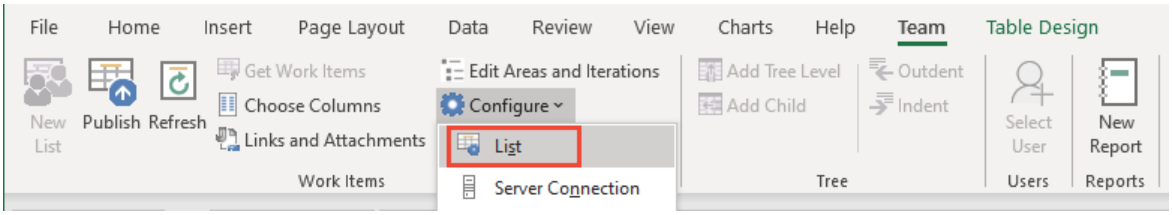
If you want to change your flat list to a tree list, you can. However, if your list is a query list, then you first need to reconfigure the list. You'll know that it's a flat list, query list as the **Tree group** commands are disabled.



To convert your query list to an input list, follow these steps.

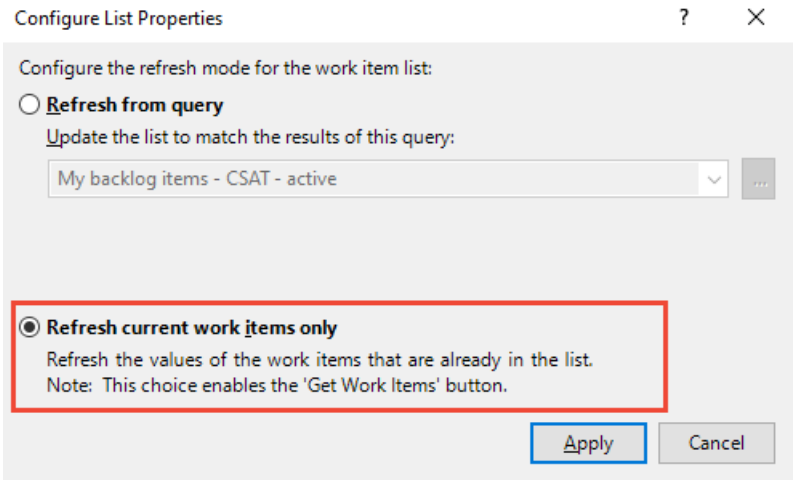
1. First, publish whatever changes you have made.

2. Next, on the **Team** ribbon, choose **Configure**, **List**.

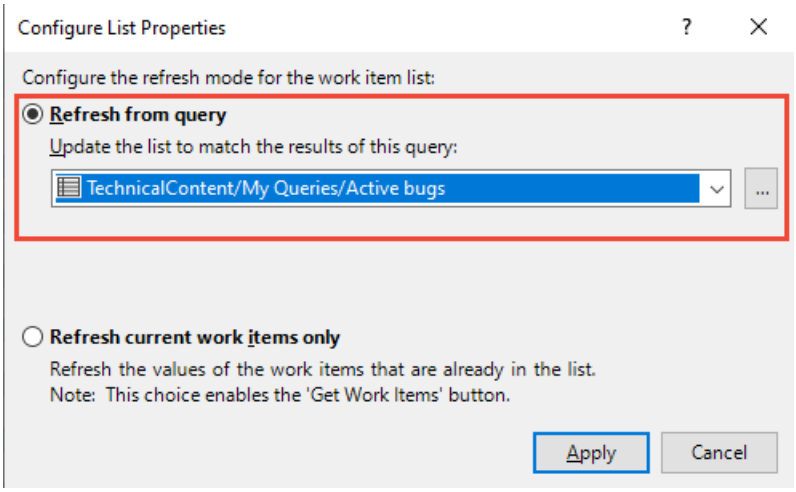


3. Choose **Refresh work items only** and then **Apply**.

   This choice changes the query list to an input list.



4. To convert from an input list to a query list, choose **Refresh from query**, select the query, and then **Apply**.

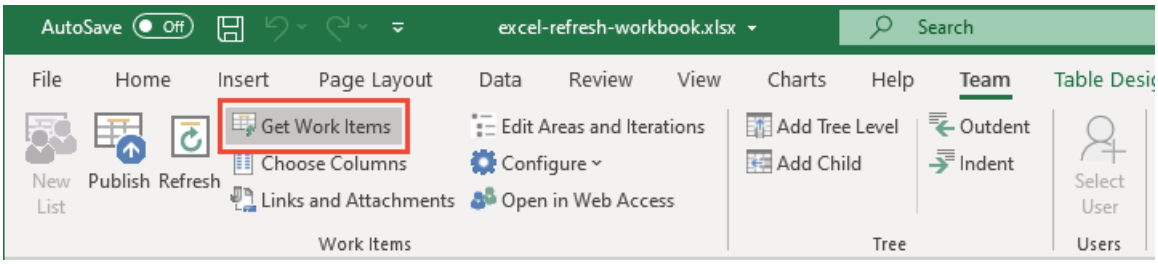


Add existing work items to your worksheet

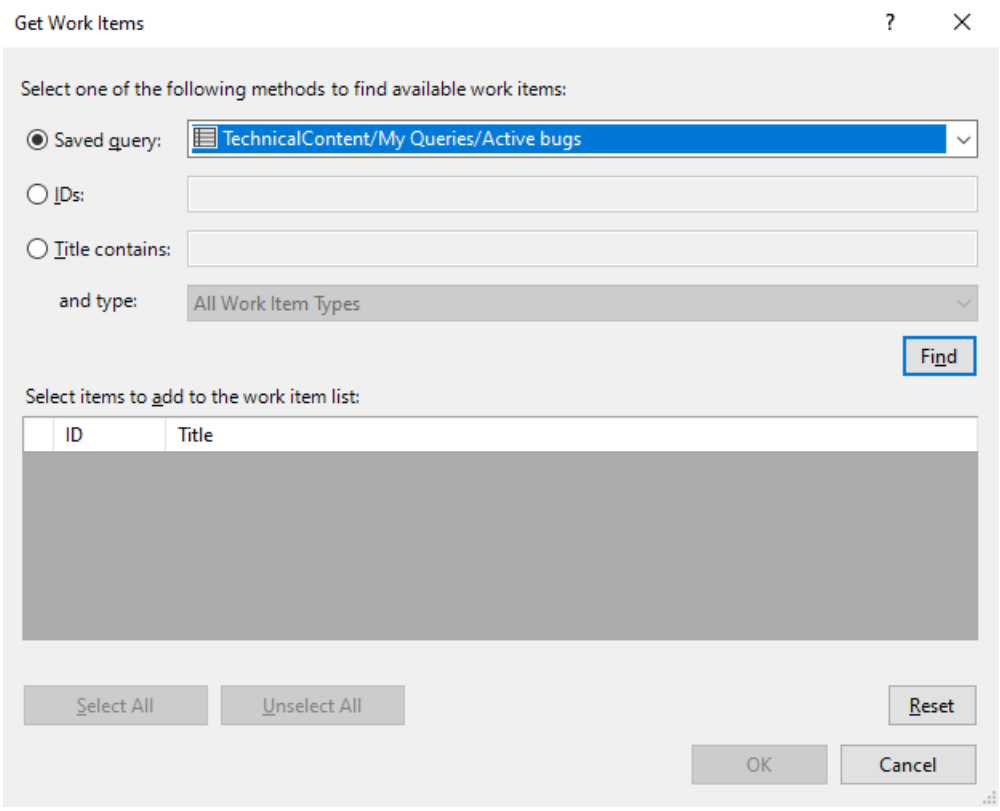
If you're working from a query, modify your query to contain the work items you want. Then refresh your list. The other work items will appear in your list.

If you're working with an input list, complete these steps.

1. From the **Team** ribbon, choose **Get Work Items**.



2. Choose the method you want from the three options available.



If the work items are defined in another project, then first select the Project. Then, make your selections:

- **Query**. Use this method when you've defined a query that contains the set or superset of work items you want.
- **IDs**. Use this method when you know the IDs of the work items that you want to link to. In the **IDs** box, type the IDs of the work items that you want to find, separated by commas or spaces.
- **Title contains**. Use this method to find work items that have a common word or phrase in the title field. In the **and type** list, select the type of work item that you want to retrieve.

> **NOTE**
> To minimize the time required to run the query, narrow the filter criteria of the search.

3. Choose **Find**.

   Only those work items defined for the selected project and specified work item type are listed. To sort on a column field, choose the column **Title**.

4. In the list of returned work items, select the check-box of one or more work items.

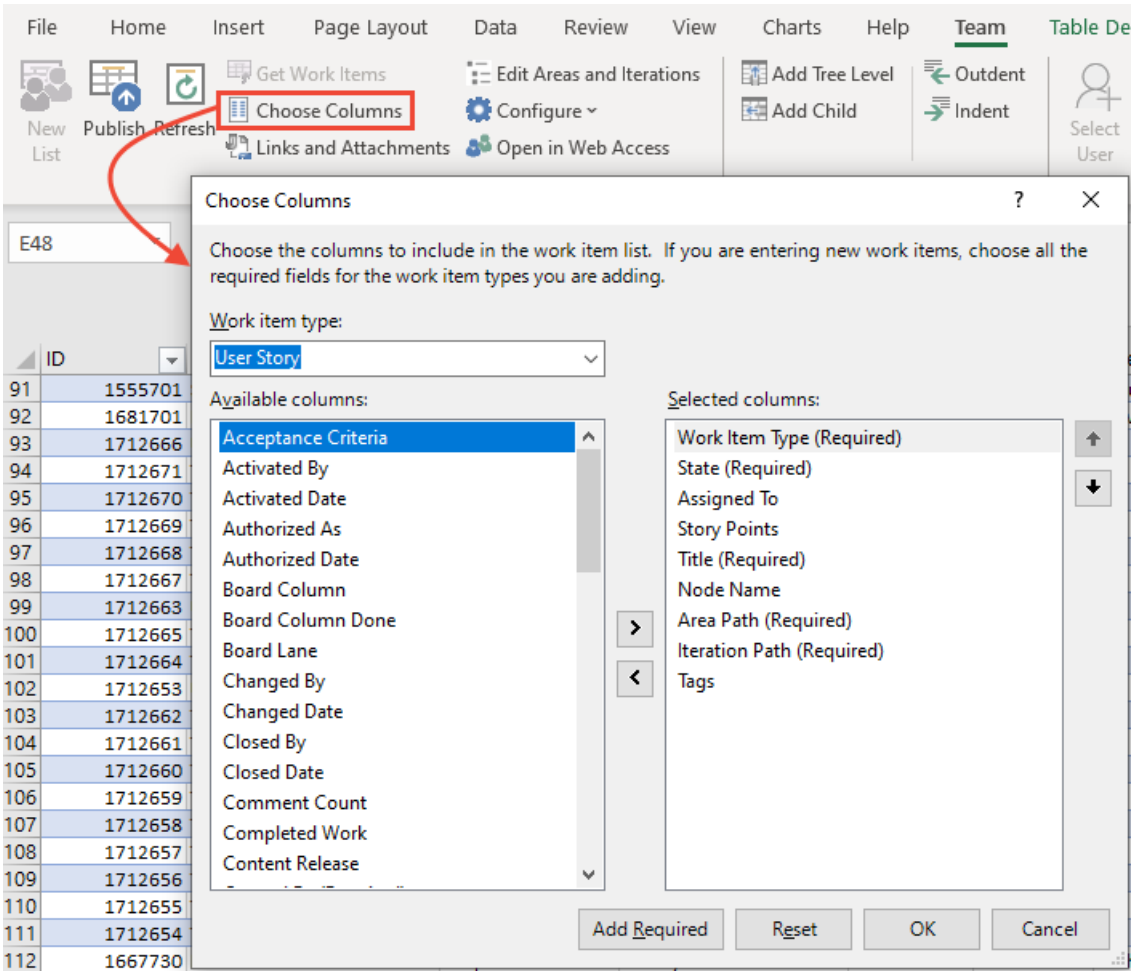
   - Select each work item that should link to the current work item. You can also press the SHIFT key while clicking to select a range of work items, or press the CTRL key while clicking to select multiple work items.
   - Choose **Select All** to select all work items in the list.

Add or remove column fields

If you start your worksheet with a **New List**, you'll see only a set of default field columns. You can add columns using the **Choose Columns** on the Team ribbon.

If you start your worksheet from an existing query, you'll see all the column fields defined for the query. From there, you can add columns using the **Choose Columns**. However, your additions don't modify the underlying query.

1. To assign values to other fields, choose **Column Options** to add the fields of interest.



   - To filter the fields based on work item type, select the **Work item type**.
   - To move or remove a field, choose the field and then select the > or < icons.
   - To change the field sequence, move the field up or down in the list using the up and down arrows.
   - You can add a rich-text field, such as the **Description** field, however you may lose some of the formatting upon publish.
2. Once the fields appear in the worksheet, assign values and publish your updates. When working with identity fields, ones that accept user accounts, see the next section, Select user accounts.
3. Save your worksheet.

Select user accounts

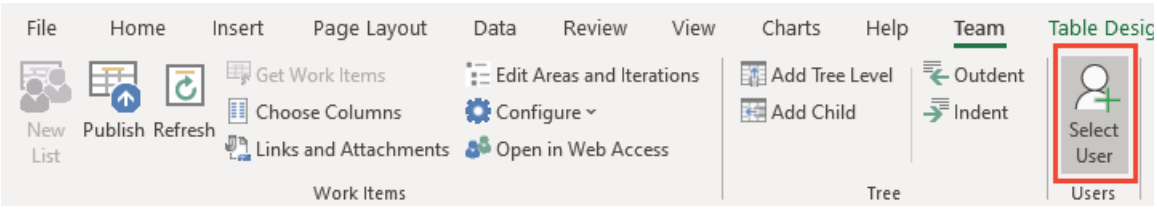
You can use the Select User feature to find user accounts and assign values to person named fields. Also, this feature provides access to the most recently used (MRU) values. If your team contains several hundreds or thousands of user accounts, you'll want to use this feature.

TIP

Without the **Select User** feature, you must enter user names exactly as they are in the database, or you'll receive data validation errors upon trying to publish.

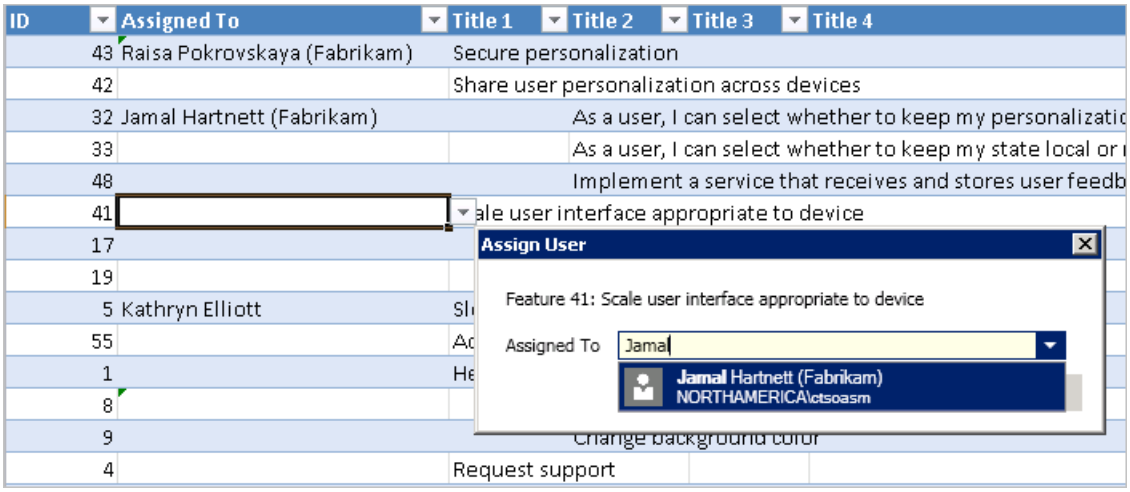
1. If you haven't installed or updated to the latest version of Visual Studio (at least VS 2015.1 or later version, do that now. You need the latest update to access the Select User feature.

2. Choose an identity or person-named field to activate the **Select User** feature in the Team ribbon.



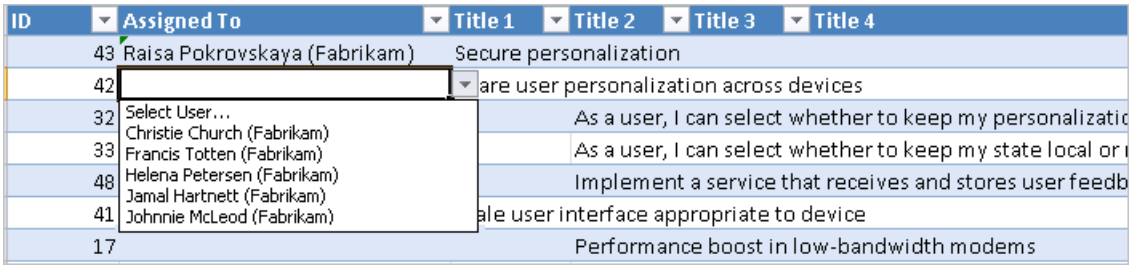
An identity or person-named field is a field that contains a user identity. These fields are typically synchronized to a database of user accounts, such as Azure Active Directory, Active Directory, or a Workgroup.

3. Begin typing the name of the user account and the Assign User dialog will automatically filter the results until you can select the account of interest.



Enter a letter to tab to the start of names beginning with that letter. Enter only user names as account aliases aren't recognized.

You'll notice that as you select user names, Excel remembers your recent selections and you can select the user accounts directly from the field.



Link work items

You can complete many actions from the **Links** tab of the **Links and Attachments** dialog. Specifically, you can:

- Review the existing links defined for the selected work item

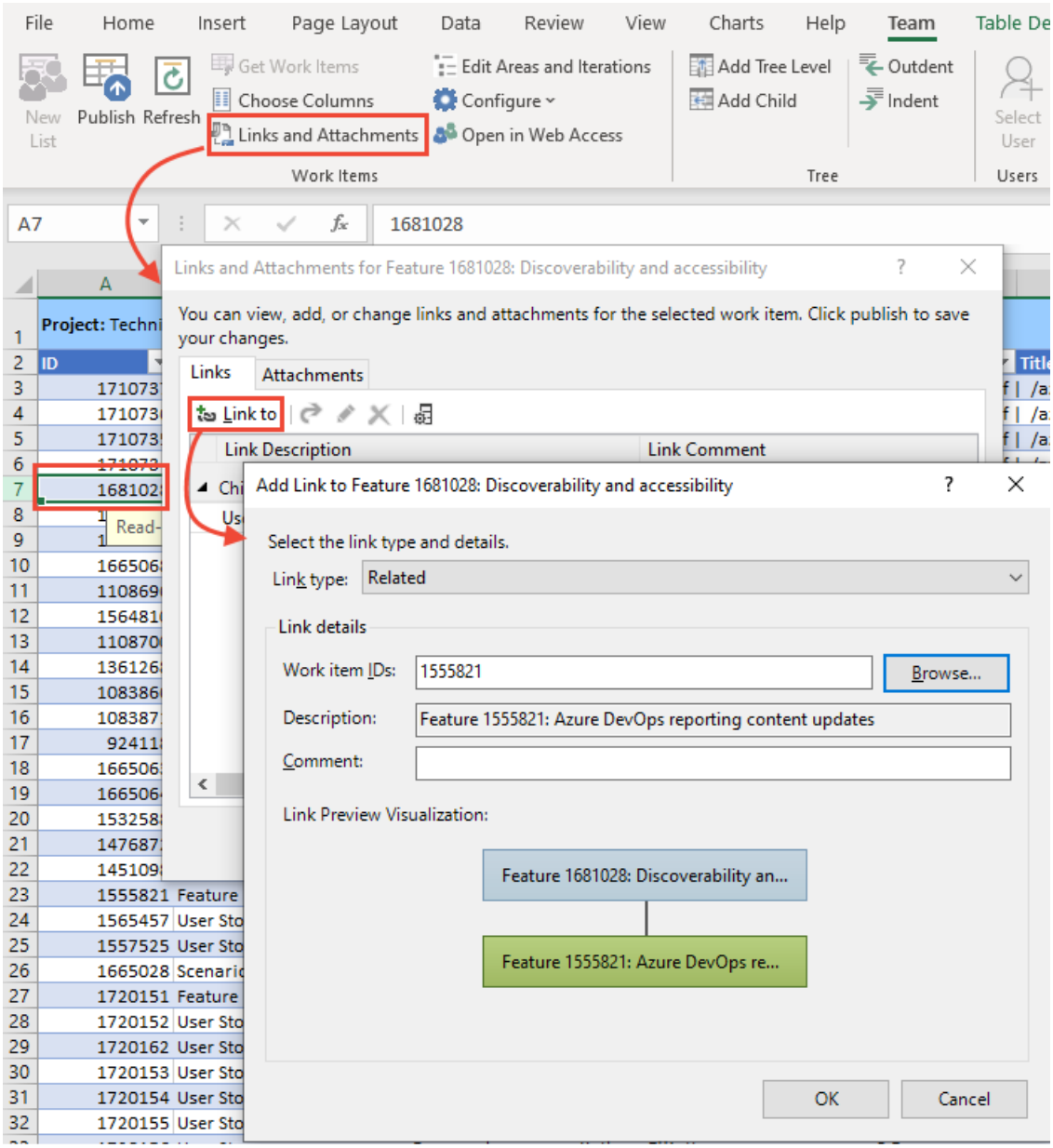
- Add links to selected work items to one or more work items or select objects
- Delete links
- Open a linked work item (opens in the web portal)
- Edit the link type of an existing link
- Add columns to the Link list and sort on that list

For more information on linking work items, see Link user stories, issues, bugs, and other work items.

View and add links

You can't use the Links and Attachments dialog to bulk update work item links. You can only bulk update tree-topology link types using a tree list.

1. To link a work item to other work items, choose the work item and then choose **Links and Attachments**. From the Links tab, choose **Link to** and then choose the **Link Type** and work item(s) you want to link to. Choose **OK** and then **Publish**.



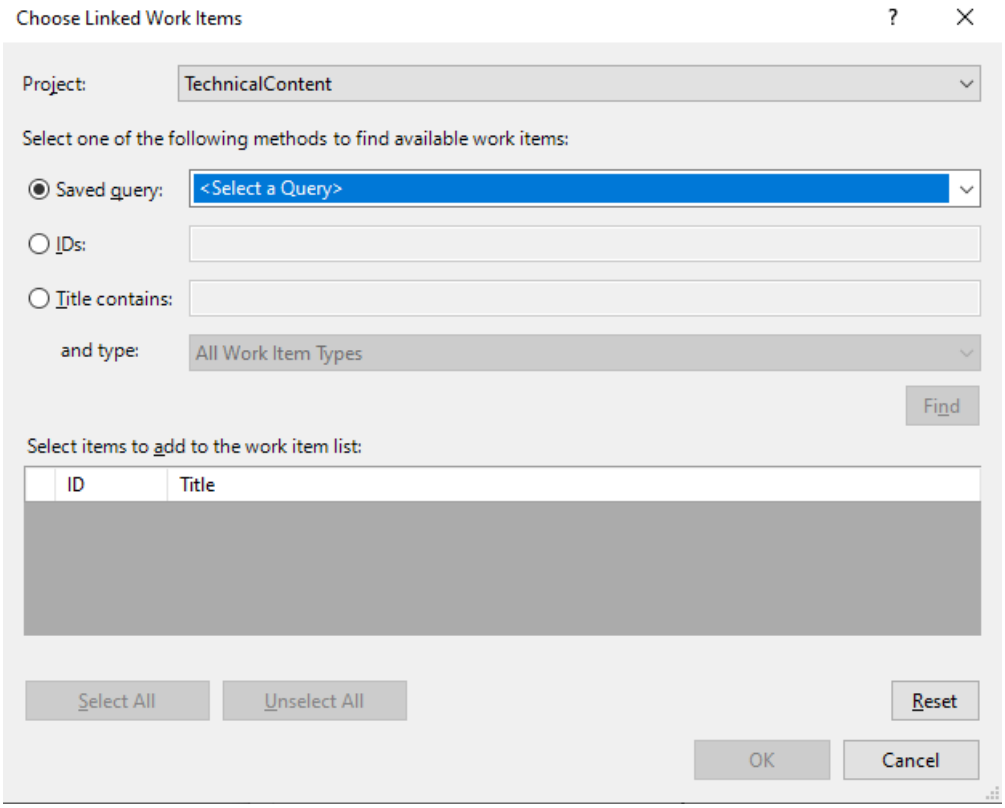
When finished, choose **Close** to dismiss the dialog.

2. To link several work items to the same work item(s), multi-select them by using **Ctrl-click** for consecutive rows, or **Shift-click** for non-consecutive rows.

Find work items to link to

From the Add link dialog, you can open a secondary dialog to choose one or more work items to link to. If you're going to find and list work items to link to by using a saved query, first define the query to use.

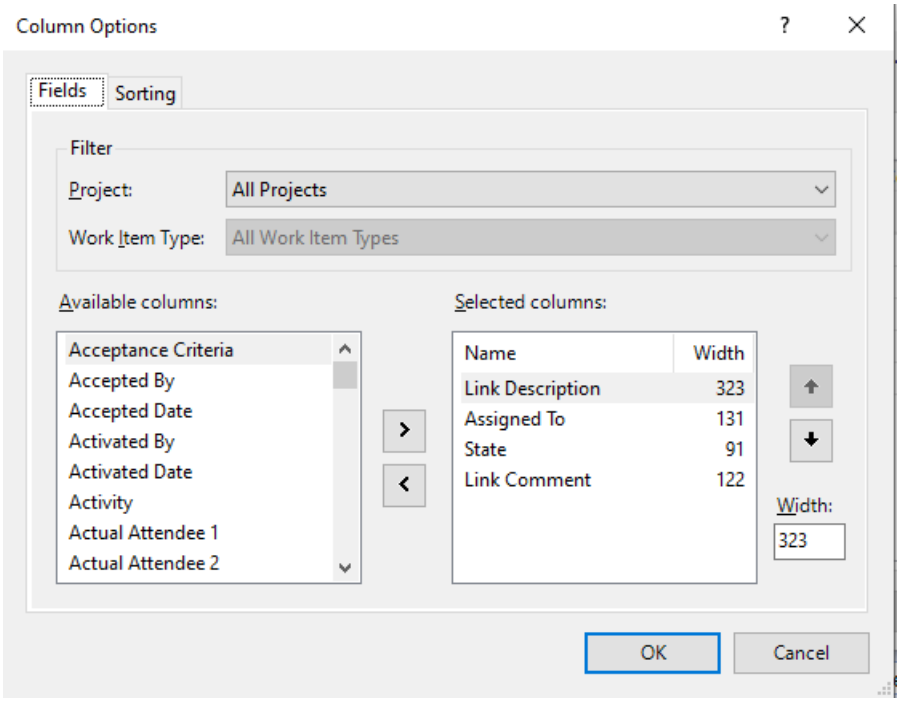
From the Add link dialog, choose the **Browse** button (Visual Studio) to open the following dialog.



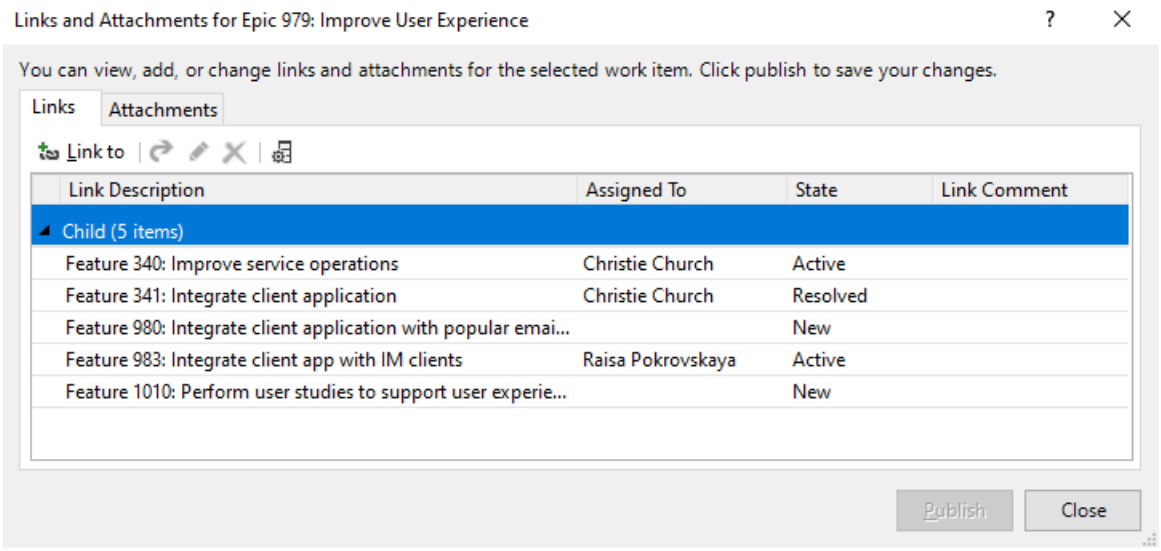
The **Choose Linked Work Items** dialog works in the same way as the **Get Work Items** dialog. To learn more, see Add existing work items to your worksheet described earlier in this article.

Add columns to the links list

1. From the **Links** tab, choose the **Columns** icon, and add the fields you want displayed. Here we add the Assigned to and State fields.



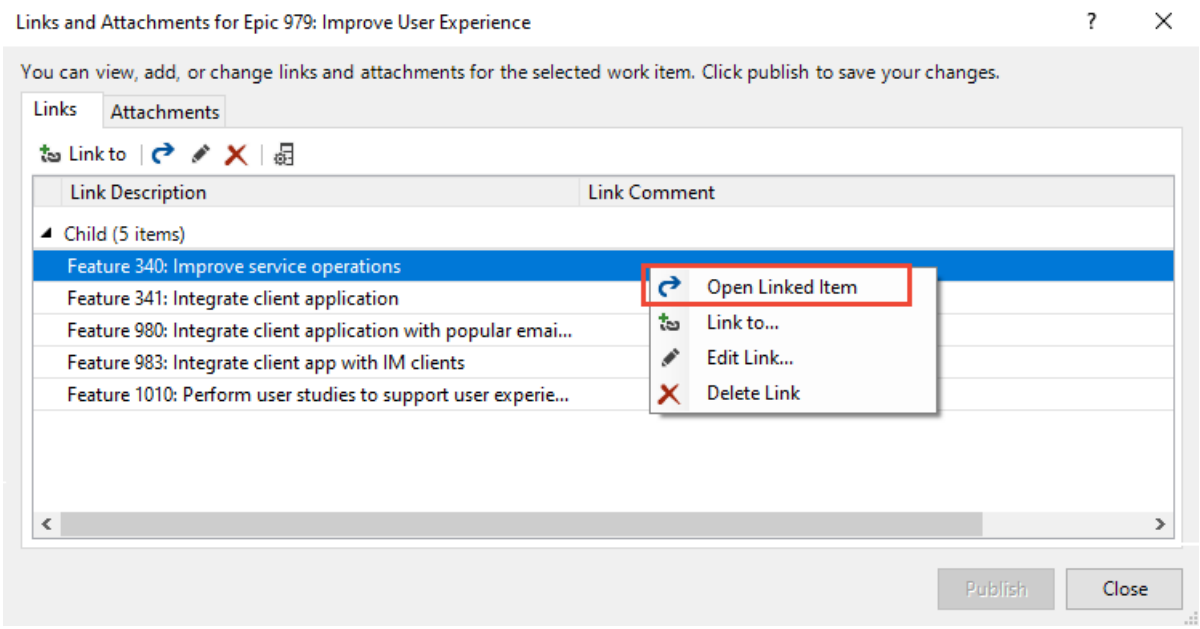
2. To reorder the links, choose the field to sort the list on that field.



This dialog works in the same way as the **Get Work Items** dialog. See Add existing work items to your worksheet described earlier in this article.

Open a linked work item

From the **Links** tab, choose the linked work item, right-click to open the context menu, and choose **Open Linked Item**.

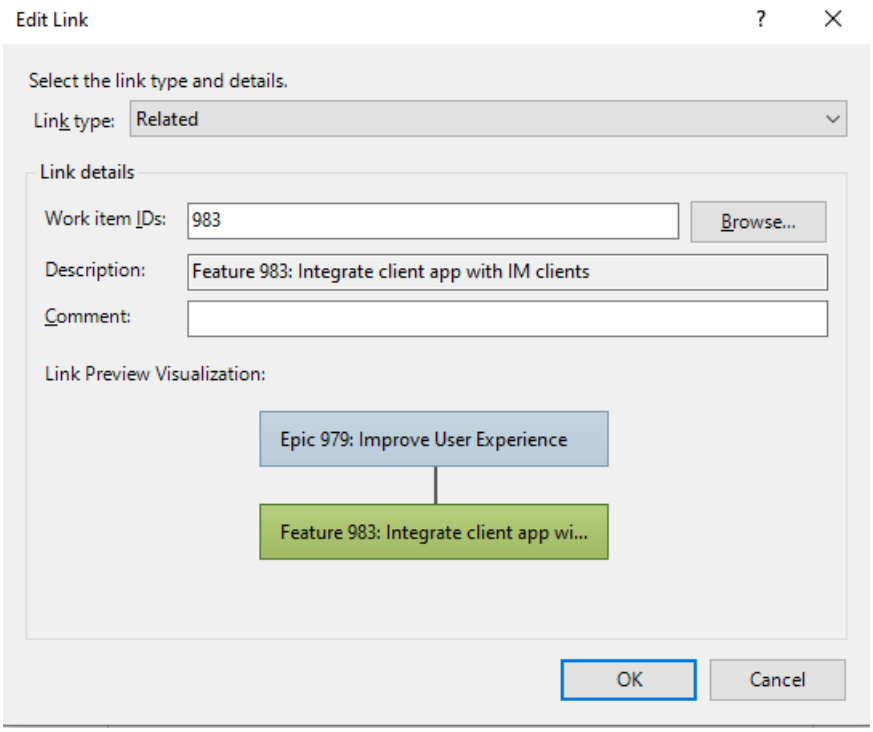


The work item opens in your web portal.

Edit the link and change the link type

You can edit any link listed. You can change the link type and the work items linked to.

1. Choose the link and choose the **Edit** icon.
2. Change the link type as needed.



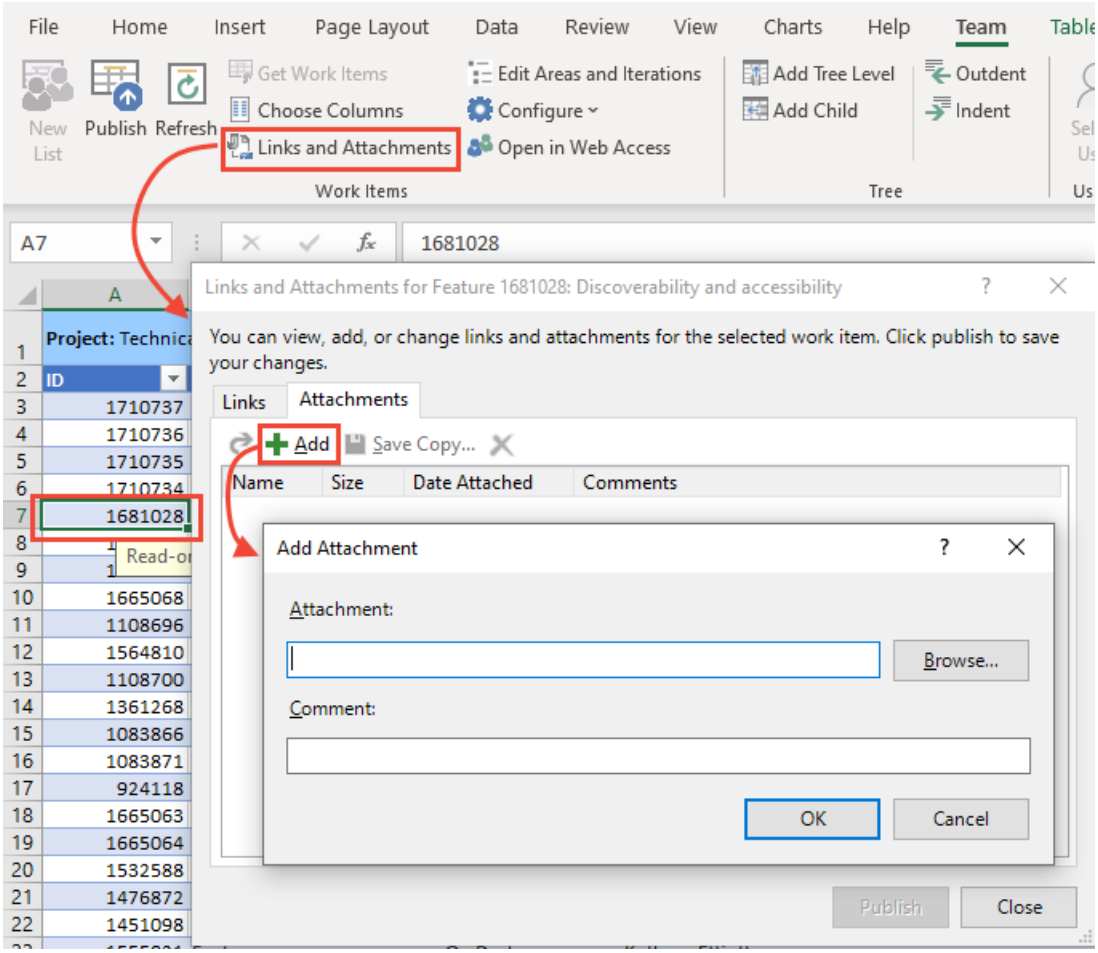
3. To change the work item linked to, enter the ID of the work item, or choose **Browse** to find the work item(s) to link to.

   The **Choose Linked Work Items** dialog works in the same way as the **Get Work Items** dialog. To learn more, see Add existing work items to your worksheet described earlier in this article.

Add attachments

- To add attachments, choose the work item, then **Links and Attachments**, and then the **Attachments** tab.

  Choose the file you want to attach, then choose **OK** and then **Publish**.



When finished, choose **Close** to dismiss the dialog.

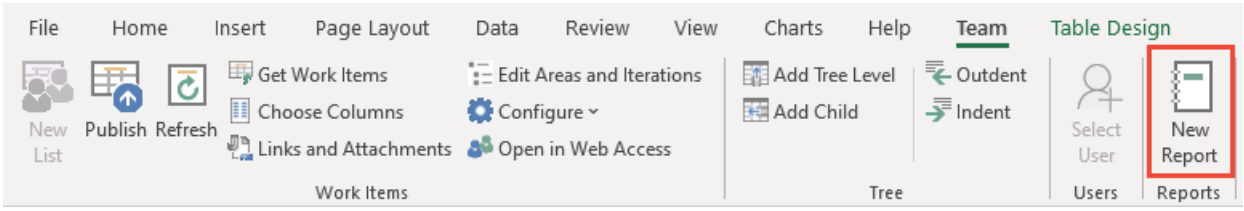
- To add the same attachment(s) to several work items, multi-select them by using **Ctrl-click** for consecutive rows, or **Shift-click** for non-consecutive rows.

Create a report

You can create a report or chart from the web portal for flat-list queries. See Track progress by creating status and trend query-based charts.

> **IMPORTANT**
> You can only create an Excel report using **New Report** from an on-premises Azure DevOps Server. These reports require that your project's project collection is configured to support SQL Server Analytics Server.

You can create a report using the **New Report** feature based on a flat list of work items.



To learn more, see Create Excel reports from a work item query.

Resolve publishing errors

To resolve publishing errors that arise when working in Excel, see one of the following articles:

- Resolve data conflicts: A data conflict occurs when a field value is changed in Azure Boards since the last time you published from Excel.
- Resolve data validation errors: A data validation error occurs if a field value violates the rules for that field and work item type.
- Resolve invalid links in a tree hierarchy: An invalid link occurs if a team member view work items in Excel as a hierarchy or tree list, and then moves a work item or sorts the list such that it breaks the dependencies between work items. You can resolve this error by reviewing the error message and repositioning work items to reflect the work item structure.
- Address Error TF208104: Hierarchical Link Relationship Is Locked:
  If you receive error TF208104, the changes you made to the fields are published. But your changes to the link hierarchy aren't published. At least one of the link relationships defined for the work item is locked by another process, such as Project Server integration.

Related articles

- Bulk modify work items (web portal)
- Azure DevOps Office integration issues
- FAQs: Work in Excel connected to Azure Boards
- Bulk import or update work items using CSV files
- View and add work items
- Basic Excel tasks

- Bulk modify work items (web portal)
- Azure DevOps Office integration issues
- FAQs: Work in Excel connected to Azure Boards

- Create Excel reports from a work item query

- Basic Excel tasks

Define a query as a hyperlink in Azure Boards and Azure DevOps

10/4/2022 • 2 minutes to read • Edit Online

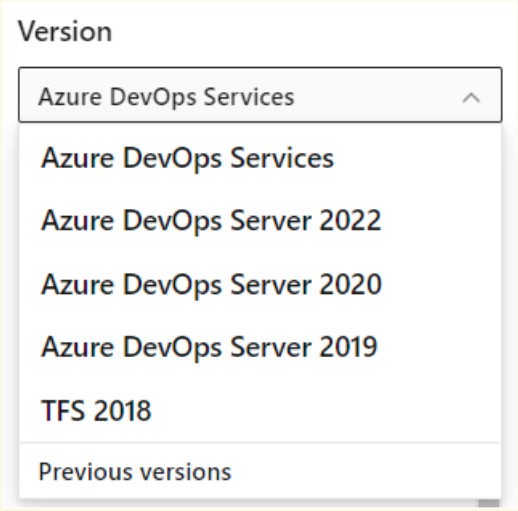
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

A query hyperlink uses the Work Item Query Language (WIQL), which resembles Transact-SQL. For details about constructing WIQLs, see Syntax for the Work Item Query Language (WIQL).

NOTE

Most browsers enforce a limit of between 2000 and 2083 characters for a URL string.

IMPORTANT



To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.
To learn which on-premises version you are using, see Look up your Azure DevOps platform and version

Query hyperlink syntax for Azure DevOps Services

Encode the WIQL portion of the URL syntax. You can use any URL encoder tool to encode your URL.

```
https://dev.azure.com/OrganizationName/ProjectName/_workitems?_a=query&wiql={Encoded WorkItemQueryLanguage}
```

For example, the following hyperlink lists the ID and title of all active bugs defined under the FabrikamFiber/Web area path for the fabrikam organization.

```
https://dev.azure.com/fabrikam/FabrikamFiber/_workitems?
_a=query&wiql=SELECT%20%5BSystem.ID%5D%2C%20%5BSystem.Title%5D%20FROM%20WorkItems%20WHERE%20%5BSystem.TeamPr
oject%5D%3D'FabrikamFiber'%20AND%20%5BSystem.WorkItemType%5D%3D'Bug'%20AND%20%5BSystem.State%5D%3D'Active'%2
0AND%20%5BSystem.AreaPath%5D%3D'FabrikamFiber%5CWeb'
```

The decoded WIQL conforms to:

```
SELECT [System.ID], [System.Title]
   FROM WorkItems
   WHERE [System.TeamProject]='FabrikamFiber'
   AND [System.WorkItemType]='Bug'
   AND [System.State]='Active'
   AND [System.AreaPath]='FabrikamFiber\Web'
```

> **NOTE**
> For queries made against Azure Boards, the WIQL length must not exceed 32K characters. The system won't allow you to create or run queries that exceed that length.

Query hyperlink syntax for TFS 2018 through Azure DevOps Server 2020

```
https://{ServerName}/{CollectionName}/{ProjectName}/_workitems?_a=query&wiql={Encoded WorkItemQueryLanguage}
```

For example, the following hyperlink lists the ID, title, and state of all bugs under the FabrikamFiber/Web area path.

```
http://fabrikam:8080/tfs/DefaultCollection/FabrikamFiber/_workitems?
_a=query&wiql=SELECT%20%5BSystem.ID%5D%2C%20%5BSystem.Title%5D%2C%20%5BSystem.State%5D%20FROM%20WorkItems%20
WHERE%20%5BSystem.TeamProject%5D%3D'FabrikamFiber'%20AND%20%5BSystem.WorkItemType%5D%3D'Bug'%20AND%20%5BSyst
em.AreaPath%5D%3D'FabrikamFiber%5CWeb'%20%20
```

Which is comparable to the non-encoded entry shown below.

```
http://fabrikam:8080/tfs/DefaultCollection/FabrikamFiber/_workitems?_a=query&wiql=
SELECT [System.ID], [System.Title], [System.State]
   FROM WorkItems
   WHERE [System.TeamProject]='FabrikamFiber'
   AND [System.WorkItemType]='Bug'
   AND [System.AreaPath]='FabrikamFiber\Web'
```

Related articles

- Syntax for the Work Item Query Language (WIQL)
- Wiql Editor, a Marketplace extension
- REST API, Wiql

Query by titles, IDs, and rich-text fields in Azure Boards and Azure DevOps

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you want to find work items based on a keyword or phrase or a null text field, you can do so by filtering on single-line text (String), multi-line text (PlainText), and rich-text (HTML) fields. If you find that your queries take too long to return results, review the Guidance to create high-performing queries.

Supported operators and macros

Query clauses that specify a text or rich-text field can use the operators and macros listed in the following table.

Data type	Supported operators and macros
Rich-text (HTML) Multi-line text strings (PlainText)	Contains Words, Does Not Contain Words, Is Empty (Note 1), Is Not Empty(Note 1)
Multi-line text strings (PlainText)	Supported operators and macros
Single text (String)	= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever **Macros**: **[Any]**, valid with the **Work Item Type** field **@Project** (Note 2), valid with the **Team Project** field
GUID	= , <> , > , < , >= , <= , In, Not In **Macros**: **@Follows**, **@MyRecentActivity**, **@RecentMentions**, **@RecentProjectActivity** valid with the **ID** field and **In** and **Not In** operators **@Project** (Note 2), valid with the **Team Project** field
State and **Work Item Type** fields	= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever **Macros**: **[Any]** valid with both fields

NOTE

1. The **Is Empty** and **Is Not Empty** operators are supported for Azure DevOps Server 2019 RC2 and later versions
2. The **@Project** macro is supported for Azure Boards and TFS 2015.1 and later versions. The system automatically defaults to filtering based on the current project. To learn more, see Query across projects.

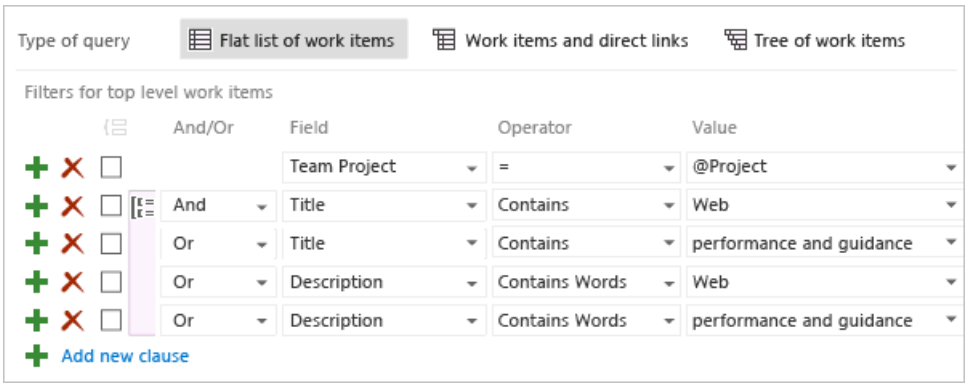
Use `Contains words` for string matches

When you want to filter on a string match, try using the `Contains Words` operator instead of `Contains`. The `Contains Words` operator runs a full-text search on the specified field, which is faster in most cases. Text string is limited to 100 characters.

While the `Contains` operator runs a table scan, which isn't only slower, but also consumes more CPU cycles. These CPU cycles contribute towards your resource consuming rate limit.

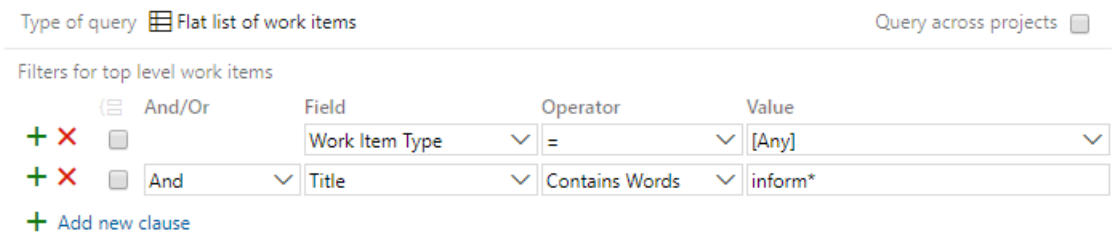
Keyword or phrase query with wildcards

Use **Contains** or **Contains Words** to list items that partially or exactly match the words or phrase that you enter.



Choose **Contains** or **Does Not Contain** to search against exact or partial matches of a word or phrase. Choose **Contains Words** or **Does Not Contain Words** to search against an exact phrase or to use the wildcard character, *. These operators use the full-text search index.

For example, specify **Contains Words** and **inform*** to filter on a text field that contains *inform* or *information* or *informational*.



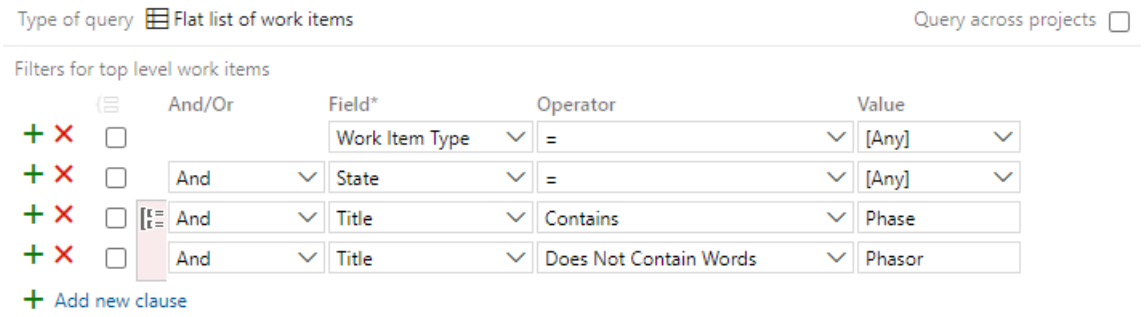
TIP

To understand how `AND/OR` clauses are grouped, see Create and save managed queries, Group clauses. To view the WIQL syntax for a query, install the WIQL query editor extension which allows you to see the WIQL version of any query editor entry.

Query for specific words and not others

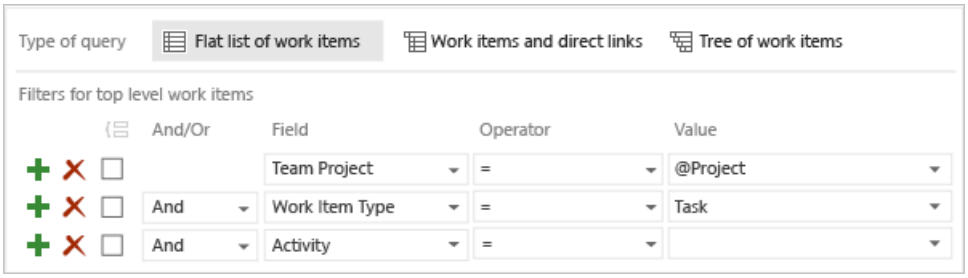
Use **Contains Words** and **Does Not Contain Words** operators to list items that exactly match the words or phrase that you enter, and exclude other words or phrases. You can use these operators in combination and with the wildcard character (*).

In the following example, these operators filter work items for those items that contain the work *Phase* but not the word *Phasor*.



Undefined field value queries

You can find work items that have an undefined field value by using the equals operator (=) and leaving the Value for the field blank. For example, the following filters will list all work items of type Task whose Activity field is blank.

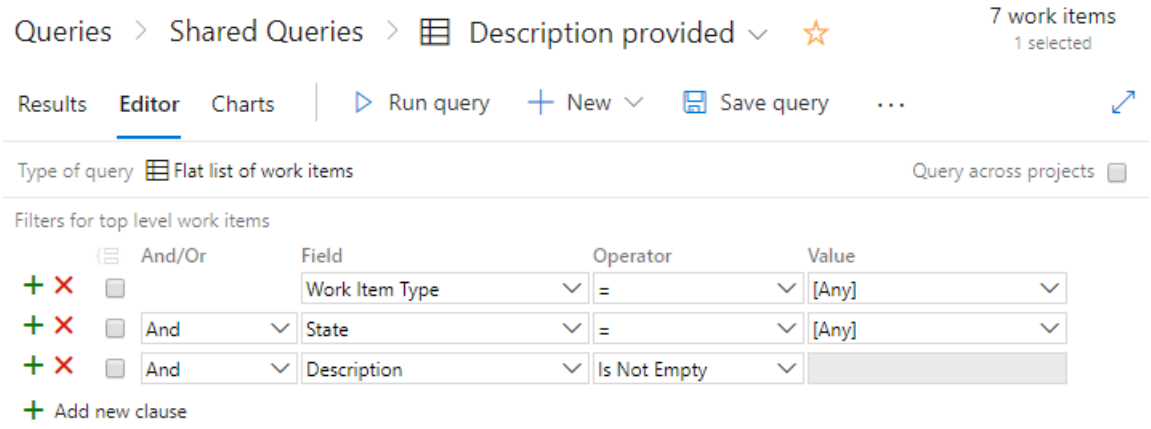


To list work items based on a field that isn't blank, use the not operator (<>) and leave the Value blank.

Empty or not empty HTML field queries

You can find work items where no **Description** has been entered. Using the **Is Empty** or **Is Not Empty** with an HTML field supports listing work items with empty or not empty rich text fields. You don't specify a value with this operator.

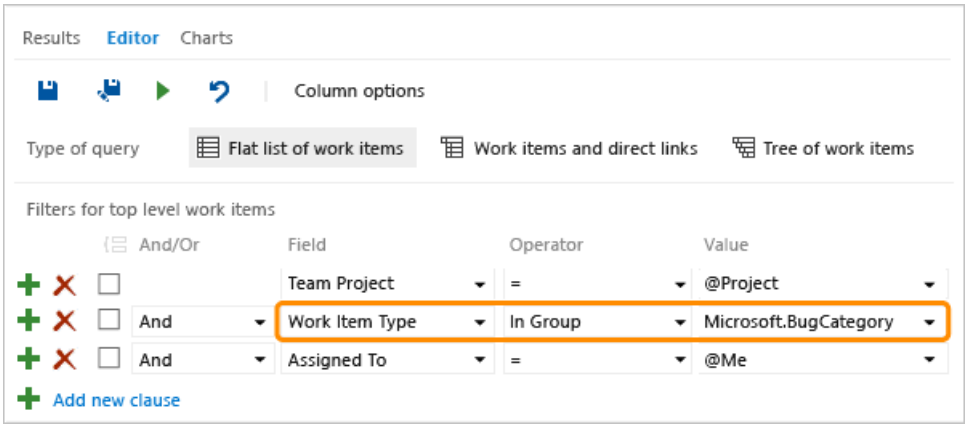
For example, the following query filters will list all work items where some entries have been made into the **Description** field.



> **NOTE**
> The ability to query for work items that don't have any tags attached to them is not a supported feature. If you'd like to up vote the request to support this feature, you can do so on our Developer Community page, Be able to search for empty tags.

Category-based queries

To filter work items based on the category they belong to, use the **In Group** operator. For example, the following filter criteria will return all work items that are in the current project, assigned to the team member, and defined as belonging to the Bug Category.



What items appear in the Requirement or Task categories?

The default assignments of work item types to each category are listed below for each process.

PROCESS	REQUIREMENT CATEGORY	TASK CATEGORY
Basic	Issue	Task
Agile	User Story	Task
Scrum	Product Backlog Item, Bug	Task
CMMI	Requirement	Task

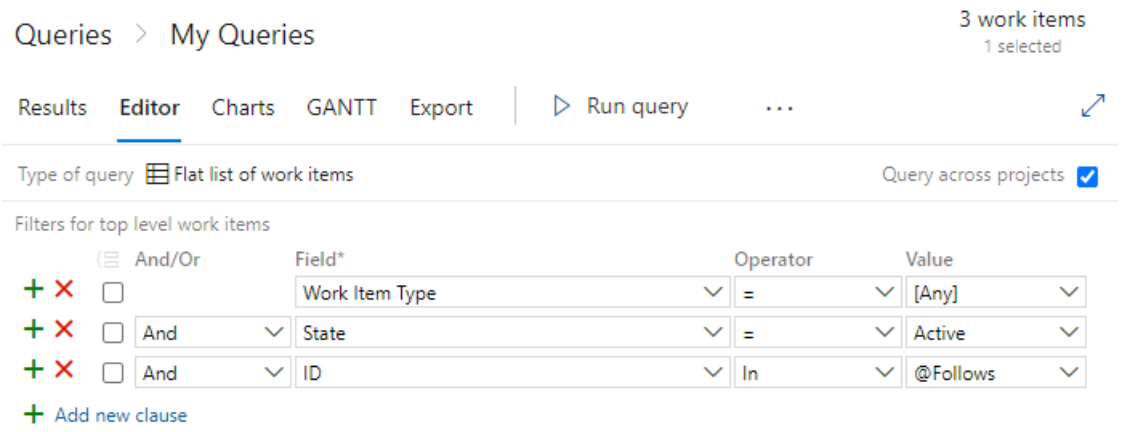
Each team can determine if the Bug work item type appears in either the Requirement or Task category. See Show bugs on backlogs and boards. You can add custom work item types to a backlog. For details, see Add or

modify a work item type, Add a custom WIT to a backlog or board.

Query for work items that you're following

You can use the **@Follows** macro to filter a list based on work items you're following along with other query filters.

For example, the following query shows how to query across all projects for active work items that you're following. You use the ID field and the In operator with the **@Follows** macro.



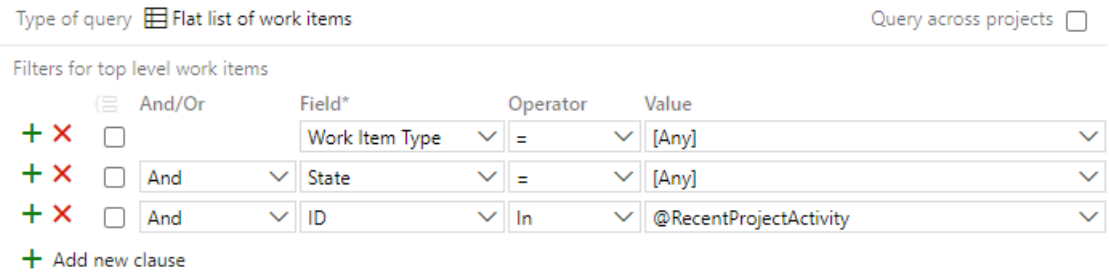
Query for recent work item activity

You can use the following macros to list work items based on recent activity:

- **@MyRecentActivity**: List items you've recently viewed or modified.
- **@RecentMentions**: List items you were added to via an **@mention** in the last 30 days.
- **@RecentProjectActivity**: List items that have been recently created or modified in your project.

Specify the **ID** field and either the **In** or **Not In** operators.

For example, the following query shows how to query for work items that you've recently viewed or modified.



Common fields for most work item types

The following table describes common fields used to filter queries. The **ID** fields uniquely identify work items in a list. Use the **Title** field to distinguish the work item from all others of the same type. The **Description** and other rich-text (data type=HTML) fields provide additional information that is needed to implement work and track changes. After a work item is created, you can modify all fields except for the **ID**. When you add and save a work item, the ID is assigned by the system and cannot be changed.

> **NOTE**
> The system automatically indexes all long-text fields with a data type of **PlainText** and **HTML** fields for full-text search. This includes the **Title**, **Description**, and **Steps to Repro** fields. For more information and server and collation requirements applicable to on-premises Azure DevOps, see Query fields, operators, values, and variables - Full-text and partial word searches.

Field name

Description

Work item type

Acceptance Criteria [1]

A description of the criteria to be met before the bug or product backlog item can be closed.

Before work begins on a bug or product backlog item, the criteria for customer acceptance should be described as clearly as possible. Conversations between the team and customers to define the acceptance criteria will help ensure that your team understands your customers' expectations. The acceptance criteria can be used as the basis for acceptance tests so that you can more effectively evaluate whether an item has been satisfactorily completed.

Reference name=Microsoft.VSTS.Common.AcceptanceCriteria, Data type=HTML

Bug, Epic, Feature, Product backlog item (Scrum)

Description [1, 2]

Use this field to provide in-depth information about a work item.

Reference name=System.Description, Data type=HTML

All

ID

The unique identifier that is assigned to a work item. Work item IDs are unique across all projects and within a project collection.

Reference name=System.Id, Data type=Integer

All

Repro Steps (or Steps to reproduce) [1]

The steps that are required to reproduce unexpected behavior. Capture enough information so that other team members can understand the full impact of the problem as well as whether they have fixed the bug. This includes actions taken to find or reproduce the bug and expected behavior.
Reference name=Microsoft.VSTS.TCM.ReproSteps, Data type=HTML

Bug

Resolution

Describes how an impediment was resolved.

Reference name=Microsoft.VSTS.Common.Resolution, Data type=HTML

Impediment (Scrum)

System Info[1]

Information about the software and system configuration that is relevant to the bug, code review, or feedback.

Reference name=Microsoft.VSTS.TCM.SystemInfo, Data type=HTML

Bug, Code Review Request, Feedback Request

Team Project

The project to which a work item belongs. Add this field to a query when you want to filter your list to items in one or more projects.

> **NOTE**
> For Azure Boards or for TFS 2015.1 and later versions, you must check the Query across projects option in the query editor for this field to appear in the drop down field list. To learn more, see Example queries, query across projects.

Reference name=System.TeamProject, Data type=String

All

Title

A short description that summarizes what the work item is and helps team members distinguish it from other work items in a list.

Reference name=System.Title, Data type=String

All

Work Item Type

The name of the work item type. Work item types are defined based on the process used when you created your project. For an overview, see Choose process. To learn how to add a custom work item type, see Add or modify a work item type.

To filter work items based on their category assignment, you can use the **In Group** and **Not In Group** operators and select a category from the drop-down list.

Reference name=System.WorkItemType, Data type=String

All

> **NOTE**
> 1. To learn more about working with rich-text fields, see Share information within work items.
> 2. Upon upgrade to Team Foundation Server 2012, the Description field was changed from a field type of PlainText to **HTML**. Using the **witadmin changefield** command you can revert the data type for this field. See Manage work item fields (witadmin).

Related articles

- Query editor
- Add work items
- Work item field index
- About managed queries

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Query by assignment or workflow changes in Azure Boards

10/4/2022 • 17 minutes to read • Edit Online

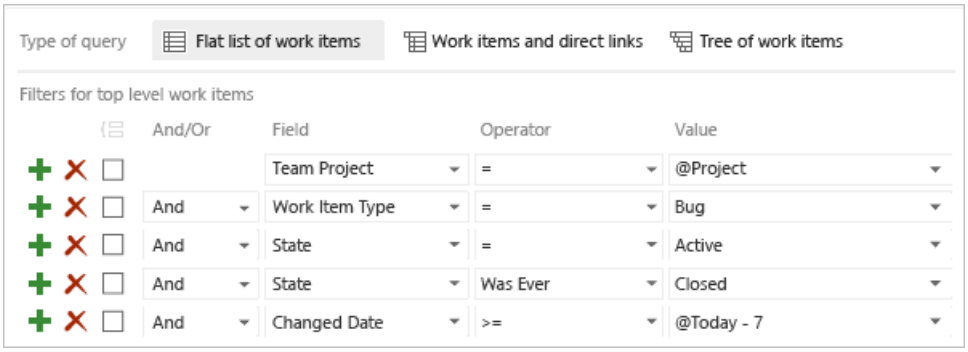
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The states in the workflow support tracking work status as it moves from a new state to a closed or done state. Kanban query fields support tracking the status of work as it moves from one column or swimlane to another on the Kanban board.

Each workflow consists of a set of states, valid transitions between states, and reasons for transitioning the work item to the selected state. Workflow states and reasons differ among the work item types and default processes used to create your project.

Most work items move from a *New*, *Active*, or *Proposed* state to a *Done* or *Closed* state. As each work item moves from one state to another, the item might also be reassigned to various members of the team. For example, a tester might create a bug that is assigned to another team member during triage. When the other team member resolves the bug, it's reassigned to the tester who created it.

For example, you can find all work items that were closed but then reactivated. By specifying the **Changed Date** field, you can focus on reactivations that occurred today, yesterday, or in the last week.



You can also use the **Activated By** and **Activated Date** fields, or other workflow fields.

> **TIP**
> Not all fields are valid for all work item types. Jump to Workflow and Kanban query fields for the set of fields you can include in queries and which work item types they apply to.

If you're new to creating queries, see Use the query editor to list and manage queries.

Supported operators and macros

Query clauses that specify an identity or workflow-associated field can use the operators and macros listed in the following table. To learn about the field data type, see Workflow and Kanban board fields provided later in this article.

Data type	Supported operators and macros

Boolean [1]

= , <> , =[Field] , <>[Field]

DateTime

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever
Macros: **@Today**, **@Today +/- n** valid with any DateTime field

Identity

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever
Macros: **@me** valid for all Identity fields

Single text (String) [2]

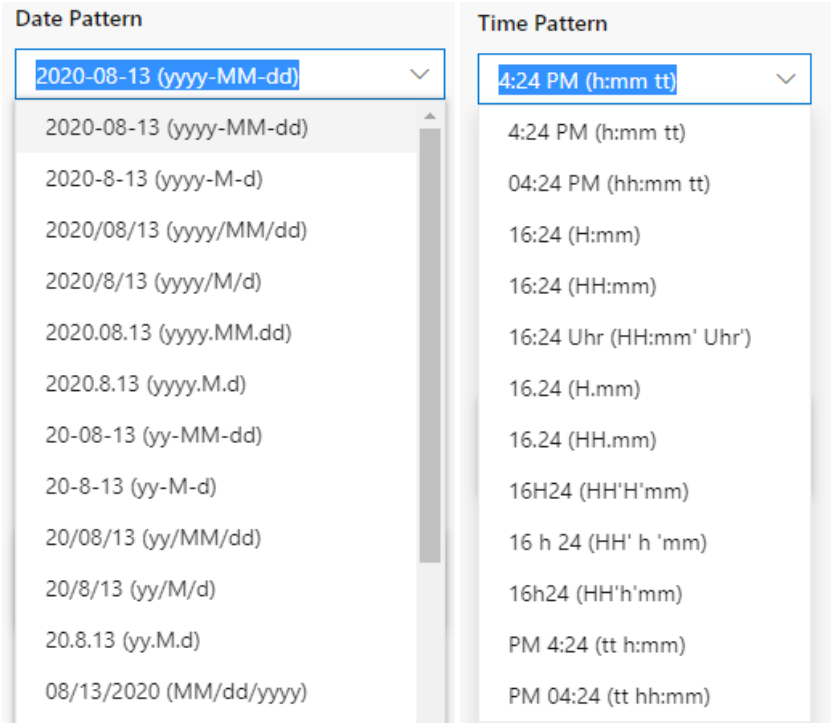
= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever

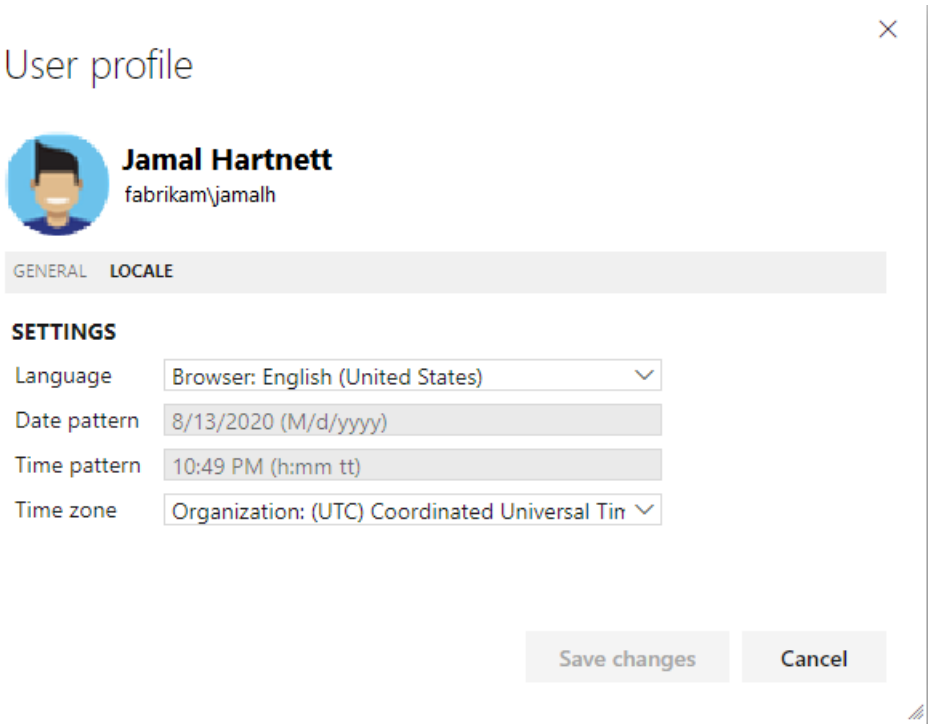
> **NOTE**
> 1. The **Boolean** data type field is supported for TFS 2017 and later versions.
> 2. The **Was Ever** operator is only valid for Kanban board columns for Azure DevOps Services at this time.

Use the **In** and **Not In** operators to filter for or exclude two or more picklist entries or a delimited set of items. Use the **In Group** or **Not In Group** operators to filter for items that belong or don't belong within a category group or security group. For more information, see Query fields, operators, and macros.

Date and time pattern

The date and time pattern you enter for **DateTime** fields should match that which you select through your profile. To view or change your selection, see Set user preferences, Time and Locale.





Identity-based queries

Use the search box or query editor to quickly find work items based on an assignment made to an **Identity** field. Also, you can filter for work items based on who changed, resolved, or closed a work item. By specifying a time period, you can scope your query even further, which can help with performance.

Use = to find current assignments, **Was Ever** to list items based on past assignments, and **@Me** to scope to your user identity.

Filter for

Include these query clauses

Active items assigned to me

```
Assigned To @Me
And State = Active
```

Closed items that at some point was assigned to me

```
Assigned To Was Ever @Me
And State = Closed
```

Active user stories assigned to the Web team

```
Work Item Type = User Story
And State = Active
And Assigned To In Group [FabrikamFiber]\Web
```

Items I've modified in the last 30 days

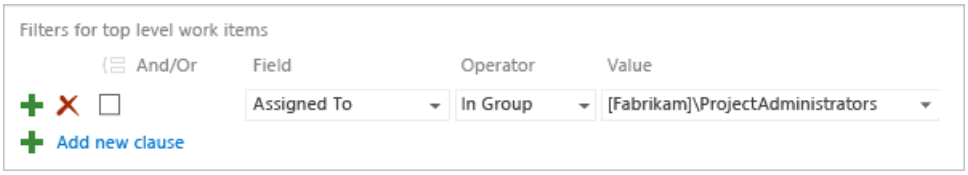
```
Changed By = @Me
And Changed Date >= @Today-30
```

Unassigned items (leave the **Value** blank)

```
Assigned To = _
```

Team or group membership queries

To filter on items assigned to someone who belongs to a team or security group, use the **In Group** operator.



You can use the **In Group** or **Not In Group** operators to filter a query based on several values that are members of a group, or that aren't members of a group. Examples of groups you can specify include the following items:

- Teams
- Built-in and custom security groups
- Azure Active Directory and Active Directory security groups
- Work item categories

Queries based on workflow changes

You use the State, Reason, and Resolved Reason fields to query for items based on workflow changes.

Filter for	Include these query clauses
Resolved stories	`Work Item Type = User Story` `And State = Resolved`
Stories, bugs, and tasks that are new or active	`Work Item Type In User Story,Bug,Task` `And State In New,Active`
Items removed as they're duplicate	`State= Removed` `And Reason = Duplicate`
Items failing acceptance tests	`Resolved Reason = Acceptance tests fail`
Items closed within the last 15 days	`State = Closed` `And Closed Date > @Today-15`

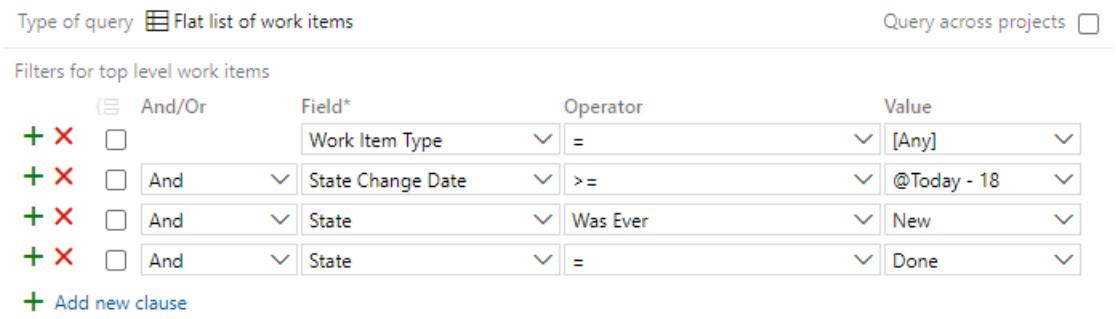
Workflow changes and identity-based queries

You can quickly find items that you changed, resolved, or closed. You can also find items that were changed by other team members. Several fields—such as the Created By, Changed By, Resolved By, and Closed By—are populated based on changes to the workflow.

Filter for	Include these query clauses
User Stories that I closed	`Work Item Type = User Story` `And Closed By = @Me`
Items I resolved in the last week	`Resolved By = @Me` `And Resolved Date >= Today-7`

Query changes in work item state

To list work items that have changed state within a specific date range, you can use the **State Change Date** field to narrow the search and then add clauses for changes to the **State** field. An example is shown in the following image.



Query changes to a Kanban board

Using the Kanban query fields—Board Column, Board Column Done, and Board Lane—you can list work items according to their flow status on the Kanban board. And, you can create a status or trend chart based on these queries.

You can list items based on the team area path, and if they are in a specific custom Kanban column and swimlane. If you rename a column or swimlane, you'll need to update the query filters to reflect the new name. For more ideas, see this blog post: New fields bring Kanban goodness to queries, and more



> **NOTE**
> Queries are now scoped to the current project by default. Check the **Query across projects** to find work items defined in other projects within the collection.

Filter for

Include these query clauses

User Stories in the **Code/Doing** column

```
Work Item Type = User Story
And Board Column = Code
And Board Column Done = False
```

Items in the **Expedite** swimlane

```
Board Lane = Expedite
```

Items in any swimlane whose label contains "Test"

```
Board Lane Contains Test
```

Items that were ever in the "In Review" column

```
Board Column Was Ever In Review
```

> **IMPORTANT**
> Work items that appear on more than one team's Kanban board can yield results that don't meet your expectations because each team can customize its Kanban board columns and swimlanes. The values assigned to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields might differ from what you expect when another team updates the work item from a different board. To learn more, see Add, review, and update work items in Azure Boards.

Workflow and Kanban board fields

The following fields are useful to filter queries. Some of these fields get updated as a work item progresses from one state to another. Or they're updated as you move a work item in the Kanban board to a different column or swimlane. Several of these fields don't appear on the work item form, but they're tracked for those work item types listed in the following table.

For more information about field attributes, see Work item fields and attributes.

Field name

Description

Work item type

Activated By [1, 2, 3]

The name of the team member who changed the status of a work item to an *In Progress* category state.

The name of the team member who changed the status of a work item from *New* to *Active* or reactivated a work item after it had been closed, completed, or done.

Reference name= `Microsoft.VSTS.Common.ActivatedBy`
Data type=String (Identity)

Bug, Change Request, Epic, Feature, Issue, Product Backlog Item, Requirement, Review, Risk, Shared Step, Task, Test Case, User Story

Activated Date [1, 3]

The date and time when the work item was changed to an *In Progress* category state.

The date and time when the work item was changed from *New* to *Active* or reactivated after it had been closed, completed, or done.

Reference name= `Microsoft.VSTS.Common.ActivatedDate`
Data type=DateTime

All

Assigned To [2]

Assigned To [2, 3, 4]

The name of the team member who currently owns the work item. For more information, see Note 1 on synchronization and person-name fields.

Reference name= `System.AssignedTo`
Data type=String (Identity)

All

Board Column

The current Kanban board column assignment of the work item, for example: Active, Closed, Committed, Done, or other custom column assignment.

Reference name= `System.BoardColumn`
Data type=String

Requirement Category [4]

Requirement Category [5]

Board Column Done

The current assignment of the work item to Doing (False) or Done (True) Kanban column. Only assigned when split-columns is enabled for a Kanban board column.

Reference name= `System.BoardColumnDone`
Data type=Boolean

Requirement Category [4]

Requirement Category [5]

Board Lane

The current Kanban board swimlane assignment of the work item, for example: Default, Expedite, Blocked, or other custom swimlane assignment. Reference name= `System.BoardLane`
Data type=String

Requirement Category [4]

Requirement Category [5]

Closed By [1, 2]

Closed By [1, 2, 3]

The name of the team member who set the state to closed, completed, or done.

Reference name= `Microsoft.VSTS.Common.ClosedBy`
Data type=String (Identity)

All

Closed Date

The date and time when a work item was closed.

Reference name= `Microsoft.VSTS.Common.ClosedDate`
Data type=DateTime

All

Created By [1, 2]

Created By [1, 2, 3]

The name of the team member who created the work item.

Reference name=`System.CreatedBy
Data type=String (Identity)

All

Created Date

The date and time when a work item was created.

Reference name= `System.CreatedDate`
Data type=DateTime

All

Reason

Reason [3, 4]

The reason why the work item is in the current state. Each transition from one workflow state to another is associated with a corresponding reason.

For On-premises XML process models, the reason values are defined within the `WORKFLOW` section of the work item type definition using the `REASON` element. To modify the defined reasons, see Change the workflow for a work item type.

Reference name= `System.Reason`
Data type=String

All (except Test Case and Shared Steps)

Resolved By [1, 2]

Resolved By [1, 2, 3]

The name of the team member who changed the status of a work item to a *Resolved* category state.

The name of the team member who changed the status of a work item to *Resolved* or done workflow state.

Reference name= `Microsoft.VSTS.Common.ResolvedBy`, Data type=String (Identity)

All

Resolved Date

Resolved Date [1, 2]

The date and time when the work item was changed to an *In Resolved* category state.

The date and time when the work item was moved into a *Resolved* or done workflow state.

Reference name= `Microsoft.VSTS.Common.ResolvedDate`, Data type=DateTime

All

Resolved Reason

Resolved Reason [3]

The reason why a work item was resolved. For example, the user story is code complete or the bug is fixed. This field is read-only and only valid for Agile and CMMI work item types.

Reference name= `Microsoft.VSTS.Common.ResolvedReason`
Data type=String

All (Agile, CMMI)

Reviewed By

The name of the team member who responded to a code review request and is cataloged in the code review response.

Reference name= `Microsoft.VSTS.Common.ReviewedBy`
Data type=String (Identity)

Code Review Response

State

State [3, 4]

The current state of the work item. This field allows you to update the status of a work item as it progresses from new or active to a done or closed state.

To modify the workflow states, see Customize the workflow for a process.

To modify the workflow states, see the following articles:

- For Inherited process model: see Customize the workflow for a process
- For On-premises XML process models: see Change the workflow for a work item type.

To modify the workflow states, see Change the workflow for a work item type.

Reference name= `System.State`
Data type=String

All

State Changed Date

The date and time when the value of the State field changed.

Reference name= `Microsoft.VSTS.Common.StateChangeDate`
Data type=DateTime

All

NOTE

1. See Date and Identity fields.
2. By default, the server synchronizes system-defined person-name or Identity-based fields with Active Directory or Azure Active Directory. These fields include: **Activated By**, **Assigned To**, **Closed By**, **Created By**, and **Resolved By**. You can grant access to a project by adding security groups that you created in AD or Azure AD or by adding accounts to existing or custom groups defined from the collection setting **Security** page. See set up Active Directory or Azure Active Directory.
3. See Activated By/Date and Resolved By/Date fields.
4. The Requirement Category applies to all work item types that appear on the product backlog and Kanban board, and may include those added to the Bug Category based on the team setting for Show bugs on boards and backlogs. For more information on work item type categories, see Use categories to group work item types.

NOTE

Even if you add a board-related field, such as Board Column or Board Lane, to a work item form, you can't modify the field from the form.

1. See Date and Identity fields.
2. By default, the server synchronizes system-defined person-name or Identity-based fields with Active Directory or Azure Active Directory. These fields include: Activated By, Assigned To, Closed By, Created By, and Resolved By. You can grant access to a project by adding security groups that you created in AD or Azure AD or by adding accounts to existing or custom groups defined from the collection setting **Security** page. See set up Active Directory or Azure Active Directory.

   For on-premises deployments, you can enable or disable synchronization for a person-name field by using the **witadmin changefields** command-line tool. You can also synchronize custom person-name fields by specifying the **syncnamechanges** attribute. See Manage work item fields and FIELD (Definition) element reference.
3. Reportable field with attribute set to Dimension. Only valid when the collection is configured to support the On-premises XML model. Reportable data is exported to the data warehouse and can be included in Excel or SQL Server reports. For on-premises Azure DevOps, use the **witadmin changefield** command to change the reportable attribute for a field.
4. Indexed field. Enabling indexing for a field may increase the performance of finding work items whose queries specify that field. For on-premises Azure DevOps, use the **witadmin indexfield** command to change the index attribute for a field.
5. The Requirement Category applies to all work item types that appear on the product backlog and Kanban board. The category includes those items added to the Bug Category based on the team setting for Show bugs on boards and backlogs. For more information on work item type categories, see Use categories to group work item types.

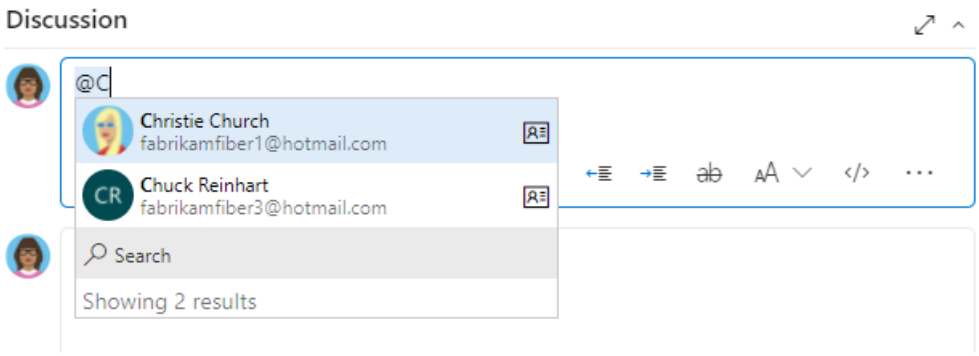
NOTE

Even if you add a board-related field, such as Board Column or Board Lane, to a work item form, you can't modify the field from the form.

People picker

The **Assigned To** field is supported by the people picker feature. For example, when you choose the **Assigned To** field from within a work item form, the people picker is activated. As shown in the following image, you

simply start typing the name of the user you want to select, and search until you find a match. Users that you've previously selected appear in the list automatically. To select users that you haven't selected previously, enter their entire name or search against the full directory.



@mention tool in Discussion showing people picker." />

For organizations that manage their users and groups using Azure Active Directory (Azure AD) or Active Directory, people pickers provide support for searching all users and groups added to the AD, not just those users and groups added to the project.

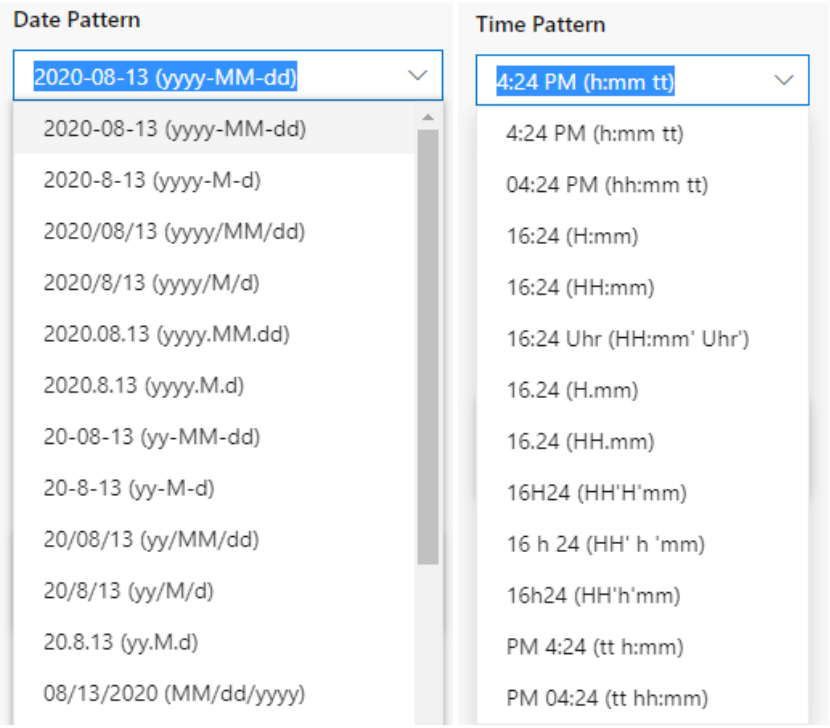
To limit the scope of identities available for selection to just those users added to the project, you can do so using the **Project-Scoped Users** group. For more information, see Manage your organization, Limit identity search and selection.

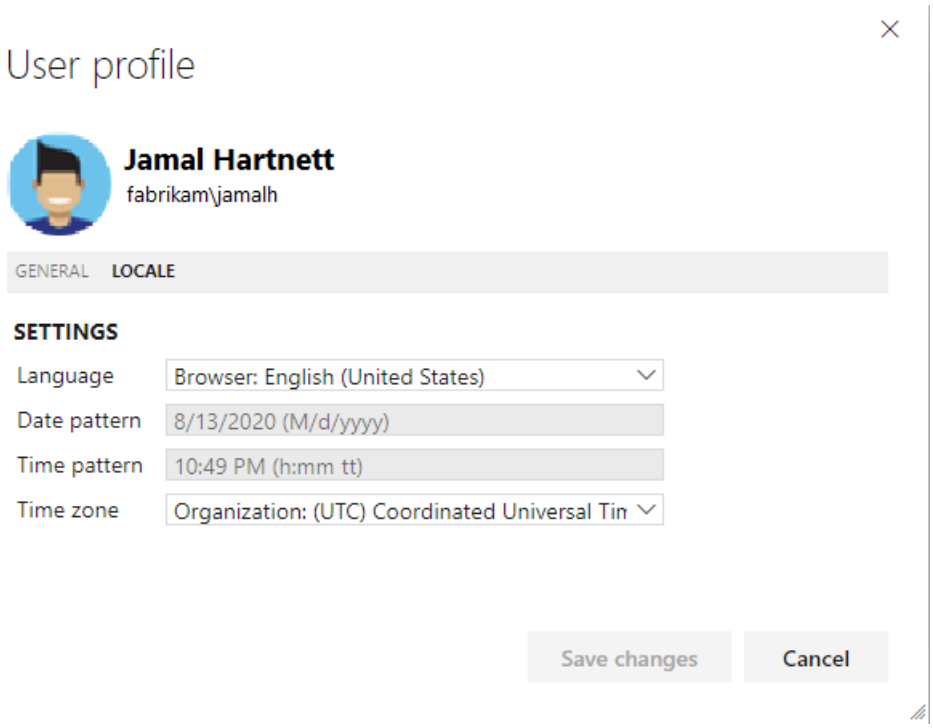
Date and identity fields

Several date and identity fields are set based on workflow states or transitions. Some fields, such as **Created By** and **Created Date**, are set by the system when a work item is added. Other fields, such as **Closed Date** and **Closed By**, are set through the workflow definition of the work item type. Additionally, customized work item types may have other rules defined that influence the date and identity field assignments.

Date and time pattern

The date and time pattern you enter for **DateTime** fields should match that which you select through your profile. To view or change your selection, see Set user preferences, Time and Locale.





State changes

The following XML syntax example illustrates rules that may be defined for a work item type that govern the values for select fields. Here, the **Resolved Date**, **Resolved By**, **Closed Date**, **Closed By**, **Activated Date**, and **Activated By** fields are set to `EMPTY` when a State value is set to New. The State value assignments are evaluated first, and then the transition assignments are evaluated next.

```
<WORKFLOW>
   <STATES>
     <STATE value="New">
       <FIELDS>
         <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ResolvedReason">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
           <EMPTY />
         </FIELD>
       </FIELDS>
     </STATE>
     <STATE value="Active">
       <FIELDS>
         <FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ResolvedBy">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ResolvedReason">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
           <EMPTY />
         </FIELD>
       </FIELDS>
     </STATE>
     <STATE value="Resolved">
       <FIELDS>
         <FIELD refname="Microsoft.VSTS.Common.ClosedDate">
           <EMPTY />
         </FIELD>
         <FIELD refname="Microsoft.VSTS.Common.ClosedBy">
           <EMPTY />
         </FIELD>
       </FIELDS>
     </STATE>
     <STATE value="Closed" />
   </STATES>
```

Activated By and Activated Date transition assignments

When the following transitions occur for a Bug work item, then the following assignments are made to the Activated By and Activated Date fields:

```
<TRANSITION from="" to="New">
<TRANSITION from="New" to="Active">
<TRANSITION from="New" to="Resolved">
<TRANSITION from="New" to="Closed">
<TRANSITION from="Resolved" to="Active">
<TRANSITION from="Closed" to="Active">
```

```
<FIELDS>
   <FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
       <COPY from="currentuser" />
           <VALIDUSER />
           <REQUIRED />
    </FIELD>
    <FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
        <SERVERDEFAULT from="clock" />
   </FIELD>
</FIELDS>
```

And when the following transitions occur for the Bug work item:

```
<TRANSITION from="Active" to="New">
<TRANSITION from="Active" to="Closed">
<TRANSITION from="Resolved" to="Closed">
```

Then the Activated By and Activated Date fields are set to `READONLY`.

```
<FIELD refname="Microsoft.VSTS.Common.ActivatedDate">
   <READONLY />
</FIELD>
<FIELD refname="Microsoft.VSTS.Common.ActivatedBy">
   <READONLY />
</FIELD>
```

Activated By/Date and Resolved By/Date fields

The system updates these fields—**Activated By**, **Activated Date**, **Resolved By**, and **Resolved Date**—when a change occurs based on corresponding workflow category states. When the workflow state changes to a *Proposed* state category, **Activated By** and **Activated Date** are updated. When the workflow state changes to a *Resolved* state category, **Resolved By** and **Resolved Date** are updated.

To learn more how workflow states map to state categories, see How workflow states and state categories are used in Backlogs and Boards.

> **NOTE**
> The logic governing the fields described here applies to Azure DevOps Services, Azure DevOps Server 2020.1 update, and later versions.

Because these fields reference the workflow state categories, custom workflow states that you add are referenced when updating the fields. To learn more about customization, see Customize the workflow for a process.

Additional notes:

- The fields get updated anytime a work item moves from any category state other than that being set. For example, if you update a work item from *New* to *Fixed*, the **Resolved By/Resolved Date** fields are updated. However, if you update from *Fixed* and *Ready for Testing*—which are in the same category state—the

Resolved By/Resolved Date fields aren't updated.

- When you transition backwards, such as going from a *Resolved* to an *Active* state, the system clears the values for **Resolved By/Resolved Date** fields. If you got from *Active* to *New*, the system clears the values for **Activated By/Activated Date** fields.
- Don't manually change the values for these fields. They are system fields that are governed by system rules. Any value you attempt to set will get over written.

Related articles

- How workflow states and state categories are used in Backlogs and Boards
- Query by date or current iteration
- Query quick reference
- Work item fields and attributes
- Query permissions

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Query by area or iteration path

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Area Path and Iteration Path are two fields that appear on the work tracking form for all work item types. You define them for a project—area paths and iteration paths—and then select the ones you want to associate with a team.

To better understand how to work with area and iteration paths, see About teams and Agile tools.

> **NOTE**
> The following macros are only supported from the web portal: **@CurrentIteration**, **@CurrentIteration +/- n**, **@Follows**, **@MyRecentActivity**, **@RecentMentions**, **@RecentProjectActivity**, and **@TeamAreas**. Queries that contain these macros won't work when opened in Visual Studio/Team Explorer, Microsoft Excel, or Microsoft Project.

Supported operators and macros

When creating queries and specifying the Area Path and Iteration Path fields, you can use the following operators:

OPERATOR	USE WHEN YOU WANT TO...
=	Specify one specific area or iteration path
< >	Filter out one, specific area or iteration path.
In	Filter for a set of area or iteration paths.
Not In	Exclude items that are assigned to a set of area or iteration paths.
Under	Specify all paths under a select area or iteration path.
Not Under	Exclude items assigned under a specific area or iteration path.

Along with these operators, you can use the following macros when you select the Iteration Path. For examples, see Query by date or current iteration.

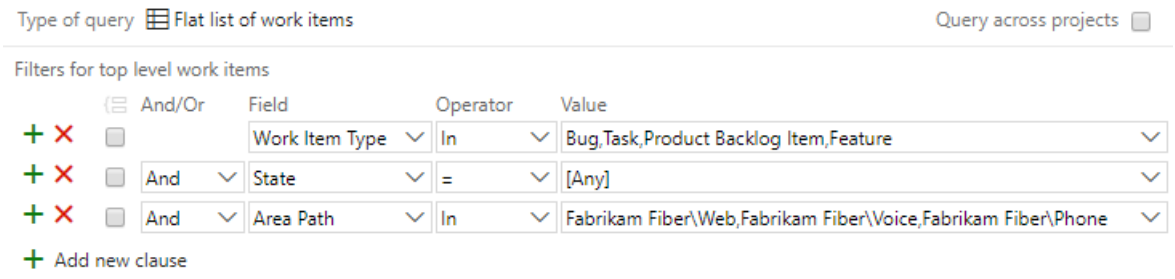
MACRO	USE WHEN YOU WANT TO...
@CurrentIteration	Specify the current iteration associated with the selected team context.
@CurrentIteration +/- n	Filter items based on assignment to a sliding window of sprints associated with the selected team context.

MACRO	USE WHEN YOU WANT TO...
@TeamAreas	Filter items based on area path(s) assigned to a specific team.

> **NOTE**
> The **@CurrentIteration** macro is supported for TFS 2015 and later versions. The **@CurrentIteration +/- n** and **@TeamAreas** macros are supported for Azure DevOps Server 2019 and later versions. These macros are only supported from the web portal. Queries that contain these macros won't work when opened in Visual Studio/Team Explorer, Microsoft Excel, or Microsoft Project.

Area path queries

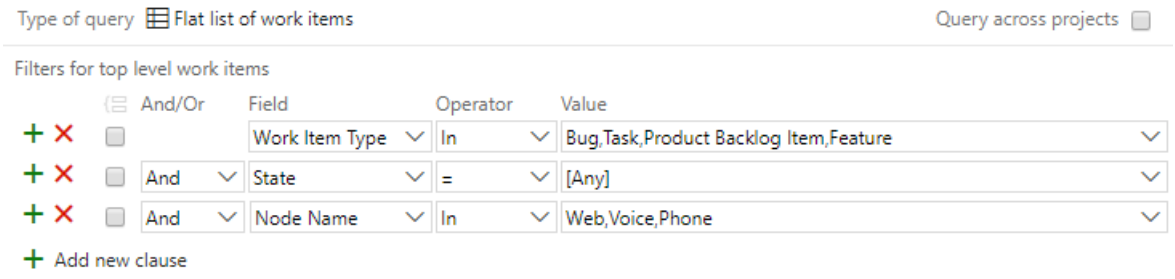
You can specify to filter for work items assigned to several area paths by using the **In** operator as shown in the following example.



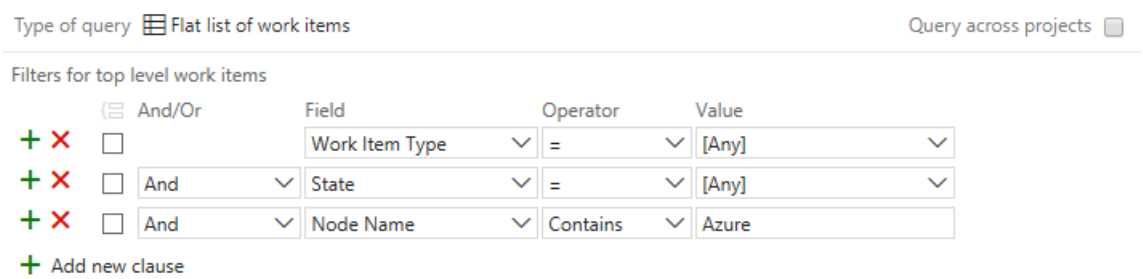
Node Name and keyword-based queries

Use the **Node Name** field to filter on work items assigned to area paths based on a keyword using the **Contains** operator. The **Node Name** specifies the last node of an **Area Path**, which corresponds to the last node in the tree hierarchy.

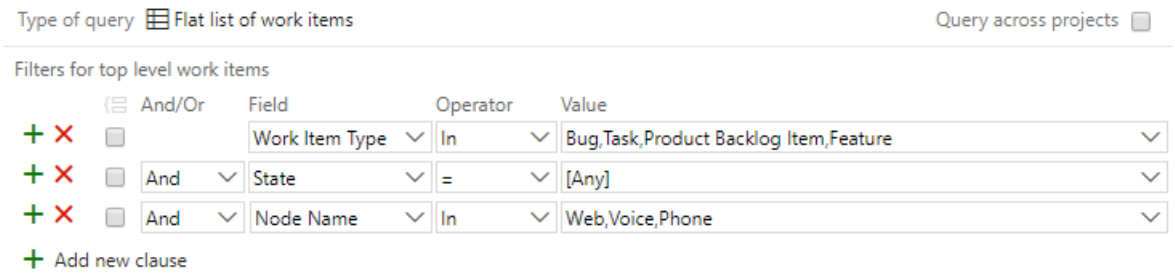
The following query yields the same result as the previous example.



In this example, the filter will return any work items assigned to an area path whose last node contains the word "Azure".

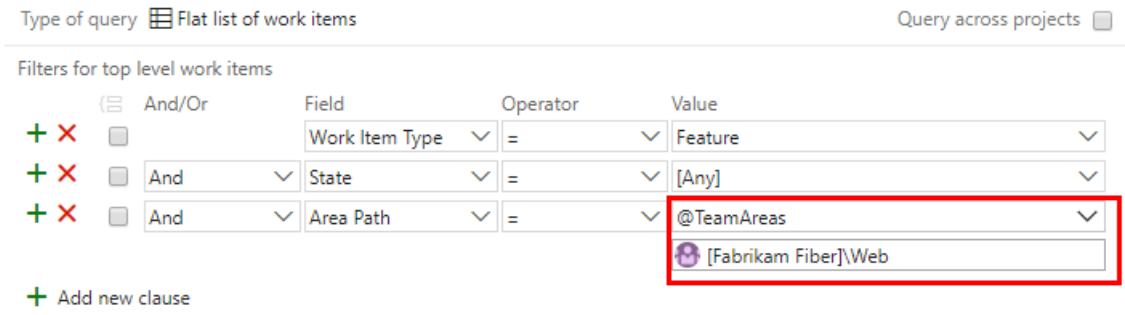


Here's another example that uses the **Node Name** and the **In** operator.



Team area path queries

Use the **@TeamAreas** macro to quickly find items assigned to the area paths assigned to a specific team. Specify the = operator. The Query Editor automatically prompts for you to enter the name of the team. You can add it by typing the name of the team and choosing the team value that appears in the search filter criteria.



Classification field reference

FIELD NAME	DESCRIPTION	REFERENCE NAME
Area Path	Groups work items into product feature or team areas. The area must be a valid node in the project hierarchy.	System.AreaPath
Iteration Path	Groups work items by named sprints or time periods. The iteration must be a valid node in the project hierarchy.	System.IterationPath

For each field, data path= `TreePath` , reportable type= `Dimension` , index attribute= `True` .

If you define a path name that is longer than 256 characters, you can't specify it in Microsoft Project. To avoid this problem, define path names of no more than 10 characters, and don't nest nodes more than 14 levels deep.

You can't apply most field rules to system fields, such as System.AreaPath and System.IterationPath fields. To learn more, see Rules and rule evaluation.

The following fields don't appear on work item forms but are tracked for each work item type. These fields provide a numeric value for each classification value that is defined for a project. You can use these fields to filter queries and create reports.

FIELD NAME	DESCRIPTION	REFERENCE NAME	DATA TYPE

FIELD NAME	DESCRIPTION	REFERENCE NAME	DATA TYPE
Area ID	The unique ID of the area to which this work item is assigned.	System.AreaId	Integer
Iteration ID	The unique ID of the iteration to which this work item is assigned.	System.IterationId	Integer
Node Name	The name of the last node of an area path. For example, if the area path is Project\A1\B2\C3, the node name is C3.	System.NodeName	String

The default reportable type is none. Area ID and Iteration ID are indexed, Node Name isn't. To learn more about field attributes, see Work item fields and attributes.

Related articles

- Query quick reference
- Define area paths and assign to a team
- Define iteration (sprint) paths and configure team iterations
- Set permissions and access for work tracking

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

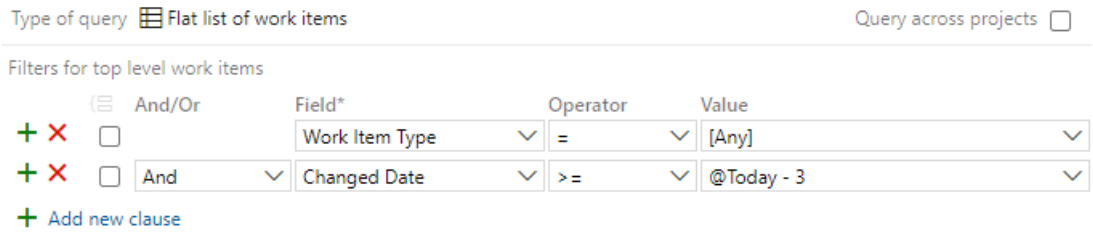
Query by date or current iteration in Azure Boards

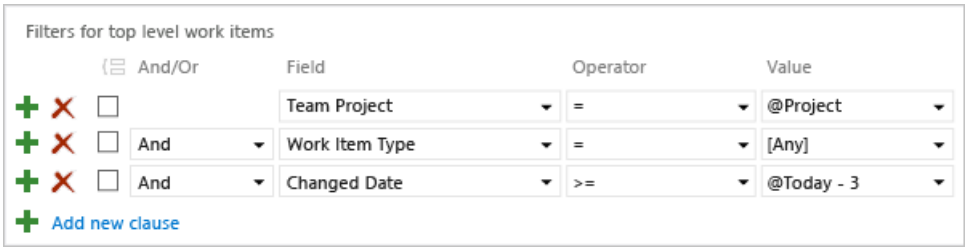
10/4/2022 • 11 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To list work items based on when they were created, closed, resolved, or changed—you can specify a date or use a supported macro. Use the **@Today** macro and specify a plus or minus number of days for relative dates. For queries that list work items based on their assignment to a team's current sprint, use **@CurrentIteration**.

For example, you can find work items that were modified in the last three days with the following query.





Also, you can use the **@CurrentIteration +/- *n*** macro to create queries based on a sliding window of team iterations.

Supported operators and macros

Query clauses that specify a **DateTime** field or the **Iteration Path** can use the operators and macros listed in the following table.

Data type	Supported operators and macros
DateTime	= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever. **Macros**: **@StartOfDay**, **@StartOfWeek**, **@StartOfMonth**, **@StartOfYear**, and **@Today**. You can use the **+/- *n*** with each of the supported macros.
TreePath	= , <> , Under, Not Under **Macros**: **@CurrentIteration**[2] and **@CurrentIteration +/- n**[3] valid with the **Iteration Path** field.
DateTime	

= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever
Macros: **@Today** which you can specify with ** +/- *n*** integer.

TreePath

= , <> , Under, Not Under **Macros**: **@CurrentIteration**[2] is valid with the **Iteration Path** field

---

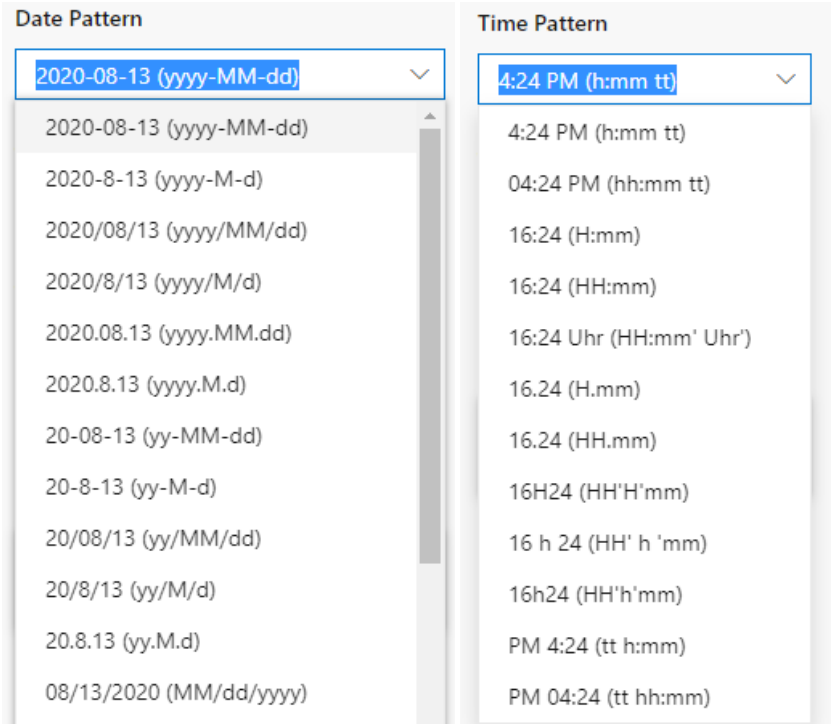
Notes:

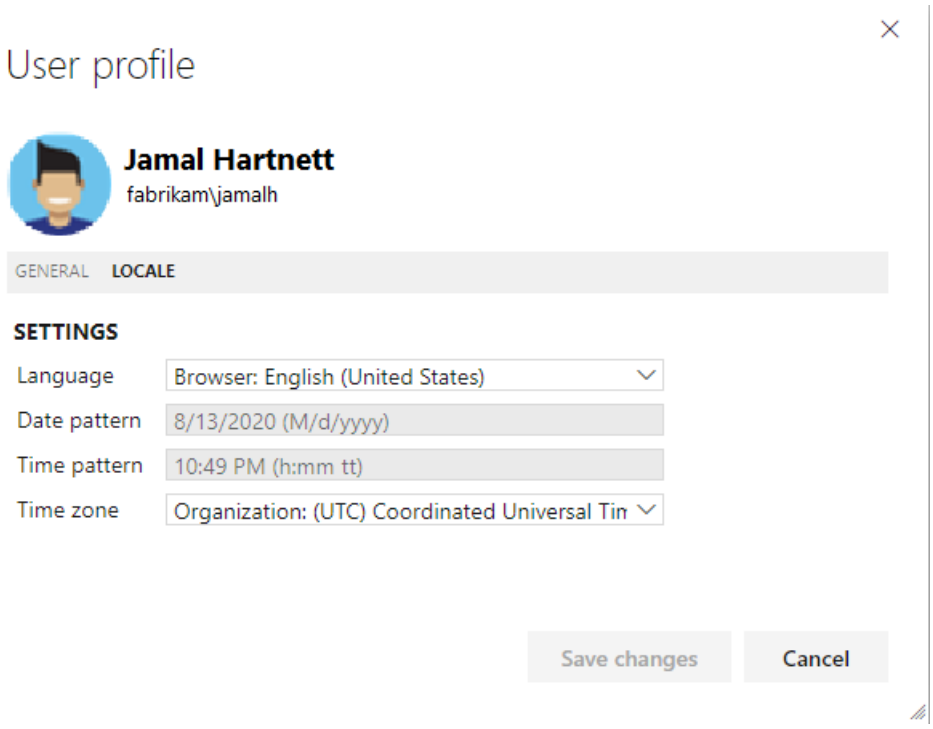
1. The **@StartOfDay**, **@StartOfWeek**, **@StartOfMonth**, **@StartOfYear** macros are supported for Azure DevOps Server 2019.1 and later versions, and only when run from the web portal.
2. The **@CurrentIteration** macro is supported for TFS 2015 and later versions, and only when run from the web portal.
3. The **@CurrentIteration +/- n** macro is supported for Azure DevOps Server 2019 and later versions, and only when run from the web portal.

> **TIP**
> The **WasEver** operator can be used with the **Iteration Path** field but only when defined through the WIQL syntax. For an example, see Work Item Query Language (WIQL) syntax reference.

Date and time pattern

The date and time pattern you enter for **DateTime** fields should match that which you select through your profile. To view or change your selection, see Set user preferences, Time and Locale.





Client restrictions on the use of the `@CurrentIteration` macros

You can use the **@CurrentIteration** in a query from the following clients:

- Web portal that connects to Azure Boards
- Web portal that connects to an on-premises Azure DevOps 2015 or later version
- Visual Studio 2015 or Team Explorer 2015 or later versions connected to Azure Boards or TFS 2015 or later versions.
- Using the REST API

You can use the **@CurrentIteration +/- *n*** macro in a query against Azure Boards, Azure DevOps Server 2019, and later versions, and with a REST API that includes the team as a parameter, for example, `@CurrentIteration('[Project]/Team')`.

An error occurs if you open a query that contains the **@CurrentIteration** macro in earlier versions of Visual Studio, or from Excel or Project. Also, you can't use the macro when copying or cloning test suites and test cases, defining alerts, or with REST APIs.

Date-based queries

You can filter for work items by the date on which they were changed or for a specific time period. Limiting the scope of your query can help with performance by only returning results that fit the date range that you include. If you're new to creating queries, see Use the query editor to list and manage queries.

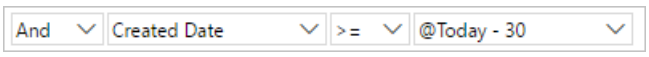
Not all fields are valid for all work item types. Jump to date fields for the set of fields you can include in queries and which work item types they apply to.

> **TIP**
> Remember to enter dates in the **Date Pattern** you set for your personal profile.

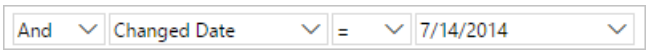
Filter for

Include these query clauses

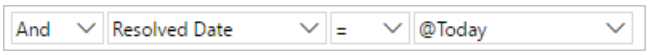
Items created in the last 30 days



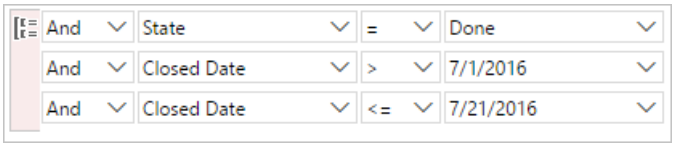
Items modified on a specific date



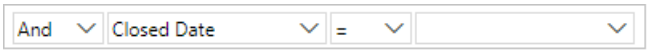
Items resolved today



Items closed within a specified time period



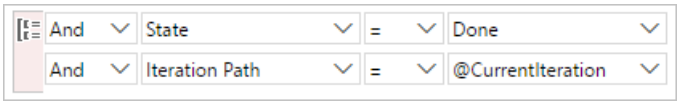
Items that haven't been closed (Closed Date is null)



Items whose status was updated within the last week



Items closed during the current sprint (the `@CurrentIteration` macro references the sprint defined for the current team context)



> **TIP**
> To understand how `AND/OR` clauses are grouped, see Create and save managed queries, Group clauses. To view the WIQL syntax for a query, install the WIQL query editor extension which allows you to see the WIQL version of any query editor entry.

Create start of day, week, month, or year date-based queries

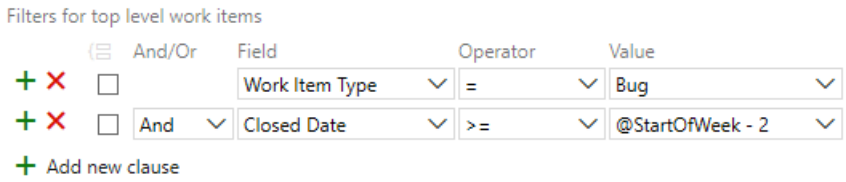
The following examples show how to use the **StartOf...** macros to filter for work items with various offsets. For more examples of using these macros, see WIQL syntax.

> **NOTE**
> Requires Azure DevOps Server 2019 Update 1 or later version.

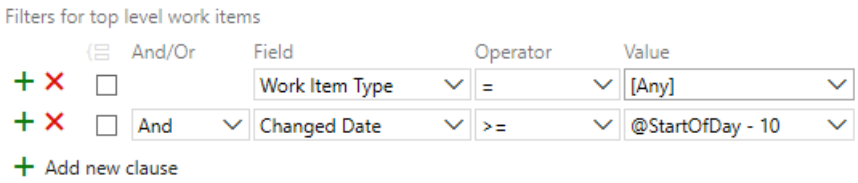
Filter for

Include these query clauses

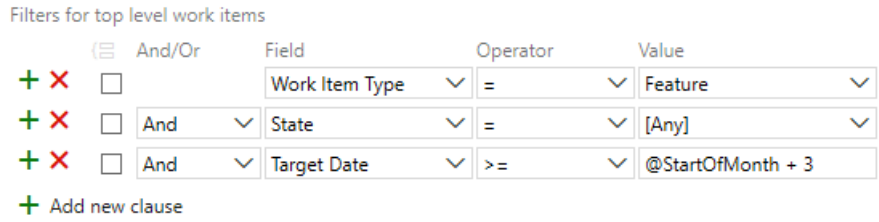
Bugs closed in the last two weeks



Items modified in the last 10 days



Features scheduled to be completed in the next three months



Not all fields are valid for all work item types. Jump to date fields for the set of fields you can include in queries and which work item types they apply to.

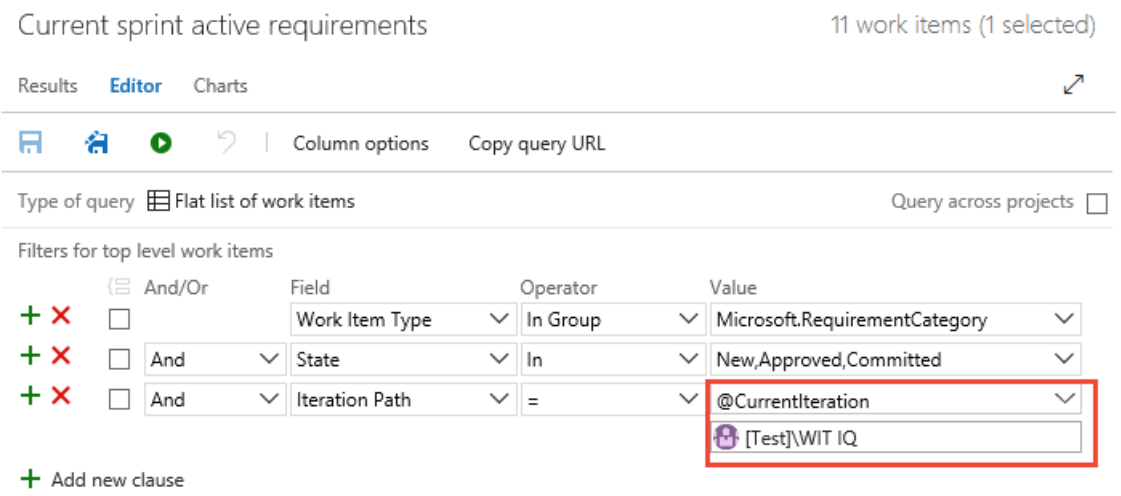
Create queries for your team's current iteration

If your team follows Scrum processes, you schedule work to be completed in sprints. You can track the progress of requirements, bugs, and other work to be completed in the current sprint using the **@CurrentIteration** macro.

Any item assigned to a sprint that corresponds to the current iteration path for the team is found. For example, if a team is on Sprint 5, the query returns items assigned to Sprint 5. Later, when the team is working in Sprint 6, the same query returns items assigned to Sprint 6.

> **NOTE**
>
> For the **@CurrentIteration** macro to work, the team must have selected an **Iteration Path** whose date range encompasses the current date. For details, see Define iteration paths (also referred to as sprints) and configure team iterations. Also, queries that contain this macro are only valid when run from the web portal.
>
> See also Client restrictions on the use of the @CurrentIteration macros later in this article.

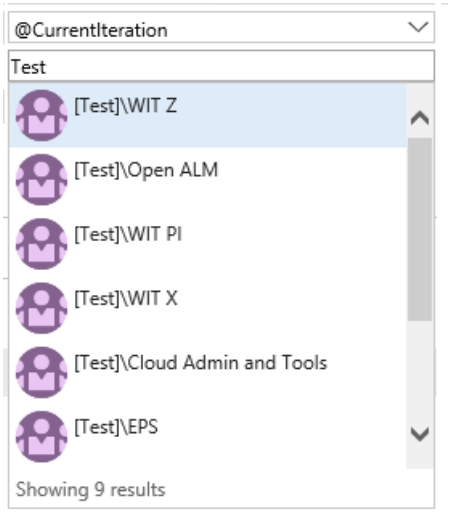
Azure Boards adds a team parameter when you select the **@CurrentIteration** or **@CurrentIteration +/- *n*** macros. The team parameter is derived from your current team context.



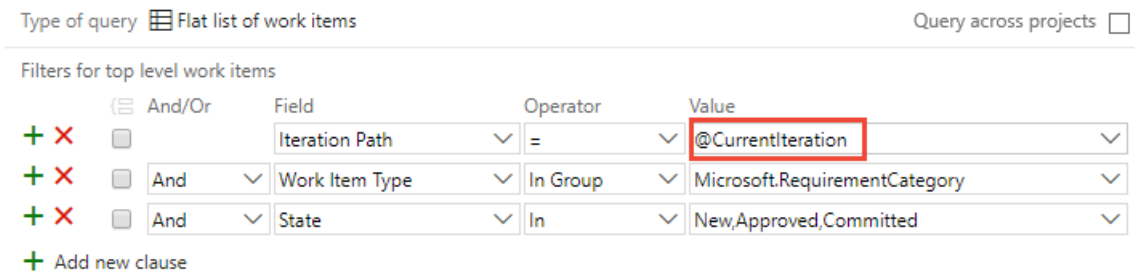
@CurrentIteration macro with team parameter" />

> **TIP**
>
> If the **@CurrentIteration** macro isn't working, check that the expected iteration is selected for your team and that dates have been set for it.

To change the team parameter the system automatically sets, you choose it by typing the name of the team into the parameter field added below the **@CurrentIteration** macro.



Before creating or updating a query to use the **@CurrentIteration** macro, make sure you select your team. The **@CurrentIteration** macro references the current team selected in the web portal.



@CurrentIteration macro" />

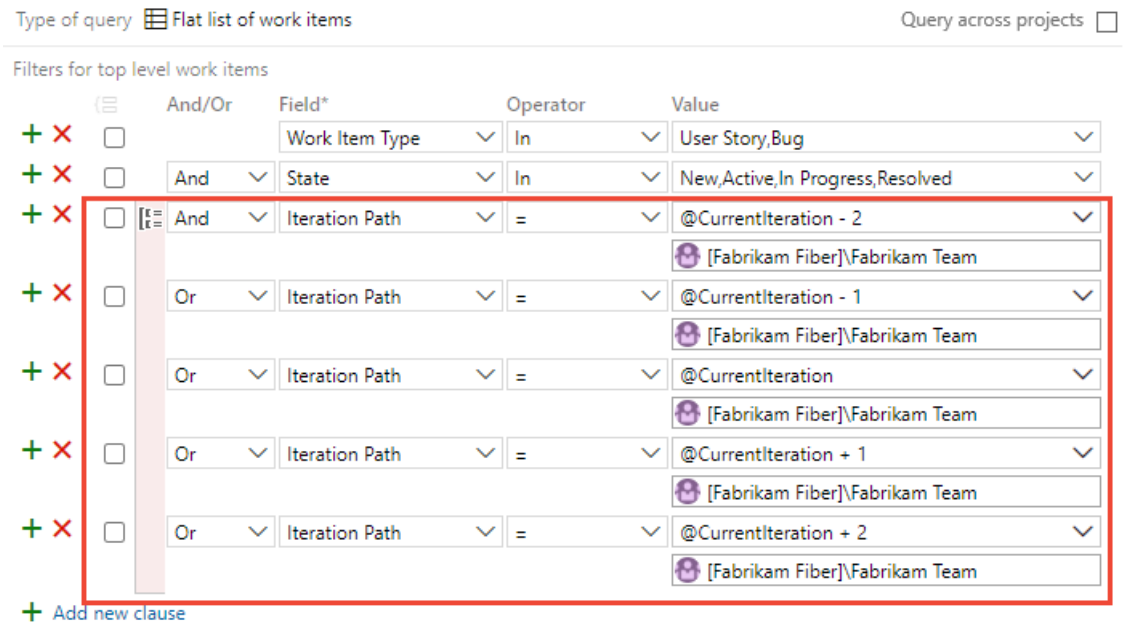
Create a sliding window of your team's iterations query

Use the **@CurrentIteration +/- *n*** macro when you want to track the work a team has planned for upcoming

sprints and for understanding work that wasn't completed in previous sprints.

> **NOTE**
>
> For the **@CurrentIteration +/- *n*** macro to work, the team must have selected **Iteration Paths** that meet the **+/- *n*** criteria and date ranges encompass the current date for the **@CurrentIteration**. For details about team selection of Iteration Paths, see Define iteration (sprint) paths and configure team iterations.
>
> See also Client restrictions on the use of the @CurrentIteration macros later in this article.

Here we show how to list all User Stories and Bugs that are assigned to the sliding window that spans the last two, the current, and the next two sprints selected for the *Cloud Admin and Tools* team.



To use this macro, the specified team must have selected a set of sprints that span the **+/- *n*** value entered for the macro.

List work items moved out of a sprint

You can list work items that were defined for a sprint but later moved out using a query with a clause that contains the **Was Ever** operator for the **Iteration Path**. You can only construct this query using the WIQL syntax. You can edit the WIQL syntax in the Query Editor by installing the Wiql Editor Marketplace extension.

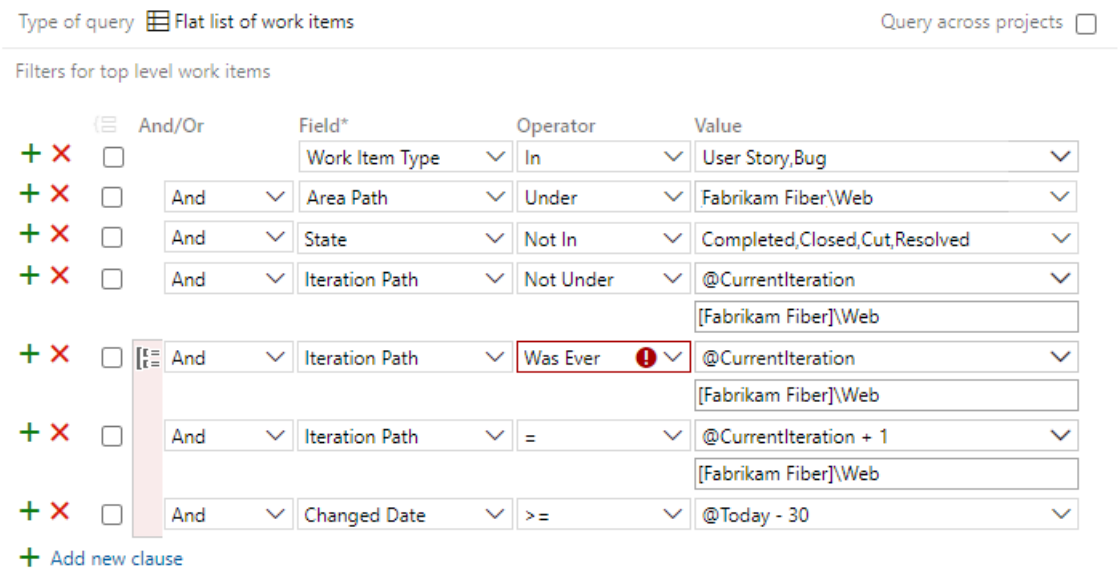
For example, the following syntax queries for work items that meet the following criteria:

1. Defined in the current project
2. Work item type equals User Story or Bug
3. Work items are under the Fabrikam Fiber Web team Area Path
4. Work items aren't in a Closed, Completed, Cut, or Resolved state
5. Not in the current iteration path for the Fabrikam Fiber Web team
6. But were assigned to the current iteration path for the Fabrikam Fiber Web team
7. Are now assigned to the Current iteration +1 for the Fabrikam Fiber Web team
8. And were changed within the last 30 days (the length of the sprint)

```
SELECT
    [System.Id],
    [System.WorkItemType],
    [System.AssignedTo],
    [System.Title],
    [System.State],
    [System.Tags],
    [System.IterationPath],
    [System.AreaPath]
FROM workitems
WHERE
    [System.TeamProject] = @project
    AND [System.WorkItemType] IN ('User Story', 'Bug')
    AND [System.AreaPath] UNDER 'FabrikamFiber\Web'
    AND NOT [System.State] IN ('Completed', 'Closed', 'Cut', 'Resolved')
    AND NOT [System.IterationPath] = @currentIteration('[FabrikamFiber]\Web <id:cdf5e823-1179-4503-9fb1-
a45e2c1bc6d4>')
    AND (
        EVER (
            [System.IterationPath] = @currentIteration('[FabrikamFiber]\Web <id:cdf5e823-1179-4503-9fb1-
a45e2c1bc6d4>')
        )
        AND [System.IterationPath] = @currentIteration('[FabrikamFiber]\Web <id:cdf5e823-1179-4503-9fb1-
a45e2c1bc6d4>') + 1
        AND [System.ChangedDate] >= @today - 30
    )
ORDER BY [System.Id]
```

The Query Editor view of the syntax appears as shown.

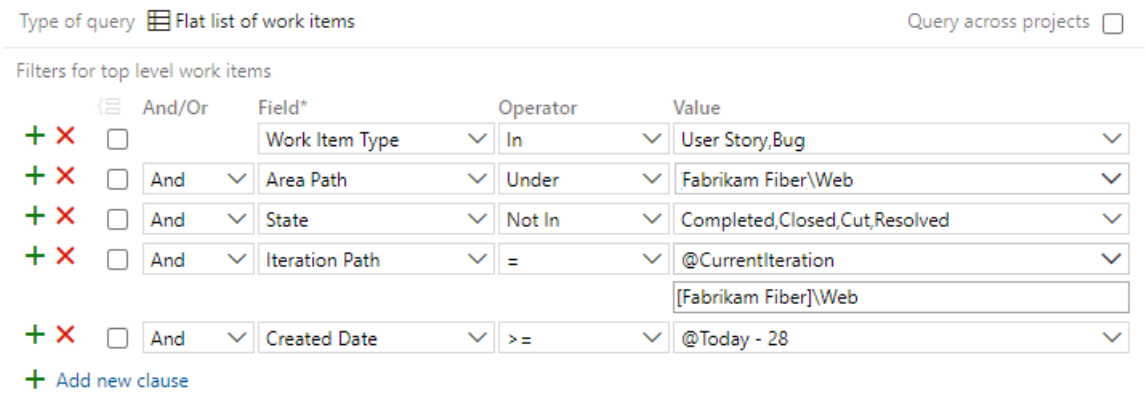
> **NOTE**
>
> The Query Editor displays a information icon next to the **Was Ever** operator, indicating an issue with the clause. However, the query will still run and you can create query charts. However, to modify the query, you need to use the WIQL editor.



List work items added to a sprint after the start date

To list newly created work items added to a sprint after its start date, use a query similar to the one shown in the following image. This query works by filtering for work items assigned to the current sprint but were created

after the start of the sprint date. In this example, this is achieved with the clause **created Date = @Today - 28**.



For other options for querying changes to sprint scope, see About Sprints, Scrum and project management, Sprint scope change.

Date and Iteration Path fields

You can use date fields to filter your queries. Some of these fields are populated with information as a work item progresses from one state to another. Several of these fields don't appear on the work item form, but are tracked for those work item types listed in the following table.

Field name

Description

Work item types

Activated Date (Notes 1, and 2)

The date and time when the work item was created or when its status was changed from closed, completed, or done to a new or active state.
Reference name=Microsoft.VSTS.Common.ActivatedDate, Data type=DateTime

Bug, Change Request, Epic, Feature, Issue, Product Backlog Item, Requirement, Review, Risk, Shared Step, Task, Test Case, User Story

Change Date

The date and time when a work item was modified.
Reference name=System.ChangedDate, Data type=DateTime

All

Closed Date (Note 2)

The date and time when a work item was closed.
Reference name=Microsoft.VSTS.Common.ClosedDate, Data type=DateTime

All

Created Date

The date and time when a work item was created.
Reference name=System.CreatedDate, Data type=DateTime

All

Due Date

The forecasted due date for an issue to be resolved.
Reference name=Microsoft.VSTS.Scheduling.DueDate, Data type=DateTime

Issue (Agile)

Finish Date (Note 3)

The date and time when the schedule indicates that the task will be completed.
Reference name=Microsoft.VSTS.Scheduling.FinishDate, Data type=DateTime

Requirement, Task, Test Plan, User Story

Iteration Path

Groups work items by named sprints or time periods. The iteration must be a valid node in the project hierarchy. You define iteration paths for a project and select iteration paths for a team define iteration paths for a project and select iteration paths for a team.
Reference name=System.IterationPath, Data type=TreePath

All

Resolved Date (Notes 1 and 2)

The date and time when the work item was moved into a Resolved state.
Reference name=Microsoft.VSTS.Common.ResolvedDate, Data type=DateTime

Bug, Change Request, Epic, Feature, Issue, Product Backlog Item, Requirement, Review, Risk, Shared Step, Task, Test Case, User Story

Start Date (Note 3)

The date and time when the schedule indicates that the task will start.

> **NOTE**
> Delivery Plans uses the **Start Date** and **Target Date** to show the span of Features, Epics, and other portfolio backlog items.

Reference name=Microsoft.VSTS.Scheduling.StartDate, Data type=DateTime

Epic, Feature, Requirement, Task, Test Plan, User Story

State Change Date

The date and time when the value of the State field changed.
Reference name=Microsoft.VSTS.Common.StateChangeDate, Data type=DateTime

All

Target Date

The date by which a feature, work item, or issue is to be completed or resolved.

> **NOTE**
> Delivery Plans uses the **Start Date** and **Target Date** to show the span of Features, Epics, and other portfolio backlog items.

Reference name=Microsoft.VSTS.Scheduling.TargetDate, Data type=DateTime

Epic, Feature

Notes:

1. See also Query by assignment or workflow changes, Date, and Identity fields.
2. For these fields to be defined for a WIT, they must be included in the `WORKFLOW` section of the WIT definition. For example, this syntax is included within the `FIELDS` definition when transitioning to a Resolved state:

```
<FIELD refname="Microsoft.VSTS.Common.ResolvedDate" />
   <SERVERDEFAULT from="clock"  />
</FIELD >
```

3. **Start Date** and **Finish Date** fields are calculated if you create a project plan in Microsoft Project and then synchronize that plan with tasks that are stored in Azure Boards. These fields might not appear on the work item form, but are calculated for the backlog items and tasks that are linked to backlog items. You can view their read-only values in results from a query or from Microsoft Excel.

> **IMPORTANT**
> Microsoft Project Integration and the **TFSFieldMapping** command are not supported for:
>
> - Visual Studio 2019 and Azure DevOps Office® Integration 2019
> - Azure DevOps Server 2019 and later versions, including Azure DevOps Services.
>
> However, full support for Microsoft Excel integration is maintained and supports bulk import and update of work items. Alternatives to using Microsoft Project include the following:
>
> - Delivery plans
> - A Marketplace extension such as Project Connect or GANTT chart.

Related articles

To query for items based on text entered in the History field, see History and auditing.

- Query by assignment or workflow changes
- Define iteration (sprint) paths and configure team iterations
- Create managed queries with the query editor
- Query operators & macros
- Query permissions
- Work item fields and attributes
- Work item query language (WIQL) syntax
- Query quick reference
- Work item field index

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Add work item tags to categorize and filter lists and boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Tagging work items helps you quickly filter the product backlog or a work item query by categories that you define. A tag corresponds to a one or two keyword phrase that you define and that supports your needs to filter a backlog or query, or define a query.

Tags are a better choice to filter work items than using text strings as described in Guidance to create high-performing queries.

You can add and modify tags from the web portal, from Team Explorer plug-in for Visual Studio. Also, you can open a query in Excel to modify tags in bulk.

> **NOTE**
> Tags are a shared resource, they're associated with a project and not a team. If your project contains multiple teams, all teams will add to and work from the same set of tags.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add new tags to add to work items, you must have **Basic** access or higher and have the project-level **Create new tag definition** permissions set to **Allow**. By default, the **Contributors** group has this permission set. Even if the permission is explicitly set for a **Stakeholder**, they won't have permission to add new tags, as they are prohibited through their access level. To learn more, see Stakeholder access quick reference.
- All project members, even those who belong to the **Readers** group, can email work items.

> **NOTE**
> Users with **Stakeholder** access for public projects are allowed to add new tags.

Add tags to a work item

Tags should be 400 characters or less and not contain separators such as a `,` (comma), `;` (semicolon), or other formatting character.

> **TIP**
> We recommend that you don't use the `@` character in a tag. Tags that start with the `@` character can't be used in a work item query. The `@` character signifies a macro within a query and therefore the tag isn't recognized as a tag.

From the web portal, open a work item and add a tag. Choose **Add tag** and type your keyword. Or, select from the list of previously assigned tags.



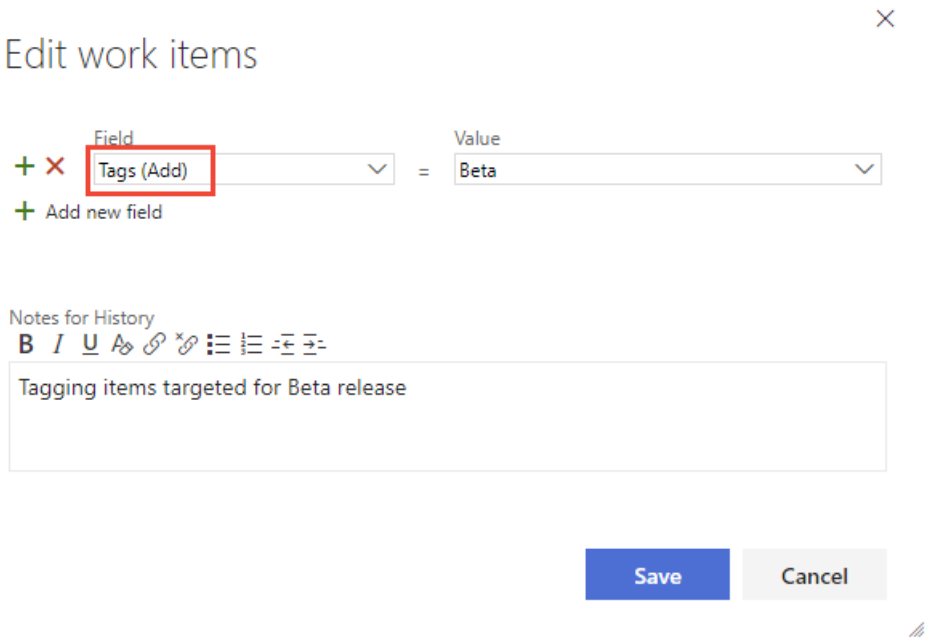
To add several tags at one time, type a comma between tags. Tags are case sensitive.

Tags that appear in the tag bar are already assigned to the work item. To unassign a tag, choose the x on the tag, Web ×.

> **NOTE**
> By default, all Contributors and Stakeholders of public projects are granted permissions to add new and existing tags. Stakeholders in private projects can add tags that are already defined, but not add new tags. To grant or restrict permissions to create new tags, you set the permission **Create tag definition** at the project-level. To learn more, see Change project-level permissions.

Bulk add or remove tags

You can bulk update work items to add or remove tags from the web portal. You bulk modify tags in the same way as you bulk modify other fields using the web portal. Or, you can use Excel to bulk add or remove tags.



NOTE

Bulk modify of tags from the Visual Studio or other supported clients isn't supported.

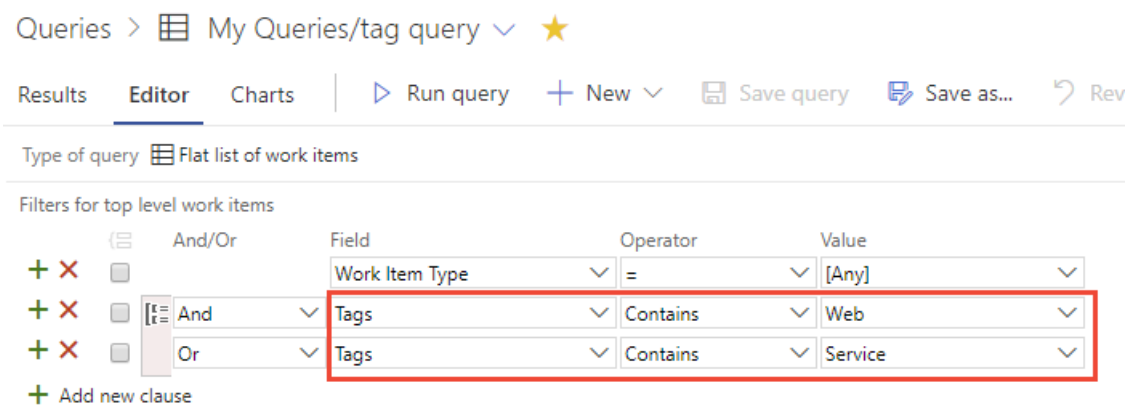
Query for work items based on tags

To query work items based on tags, add a clause for each tag you want to use to support your query.

TIP

You can use the **Contains** or **Does Not Contain** operators. Tags that start with the `@` character can't be used in a work item query as the query editor interprets the `@` character as a macro. To learn more about queries, see Create managed queries.

For example, here we query for all work items that are tagged either `Web` or `Service`.



TIP

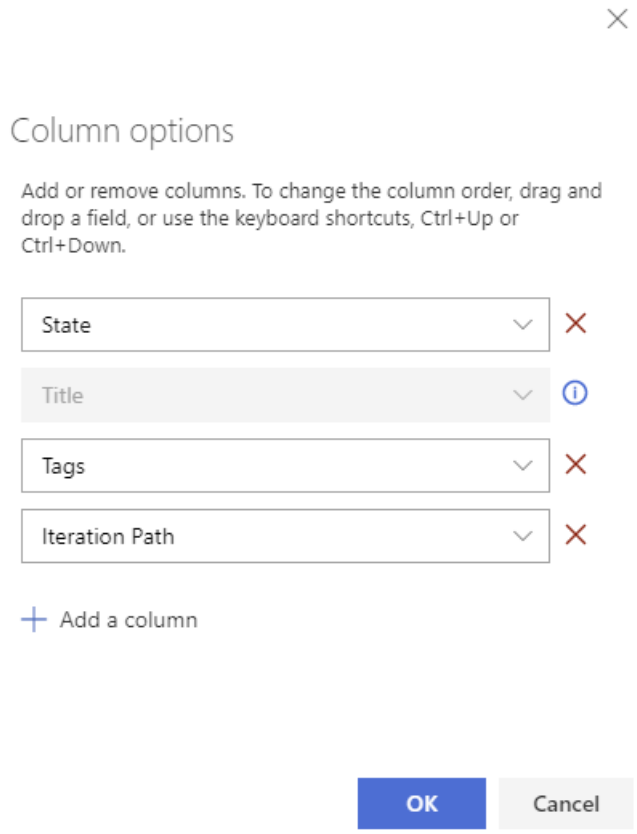
To understand how `AND/OR` clauses are grouped, see Create and save managed queries, Group clauses. To view the WIQL syntax for a query, install the WIQL query editor extension which allows you to see the WIQL version of any query editor entry.

NOTE

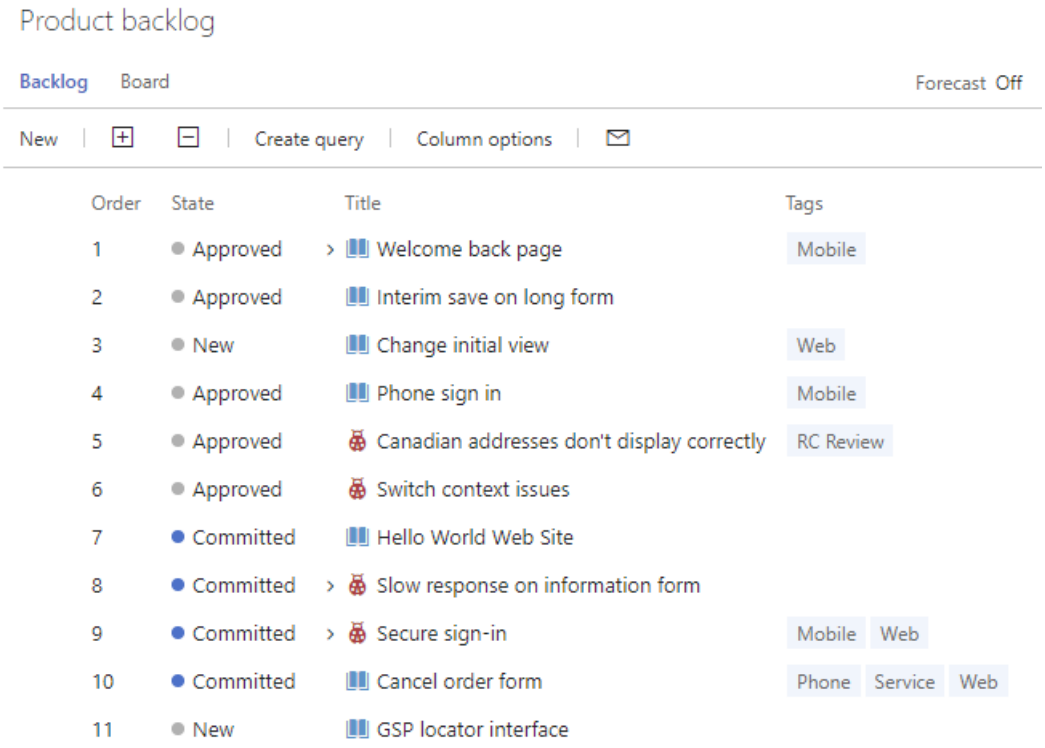
The ability to query for work items that don't have any tags attached to them is not a supported feature. If you'd like to up vote the request to support this feature, you can do so on our Developer Community page, Be able to search for empty tags.

Show tags in your backlog or query results

Choose **Column Options** to add the **Tags** field to the product backlog or a work item query. If the option doesn't appear, choose **More commands** ··· to select it from the menu of options.



All tags that have been added to the listed work items appear.

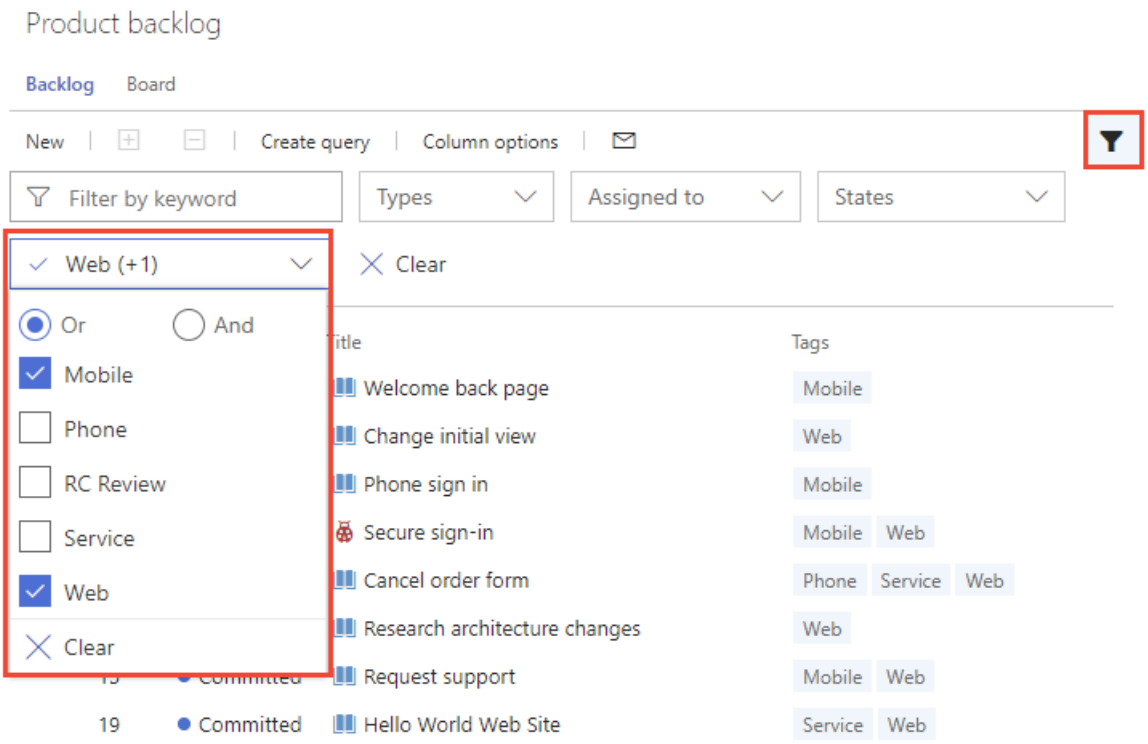


Filter lists using tags

From the web portal, you can filter backlogs, boards, and query results using tags.

Begin by choosing **Filter** .

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to run a logical OR for all the tags you selected. Or, choose the **AND** option to run a logical AND on all the selected tags.



Delete, remove, or manage tags

You can't delete a tag itself. However, if you delete a tag from all work items to which it's currently assigned, the

system will delete the tag. The system automatically deletes unassigned tags after three days of disuse.

If you misspell a tag, don't assign the misspelled tag to any work item and the system will automatically delete it within three days.

Another option is to install the Marketplace Tags Manager, which adds a **Tags** page under **Boards** or **Work** to manage tags.

Color-code tags on boards

You can highlight tags on Kanban board cards by color-coding them. These colors only appear on the Kanban board that you configure. They don't appear on backlogs or taskboards. To learn more, see Customize cards, color-code tags.

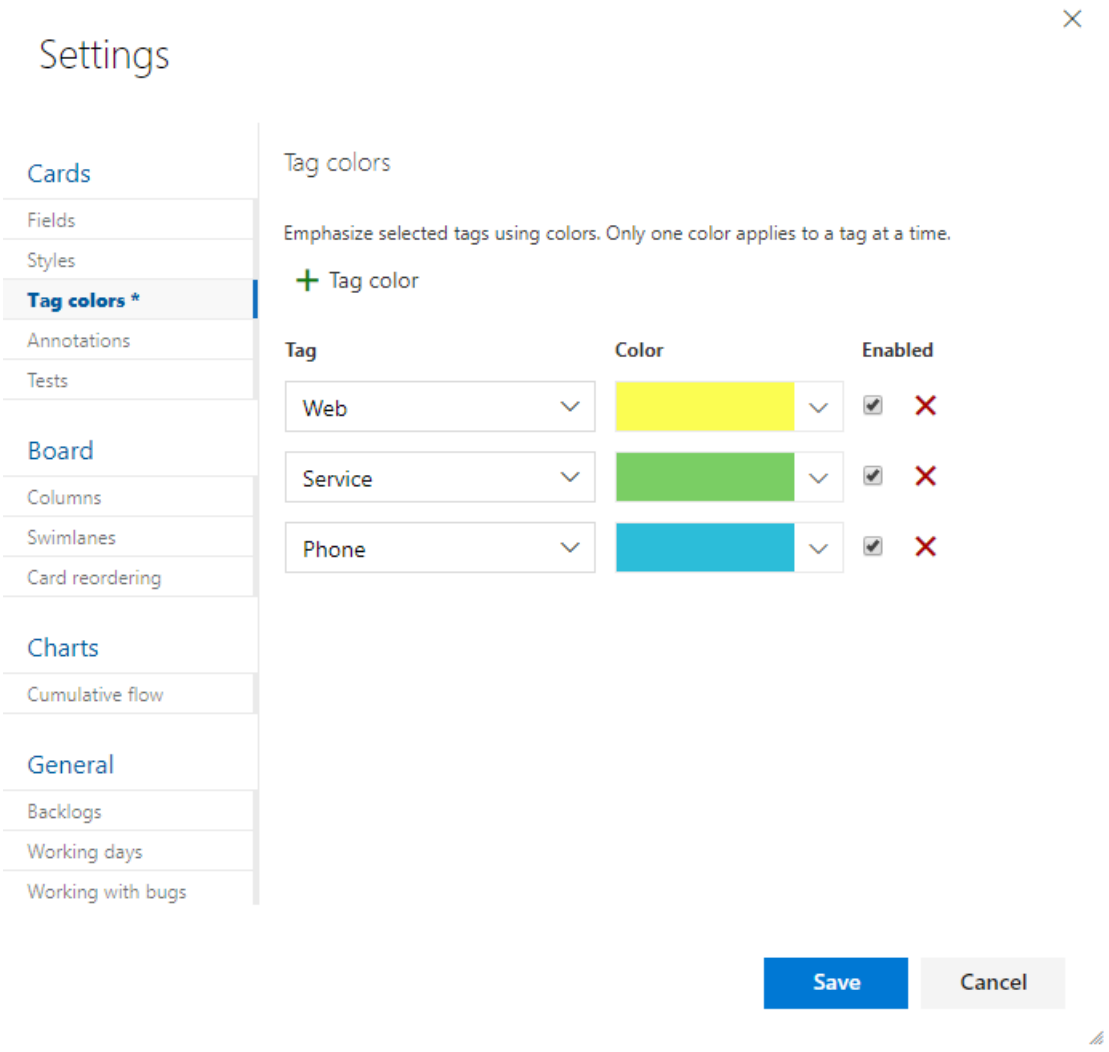
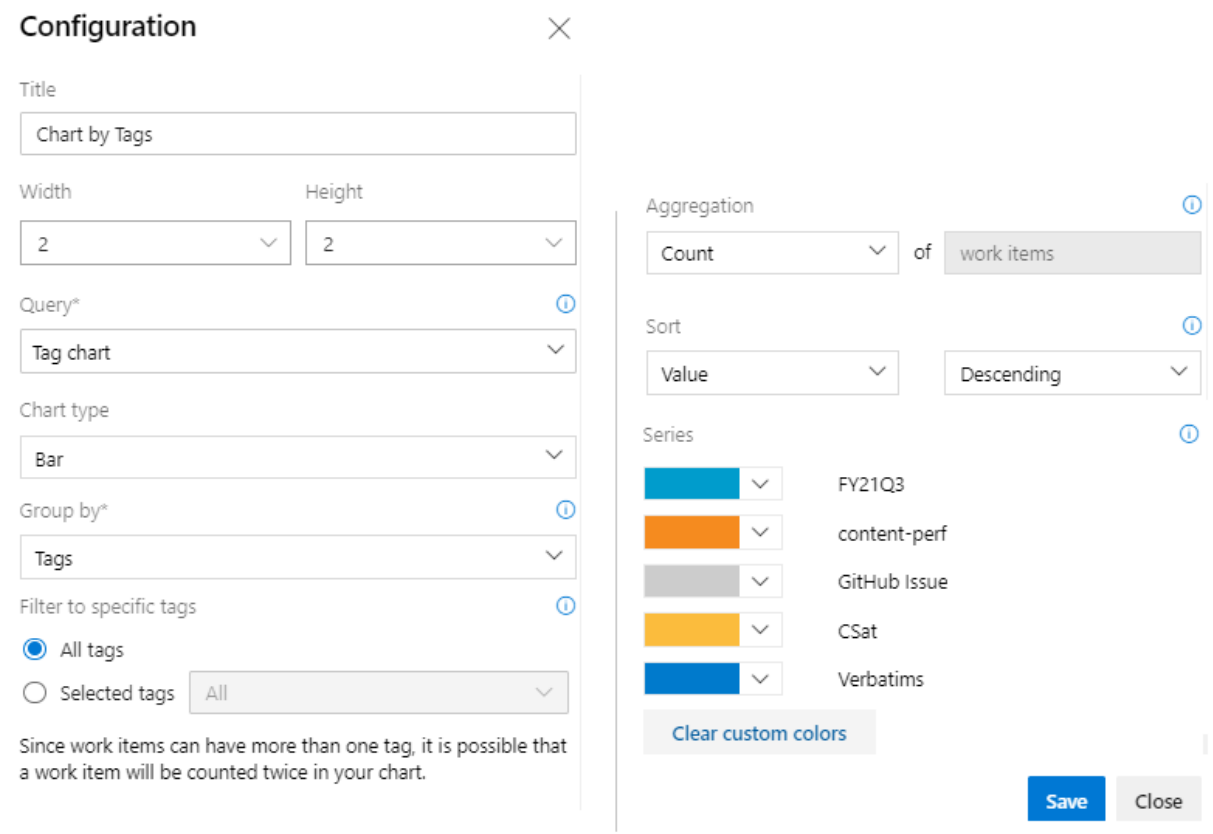


Chart work items and group by tags

> **NOTE**
> You can't group a query-based chart by tags, however, you can group a **Chart for Work Items** widget by tags that you add to a dashboard. This feature is in public preview. To enable it, see Manage or enable features and turn on **Enable group by tags for work item chart widget on dashboard**.

To group a **Chart for Work Items** widget by tags, complete the same steps provided in Track progress with status and trend query-based charts, Add a chart widget to a dashboard. Make sure that your flat-list query contains **Tags** in the query clause or as a column option. Then, choose **Tags** for the **Group by** selection. To filter

the chart to show only some tags, choose the **Selected tags** radio button and then choose the tags you want the chart to display.



Related articles

- Best tool to add, update, and link work items
- Use the query editor to list and manage queries
- Show tags on cards
- Bulk modify work items from the web portal
- Bulk modify work items from Excel

Marketplace extension

- Marketplace Tags Manager

Limits on the number of tags

While no hard limit exists, creating more than 100,000 tags for a project collection can negatively impact performance. Also, the autocomplete dropdown menu for the tag control displays a maximum of 200 tags. When more than 200 tags are defined, begin typing to cause the tag control to display relevant tags.

You can't assign more than 100 tags to a work item or you'll receive the following message:

```
TF401243: Failed to save work item because too many new tags were added to the work item.
```

Save the work item with the tags (100 or less) that you've added, and then you can add more tags.

Limit queries to fewer than 25 tags. More than that amount and the query will likely time out.

Add tags to the default column view on the product backlog

To add the **Tags** field as a column field for the product backlog, you modify the ProcessConfiguration file to include `System.Tags`. To learn how, see the Process configuration XML element reference.

Query work item history and discussion fields in Azure Boards

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The history of a work item tells you who opened the item, what changed, and why. This information helps you track how an item changes over time. When you enter information in the history field, provide as much information as possible to help the next work item owner understand what has happened and what they have to do.

> **NOTE**
> There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the History field. The full content of the text entered into the Discussion text box is added to the History field.

Supported operators and macros

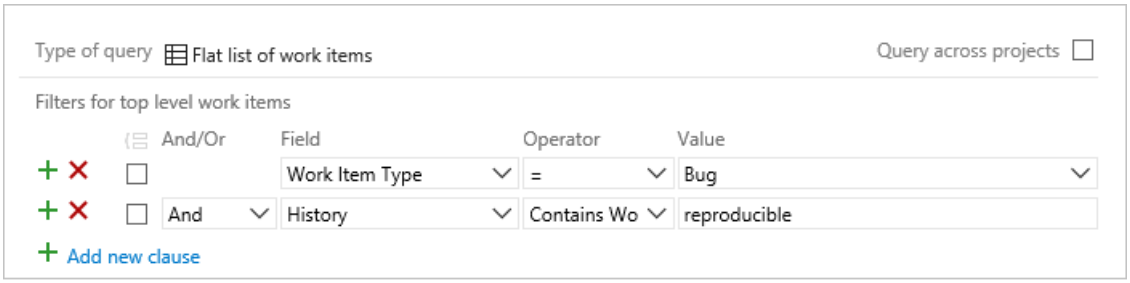
Query clauses that specify the **History** field can use the **Contains Words** and **Does Not Contain Words** operators. Search against an exact phrase or to use the wildcard character, *. You can only use the wildcard character at the end of a partial word or phrase.

The **History** field is automatically indexed for full-text search when full-text search is available. See Full-Text and partial word searches

Query a work item's history

You can use either the web portal or Team Explorer to view the history of a work item or find work items based on the contents of the **History** field. When you run a search on the contents of the **History** field, it returns only work items that have changes recorded in that field. That is, it doesn't register changes that were made to text in other fields.

- Browser
- Visual Studio 2015



List items based on the contents of the History field

You use the query editor to add the **History** field to a query clause. Comments entered into the **Discussion** area are queryable. Change history entries, such as which fields were changed, aren't queryable. To quickly find items based on words entered into the Discussion area, or **Description** or other rich-text fields, consider using

work item search.

You can filter for work items by the date on which they were changed or for a specific time period. If you limit the scope of your query, it can help with performance by only returning those results that fit the date range that you want to include.

Filter for	Include these query clauses
Items whose History field contains the word "reproducible"	`History Contains Words reproducible`
Items whose History field doesn't contain the word "beta"	`History Does Not Contain Words beta`
Items that contain the phrase "stack traces" and were closed but reactivated	`History Contains Words stack traces` `And State Was Ever Closed` `And State <> Closed`
Items closed within a specified time period	`State = Done` `And Closed Date > 7/1/2015` `And Closed Date <= 7/21/2015`
Items I've been associated with	`History Contains Words MyName` `Or Assigned To Was Ever _ @Me`

Tips for using the query editor

- Type the complete word or phrase that is specified in the **History** field of those work items that you want to find.
- Enter the full text for the word that you want to search. The **History** field is indexed for full-text search. If you enter only a partial word, the query won't return work items that contain the full word. For example, if the **History** field contains the phrase *reproducible behavior* and you search for *repro*, the work item won't be found. However, if you search for the complete word *reproducible*, the work item will be found. You can also search for the string with a wild card, such as `repro*`.
- The query editor ignores common words or stop words as defined in Configure and Manage Stopwords and Stoplists for Full-Text Search.
- On the query editor toolbar, choose or icon and confirm that your query returns expected results.
- If you don't receive the results you expect, adjust the word or phrase that you entered, and run the query again.

View the history of work items

An entry is made to the **History** field each time a work item is saved. To view the history of changes, open an existing work item, and then choose the or **History** tab, or for some work item types, choose the **Details** tab.

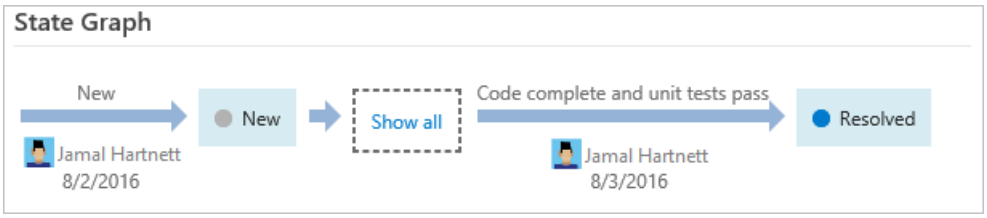
The history details shown depend on the platform, version, and client.

- Browser

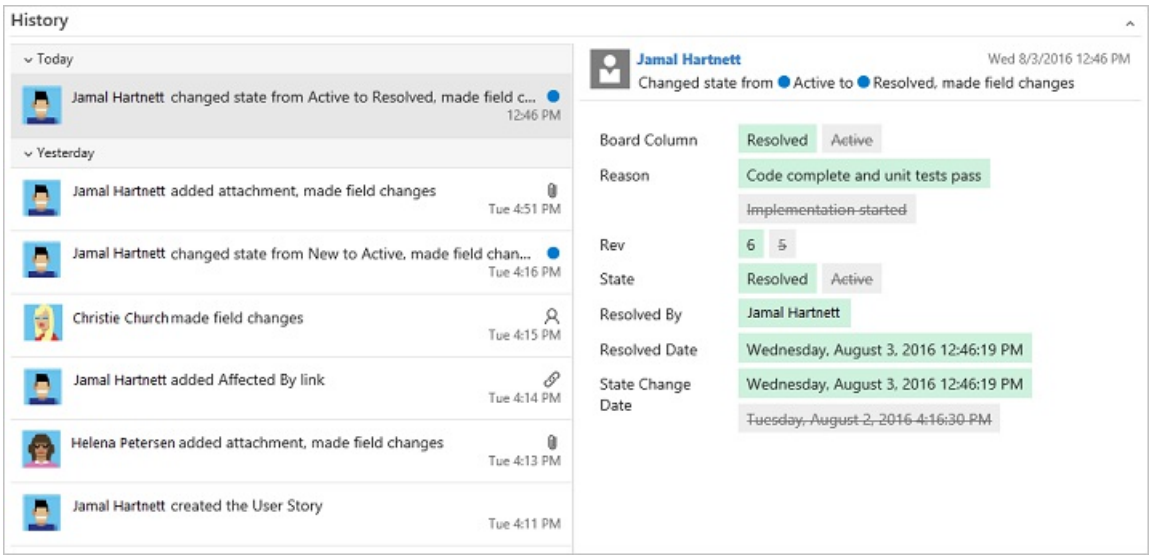
- Visual Studio



The state change history diagram appears first. To see the entire history of state changes, choose **Show all**.



Choose an entry in the left pane to view the details of changes made.

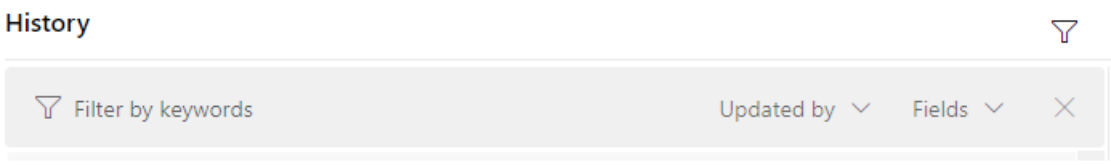


Filter the history view

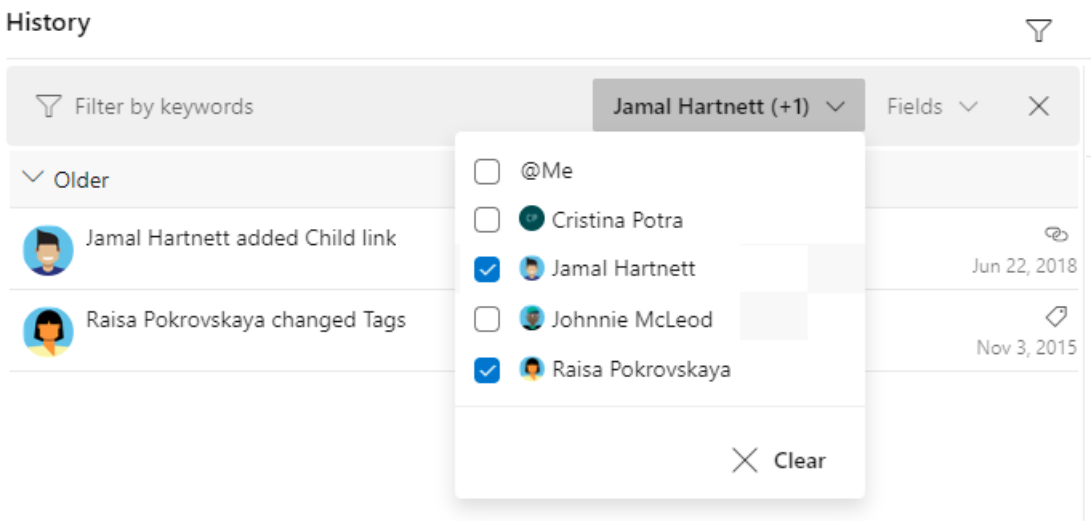
The **History** tab is designed to track all changes made to a work item to support full traceability. The long revision history that results can make it difficult to understand when changes happen to specific fields. To quickly find revisions made to a specific field or by specific people, filter the history view.

> **NOTE**
> The **Toggle filter** feature requires you to enable the **New Boards Hub** preview feature. To enable this feature, see Manage or enable features.

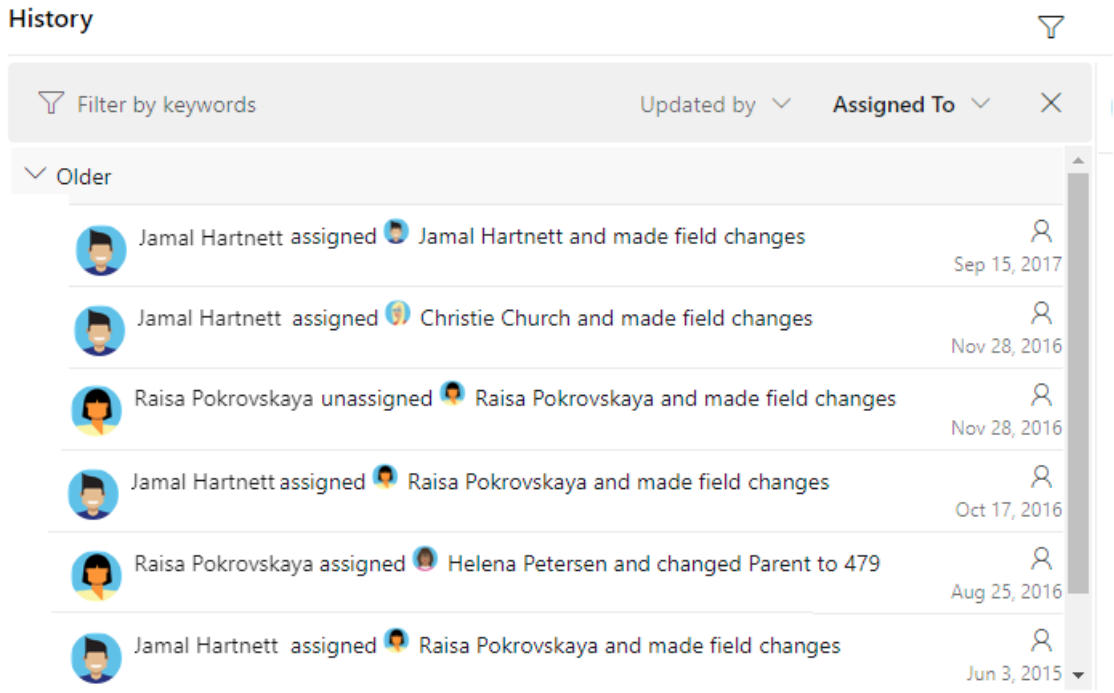
You enable the filter feature by choosing **Toggle filter**.



To review updates by specific people, select their names from the **Updated by** menu.



To review updates made to one or more fields, select the fields from the **Fields** menu.



Fields that support history, auditing, and revision tracking

You can use the following fields to filter queries and create reports. Several of these fields are populated with information as a work item progresses from one state to another. Other fields update when the work item is modified. Some fields don't appear on the work item form, but they're tracked for the WITs listed.

Field name

Description

Work item type

Changed By

The name of the team member who modified the work item most recently.

Reference name=System.ChangedBy, Data type=String

All

Change Date

The date and time when a work item was modified.

Reference name=System.ChangedDate, Data type=DateTime

All

Closed Date [1]

The date and time when a work item was closed.
Reference name=Microsoft.VSTS.Common.ClosedDate, Data type=DateTime

All

Created Date

The date and time when a work item was created.

Reference name=System.CreatedDate, Data type=DateTime

All

History

The record of changes that were made to the work item after it was created. Every time that the work item is updated, information is appended to the history, which specifies the date of the change, who made the changes, and which fields were changed.

> **NOTE**
> History field queries return work items whose **Discussion** comments or **Description** fields contain words that match the keywords entered. You can't use the History field to query on changes made to other fields.

You can't add formatted text to the history field. Once you've saved the work item, you can't alter the history. The `History` field, along with the `Description`, `Steps to Repro` and `Title` fields are automatically indexed for full-text search as described in Query fields, operators, and macros.

Reference name=System.History, Data type=History

All

Resolved Date [1]

The date and time when the work item was moved into a Resolved state.

Reference name=Microsoft.VSTS.Common.ResolvedDate, Data type=DateTime

Bug (Agile, CMMI)

Rev

A number that is assigned to the historical revision of a work item.

> **NOTE**
> A work item revision limit of 10,000 is in effect for updates made through the REST API for Azure DevOps Services. This limit restricts updates from the REST API, however, updates from the web portal are not affected.

Reference name=System.Rev, Data type=Integer

All

Revised Date

The date and time when a work item was revised or modified.

Reference name=System.RevisedDate, Data type=DateTime

Shared Parameter, Shared Step, Test Case

State Change Date

The date and time when the value of the State field changed.

Reference name=Microsoft.VSTS.Common.StateChangeDate, Data type=DateTime

All

Test Suite Audit

Tracks other operations performed when modifying a test suite, for example: adding tests to a test suite or changing configurations. This field can be viewed through the History tab or through a separate query. There will be a consolidated history view, including changes performed to work items field and changes resulting from related artifacts such as test points and configurations.
Reference name=Microsoft.VSTS.TCM.TestSuiteAudit, Data type=PlainText

Test Suite

Watermark

A system-managed field (not editable) that increments with changes made to a work item.
Reference name=System.Watermark, Data type=Integer

All

NOTE

1. For these fields to be defined for a WIT, they must be included in the `WORKFLOW` section of the WIT definition. For example, this syntax is included within the `FIELDS` definition when transitioning to a Resolved state:

```
<FIELD refname="Microsoft.VSTS.Common.ResolvedDate">
   <SERVERDEFAULT from="clock" />
</FIELD>
```

Related articles

- Query editor
- Query fields, operators, and macros
- Query by date or current iteration

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Query by field value comparisons in Azure Boards and Azure DevOps

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can create queries based on how one field's value compares to another using the comparison field operators. This query is useful to filter work items based on:

- Is the person who created the work item the same as or different than the person assigned to it? Or, who closed it
- Which Tasks were closed before or after their Target Date.

Supported data types

You can use the comparison field operators— `=[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field]` —with the following field data types. The data type you select for the **Field** and the **Value** must match.

- Boolean (supports `=[Field], <>[Field]`
- Date/Time
- Double, Integer
- GUID
- Identity
- String (excluding Tags)

> **NOTE**
> Some combinations of data type and comparison field operator might not make sense to use, such as `Title >=[Field]` or `Assigned To <=[Field]`.

Sample filters

Filter for

Include these query clauses

Work items closed by someone other than the person who created the work item

`Created By <>[Field] Closed By`
State = Closed

Tasks whose Original Estimate is less than Completed Work

`Original Estimate <=[Field] Completed Work`

Closed tasks completed before their target date

`Target Date <=[Field] Closed Date`
State = Closed

Fields that support field comparison

The following table provides an index to those fields that support field comparison queries.

> **NOTE**
> Not all fields listed are supported for all projects or work item types. However, you can customize a process or work item type by adding custom fields which you can use for the purposes of queries and field comparisons. To learn more, see Add a custom field to a work item type (Inheritance process) or Add or modify a field (Online XML process).

A

- Acceptance Criteria (Scrum)
- Accepted By
- Accepted Date
- Activated By
- Activated Date
- Activity
- Actual Attendee 1-8 (CMMI)
- Analysis (CMMI)
- Application Launch Instructions
- Application Start Information
- Application Type
- Iteration Id (System)
- Assigned To
- Associated Context
- Associated Context Code
- Associated Context Owner
- Associated Context Type
- Attached File Count
- Automated Test Id (TCM)
- Automated Test Name (TCM)
- Automated Test Storage (TCM)
- Automated Test Type (TCM)
- AutomatedTestId (TCM)
- AutomatedTestName (TCM)
- Automation Status (TCM)

B

- Backlog Priority (Scrum)
- Blocked
- Board Column
- Board Column Done
- Board Lane
- Business Value

C

- Called By (CMMI)
- Called Date (CMMI)
- Changed By (System)

- Changed Date (System)
- Closed By (System)
- Closed Date (System)
- Closed Status
- Closed Status Code
- Closing Comment
- Comment Count
- Comments (CMMI)
- Committed (CMMI)
- Completed Work
- Contingency Plan (CMMI)
- Corrective Action Actual Resolution (CMMI)
- Corrective Action Plan (CMMI)
- Created By (System)
- Created Date (System)

D-E-F

- Discipline (CMMI)
- Due Date
- Effort
- Escalate (CMMI)
- External Link Count
- Finish Date
- Found In Build (TCM)
- Found In Environment (CMMI)

H

- How Found (CMMI)
- Hyperlink Count

I

- ID (System)
- Impact Assessment (CMMI)
- Impact on Architecture (CMMI)
- Impact on Development (CMMI)
- Impact on Technical Publications (CMMI)
- Impact on Test (CMMI)
- Impact on User Experience (CMMI)
- Integrated in Build (TCM)
- Issue (TCM)
- Iteration Id (System)

J-L-M-N

- Justification (CMMI)
- Link Comment (System)
- Link Description (System)
- Local Data Source (TCM)
- Meeting Type (CMMI)
- Minutes (CMMI)

- Mitigation Plan (CMMI)
- Mitigation Triggers (CMMI)
- Node Name (System)

O-P-Q

- Optional Attendee 1-8 (CMMI)
- Original Estimate
- Parameters (TCM)
- Priority
- Probability (CMMI)
- Proposed Fix (CMMI)
- Purpose (CMMI)
- Query Text (TCM)

R

- Rating
- Reason (System)
- Related Link Count (System)
- Remaining Work
- Remote Link Count (System)
- Repro Steps
- Required Attendee 1-8 (CMMI)
- Requirement Type (CMMI)
- Requires Review (CMMI)
- Requires Test (CMMI)
- Resolution (Scrum)
- Resolved By
- Resolved Date
- Resolved Reason
- Reviewed By
- Reviewed Date
- Rev (System)
- Risk (Agile)
- Root Cause (CMMI)

S

- Severity
- Size (CMMI)
- Stack Rank
- Start Date
- State (System)
- State Change Date
- State Code
- Steps (TCM)
- Steps to Reproduce (TCM)
- Story Points (Agile)
- Subject Matter Expert (CMMI)
- Symptom (CMMI)

- System Info (TCM)

T

- Target Date
- Target Resolve Date (CMMI)
- Task Type (CMMI)
- Team Project (System)
- Test Suite Audit (TCM)
- Test Suite Type (TCM)
- Test Suite Type ID (TCM)
- Time Criticality
- Title (System)
- Triage (CMMI)

U-V-W

- User Acceptance Test (CMMI)
- Value Area
- Watermark (System)
- Work Item Type (System)

Related articles

- Query index quick reference
- Query by title, ID, or description
- Query by assignment or workflow changes
- Query by date or current iteration
- Query a numeric field
- Query by picklist value

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Query by numeric fields in Azure Boards and Azure DevOps

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

How do I determine how much work each developer has completed on my team? Is there a way to sum up the effort or story points for an iteration?

The most common numeric fields track effort for items in the Requirements category or estimated, remaining, and completed work for items in the Task category. With queries you can list the work items of interest, and then define a chart that shows either a count of work items or a sum of a numeric field.

Supported operators and macros

Query clauses that specify a numeric field can use the operators listed below.

- = , <> , > , < , >= , <=
- =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field]
- In, Not In
- Was Ever

Tips for developing chart-based queries

- You can only add charts for flat-list queries
- Chart options reference either query filters or fields displayed through column options
- Save changes you make to your query prior to adding or modifying a chart.
- To group one or more clauses, select them and then choose the group clauses icon. To ungroup, select on the grouped clause.

For more details on creating queries and chart-based queries, see Use the query editor to list and manage queries and Charts. If you want to add a custom field to track and generate sums of other numeric values, see Add or modify a field.

Useful filters

Filter for

Include these query clauses

User stories or bugs

```
Work Item Type In User Story,Bug
```

Tasks or bugs

```
Work Item Type In Task,Bug
```

Items that are Active or Closed

```
State In Active,Closed
```

Items in the Requirements category

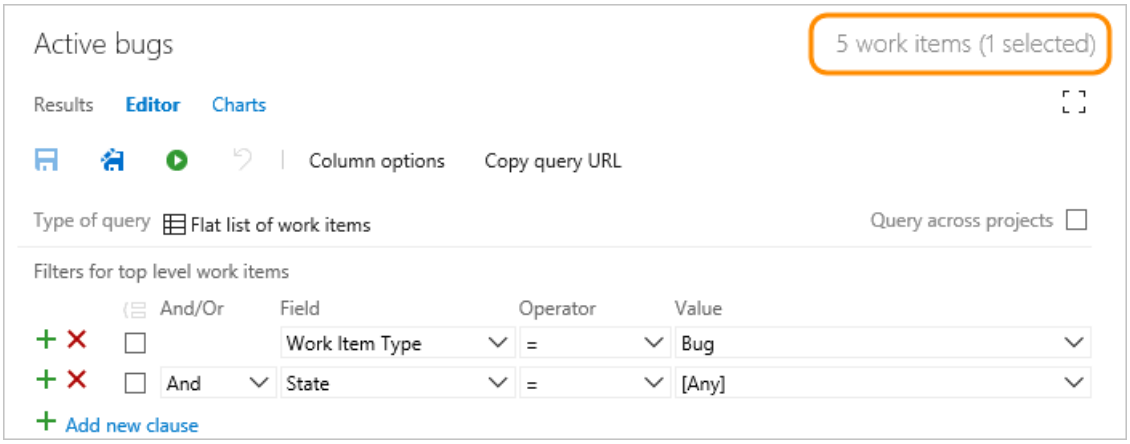
```
Work Item Type In Group Microsoft.RequirementCategory
```

Unestimated user stories

```
Story Points <> (leave Value field blank)
```

Work item count queries and charts

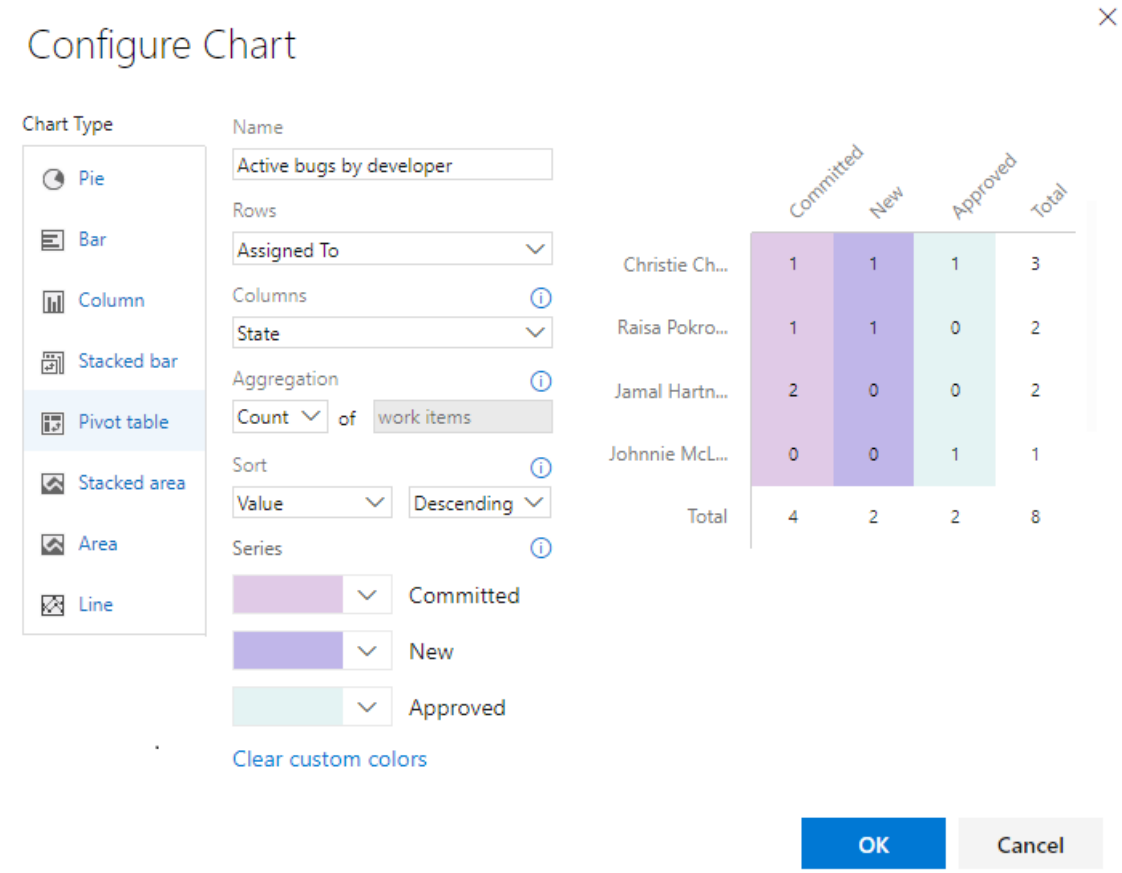
All queries show a count of items when you run the query. Here we define a flat-list query that filters for bugs in any state.

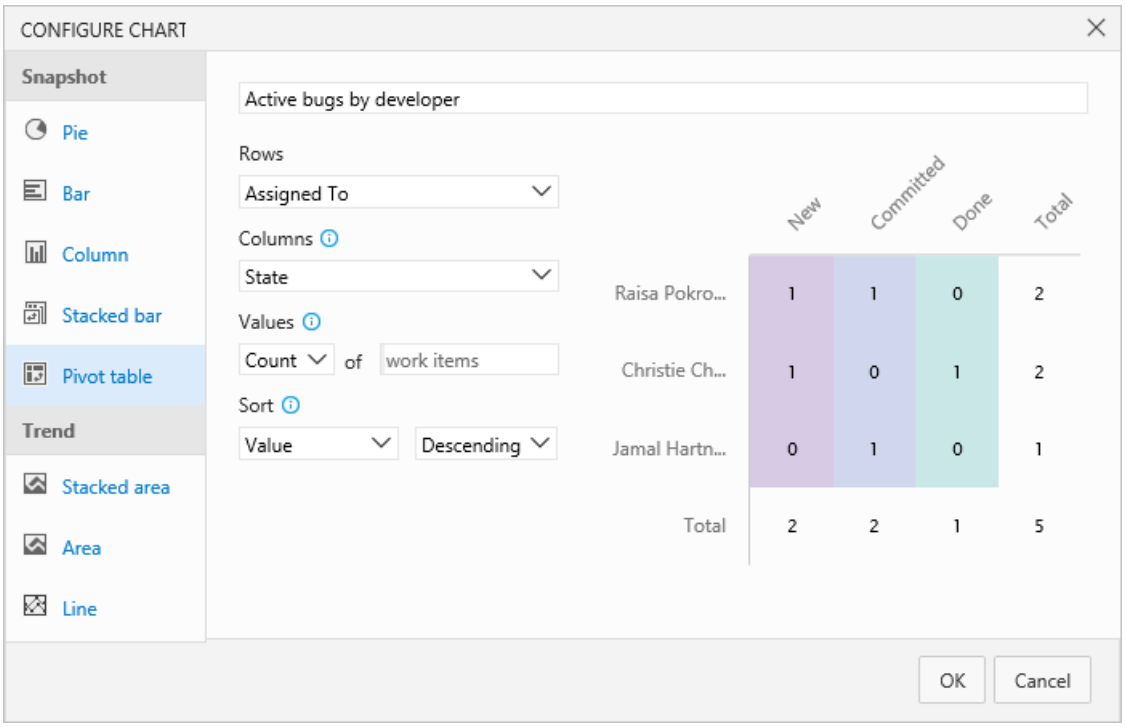


Also, all charts contain a Values selection designed to display a count of work items within the chart.

Count of bugs per developer

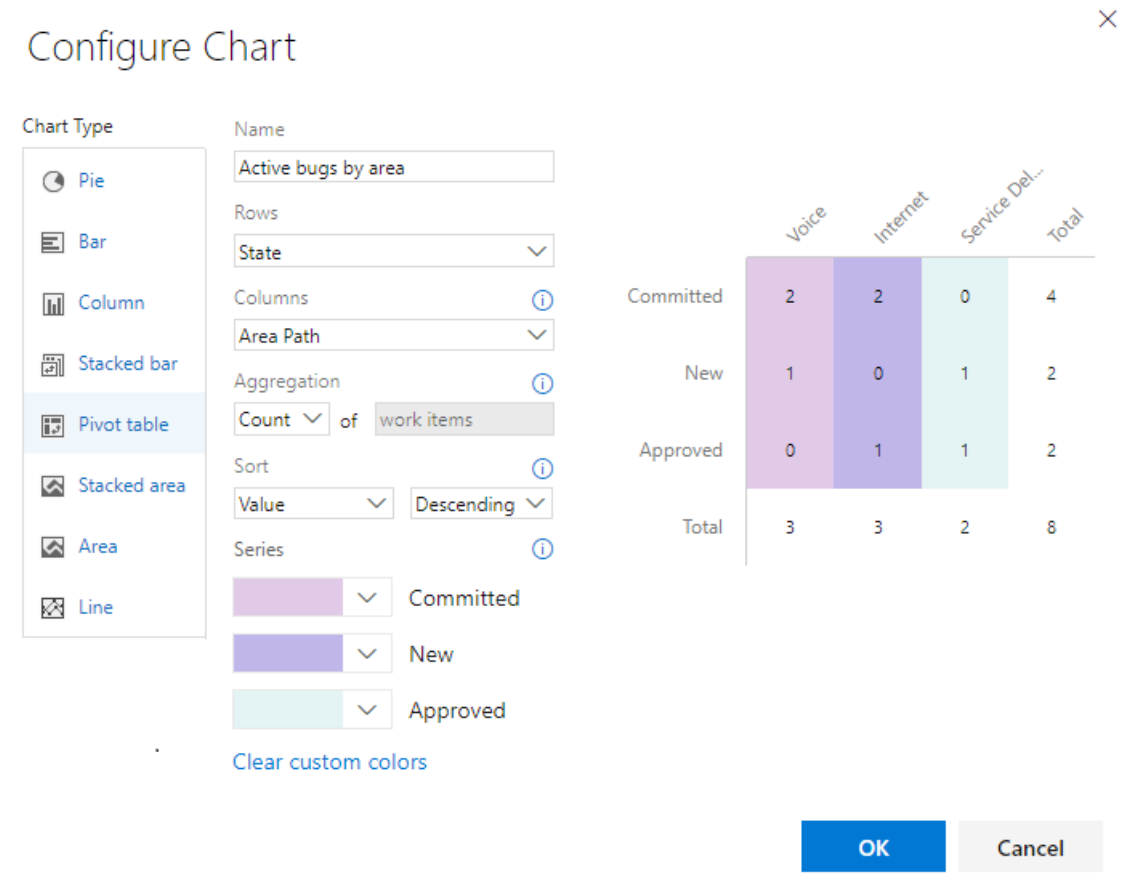
Create an active bugs query and modify the column options to show Assigned To and State. Then, add a pivot chart that displays the assignments and state.

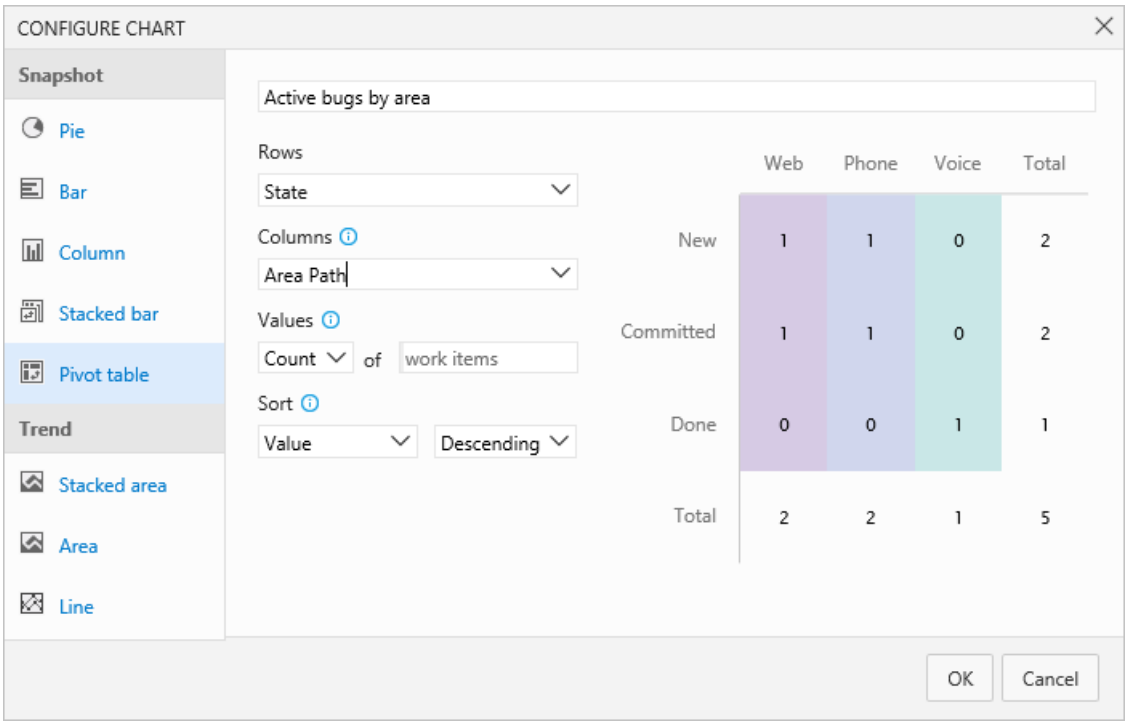




Count of bugs by state and area

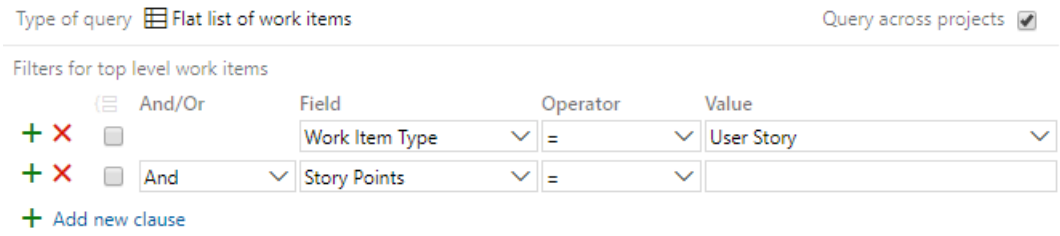
Using the same flat-list query that filters for bugs shown in the previous section, you can show a count based on area. Modify the column options to show the Area Path. Then, add a pivot chart that displays the state and area path.





Undefined field value queries

You can find work items that have an undefined field value by using the equals operator (=) and leaving the Value for the field blank. For example, the following filters will list all work items of type User Stories whose Story Points field is blank.



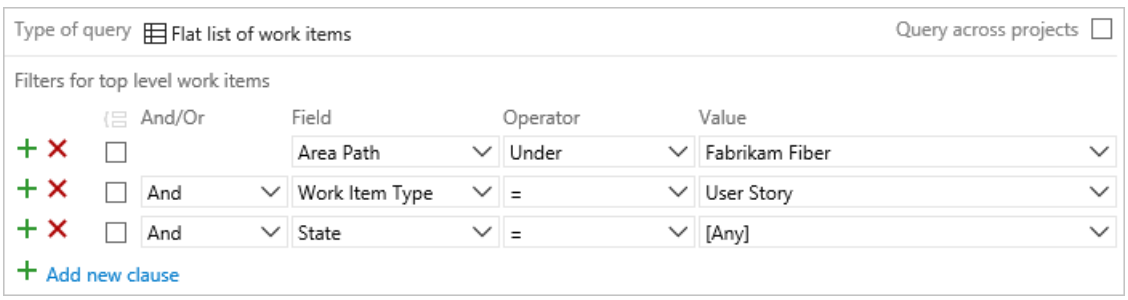
To list work items based on a field that isn't blank, use the not operator (<>) and leave the Value blank.

Effort or story point queries and charts

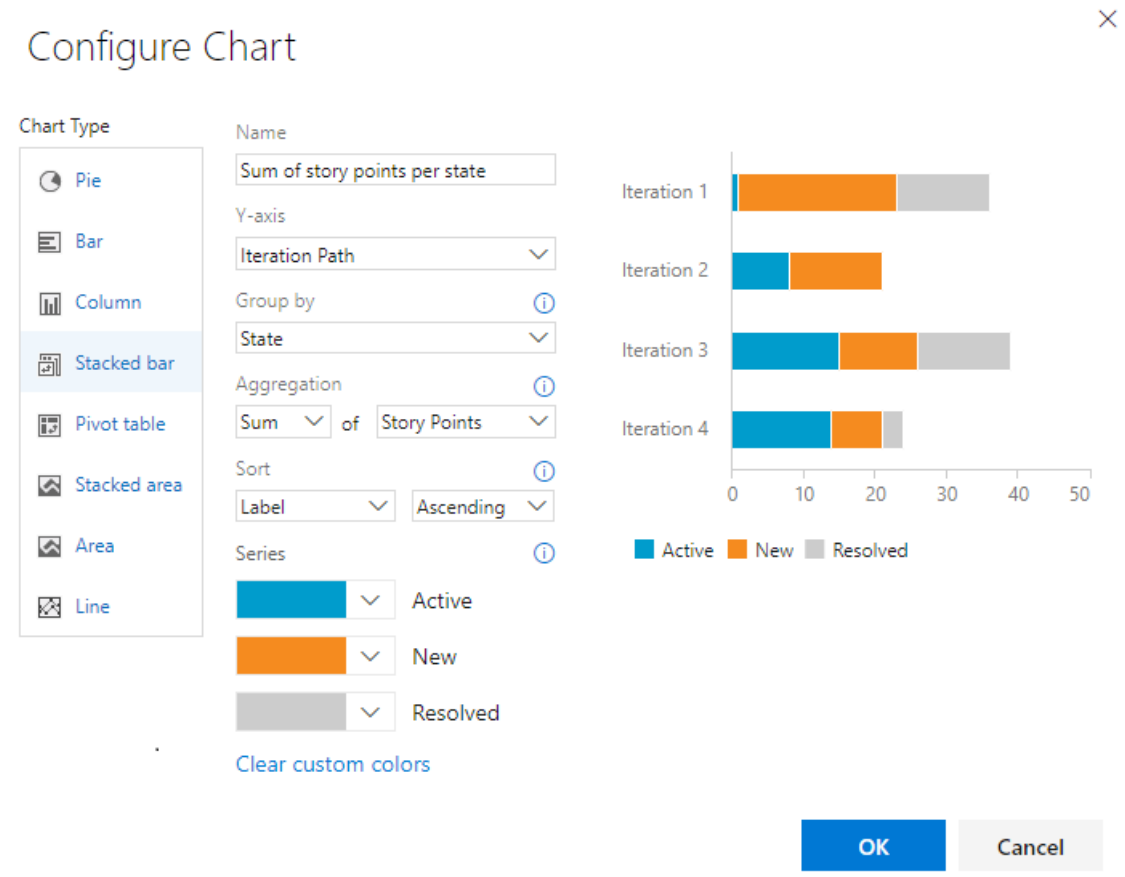
You can assign Story Points to user stories or bugs when you work in an Agile process. Or, Effort to product backlog items and bugs when you work in a Scrum process. For more information, see Basic, Agile, Scrum, or CMMI work item types and workflow articles.

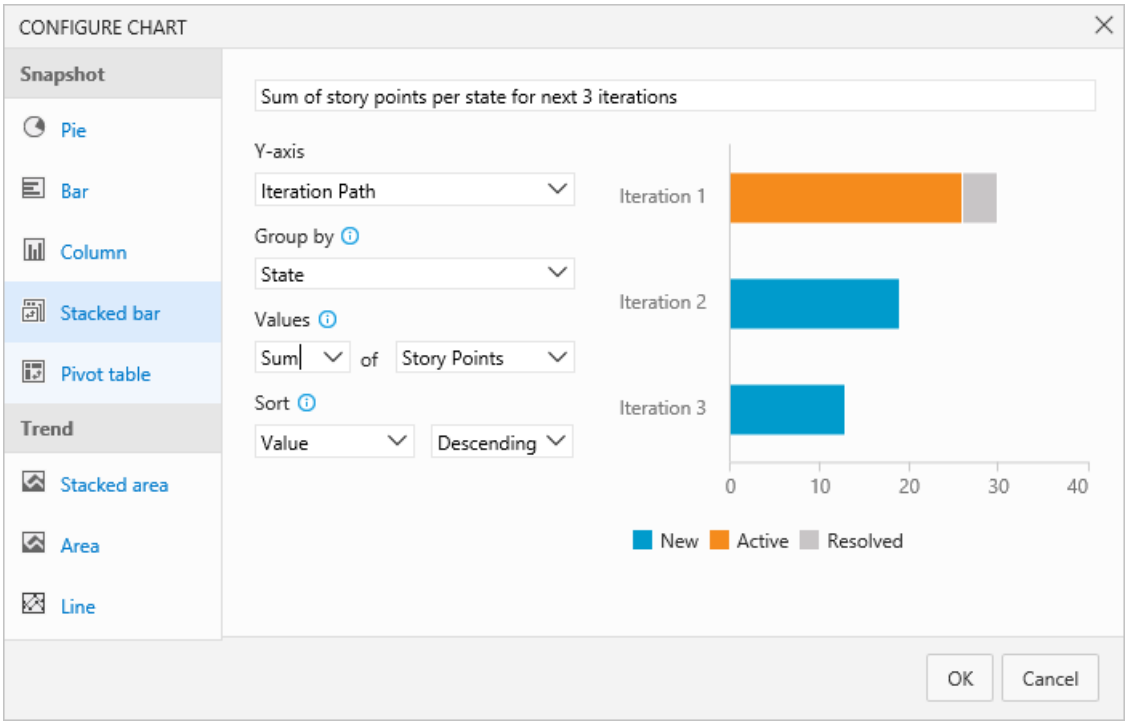
Sum of story points and their status

Create a query that filters for User Story as the work item type and modify the column options to show Story Points and State.



Then, add a stacked bar chart that sums the Story Points.

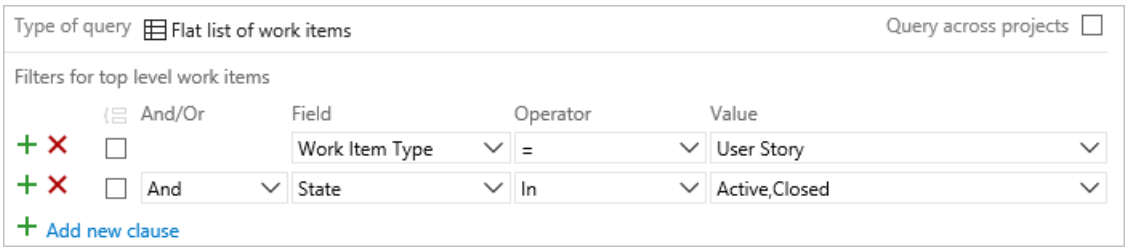




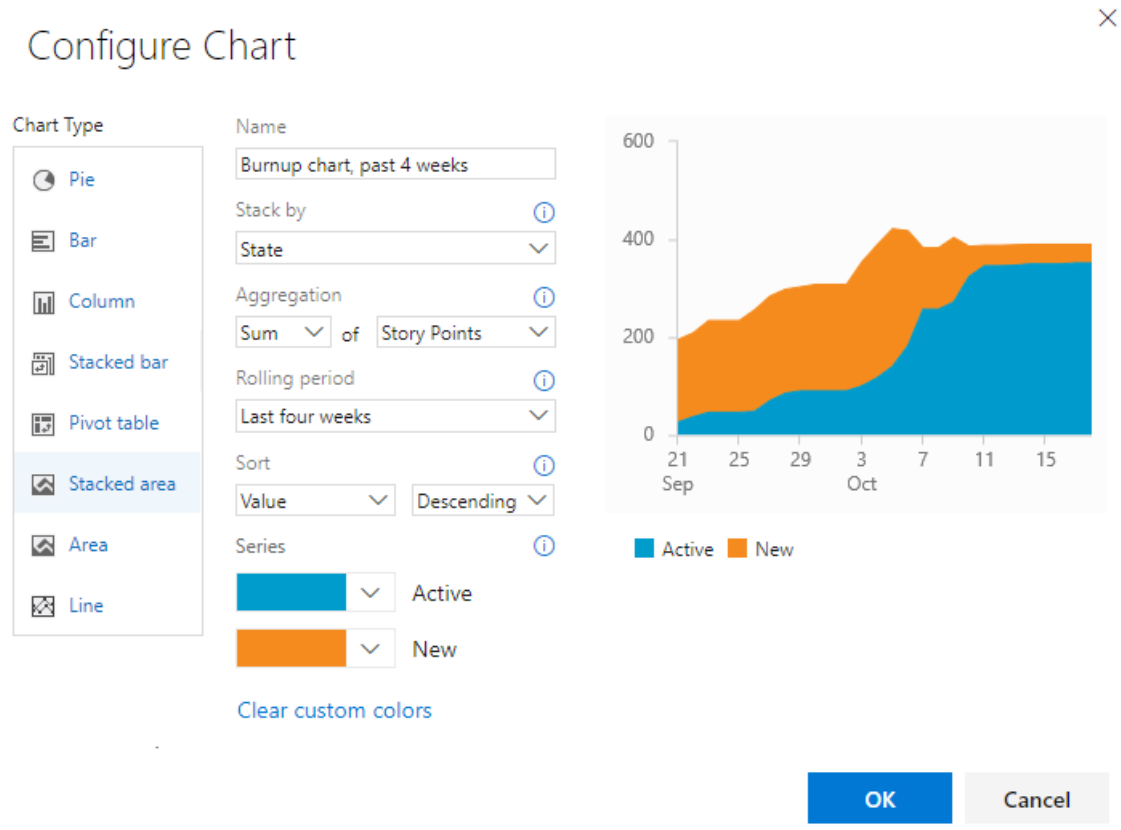
For information on system-defined cumulative flow diagrams, see Cumulative flow.

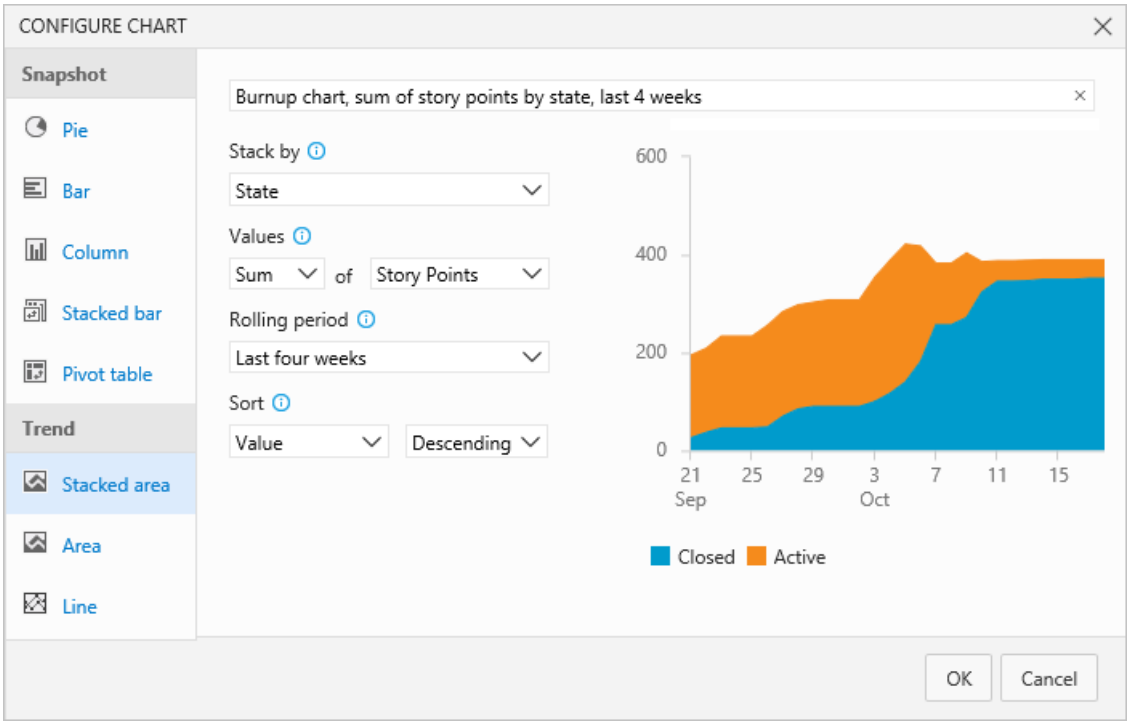
Burn up chart of user stories for an iteration

Create a query that filters for User Story as the work item type and in the Active or Closed state. Modify the column options to show Story Points.



Then, add a stacked area trend chart that sums the Story Points.





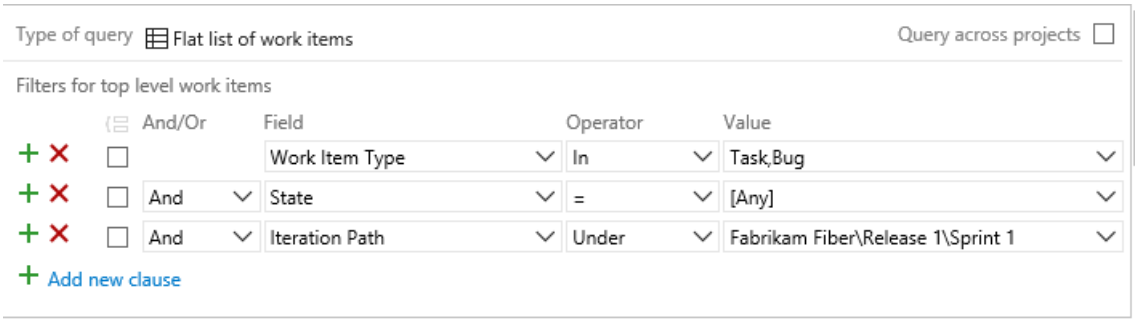
Remaining and completed work queries and charts

Based on the process your project references, you can assign the following fields to tasks or bugs.

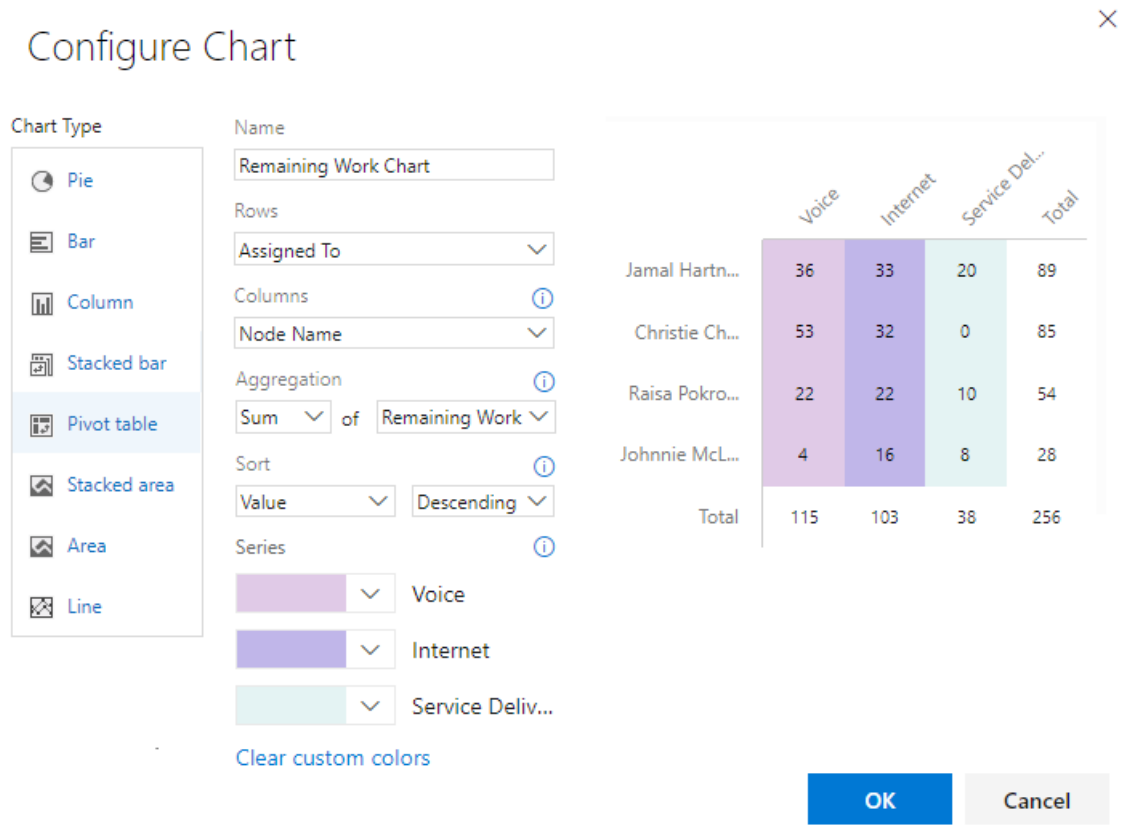
PROCESS	AVAILABLE FIELDS
Agile	Original Estimate, Remaining Work, Completed Work
Scrum	Remaining Work
CMMI	Original Estimate, Remaining Work, Completed Work

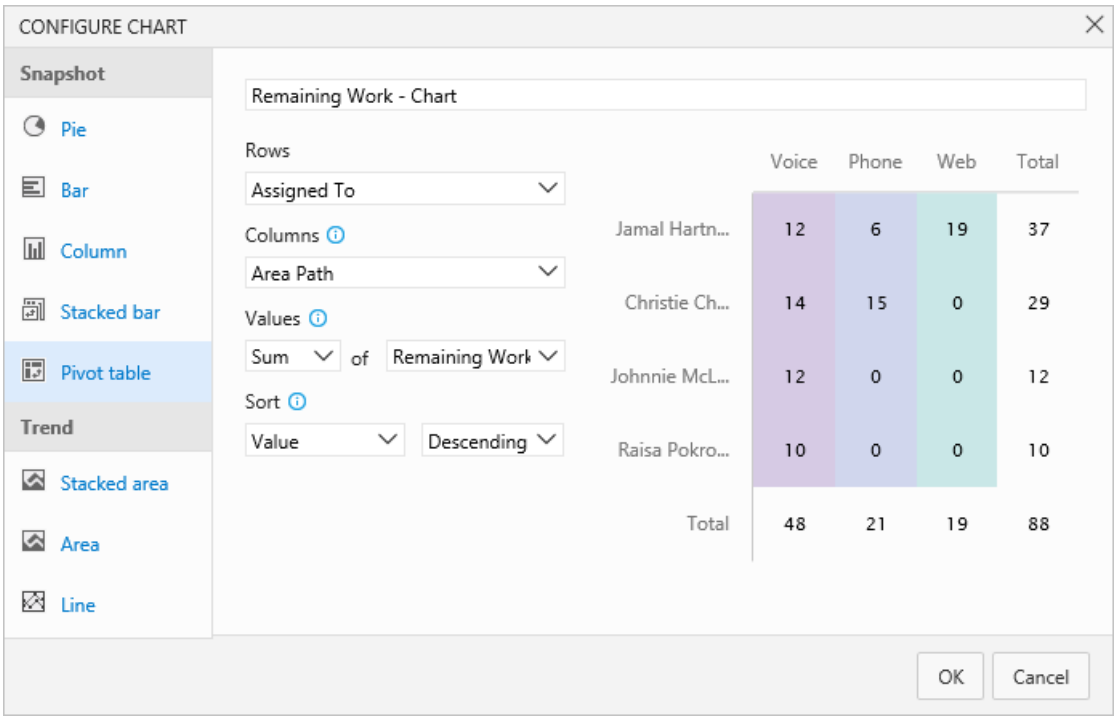
Sum of remaining work per developer

If you follow Scrum practices and estimate Remaining Work for your tasks and bugs, you can get a rollup of the amount of work remaining for each developer with the following query and chart. By using the In operator and including both Task and Bug, you include any bugs that are being tracked as tasks.



Add Remaining Work as a column option to the query and save. To view a sum of the remaining work, add a pivot chart as shown.





For information on system-defined sprint burndown charts, see Sprint burndown.

Fields used to estimate and track work

The following table describes the activity-based and numeric fields that you can use to track work. For information on date-related fields, such as Start Date, Finish Date, and Target Date, see Query by date or current iteration.

Field name

Description

Work item type

Activity [1, 2]

The type of activity that is required to complete a task.To learn more about how this field is used, see Capacity planning. Allowed values are:

- Deployment
- Design
- Development
- Documentation
- Requirements
- Testing

The Activity field is assigned to `Activity` in the ProcessConfiguration file.[3]

Reference name=Microsoft.VSTS.Common.Activity, Data type=String

Task, Bug[4] (Agile and Scrum)

Business Value

A subjective unit of measure that captures the relative business value of a product backlog item or feature compared to other items of the same type. An item that is assigned a higher number should be considered as having more business value than an item that is assigned a lower number.

Reference name=Microsoft.VSTS.Common.BusinessValue, Data type=Integer

Epic, Feature

Completed Work

The amount of work that has been spent implementing a task. You can specify work in hours or in days. There are no inherent time units associated with this field.

Reference name=Microsoft.VSTS.Scheduling.CompletedWork, Data type=Double

Task, Bug[4]

Discipline [1, 2]

The type of activity or discipline that is assigned to a task. To learn more about how this field is used, see Capacity planning. Allowed values are:

- Analysis
- Development
- Test
- User Education
- User Experience

The Discipline field is assigned to `Activity` in the ProcessConfiguration file.[3]

Reference name=Microsoft.VSTS.Common.Discipline, Data type=String

Task, Bug [4] (CMMI)

Effort

A subjective unit of measure that captures the size of a bug or product backlog item. If you assign more effort to an item, you indicate that more work is required to implement it.

This field [3] is also used to calculate team velocity and forecasting. It's assigned to `Effort` in the ProcessConfiguration file.

Reference name=Microsoft.VSTS.Scheduling.Effort, Data type=Double

Product Backlog Item, Bug [4] (Scrum)

Feature, Epic

Story Points

A subjective unit of measure that captures the size of a user story. If you assign more points to a user story, you indicate that more work is required to implement it.

This field [3] is also used to calculate team velocity and forecasting. It's assigned to `Effort` in the ProcessConfiguration file.

Reference name=Microsoft.VSTS. Scheduling.StoryPoints, Data type=Double

User Story, Bug [4] (Agile)

Size

A subjective unit of measure that captures the size of a requirement. The larger the size, the more work is required to implement it.

This field[3] is also used to calculate team velocity and forecasting. It's assigned to `Effort` in the

ProcessConfiguration file.

Reference name=Microsoft.VSTS. Scheduling.Size, Data type=Double

Requirement, Bug [4] (CMMI)

Original Estimate

The amount of work required to complete a task. You can specify work in hours or in days. There are no inherent time units associated with this field.

Reference name=Microsoft.VSTS.Scheduling.OriginalEstimate, Data type=Double

Task, Bug [4] (Agile and CMMI)

Remaining Work

The amount of work that remains to finish a task. You can specify work in hours or in days. There are no inherent time units associated with this field. This field [3] is also used to calculate burn down. It is assigned to `type="RemainingWork"` in the ProcessConfiguration file.

> **NOTE**
> For Azure Boards, the taskboard always shows "h" for hours in relationship to Remaining Work. For TFS, you can modify the ProcessConfiguration file for the Remaining Work type field to specify "d" for days, or other preferred label.

Reference name=Microsoft.VSTS.Scheduling.RemainingWork, Data type=Double

Task, Bug[4]

Requires Review

Indicates the task requires review. You can specify **Yes** or **No** (default).

Reference name=Microsoft.VSTS.CMMI.RequiresReview, Data type=String

Task (CMMI)

Requires Test

Indicates the task requires a test. You can specify **Yes** or **No** (default).

Reference name=Microsoft.VSTS.CMMI.RequiresTest, Data type=String

Task (CMMI)

Task Type[1]

Specifies the kind of task to implement. Allowed values are:

- Corrective Action
- Mitigation Action
- Planned

Reference name=Microsoft.VSTS.CMMI.TaskType, Data type=String

Task, Bug[4] (CMMI process)

NOTE

1. To change the menu selection: for cloud services or an Inherited process, see Add and manage fields; and for On-premises XML process, see Add or modify a field, customize a picklist.
2. The values displayed in the Capacity page for **Activity** (Agile or Scrum) or **Discipline** (CMMI) reflect a union of all values defined for the field in all projects within the project collection instance. Therefore, to restrict the values that appear for Capacity on the sprint backlog pages, you must make the values match in all the projects for the field assigned to `type="Activity"`.
3. To change the ProcessConfiguration field assignment (on-premises only), see Process configuration XML element reference.
4. Each team can configure their Agile tools to determine if bugs are treated similar to requirements or tasks. Since bugs can appear either with requirements or tasks, fields used to estimate effort at the requirement-level and the task-level are included in the work item form.

Related articles

For information on adding custom fields, see Customize your work tracking experience.

The main tools you use to plan and track work are described here:

- Create your backlog
- Sprint planning
- Capacity planning
- Taskboard
- Kanban board

For more information on using work items and queries, see:

- Query editor
- Query fields, operators, and macros
- Add work items
- Work item field index
- About managed queries

Rollup numeric values across work item types

Rollup provides summed values of select fields for all child work items of a parent. Natively, Azure Boards and TFS provide rollup of Remaining Work for tasks on the taskboard. For other rollup requirements, see the following articles:

- Support rollup of work and other fields
- Create rollup charts with Power BI

What items appear in the Requirement or Task categories?

The default assignments of work item types to each category are listed below for each process.

PROCESS	REQUIREMENT CATEGORY	TASK CATEGORY
Agile	User Story	Task
Scrum	Product Backlog Item	Task
CMMI	Requirement	Task

However, each team can determine if the Bug work item type appears in either the Requirement or Task category. See Show bugs on backlogs and boards.

You can add custom work item types to a backlog. For details, see Add or modify a work item type, Add a custom WIT to a backlog or board.

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Query by rank and picklist value in Azure DevOps and Azure Boards

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You use planning, ranking, and priority fields to specify which work the team should complete first. By ranking and prioritizing work items, all team members gain an understanding of the relative importance of the work that they must accomplish.

You rank and prioritize work items when you Create your backlog.

Supported operators and macros

Query clauses that specify a string or integer field can use the operators listed below.

- = , <> , > , < , >= , <=
- =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field]
- In, Not In
- Was Ever

Picklist query examples

Most of the planning fields described in the next section are either an integer or string field. For example queries of numeric or rich-text fields, see Query by numeric fields and Query by titles, IDs, and rich-text fields.

To use the **In** and **Not In** operators, enter the names or labels of items that correspond to the selected field separated by a comma. For example, to filter for **Work Item Types** that are either user stories or bugs, add the clause: `Work Item Types In User Story,Bug`.

Filter for

Include these query clauses

List blocked tasks (Scrum)
(Blocked field is type String)

```
Work Item Type In Task
And Blocked = Yes
```

Priority 1 bugs (**Priority** field is type Integer)

```
Work Item Type In Bug
And Priority = 1
```

Features and stories that address Architectural areas
(**Value Area** field is type String)

```
Work Item Type In Feature,User Story
And Value Area = _ Architectural
```

Fields used to plan and prioritize work

The following table describes the fields that you can use to plan and prioritize work. Some fields are only valid for a specific process—Basic, Agile, Scrum, or Capability Maturity Model Integration (CMMI).

Field name

Description

Work item type

Backlog Priority [1]

A number assigned by a background process used to track the sequence of items on a backlog or board. To learn more about how this field is used, see Use backlogs for effective project management, Backlog priority or stack rank order.

Reference name=Microsoft.VSTS.Common.BacklogPriority, Data type=Double

Bug, Epic, Feature, Product backlog item, Task (Scrum)

Blocked

Indicates that no further work can be performed on the work item. If an issue has been opened to track a blocking problem, a link should be made to the issue.

- For the Scrum process, task work items: You can specify **Yes** or clear the field.
- For the CMMI process work items: You can specify **Yes** or **No**.

Reference name=Microsoft.VSTS.CMMI.Blocked, Data type=String

Bug, Change Request, Requirement, Risk, Task (CMMI, Scrum)

Committed

Indicates if the requirement is committed in the project. You can specify **Yes** or **No**.

Reference name=Microsoft.VSTS.CMMI.Committed, Data type=String

Requirement (CMMI)

Escalate

Indicates if the issue is affecting the critical path of the project plan. You can specify **Yes** or **No**.

Reference name=Microsoft.VSTS.CMMI.Escalate, Data type=String

Issue (CMMI)

Priority [1]

A subjective rating of the bug, issue, task, or test case as it relates to the business. You can specify the following values:

- **1**: Highest priority, implement feature or fix as soon as possible. Product cannot ship without successful resolution.
- **2**: Medium priority. Product cannot ship without successful resolution, but it does not need to be addressed immediately.
- **3**: Low priority. Implementation or fix is optional based on resources, time, and risk. If product ships without successful resolution, document the issue in release notes as known issues.
- **4**: Lowest priority. Tracks an issue that basically does not affect usage (such as a small typo).

Reference name=Microsoft.VSTS.Common.Priority, Data type=Integer

Bug, Change Request, Epic, Feature, Impediment, Issue, Product backlog item, Requirement, Risk, Shared Step, Task, Test Case, User Story

Risk

A subjective rating of the relative uncertainty around the successful completion of a user story. Defined allowed values are:

- **1 - High**
- **2 - Medium**
- **3 - Low**

Reference name=Microsoft.VSTS.Common.Risk, Data type=String

Epic, Feature, User Story (Agile)

Severity [1]

A subjective rating of the impact of a bug on the project. You can specify the following values:

- **1 - Critical**
- **2 - High**
- **3 - Medium**
- **4 - Low**

Reference name=Microsoft.VSTS.Common.Severity, Data type=String

Bug, Issue (CMMI), Risk (CMMI)

Stack Rank [2]

A number, assigned by a background process, used to track the list order of items on a backlog or board in the web portal. To learn more about how this field is used, see Use backlogs for effective project management, Backlog priority or stack rank order.

Reference name=Microsoft.VSTS.Common.StackRank, Data type=Double

Bug, Epic, Feature, Requirement (CMMI), Risk (CMMI), Task, User Story (Agile)

Time Criticality

A subjective unit of measure that captures how the business value lessens over time. Higher values indicate that the epic or feature is inherently more time critical than those items with lower values.

Reference name=Microsoft.VSTS.Common.TimeCriticality, Data type=Double

Epic, Feature

Triage

Indicates the type of triage decision that is pending for the work item. You use this field when the work item is in the **Proposed** state.

You can specify one of the following values:

- **Pending** (default)
- **More Info**
- **Info Received**
- **Triaged**

Reference name=Microsoft.VSTS.Common.Triage, Data type=String

CMMI only: Bug, Change Request, Epic, Feature, Issue, Requirement, Task

Value Area [1]

The area of customer value addressed by the epic, feature, or backlog item. Values include:

- **Architectural**: technical services to implement business features that deliver solution
- **Business**: services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default)

Reference name=Microsoft.VSTS.Common.ValueArea, Data type=String

Bug, Epic, Feature, Product Backlog Item (Scrum) Requirement (CMMI), User Story (Agile)

Notes:

1. To change the menu selection, see Add and manage fields (Inherited process) or Add or modify a field, customize a picklist (On-premises XML process).
2. The sequence of items on a product backlog page is determined according to where you have added or dragged the items. As you drag items, a background process updates either the Backlog Priority (Scrum) or Stack Rank (Agile, Basic, CMMI) field. These fields determine the order in which backlog items appear on a backlog page. They are assigned to `type="Order"` in the ProcessConfiguration file.

More about Backlog Priority or Stack Rank fields

The Backlog Priority and Stack Rank fields don't appear on the work item forms. (To learn why, see Where is the field on the work item form to order the backlog?.

To add the field to the form:

- For an Inherited process, add the Stack Rank field to a work item type (for the custom process that your project references).
- For an On-premises XML process, add the field to the form, modify the WIT XML definition to add the following control element:

  ```
  <Control FieldName="Microsoft.VSTS.Common.StackRank" Type="FieldControl" Label="Stack Rank"
  LabelPosition="Left" />
  ```

  or, for Scrum:

  ```
  <Control FieldName="Microsoft.VSTS.Common.BacklogPriority" Type="FieldControl" Label="Stack Rank"
  LabelPosition="Left" />
  ```

Related articles

- Query by a numeric field
- Work item field index
- Work item fields and attributes.

Query work items by link or attachment count

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can link work items to track related work and dependencies and attach files to share information with your team. You can then list work items based on one or more of the following fields:

- Attachment File Count
- (Discussion) Comment Count
- External Link count
- Hyperlink Count
- Link Comment
- Related Link Count
- Remote Link Count

- Attachment File Count
- (Discussion) Comment Count
- External Link count
- Hyperlink Count
- Link Comment
- Related Link Count

For descriptions of each of these fields, see the table provided later in this article.

Supported operators and macros

Query clauses that specify an integer field can use the operators listed below.

`= , <> , > , < , >= , <= ,`
`=[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field],`
`In, Not In,`
`Was Ever`

Link or attachment count queries

You can filter for work items by the link type, link count, or attachment count.

Filter for	Include these query clauses
Items with attachments	`Attached File Count >= 1`
Items with two or more hyperlinks	`Hyperlink Count >= 2`
Items containing external links, links to objects other than work items	

`External Link Count >= 1`

Items that contain between three and seven related links

`Related Link Count >= 3` `And` `Related Link Count <= 7`

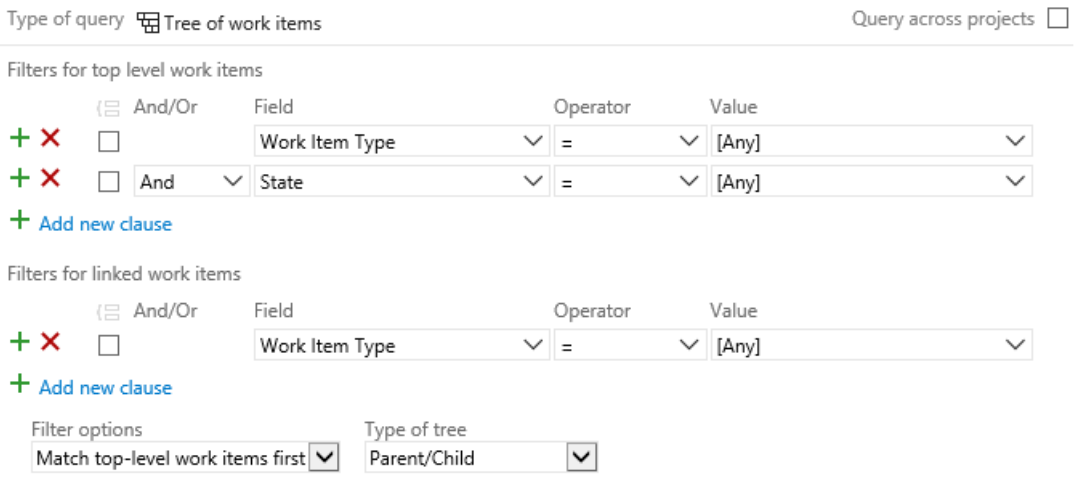
Items that contain remote links

`Remote Link Count > 0`

List hierarchical items in a tree view

Add a query and select **Tree of work items** to begin your query. You should see something similar to the following examples:

- Browser
- Visual Studio 2015



> **NOTE**
> You can't construct a query that shows a hierarchical view of Test Plans, Test Suites, and Test Cases. These items aren't linked together using parent-child link types. However, you can create a Direct links query that lists test-related work items. Also, you can, view the hierarchy through the Test>Test Plans page.

From there, you can add query clauses or change the filter options for linked work items.

Filter for

Include these query clauses

View only child items of work item 645

Add to Filters for top-level work items:

`ID = 645`

Tasks or bugs

Add to Filters for linked work items:

`Work Item Type In Task,Bug`

Items assigned to my team (Web)

Add to both top and bottom filters:

```
Assigned to In Group [Fabrikam Fiber]\Web
```

Parent items of tasks assigned to me

Change Filter options to **Match linked work items first**
Add to Filters for linked work items:

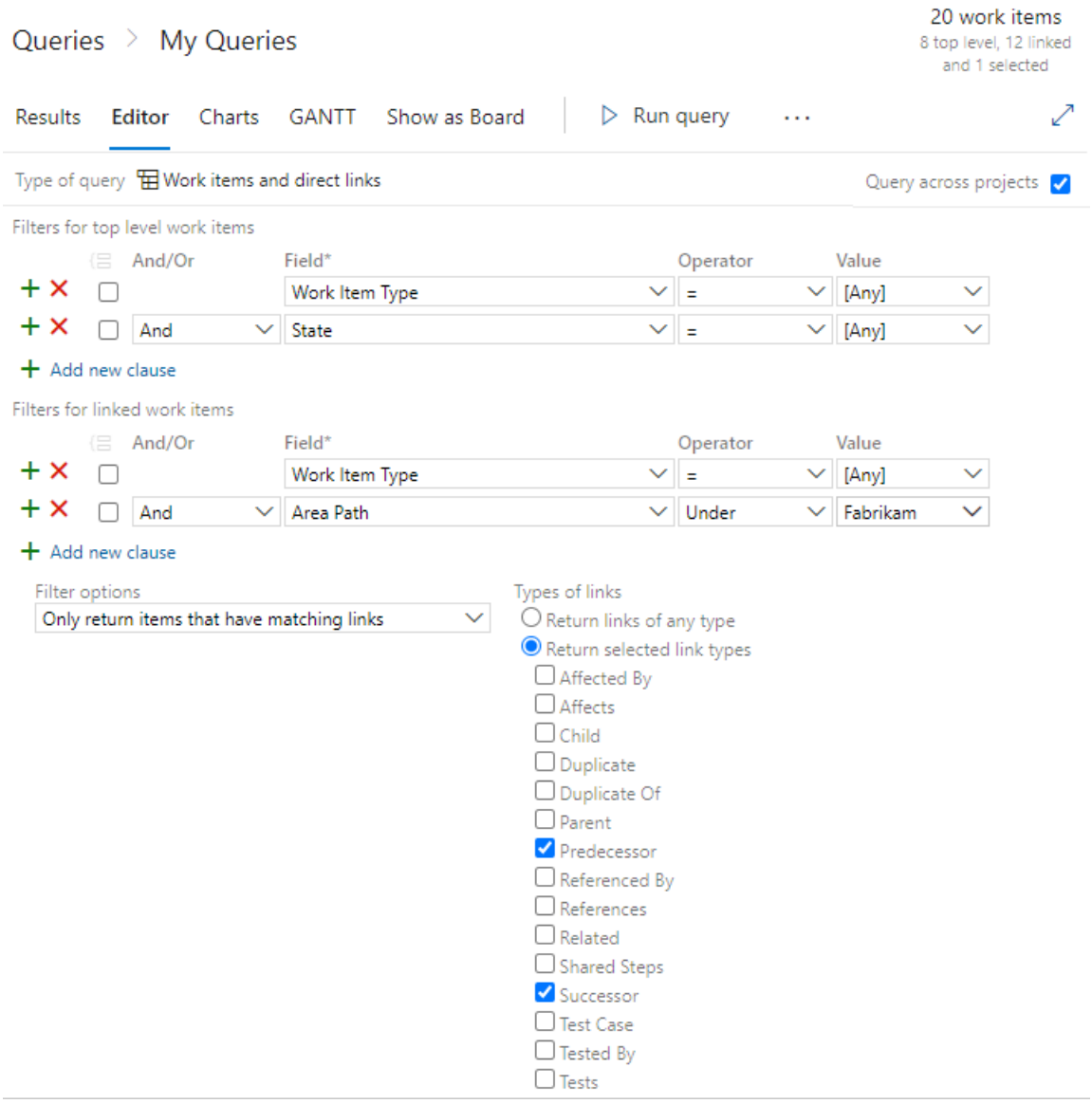
```
Assigned To = @Me
```

List items based on linked dependents

The following example shows a dependent linked query that returns items with dependencies on work managed by other teams and projects.

- Browser
- Visual Studio 2015

The following query finds work items in all projects that are linked to work items under the **Fabrikam** area path and project using Predecessor and Successor link types.



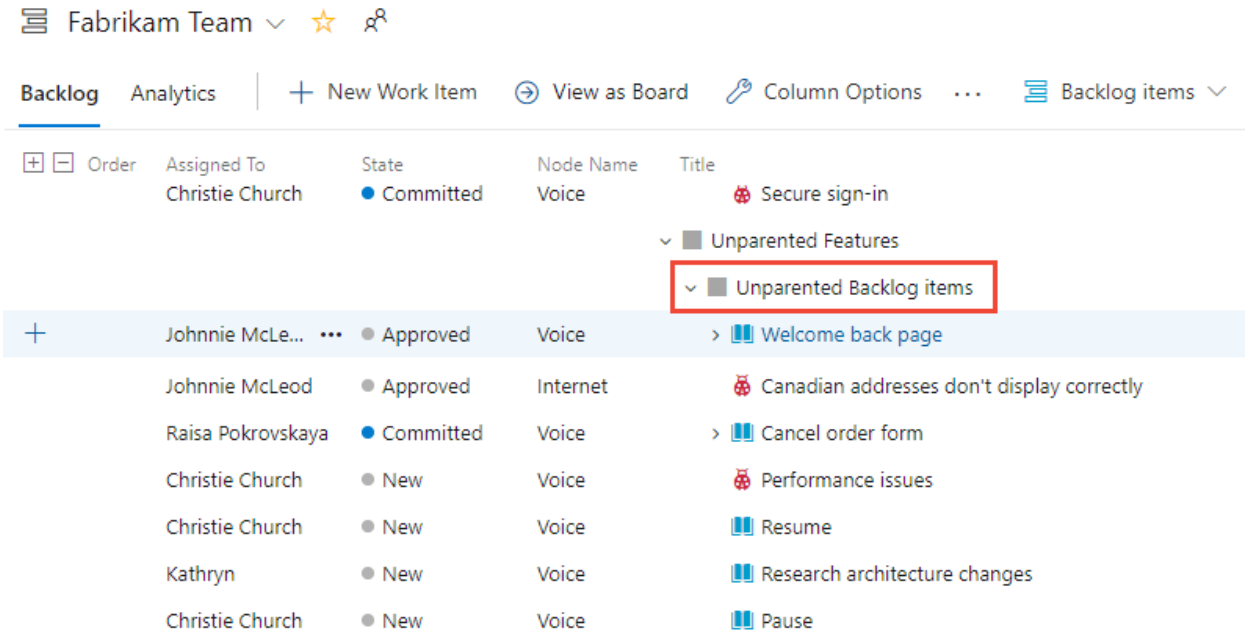
Why this works:

- Checking the **Query across projects** checkbox enables all dependent linked work items that match the filter criteria to be listed, no matter which project they belong to.

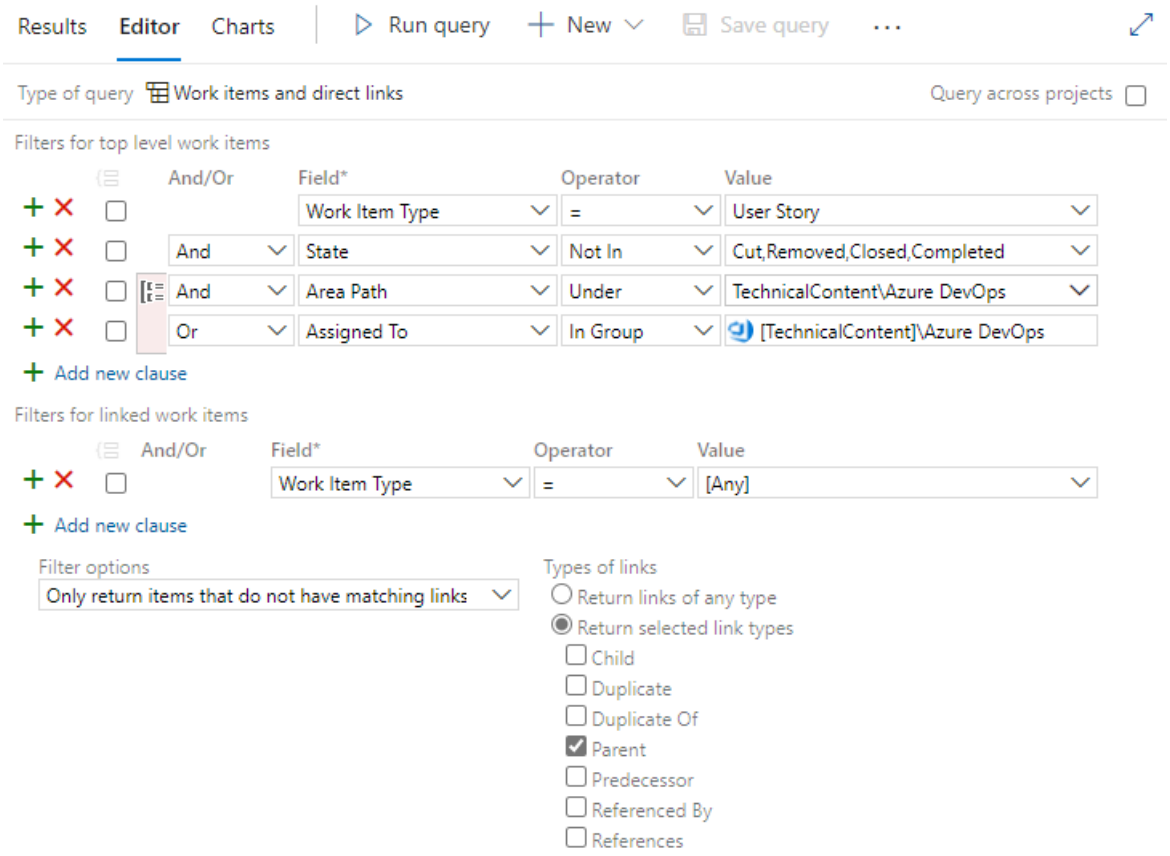
- Specifying the **Area Path Under Fabrikam** clause indicates to find only work items that are linked to work items defined under the Fabrikam project.
- Specifying **Only return items that have matching links**, and **Return selected link types** causes the query to return only work items that are linked based on the **Predecessor** and **Successor** link types.

List orphan user stories

If you typically organize your user stories under features, you can quickly find user stories that are orphan by opening the product backlog, enable **Parents On** view option, and scroll down to the section that lists **Unparented Stories** (Agile) or **Unparented Backlog items** (Scrum).



Or, you can find unparented backlog items using a **Work items and direct links** query. For example, the following query lists active user stories for the Azure DevOps team that don't have a Parent link.



Link, attachment count, and comment fields

The following table describes fields associated with links and attachments. Most of these fields don't appear within the work item form, but are tracked for all work item types.

Field name

Description

Work item type

Attachment File Count

The number of files attached to the work item and stored in the work item tracking database.
Reference Name=System.AttachedFileCount, Data type=Integer

> **NOTE**
> For Azure Boards (cloud service), you can add up to 100 attachments to a work item. Attempts to add more result in an error message upon saving the work item.

All

Comment Count

The number of comments added to the **Discussion** section of the work item.
Reference Name=System.CommentCount, Data type=Integer

> **NOTE**
> Available for TFS 2017 with the new work item form which supports the **Discussion** section and later versions.

All

External Link Count

The number of links from the work item to artifacts that are not work items. such as pull requests, commits, changesets, or other link types.
Reference Name=System.ExternalLinkCount, Data type=Integer

All

Hyperlink Count

The number of hyperlinks that are defined for the work item.

Reference Name=System.HyperLinkCount, Data type=Integer

All

Link Comment

Contains comments from the team member who created the link. You can configure this field to appear as a column in a list of links on a work item form. (Not supported in query editor.)

Reference Name=System.Links.Comment, Data type=PlainText

All

Link Description

Contains the work item type, ID, and title of the work item that is the target of the link. You can configure this field to appear as a column in a list of links on a work item form. (Not supported in query editor.)

Reference Name=System.Links.Description, Data type=PlainText

All

Parent

When included as a column option in a backlog or query results list, the **Title** of the parent work item is displayed. Internally, the system stores the **ID** of the work item within an Integer field.

> **NOTE**
> You can add the **Parent** field as a column or specify it within a query clause by specifying the parent work item ID.
> Reference Name=System.Parent, Data type=Integer

All

Parent

When included as a column option in a backlog or query results list, the **Title** of the parent work item is displayed. Internally, the system stores the **ID** of the work item within an Integer field.

> **NOTE**
> The **Parent** field is available from Azure DevOps Server 2020 and later versions. You can't specify this field within a query clause.
> Reference Name=System.Parent, Data type=Integer

All

Related Link Count

The number of links defined for a work item which use a work link type, such as Parent-Child, Predecessor-Successor, and Related. For a full list, see Link type reference.
Reference Name=System.RelatedLinkCount, Data type=Integer

All

Remote Link Count

Available for Azure DevOps Services only. The number of links from a work item to work items defined in another organization. Organizations must be managed by the same Azure Active Directory. Supported link types include Consumes From, Produced For, and Remote Related. To learn more, see Add link to work items, Link to a remote work item.
Reference Name=System.RemoteLinkCount, Data type=Integer

All

Related articles

- Add a link to multiple work items
- Linking, traceability, and managing dependencies
- Track dependencies using Delivery Plans
- Query quick reference
- Query editor
- Query fields, operators, and macros
- Add work items
- Work item field index

Visualize related work and other objects

You can view related work items and object within a work item form by installing the Work item visualization extension available from the Visual Studio Marketplace, Azure DevOps tab.

Add custom link types or customize the links controls

To add link types, see Manage link types [witadmin].

All tabs that support creating links between work items are implemented by using the **LinksControl** element on the work item form. This element controls filtering and restricting the types of work items to which you can link, the types of links that you can create, and whether you can link to work items in another project. To customize the link controls and restrictions, you modify the definition of the `LinksControlOptions` for a work item type, see LinksControlOptions XML elements.

Default data fields in lists of links

You can add or remove columns from the list of links, and you can customize the default columns and the column order. For more information, see LinksControlOptions XML elements.

Create a query based on build and test integration fields

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Work item fields that support build and test integration support the following actions:

- Associate bugs with the builds where they were found or fixed
- Query for bugs associated with a build
- Mark test cases as either manual or automated, and store information to support automated test cases
- For test cases and shared steps, define the action and validation steps and the data that are used to run tests.

Supported operators and macros

Most build and test integration fields have a data type of String, PlainText, or HTML. Query clauses that specify a text or rich-text field can use the operators and macros listed in the following table.

Data type	Supported operators and macros
Rich-text (HTML)	Contains Words, Does Not Contain Words, Is Empty[1], Is Not Empty[1]
Multi-line text strings (PlainText)	Contains Words, Does Not Contain Words, Is Empty[1], Is Not Empty[1]
Single text (String)	= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever **Macros**: **[Any]**, valid with the **Work Item Type** field **Project**[2], valid with the **Team Project** field

> **NOTE**
>
> 1. The **Is Empty** and **Is Not Empty** operators are supported for Azure DevOps Server 2019 RC2 and later versions
> 2. The **@Project** macro is supported for Azure Boards and TFS 2015.1 and later versions. The system automatically defaults to filtering based on the current project. To learn more, see Query across projects.

Useful filters

Filter for	Include these query clauses
Automated test cases	`Work Item Type = Test Case` `And Automation Status = Automated`

Query-based test suites

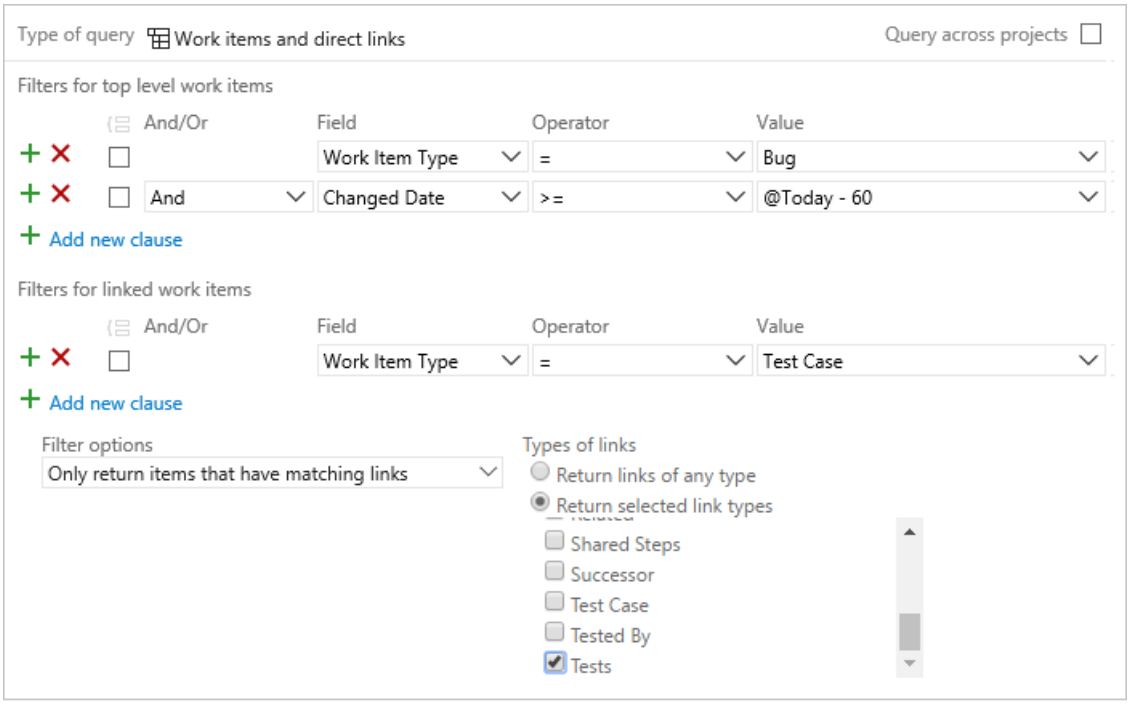
```
Work Item Type = Test Suite And Test Suite Type = Query Based
```

Requirement-based test suites

```
Work Item Type = Test Suite And Test Suite Type = Requirement Based
```

List bugs and the test cases that test them

Open a new query, set the query type to Work items and direct links. Filter for bugs in the top level and add the filter for Test Cases in the linked work items filter.



> **NOTE**
> You can't construct a query that shows a hierarchical view of Test Plans, Test Suites, and Test Cases. These items aren't linked together using parent-child link types. You can view the hierarchy through the **Test > Test Plans** page.

Build and test data fields

The following table describes the fields that are defined in one or more of the test WITs. For information about data types and field attributes, see Work item fields and attributes.

To customize a field or picklist, see Add or modify a field to support queries, reports, and workflow.

Field name

Description

Work item type

Automation Status [1]

The status of a test case. You can specify the following values:

- **Automated**

- **Not Automated**
- **Planned**
  To run automated tests, see Run automated tests from test plans.
  Reference name=Microsoft.VSTS.TCM.AutomationStatus, Data type=String

Test Case

Found In [2]

Product build number, also known as a revision, in which a bug was found.
Reference name=Microsoft.VSTS.Build.FoundIn, Data type=String

> **NOTE**
> You can also use the **Found in build** link type to link a work item to a build. This link type is available from Azure DevOps and only works with the current build processes (not XAML builds).

Bug

Integration Build [2]

Product build number that incorporates the code or fixes a bug.
Reference name=Microsoft.VSTS.Build.IntegrationBuild, Data type=String

> **NOTE**
> You can also use the **Integrated in build** link type to link a work item to a build. This link type is available from Azure DevOps and only works with the current build processes (not XAML builds).

All

Issue

Indicates that the Shared Steps are associated with an expected result. Allowed values are **Yes** and **No**. Reference name=Microsoft.VSTS.Common.Issue, Data type=String

Shared Steps

Parameters [3]

Contains the parameters to use when running a manual test.
Microsoft.VSTS.TCM.Parameters, Data type=HTML

Shared Parameters, Shared Steps, Test Case

Steps

The action and validation steps that are required to run the test. Microsoft.VSTS.TCM.Steps, Data type=HTML

Shared Steps, Test Case

System Info

Information about the software and system configuration that is relevant to the test.
Microsoft.VSTS.TCM.SystemInfo, Data type=HTML

Bug, Feedback Response

Repro Steps (or Steps to reproduce)

The steps that are required to reproduce unexpected behavior. Capture enough information so that other team members can understand the full impact of the problem and whether they've fixed the bug. This includes actions taken to find or reproduce the bug and expected behavior. Reference name=Microsoft.VSTS.TCM.ReproSteps, Data type=HTML

Bug

Test Suite Type [1,4]

The test suite category. Allowed values are:

- **Query Based**: Use to group together test cases that have a particular characteristic - for example, all the tests that have Priority=1. The suite will automatically include every test case that is returned by the query that you define.
- **Static**: Use to group together test cases designed to track the test status of backlog items. Each test case that you add to a requirement-based test suite is automatically linked to the backlog item.
- **Requirement Based**: Use to group together test cases with any characteristics or test suites.
  For more information, see Create a test plan.
  Reference name=Microsoft.VSTS.TCM.TestSuiteType, Data type=String

Test Suite

> **NOTE**
> 1. Do not customize the picklist for these fields. The system accepts only those values listed.
> 2. By adding a `GLOBALLIST` element to the `FIELD` definition, you can provide a drop-down menu of builds that users can choose from. To learn how, see Builds and global list auto-population later in this article.
> 3. Requires TFS 2013.2 or later version to be installed on the application-tier server and existing projects to be updated to support Shared Parameters. To learn more, see Configure features after a TFS upgrade.
> 4. Requires TFS 2013.3 or later version to be installed on the application-tier server and existing projects to be updated to support Test Plan and Test Suite. To learn more, see Configure features after a TFS upgrade.

Other fields

The following fields don't appear on work item forms, but these fields are tracked for test cases or test suites. You can use some of these fields to filter queries and create reports. (None of these fields are added to the data warehouse nor indexed.)

Field name

Description

Work item type

Automated Test Storage

The assembly that contains the test that automates the test case.

Reference name=Microsoft.VSTS.TCM.AutomatedTestStorage, Data type=String

Test Case

Automated Test Type

The type of test that automates the test case.

Reference name=Microsoft.VSTS.TCM.AutomatedTestType, Data type=String

Test Case

AutomatedTestId

The ID of the test that automates the test case.

Reference name=Microsoft.VSTS.TCM.AutomatedTestId, Data type=String

Test Case

AutomatedTestName

The name of the test that is used to automate the test case.

Reference name=Microsoft.VSTS.TCM.AutomatedTestName, Data type=String

Test Case

LocalDataSource

The local data source that supports the test.

Reference name=Microsoft.VSTS.TCM.LocalDataSource, Data type=HTML

Test Case

Query Text

Field used to capture the query defined for a Query-based suite type.

Reference name=Microsoft.VSTS.TCM.QueryText, Data type=PlainText

Test Suite

Test Suite Audit [1]

Tracks other operations run when modifying a test suite, for example: adding tests to a test suite or changing configurations. This field can be viewed through the History tab or through a separate query. There will be a combined history view, including changes done to work items field and changes resulting from related artifacts such as test points and configurations.

Reference name=Microsoft.VSTS.TCM.TestSuiteAudit, Data type=PlainText

Test Suite

Test Suite Type ID [1, 2]

A system assigned value that corresponds to the test suite category and only applicable to test suites. Assigned values are:

- **1** (Static)
- **2** (Query-based)
- **3** (Requirement- based)

Reference name=Microsoft.VSTS.TCM.TestSuiteTypeId, Data type=Integer

Test Suite

NOTE

1. Requires TFS 2013.3 or later version to be installed on the application-tier server and existing projects to be updated to support Test Plan and Test Suite.
2. Do not customize the picklist for these fields. The system accepts only those values listed.

Fields that integrate with Team Foundation Build

Team Foundation Build is the on-premise build system you can use with Azure DevOps Server and TFS. You can configure your build process by using Team Foundation Build, and Team Foundation Build can generate work items when a build fails. It can also add build information to work items that were resolved in a particular build. For this to work, Team Foundation Build requires that the following two fields be added to the work item type definition: **Found In** and **Integration Build**.

Found In and **Integrated in Build** fields are defined for Bugs in the default processes. These fields associate bugs with the builds where they were found or fixed.

You can use the following code snippet to add these fields to a WIT definition.

```
<FIELD name="Found In" refname="Microsoft.VSTS.Build.FoundIn" type="String" reportable="dimension">
    <HELPTEXT>Product build number (revision) in which this item was found</HELPTEXT>
        <SUGGESTEDVALUES>
          <LISTITEM value="&lt;None&gt;" />
        </SUGGESTEDVALUES>
</FIELD>
<FIELD name="Integration Build" refname="Microsoft.VSTS.Build.IntegrationBuild" type="String"
reportable="dimension">
    <HELPTEXT>Product build number this bug was fixed in</HELPTEXT>
        <SUGGESTEDVALUES>
          <LISTITEM value="&lt;None&gt;" />
        </SUGGESTEDVALUES>
</FIELD>
```

When the **Found In** field is present in a WIT definition, Team Foundation Build creates a work item when a build fails, and sets the **Found In** field to the build number of the build that failed. If the **Found In** field is missing, Team Foundation Build doesn't create a work item for the failed build, and everything else works as expected.

When the **Integration Build** field is present in the WIT definition, Team Foundation Build identifies work items that were resolved with each build and then updates those work items to set the build number in which they were resolved in the **Integration Build** field. If the **Integration Build** field is missing, Team Foundation Build doesn't store the build number in the work items, and everything else works as expected.

Builds and global list autopopulation

The first time you queue a build for a project using Team Foundation Build, TFS automatically adds a global list labeled **Build - *ProjectName***. Each time a build is run, a **LISTITEM** is added to this global list with the name of the build.

By adding a **GLOBALLIST** element to the **FIELD** definition, you can provide a drop-down menu of builds that users can choose from. For example:

```
<FIELD name="Found In" refname="Microsoft.VSTS.Build.FoundIn" type="String" reportable="dimension">
    <HELPTEXT>Product build number (revision) in which this item was found</HELPTEXT>
        <SUGGESTEDVALUES>
          <LISTITEM value="&lt;None&gt;" />
        </SUGGESTEDVALUES>
        <SUGGESTEDVALUES expanditems="true" filteritems="excludegroups">
          <GLOBALLIST name="Builds - TeamProjectName" />
        </SUGGESTEDVALUES>
</FIELD>
```

Fields that Integrate with Test Plans

With Test Plans, you can automate the creation of a bug or other type of work item when a test fails. For more information, see Add findings to existing bugs with exploratory testing.

When a work item has been created in this manner, information about the system and the steps to reproduce the bug are captured in the **System Info** and **Repro Steps** fields.

You can add these fields to work item types that you create for tracking defects using the following code snippet.

```
<FIELD name="System Info" refname="Microsoft.VSTS.TCM.SystemInfo" type="HTML" />
<FIELD name="Repro Steps" refname="Microsoft.VSTS.TCM.ReproSteps" type="HTML" />
```

Fields that integrate with Team Foundation Version Control

One of the features available in Team Foundation version control (TFVC) enables you to associate or resolve work items when you check in code. You might have worked on a particular work item when you make a code change and you can set that association from within the source-control check-in window when you're finished working on the code.

The ability of Team Foundation version control to resolve a work item requires that work items contain a particular action. The source control system then queries work item tracking to determine whether the work item supports that action, and if it does support that action, it also queries for the source and destination states of the transition. If the action is found, the source control system can transition the work item according to the set transition when it checks in the code.

> **NOTE**
> When you use the **Checkin** action, you must set appropriate *from* and *to* states to reflect the state transition that you want.

For more information about Actions, see Automate field assignments based on State, Transition, or Reason.

Related articles

- Work item field index
- Drive Git development from a work item
- Linking, traceability, and managing dependencies
- Link and attachment queries

Query fields, operators, and macros in Azure Boards

10/4/2022 • 15 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Here you'll find detailed descriptions of each field data type, query operators, and query macros. Some data types, operators, and macros are only valid for the indicated Azure DevOps Server or Team Foundation Server (TFS) version.

For a quick reference of query tasks and operators and macros supported for each data type, see Query quick reference. See also Guidance to create high-performing queries for tips on constructing high-performing queries.

Query field data types and values

The value you specify for a field must conform to the data type for that field. The following table lists the supported data types:

> **NOTE**
> For Azure Boards cloud service, the data type corresponds to that listed for the field on the Process>Fields page. For on-premises deployments, the data type corresponds to the `type` attribute assigned to a `FIELD` definition. For more information, see Work item fields and field attributes.

Data type

Description

Boolean[1]

Specifies a field that takes on a True/False value.

DateTime or **Date/Time**

A date field in which you can specify a variable, such as **@Today** or **@Today-1**, or a value, such as 1/1/2012. Enter dates in the Date Pattern you set for your personal profile. (See Set personal preferences for details.) For query examples, see Query by date or@CurrentIteration.

For WIQL queries, you can also specify the date in the Coordinated Universal Time (UTC) pattern. For details, see Syntax for the Work Item Query Language (WIQL).

Double or **Decimal**

A real number, such as 0.2 or 3.5. For query examples, see Query by numeric fields.

GUID

A character string that represents a unique ID.

History

Custom formatted field used to track historical information. This data type is only used to support the **History** field. This field is automatically indexed for full-text search when full-text search is available. See Full-Text and partial word searches described later in this article. For query examples, see History and auditing.

HTML

Text strings that support formatted descriptions, such as the **Description** or **Repro Steps** fields. These fields are automatically indexed for full-text search when full-text search is available. See Full-Text and partial word searches described later in this article. To query rich-text fields, see Query by titles, IDs, and rich-text fields.

Identity

Short text string that identifies a user identity.

Integer

A 32-bit integer that is signed, such as 0, 1, 2, 34.

PlainText or Text field (multi-line)

Text strings that support long descriptions, such as the **Application Start Information** field. These fields are automatically indexed for full-text search, when full-text search is available. See Full-Text and partial word searches described later in this article. To query plain-text fields, see Query by titles, IDs, and rich-text fields.

picklistDouble[2]

Custom field defined to contain a pick list of Decimal values.

picklistInteger[2]

Custom field defined to contain a pick list of Integer values.

picklistString[2]

Custom field defined to contain a pick list of short text string (255 characters or less) values.

String or Text field (single line)

Short text string that can contain up to 255 Unicode characters. String text fields are often used to support picklists or drop-down menus.

TreePath

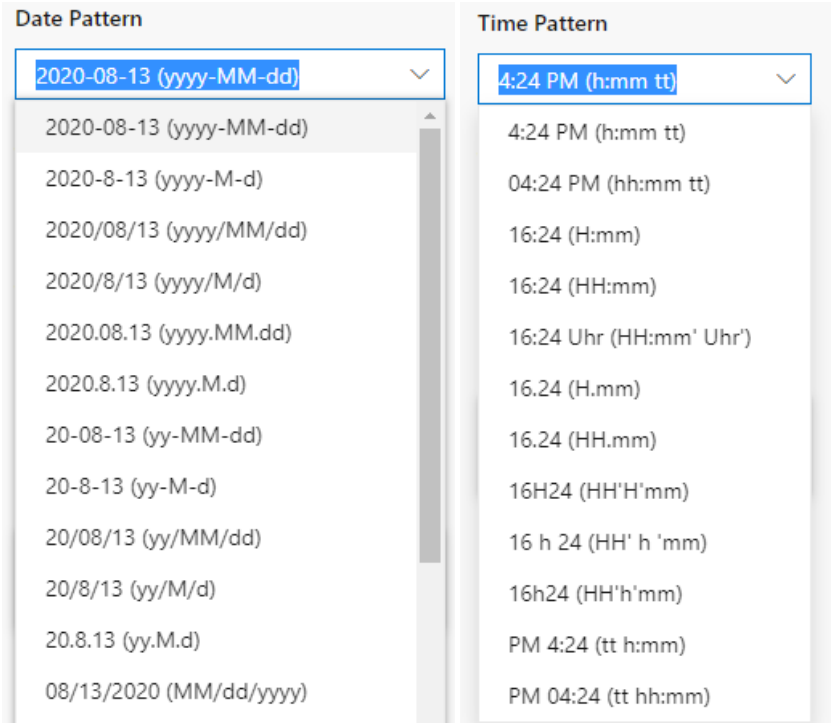
A branching tree structure, such as an Area Path or Iteration path. Choose an item from a list of valid values. Find work items that equal, not equal, under or not under a tree structure, or use the In or Not In operators to specify several values. You define the tree structure for a project—area paths and iteration paths—and then select the ones you want to associate with a team.

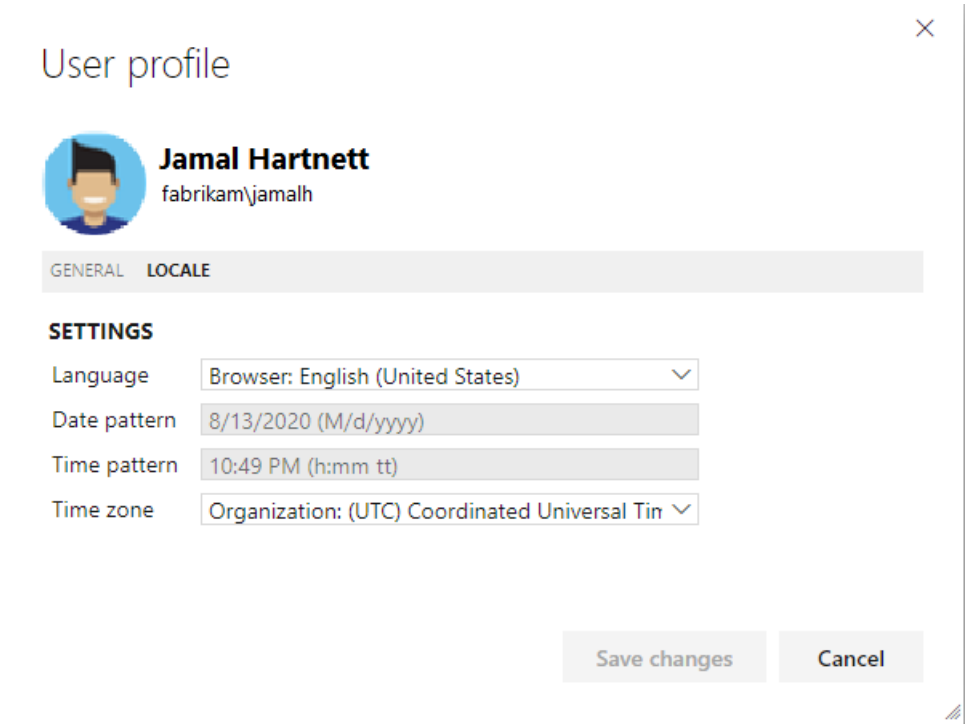
For more information on constructing queries, see Query by area or iteration path or Query by date or current iteration.

> **NOTE**
>
> 1. The **Boolean** data type field is supported for TFS 2017 and later versions.
> 2. The **picklist...** data types are only assigned to custom fields defined for an inherited process. The Inherited process model is only supported for Azure DevOps Services and Azure DevOps Server 2019.

Date and time pattern

The date and time pattern you enter for **DateTime** fields should match that which you select through your profile. To view or change your selection, see Set user preferences, Time and Locale.





Query operators

You can use query operators in the following table to specify how each value in a clause must relate to the corresponding value in a work item. For information about the data type that is assigned to work item fields, see Work item field reference.

To learn about adding clauses and use of the And/Or operators, see Define a query, And/Or logical expression.

Query operator

Returns work items if the value in the work item matches the criteria listed

Applicable data types

=

Matches the value in the clause.

Number—which includes **Double**, **GUID**, **Integer**—and **String**, **DateTime**, and **TreePath**

< >

Doesn't match the value in the clause.

Number, **String**, **DateTime**, and **TreePath**

>

Is larger than the value in the clause.

Number, **String**, and **DateTime**

<

Is less than the value in the clause.

Number, **String**, and **DateTime**

> =

Is larger than or equal to the value in the clause.

Number, **String**, and **DateTime**

< =

Is less than or equal to the value in the clause.

Number, **String**, and **DateTime**

=[Field]

Matches the value that is contained in the specified field.

Name of a field that is of the same data type as the specified field

< >[Field]

Doesn't match the value that is contained in the specified field.

Name of a field that is of the same data type as the specified field

>[Field]

Is larger than the value that is contained in the specified field.

Name of a field that is of the same data type as the specified field

<[Field]

Is less than the value that is contained in the specified field.

Name of a field that is of the same data type as the specified field

> =[Field]

Is larger than or equal to the value that is contained in the specified field.

Name of a field that is of the same data type as the specified field

< =[Field]

Is less than or equal to the value that is contained in the specified field.

Name of a field that is of the same data type as the specified field

Contains

Contains an exact or partial match of the text string within the field you selected for filtering.

String

Does Not Contain

Doesn't contain an exact or partial match of the text string within the field you selected for filtering.

String

Contains Words

Contains the exact text string or words within the field you selected for filtering. You can also enter partial words or phrases that contain the wildcard character, *. Text string is limited to 100 characters. For restrictions, see Full-text searches for server and collation requirements.

Long-text fields that are indexed for full-text search, which correspond to all **PlainText** and **HTML** fields, and the **History** and **Title** fields.

Does Not Contain Words

Doesn't contain the exact text string or words within the field you selected for filtering. Text string is limited to 100 characters.

Use this operator in combination with a clause with the **Contains Words** operator to include and exclude specific keywords.

Text fields that are indexed for full text search.

In

Matches any value in a delimited set. For example, you can find work items whose IDs are 100, 101, and 102 if you specify those values for the ID field. Separate values with the list separator that corresponds to the regional settings that are defined for your client computer. For example, you might use a comma(,).

Number, **String**, **DateTime**, **TreePath**

Is Empty

Lists work items that contain an empty HTML field. You don't specify a value with this operator. This operator is supported for Azure Boards (cloud service), Azure DevOps Server 2019, and later versions.

HTML

Is Not Empty

Lists work items that contain some content in the HTML field. You don't specify a value with this operator. This operator is supported for Azure Boards (cloud service), Azure DevOps Server 2019, and later versions.

HTML

Not In

Doesn't match any value in a delimited set. You can exclude work items whose states are not Resolved, Completed, or Closed from query results if you specify those values for the State field. Separate values with the list separator that corresponds to the regional settings that are defined for your client computer. For example, you might use a comma(,).

> The **Not In** operator is available from Azure Boards and TFS 2018.2 and later versions.

Number, **String**, **DateTime**, **TreePath**

In Group

Matches a value that is a member of the group in the clause. Groups correspond to the name of a team, security group, or work tracking category. For example, you can create a query to find all work items that are assigned to members of the Contributors group or to a team. Team groups are created when you create a team. The name of team groups follows the pattern [*Team Project Name*]*Team Name*.

For example queries, see Query by assignment or workflow changes.

String that matches the name of a team, security group, or category defined in the system.

> **NOTE**
> You can use the **In Group** operator only with fields that use the **String** data type or the **Work Item Type** field. You can also use groups defined in Azure Active Directory (Azure AD) when your Azure Boards account is backed by Azure AD, or Active Directory when your on-premises server instance is backed by Active Directory.

For information about category groups, see Use categories to group work item types.

Not in Group

Doesn't match a value that is a member of the group in the clause.

String that matches the name of a user group in Team Foundation Server or a category group defined for a project.

> **NOTE**
> You can use the **Not In Group** operator only with fields that use the **String** data type or the **Work Item Type** field. You can also use groups defined in Azure AD when your Azure Boards account is backed by Azure AD, or Active Directory when your on-premises server instance is backed by Active Directory.

Not Under

Doesn't match the value in the clause and isn't contained under the node in the clause.

TreePath

Under

Matches the value in the clause or is contained under the node in the clause.

TreePath

Was Ever

Matches the value in the clause at any previous point.

String , **DateTime**

TIP

It's possible to contruct a query using WIQL syntax that uses an operator, such as **Was Ever**, for other data type fields than those listed. For example, you can use **Was Ever** within a clause using the **Iteration Path**. For an example, see Query by date or current iteration, List work items moved out of a sprint.

Query macros or variables

You can use the macros described in the following table to filter your queries based on specific fields.

NOTE

The following macros are only supported from the web portal: **@CurrentIteration**, **@CurrentIteration +/- n**, **@Follows**, **@MyRecentActivity**, **@RecentMentions**, **@RecentProjectActivity**, and **@TeamAreas**. Queries that contain these macros won't work when opened in Visual Studio/Team Explorer, Microsoft Excel, or Microsoft Project.

Macro	Description
[Any]	Use with the **Work Item Type** or **State** fields to search across all work item types or across all states. For example, `Work Item Type=[Any]` won't place any filters based on the work item type.
@CurrentIteration	Use with the **Iteration Path** field to automatically filter for work items assigned to the current sprint based on the current team focus or context. For specific examples, see Query by date or current iteration. The **@CurrentIteration** macro is supported for Azure Boards and TFS 2015 and later versions. This macro only works when run from the web portal. You can't use the macro when copying or cloning test suites and test cases, defining alerts, or with REST APIs.
@CurrentIteration +/- n	Use with the **Iteration Path** field to filter the set of work items assigned to the current sprint +/- *n* sprints based on the current team focus or context. For specific examples, see Query by date or current iteration. The **@CurrentIteration +/- n** macro is supported for Azure Boards, Azure DevOps Server 2019 and later versions, and only when run from the web portal.
@Follows	Use with the **ID** field and **In** operator to list all work items that you are following in the project. To learn more about the Follow feature, see Follow a work item or pull request. You can view this same list from the Work Items page, **Following** pivot view. The **@Follows** macro is supported for Azure Boards and TFS 2017 and later versions, and only when run from the web portal.
@Me	Use with an identity or user account field to automatically search for items associated with your user or account name. For example, you can find work items that you opened with the clause `Created By=@Me`. For more examples, see Query by assignment, workflow, or Kanban board changes.

@MyRecentActivity [1]

Use with the **ID** field and **In** operator to list work items that you have viewed or updated in the project within the last 30 days. You can view this same list from the Work Items page, **My activity** pivot view.

@Project

Use with the **Team Project** field to filter for work items in other projects. For example, you can find all the work items in the currently selected project with the clause `Team Project=@Project`.
The **@Project** macro is supported for Azure Boards and TFS 2015.1 and later versions. The system automatically defaults to filtering based on the current project. To learn more, see Define a query, Query across projects.

@RecentMentions [1]

Use with the **ID** field and **In** operator to list work items where you have been mentioned in the Discussion section. You can view this same list from the Work Items page, **Mentioned** pivot view.

@RecentProjectActivity [1]

Use with the **ID** field and **In** operator to list work items that have been recently updated. The number of work items listed depends on the work tracking activity of the project. For highly active projects, the macro will list work items that have been updated in the project within the last 30 days or so. For less active projects, however, this list could include work items older than 30 days. You can view similar lists from the Work Items page, **Recently created**, **Recently updated** and **Recently completed** pivot views. The number of work items returned is capped at 5000.

@StartOfDay [2]

Use with a `DateTime` field to filter for work items that relate to the current date or with a plus/minus offset. For example, you can find all items closed in the last week with the clause `Closed Date>=@StartOfDay-7`. For more examples, see Query by date or current iteration.

@StartOfMonth [2]

Use with a `DateTime` field to filter for work items that relate to the current month or with a plus/minus offset. For example, you can find all items created in the last three months with the clause `Created Date>=@StartOfMonth-3`. For more examples, see Query by date or current iteration.

@StartOfWeek [2]

Use with a `DateTime` field to filter for work items that relate to the current week or with a plus/minus offset. For example, you can find all items changed in the last two weeks with the clause `Changed Date>=@StartOfWeek-2`. For more examples, see Query by date or current iteration.

@StartOfYear [2]

Use with a `DateTime` field to filter for work items that relate to the current year or with a plus/minus offset. For example, you can find all features that have a Target Date scheduled within the current year with the clause `Target Date>=@StartOfYear`. For more examples, see Query by date or current iteration.

@TeamAreas

Only use with the **Area Path** field to filter for work items whose area path corresponds to one assigned to a specific team. Requires you use the = operator. For example, you can find all items assigned to the area paths assigned to the Web team with the clause `Area Path=@TeamAreas [Fabrikam Fiber]\Web`. For more examples, see

Query by area or iteration path.
The **@TeamAreas** macro is supported for Azure DevOps Server 2019 and later versions, and only when run from the web portal.

@Today

Use with a `DateTime` field to filter for work items that relate to the current date or to an earlier date. You can also modify the **@Today** macro by subtracting days. For example, you can find all items created in the last week with the clause `Created Date>=Today-7`. For more examples, see Query by date or current iteration.

> **NOTE**
> 1. The **@MyRecentActivity**, **@RecentMentions**, and **@RecentProjectActivity** macros are supported for TFS 2018.2 and later versions.
> 2. The **@StartOfDay**, **@StartOfWeek**, **@StartOfMonth**, and **@StartOfYear** macros are supported for Azure DevOps Server 2019 Update 1 and later versions.

Full-text and partial word searches

Specify **Contains** or **Does Not Contain** to search against exact or partial matches of a word or phrase. These operators filter items based on the full-text search index created for long-text fields. Specify **Contains Words** or **Does Not Contain Words** to search against an exact phrase or to use the wildcard character, *. These operators use the full-text search index. You can only use the wildcard character at the end of a partial word or phrase.

For examples, see Example work item queries and Query for work items using the History field.

> **NOTE**
> Not all deployments support full-text searches. For example, SQL Express and SQL Azure, which support the cloud service, do not support full-text search. In these instances, you will only see the **Contains** and **Does not Contain** operators.

Azure DevOps Server and Team Foundation Server automatically index all long-text fields with a data type of **PlainText** and **HTML** and the **Title** field for full-text search. The index and operators are only available when the SQL Server that supports Team Foundation Server supports full-text search.

Full-text searches require a SQL collation that corresponds to a language that has a word breaker registered with SQL Server. If the collation settings for the project collection database used for your Team Foundation Server instance don't correspond to a supported language, your search results may not match your expectations. In these cases, you might try using the **Contains** or **Does Not Contain** operators.

For more information, see Full-Text Search Queries and Collation Settings.

Related articles

- Query quick reference
- About managed queries
- Work item field index
- Syntax for the Work Item Query Language (WIQL)

REST API

To programmatically interact with queries, see one of these REST API resources:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Fetch work items with queries programmatically

Work Item Query Language (WIQL) syntax reference

10/4/2022 • 19 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can use the WIQL syntax to define a query as a hyperlink or when using the Work Item Query Language (REST API).

The WIQL syntax supports all functions available through the web portal Query Editor plus a few more. You can specify the fields to return and specify logical grouping of query clauses. In addition, you can use an `ASOF` clause to filter based on assignments based on a previous date.

> **IMPORTANT**
> The WIQL syntax is used to execute the Query By Wiql REST API. Currently, there is no way to call the API to return the detailed work item information from a WIQL query directly. No matter which fields you include in the SELECT statement, the API only returns the work item IDs. To get the full information, you need to perform two steps: (1) get the ID of the work items from a WIQL, and (2) get the work items via Get a list of work items by ID and for specific fields.

Prerequisites

A query returns only those work items for which you have the **View work items** or **View work items in this node** permission. Typically, these permissions are granted to members of the **Readers** and **Contributors** groups for each team project. For more information, see Permissions and groups.

Query language overview

The work item query language has five parts shown in the following syntax snippet and described in the following table. WIQL syntax isn't case-sensitive.

```
SELECT
    [System.Id],
    [System.AssignedTo],
    [System.State],
    [System.Title],
    [System.Tags]
FROM workitems
WHERE
    [System.TeamProject] = 'Design Agile'
    AND [System.WorkItemType] = 'User Story'
    AND [System.State] = 'Active'
ORDER BY [System.ChangedDate] DESC
ASOF '02-11-2020'
```

> **TIP**
> By installing the Wiql Editor Marketplace extension, you can construct your queries using the Query Editor and then view the WIQL syntax. You can then copy and modify the WIQL syntax and run the query using the **Wiql Playground** hub added to **Boards**.

Clause

Example

`SELECT`

Identifies the fields to return for each work item returned by the query. You can specify either the friendly name or reference name. Use square brackets ([]) if the name contains blanks or periods.

`FROM`

Indicates whether you want the query to find work items or links between work items.

- Use `FROM WorkItems` to return work items.
- Use `FROM workItemLinks` to return links between work items. For more information, see Queries for links between work items later in this article.

`WHERE`

Specifies the filter criteria for the query. For more information, see Filter conditions (WHERE) later in this article.

`ORDER BY`

Specifies the sort order of the work items returned. You can specify Ascending (Asc) or Descending (Desc) for one or more fields. For example:
`ORDER BY [State] Asc, [Changed Date] Desc`

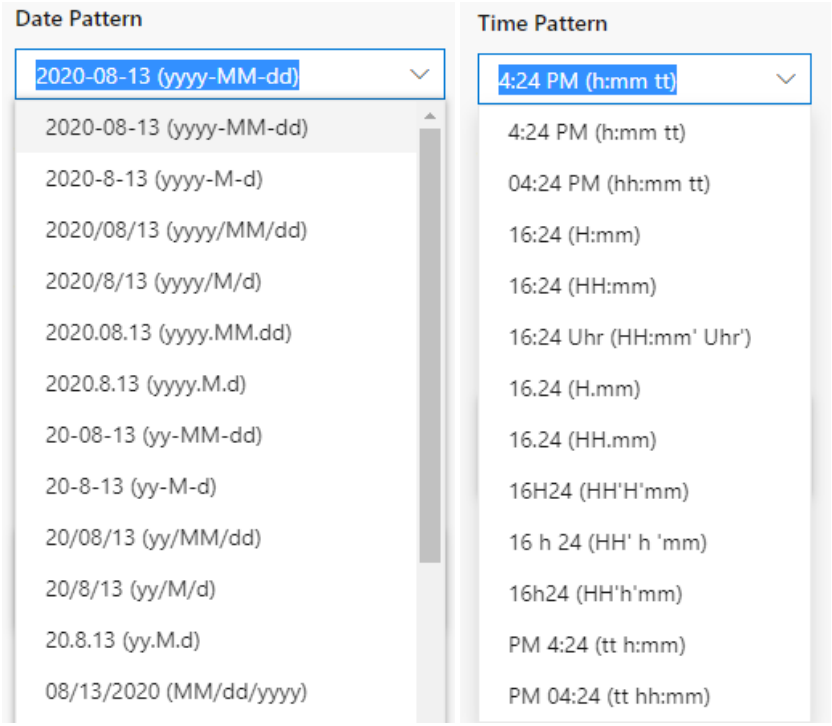
`ASOF`

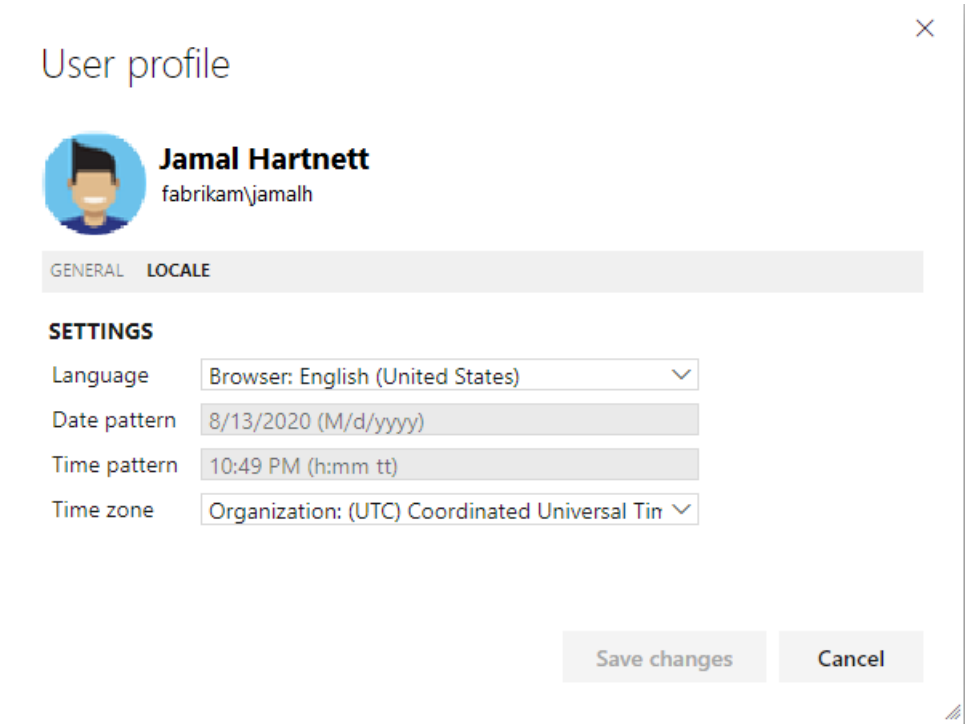
Specifies a historical query by indicating a date for when the filter is to be applied. For example, this query returns all user stories that were defined as *Active* on February 11, 2020. Specify the date according to the guidance provided in Date and time pattern. `ASOF '02-11-2020'`

> **NOTE**
> The WIQL length of queries made against Azure Boards must not exceed 32K characters. The system won't allow you to create or run queries that exceed that length.

Date and time pattern

The date and time pattern you enter for **DateTime** fields should match that which you select through your profile. To view or change your selection, see Set user preferences, Time and Locale.





Quote (single or double quotes are supported) DateTime literals used in comparisons. They must be in the .NET DateTime format of the local client computer running the query. Unless a time zone is specified, DateTime literals are in the time zone of the local computer.

```
WHERE [System.ResolvedDate] >= '01-18-2019 GMT' and [Resolved Date/Time] < '01-09-2019 GMT'
WHERE [Resolved Date] >= '01-18-2019 14:30:01'
```

When the time is omitted in a DateTime literal and the dayPrecision parameter equals false, the time is assumed to be zero (midnight). The default setting for the dayPrecision parameter is false.

Custom fields

You can add a custom field to a query clause. With WIQL, you must specify the reference name for the custom field. For projects that use an Inherited process model, custom fields are typically labeled with *Custom.* prepended to their name, and spaces removed. For example:

FRIENDLY NAME	REFERENCE NAME
Approver	Custom.Approver
Request Type	Custom.RequestType
Scope Estimate	Custom.CustomEstimate

For projects that use the On-premises XML process model, the reference name is as defined by the XML work item type definitions.

To learn more, see Work item fields and attributes.

You can add a custom field to a query clause. With WIQL, you must specify the reference name for the custom field.

To learn more, see Add or modify a field to track work.

Specify filter clauses (`WHERE`)

The `WHERE` clause specifies the filter criteria. The query returns only work items that satisfy the specified criteria. For example, the following example `WHERE` clause returns user stories that are active and that are assigned to you.

```
WHERE [Work Item Type] = 'User Story'
   AND [State] = 'Active'
   AND [Assigned to] = @Me
```

You can control the order in which logical operators are evaluated by enclosing them within parentheses to group the filter criteria. For example, to return work items that are either assigned to you or that you closed, change the query filter to match the following example.

```
WHERE
    [System.TeamProject] = @project
    AND (
        [System.WorkItemType] = 'Product Backlog Item'
        AND (
            [System.AssignedTo] = @me
            OR [Microsoft.VSTS.Common.ClosedBy] = @me
        )
    )
```

Filter conditions

Each filter condition is composed of three parts, each of which must conform to the following rules:

- **Field**: You can specify either the reference name or friendly name. The following examples are valid WIQL syntax:
  - Reference name: `SELECT [System.AssignedTo] ...`
  - Friendly name with spaces: `SELECT [Assigned To] ...`
  - Names without spaces don't require square brackets: `SELECT ID, Title ...`
- **Operator**: Valid values are specified in the Operators section later in this article.
- **Field value**: You can specify one of the following three values depending on the field specified.
  - A *literal value* must match the data type of the field value.

- A *variable or macro that indicates a certain value. For example, **@Me** indicates the person who is running the query. For more information, see Macros and variables later in this article.
- The name of another *field*. For example, you can use `[Assigned to] = [Changed by]` to find work items that are assigned to the person who changed the work item most recently.

For a description and reference names of all system-defined fields, see Work item field index.

Operators

Queries use logical expressions to qualify result sets. These logical expressions are formed by one or more conjoined operations.

Some simple query operations are listed below.

```
WHERE
    [System.TeamProject] = @project
    AND [System.WorkItemType] <> ''
    AND [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
    AND [Microsoft.VSTS.Common.Severity] <> '1 - Critical'
```

The table below summarizes all the supported operators for different field types. For more information on each field type, see Work item fields and attributes.

The `=, <>, >, <, >=, and <=` operators work as expected. For instance, `System.ID > 100` queries for all work items with an **ID** greater than 100. `System.ChangedDate > '01-01-19 12:00:00'` queries for all work items changed after noon of January 1, 2019.

Beyond these basic operators, there are some behaviors and operators specific to certain field types.

> **NOTE**
> The operators available to you depend on your platform and version. For more information, see Query quick reference.

Field type

Supported operators

Boolean

`= , <> , =[Field] , <>[Field]`

DateTime

`= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever`

Double, GUID, Integer

`= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], In, Not In, Was Ever`

Identity

`= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever`

PlainText

`Contains Words, Does Not Contain Words, Is Empty, Is Not Empty`

String

`= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever`

TreePath

`=, <>, In, Not In, Under, Not Under`

Logical groupings

You can use the terms `AND` and `OR` in the typical Boolean sense to evaluate two clauses. You can use the terms `AND EVER` and `OR EVER` when specifying a `WAS EVER` operator. You can group logical expressions and further conjoin them, as needed. Examples are shown below.

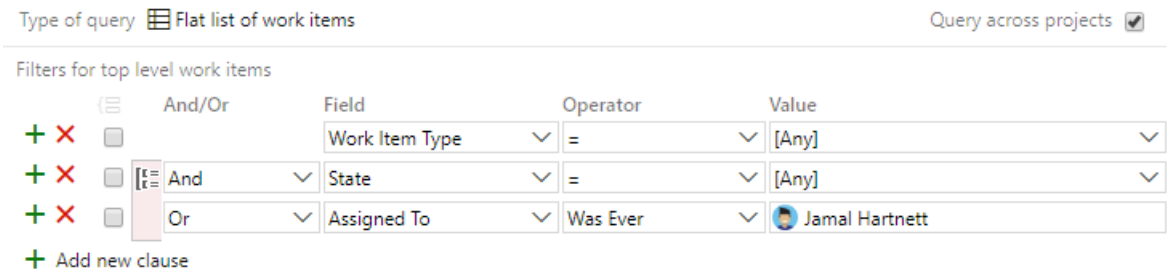
```
WHERE
    [System.TeamProject] = @project
    AND (
        [System.WorkItemType] <> ''
        AND [System.State] IN ('Active', 'Approved', 'Committed', 'In Progress')
        AND (
            [System.CreatedBy] = ''
            OR [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
        )
    )
```

You can negate the `contains, under,` and `in` operators by using `not`. You can't negate the `ever` operator. The example below queries for all work items that aren't assigned under the subtree of `Fabrikam Fiber\Account Management`.

```
WHERE
    [System.TeamProject] = @project
    AND [System.WorkItemType] <> ''
    AND NOT [System.AreaPath] UNDER 'Fabrikam Fiber\Account Management'
```

Example query, Was Ever Assigned To

The following Query Editor example finds all work items that were ever assigned to *Jamal Hartnett*.



And, here is the corresponding WIQL syntax.

```
SELECT
    [System.Id],
    [System.Title],
    [System.State],
    [System.IterationPath]
FROM workitems
WHERE
    [System.TeamProject] = @project
    AND [System.WorkItemType] <> ''
    AND EVER [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
```

Macros or variables

The following table lists the macros or variables you can use within a WIQL query.

MACRO	USAGE
@Me	Use this variable to automatically search for the current user's alias in a field that contains user aliases. For example, you can find work items that you opened if you set the **Field** column to **Activated By**, the **Operator** column to **=**, and the **Value** column to **@Me**.
@CurrentIteration	Use this variable to automatically filter for work items assigned to the current sprint for the selected team based on the selected team context.
@Project	Use this variable to search for work items in the current project. For example, you can find all the work items in the current project if you set the **Field** column to **Team Project**, the **Operator** column to **=**, and the **Value** column to **@Project**.
@StartOfDay **@StartOfWeek** **@StartOfMonth** **@StartOfYear**	Use these macros to filter DateTime fields based on the start of the current day, week, month, year, or an offset to one of these values. For example, you can find all items created in the last 3 months if you set the **Field** column to **Created Date**, the **Operator** column to **>=**, and the **Value** column to **@StartOfMonth - 3**.
@Today	Use this variable to search for work items that relate to the current date or to an earlier date. You can also modify the **@Today** variable by subtracting days. For example, you can find all items activated in the last week if you set the **Field** column to **Activated Date**, the **Operator** column to **>=**, and the **Value** column to **@Today - 7**.
[Any]	Use this variable to search for work items that relate to any value that is defined for a particular field.

MACRO	USAGE
@Me	Use this variable to automatically search for the current user's alias in a field that contains user aliases. For example, you can find work items that you opened if you set the **Field** column to **Activated By**, the **Operator** column to **=**, and the **Value** column to **@Me**.
@CurrentIteration	Use this variable to automatically filter for work items assigned to the current sprint for the selected team based on the selected team context.
@Project	Use this variable to search for work items in the current project. For example, you can find all the work items in the current project if you set the **Field** column to **Team Project**, the **Operator** column to **=**, and the **Value** column to **@Project**.

MACRO	USAGE
@Today	Use this variable to search for work items that relate to the current date or to an earlier date. You can also modify the **@Today** variable by subtracting days. For example, you can find all items activated in the last week if you set the **Field** column to **Activated Date**, the **Operator** column to **>=**, and the **Value** column to **@Today - 7**.
[Any]	Use this variable to search for work items that relate to any value that is defined for a particular field.

@me macro

The `@me` macro replaces the Windows Integrated account name of the user who runs the query. The example below shows how to use the macro and the equivalent static statement. The macro is intended for use with identity fields such as `Assigned To`.

```
WHERE
   [System.AssignedTo] = @Me
```

@today macro

You can use the `@today` macro with any **DateTime** field. This macro replaces midnight of the current date on the local computer that runs the query. You can also specify `@today+x` or `@today-y` using integer offsets for x days after `@today` and y days before `@today`, respectively. A query that uses the `@today` macro can return different result sets depending on the time zone in which it's run.

The examples below assumes that today is 1/3/19.

```
WHERE
   [System.CreatedDate] = @today
```

Is the equivalent of:

```
WHERE
   [System.CreatedDate] = '01-03-2019'
```

And

```
WHERE
   [System.CreatedDate] > @today-2
```

Is the equivalent of:

```
WHERE
   [System.CreatedDate] > '01-01-2019'
```

@StartOfDay, @StartOfWeek, @StartOfMonth, @StartOfYear macros

You can use the `@StartOf...` macros with any **DateTime** field. This macro replaces midnight of the current day, start of week, start of month, or start of year on the local computer that runs the query.

> **NOTE**
> Requires Azure DevOps Server 2019 Update 1 or later version.

These macros accept a modifier string that has a format of `(+/-)nn(y|M|w|d|h|m)`. Similar to the `@Today` macro, you can specify plus or minus integer offsets. If the time unit qualifier is omitted, it defaults to the natural period of the function. For example, `@StartOfWeek("+1")` is the same as `@StartOfWeek("+1w")`. If the plus/minus (+/-) sign is omitted, plus is assumed.

This syntax allows you to nest modifiers and offset your query twice. For example, the clause `Closed Date >= @StartOfYear - 1`, filters work items that have been closed since last year. By modifying it to `Closed Date >= @StartOfYear('+3M') - 1`, it excludes work items closed within the first three months of the last year. The WIQL syntax is as shown in the following example.

```
WHERE
   [Microsoft.VSTS.Common.ClosedDate] >=@StartOfYear('+3M') - 1
```

The following examples assume that today is 4/5/19.

```
WHERE
   [Microsoft.VSTS.Common.CreatedDate] >= @StartOfMonth-3
```

Is the equivalent of:

```
WHERE
   [Microsoft.VSTS.Common.CreatedDate] >= '01-01-2019'
```

And

```
WHERE
   [Microsoft.VSTS.Scheduling.TargetDate] > @StartOfYear
```

Is the equivalent of:

```
WHERE
   [Microsoft.VSTS.Scheduling.TargetDate]  > '01-01-2019'
```

Custom macros

WIQL also supports arbitrary custom macros. Any string prefixed by an `@` is treated as a custom macro and will be substituted. The replacement value for the custom macro is retrieved from the context parameter of the query method in the object model. The following method is the API used for macros:

```
public WorkItemCollection Query(string wiql, IDictionary context)
```

The context parameter contains key-value pairs for macros. For example, if the context contains a key-value pair of (project, MyProject), then **@project** will be replaced by `MyProject` in the WIQL. This replacement is how the work item query builder handles the **@project** macro in Visual Studio.

Specify historical queries (`ASOF`)

You can use an `ASOF` clause in a query to filter for work items that satisfy the specified filter conditions as they were defined on a specific date and time.

> **NOTE**
> You can't create `ASOF` queries in the query builder in Visual Studio. If you create a query file (.wiq) that includes an `ASOF` clause, and then load that in Visual Studio, the `ASOF` clause is ignored.

Suppose a work item was classified under an **Iteration Path** of `Fabrikam Fiber\Release 1` and assigned to 'Jamal Hartnett' prior to 5/05/2022. However, the work item was recently assigned to 'Raisa Pokrovskaya' and moved to a new iteration path of Release 2. The following example query will return this work items assigned to Jamal Hartnett because the query is based on the state of work items as of a past date and time.

```
SELECT
    [System.Id],
    [System.Title],
    [System.State],
    [System.IterationPath]
FROM workitems
WHERE
    [System.TeamProject] = @project
    AND [System.WorkItemType] <> ''
    AND ([System.IterationPath] UNDER 'Fabrikam Fiber\Release 1'
    AND [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>')
    ASOF  '01-05-2022T00:00:00.0000000'
```

> **NOTE**
> If no time is specified, WIQL uses midnight. If no time zone is specified, WIQL uses the time zone of the local client computer.

Set the sort order (`ORDER BY`)

You can use the `ORDER BY` clause to sort the results of a query by one or more fields in ascending or descending order.

> **NOTE**
> The sorting preferences of the SQL server on the data tier determine the default sort order. However, you can use the `asc` or `desc` parameters to choose an explicit sort order.

The following example sorts work items first by **Priority** in ascending order (default), and then by **Created Date** in descending order (`DESC`).

```
SELECT
    [System.Id],
    [System.Title],
    [System.State],
    [System.IterationPath]
FROM workitems
WHERE
    [System.TeamProject] = @project
    AND [System.WorkItemType] <> ''
    AND [System.State] =  'Active'
    AND [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
ORDER BY [Microsoft.VSTS.Common.Priority],
    [System.CreatedDate] DESC
```

Query for links between work items

To return links between work items, you specify `FROM WorkItemLinks`. Filter conditions in the `WHERE` clause may apply to the links or to any work item that is the source or the target of a link. For example, the following query returns the links between Product Backlog Items and their active child items.

```
SELECT
    [System.Id],
    [System.Title],
    [System.State],
    [System.IterationPath]
FROM workitemLinks
WHERE
    (
        [Source].[System.TeamProject] = @project
        AND [Source].[System.WorkItemType] = 'Product Backlog Item'
    )
    AND (
        [System.Links.LinkType] = 'System.LinkTypes.Hierarchy-Forward'
    )
    AND (
        [Target].[System.TeamProject] = @project
        AND [Target].[System.WorkItemType] <> ''
        AND [Target].[System.State] <> 'Closed'
    )
ORDER BY [Microsoft.VSTS.Common.Priority],
    [System.CreatedDate] DESC
MODE (Recursive)
```

The following table summarizes the differences between work item queries and queries for links between work items.

Clause

Work items

Links between work items

`FROM`

`FROM WorkItems`

`FROM WorkItemLinks`

`WHERE`

`[FieldName] = Value`

`Specify one or more of the following:`
`[Source].[FieldName] = Value`
`[Target].[FieldName] = Value`
`[System.Links.LinkType] = 'LinkName'`

`MODE`

not applicable

Specify one of the following:

- `MODE (MustContain)`: (Default) Returns only WorkItemLinkInfo records where the source, target, and link criteria are all satisfied.
- `MODE (MayContain)`: Returns WorkItemLinkInfo records for all work items that satisfy the source and link criteria, even if no linked work item satisfies the target criteria.
- `MODE (DoesNotContain)`: Returns WorkItemLinkInfo records for all work items that satisfy the source, only if no linked work item satisfies the link and target criteria.
- `MODE (Recursive)`: Use for Tree queries(`[System.Links.LinkType] = 'System.LinkTypes.Hierarchy-Forward'`). Link type must be Tree topology and forward direction. Returns WorkItemLinkInfo records for all work items that satisfy the source, recursively for target. `ORDER BY` and `ASOF` aren't compatible with tree queries.

`RETURNS`

`WorkItemQueryResult`

`WorkItemLink`

You can specify one of the following system link type names.

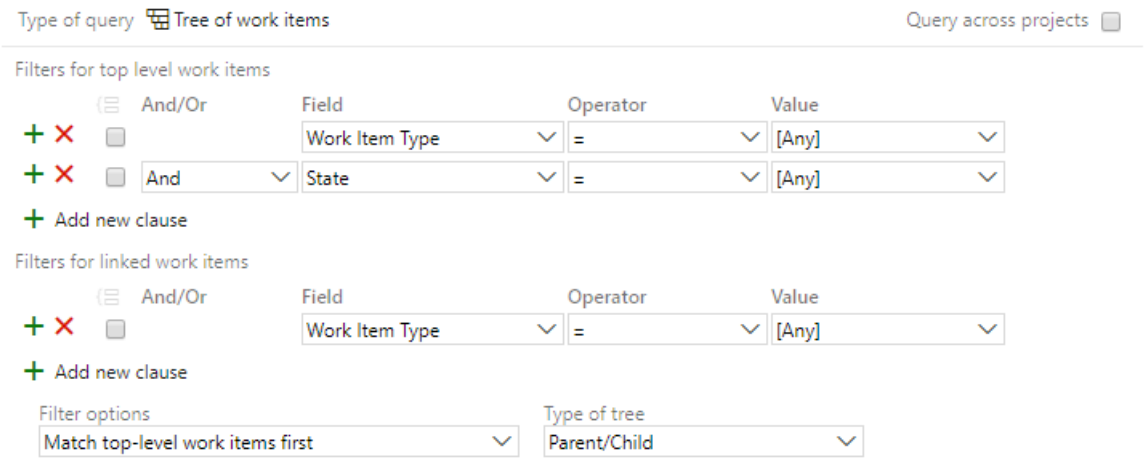
You can specify one of the system link type names, listed below, or a custom link type you've defined with the On-premises XML process.

- `System.LinkTypes.Hierarchy-Forward`
- `System.LinkTypes.Related`
- `System.LinkTypes.Dependency-Predecessor`
- `System.LinkTypes.Dependency-Successor`
- `Microsoft.VSTS.Common.Affects-Forward` (CMMI process)

For more information, see Link type reference.

Tree type query example

The following query returns all work item types define in the current project. The query as shown in the Query Editor appears as shown in the following image.

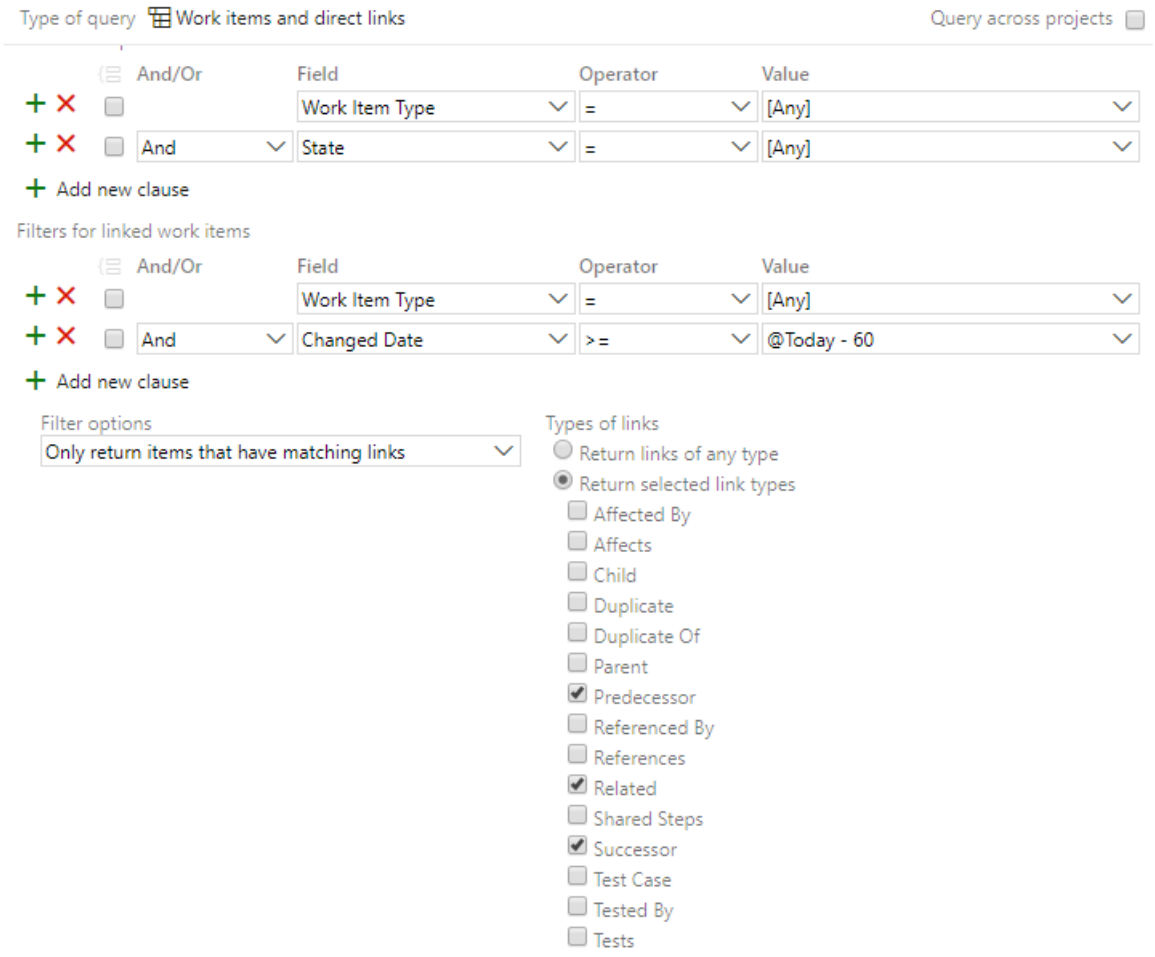


The equivalent WIQL syntax is shown below.

```
SELECT
    [System.Id],
    [System.Title],
    [System.State],
    [System.IterationPath]
FROM workitemLinks
WHERE
    (
        [Source].[System.TeamProject] = @project
        AND [Source].[System.WorkItemType] <> ''
        AND [Source].[System.State] <> ''
    )
    AND (
        [System.Links.LinkType] = 'System.LinkTypes.Hierarchy-Forward'
    )
    AND (
        [Target].[System.TeamProject] = @project
        AND [Target].[System.WorkItemType] <> ''
    )
MODE (Recursive)
```

Direct-link query example

The following query returns all work item types define in the current project. The query as shown in the Query Editor appears as shown in the following image.



The equivalent WIQL syntax is as shown.

```
SELECT
    [System.Id],
    [System.WorkItemType],
    [System.Title],
    [System.AssignedTo],
    [System.State]
FROM workitemLinks
WHERE
    (
        [Source].[System.TeamProject] = @project
        AND [Source].[System.WorkItemType] <> ''
        AND [Source].[System.State] <> ''
    )
    AND (
        [System.Links.LinkType] = 'System.LinkTypes.Dependency-Reverse'
        OR [System.Links.LinkType] = 'System.LinkTypes.Related-Forward'
        OR [System.Links.LinkType] = 'System.LinkTypes.Dependency-Forward'
    )
    AND (
        [Target].[System.TeamProject] = @project
        AND [Target].[System.WorkItemType] <> ''
        AND [Target].[System.ChangedDate] >= @today - 60
    )
ORDER BY [System.Id]
MODE (MustContain)
```

More query examples

The following typical WIQL query example uses reference names for the fields. The query selects work items (no work item type specified) with a **Priority=1**. The query returns the **ID** and **Title** of the return set as columns.

The results are sorted by **ID** in ascending order.

```
SELECT
    [System.Id],
    [System.Title],
    [System.State],
    [System.IterationPath]
FROM workitems
WHERE
    [System.TeamProject] = @project
    AND [Microsoft.VSTS.Common.Priority] <> ''
ORDER BY [System.Id]
```

Date-time pattern

You specify the date-time pattern according to one of two patterns:

- The Date Pattern and Time Pattern format comes from your user preferences, Time and Locale
- The pattern specified by UTC, which follows this pattern (with Z appended to the date-time).

`AND [System.ChangedDate] >= '1/1/2019 00:00:00Z'`

Example clauses

The following example statements show specific qualifying clauses.

Clause

Example

`AND`

```
SELECT [System.Id], [System.Title]
FROM WorkItems
WHERE [System.TeamProject] = @project
AND [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
```

`OR`

```
SELECT [System.Id], [System.Title]
FROM WorkItems
WHERE [System.TeamProject] = @project
AND ( [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
OR [System.AssignedTo] = ''Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>' )
```

`NOT`

```
SELECT [System.Id], [System.Title]
FROM WorkItems
WHERE [System.TeamProject] = @project
AND [System.AssignedTo] EVER 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
AND [System.AssignedTo] NOT CONTAINS 'Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>'
```

`EVER`

```
SELECT [System.Id], [System.Title]
FROM WorkItems
WHERE [System.TeamProject] = @project
AND [System.AssignedTo] EVER 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
```

`UNDER`

```
SELECT [System.Id], [System.Title]
FROM WorkItems
WHERE [System.TeamProject] = @project
AND [System.AssignedTo] EVER 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
AND [System.AreaPath] UNDER 'Agile1\Area 0'
```

`ORDER BY`

```
SELECT [System.Id], [System.Title]
FROM WorkItems
WHERE [System.TeamProject] = @project
AND [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
ORDER BY [System.Id] [asc | desc]
```

`ASOF` (Time filter)

```
SELECT [System.Title]
FROM workitems
WHERE [System.IterationPath] = 'MyProject\Beta'
AND [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
ASOF '3/16/19 12:30'
```

String and PlainText

Quote string literals (single or double quotes are supported) in a comparison with a string or plain text field. String literals support all Unicode characters.

```
WHERE [Custom.Blocking] = 'Not Blocking'
WHERE [Custom.Blocking] <> 'Blocked'
```

You can use the contains operator to search for a substring anywhere in the field value.

```
WHERE [System.Description] contains 'WIQL'
```

Area and Iteration (TreePath)

You can use the `UNDER` operator for the Area and Iteration Path fields. The `UNDER` operator evaluates whether a value is within the subtree of a specific classification node. For instance, the expression below would evaluate to true if the Area Path were 'MyProject\Server\Administration', 'MyProject\Server\Administration\Feature 1', 'MyProject\Server\Administration\Feature 2\SubFeature 5', or any other node within the subtree.

```
WHERE [System.AreaPath] UNDER 'MyProject\Server\Administration'
```

Modifiers and special operators

You can use some modifiers and special operators in a query expression.

Use the `IN` operator to evaluate whether a field value is equal to any of a set of values. This operator is supported for the String, Integer, Double, and DateTime field types. See the following example along with its semantic equivalent.

```
WHERE
    [System.TeamProject] = @project
    AND [System.CreatedBy] IN ('Jamal Hartnett <fabrikamfiber4@hotmail.com>', 'Raisa Pokrovskaya
<fabrikamfiber5@hotmail.com>', 'Christie Church <fabrikamfiber1@hotmail.com>')

or

WHERE
    [System.TeamProject] = @project
    AND (
        [System.CreatedBy] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
        OR [System.CreatedBy] = 'Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>'
        OR [System.CreatedBy] = 'Christie Church <fabrikamfiber1@hotmail.com>'
    )
```

The `EVER` operator is used to evaluate whether a field value equals or has ever equaled a particular value throughout all past revisions of work items. The String, Integer, Double, and DateTime field types support this operator. There are alternate syntaxes for the `EVER` operator. For example, the snippets below query whether all work items were ever assigned to Jamal, Raise, or Christie.

```
WHERE
    [System.TeamProject] = @project
    AND (
        EVER [System.AssignedTo] = 'Jamal Hartnett <fabrikamfiber4@hotmail.com>'
        OR EVER [System.AssignedTo] = 'Raisa Pokrovskaya <fabrikamfiber5@hotmail.com>'
        OR EVER [System.AssignedTo] = 'Christie Church <fabrikamfiber1@hotmail.com>'
    )
```

Related articles

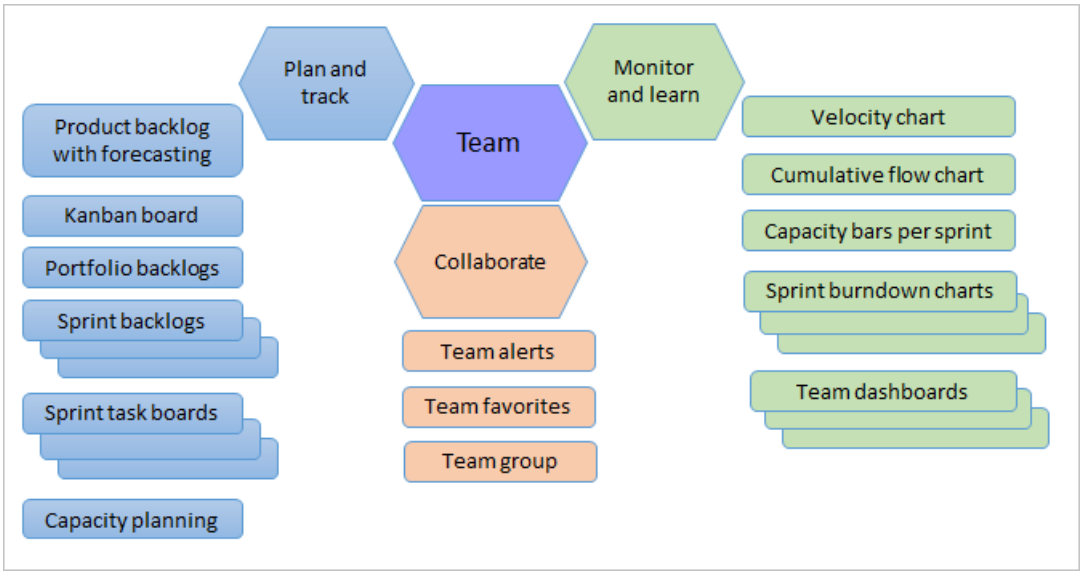
- Query fields, operators, values, and variables
- Work item fields and attributes
- About managed queries
- Cross-service and enhanced query operations
- Define a query

Plan for Agile at scale

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As your organization grows, you want your tools to scale to support your growing business needs. Azure Boards tools scale primarily by supporting the addition of teams. Each team provides configurable tools that allow teams to focus on their set of work.



For guidance on adding teams, see About teams and Agile tools.

Video: Agile at Scale

Portfolio management

Enterprise project managers often have a portfolio of projects that they manage. These projects are typically developed by several teams. By creating a hierarchy of teams, portfolio managers can gain insight into the tools being developed by their teams.

To learn more, see Manage portfolios.

Delivery plans

Delivery plans provide visibility into features under development by several teams across several sprints. With Delivery Plans, portfolio managers can review the schedule of stories or features their teams plan to deliver. Delivery Plans show the scheduled work items by sprint (iteration path) of selected teams against a calendar view.

To learn more, see Review delivery plans.

Team dashboards

Each team can construct several dashboards to track and monitor their progress. Also, portfolio managers can create dashboards to monitor progress across several teams.

To learn more, see Add and manage dashboards.

Related articles

- Visibility across teams
- Agile culture and scale
- Practices that scale
- Agile culture
- Scale Agile to large teams
- Creating productive teams

Use Azure Boards to manage your product and portfolio backlogs

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

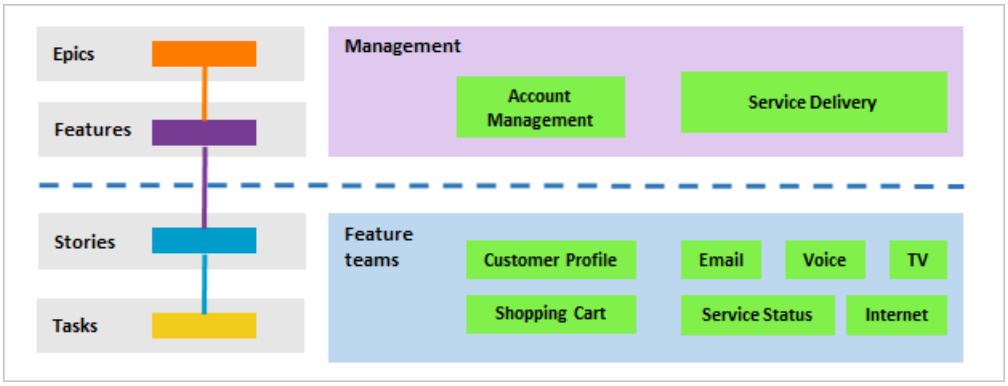
Portfolio backlogs provide product owners insight into the work done by several agile feature teams. Product owners can define the high-level goals as Epics or Features. Feature teams can break down these work items into the user stories they'll prioritize and develop.

In this article you'll learn:

- How to support a management view of multiple team progress
- How feature teams can focus on their team backlog progress
- How to assign work from a common backlog
- How to set up a hierarchical set of teams and backlogs

> **NOTE**
> For guidance on configuring and customizing your project and teams to support your business needs, review Configuration and customization of Azure Boards.

By setting up a team structure like the one shown, you provide each feature team with their distinct backlog to plan, prioritize, and track their work. And, portfolio or product owners can create their vision, roadmap, and goals for each release, monitor progress across their portfolio of projects, and manage risks and dependencies.



Set up a hierarchical team and backlog structure when you want to support the following elements:

- Autonomous feature teams that can organize and manage their backlog of work
- Portfolio management views for planning epics and features and monitoring progress of subordinate feature teams
- Assign backlog items to feature teams from a common backlog

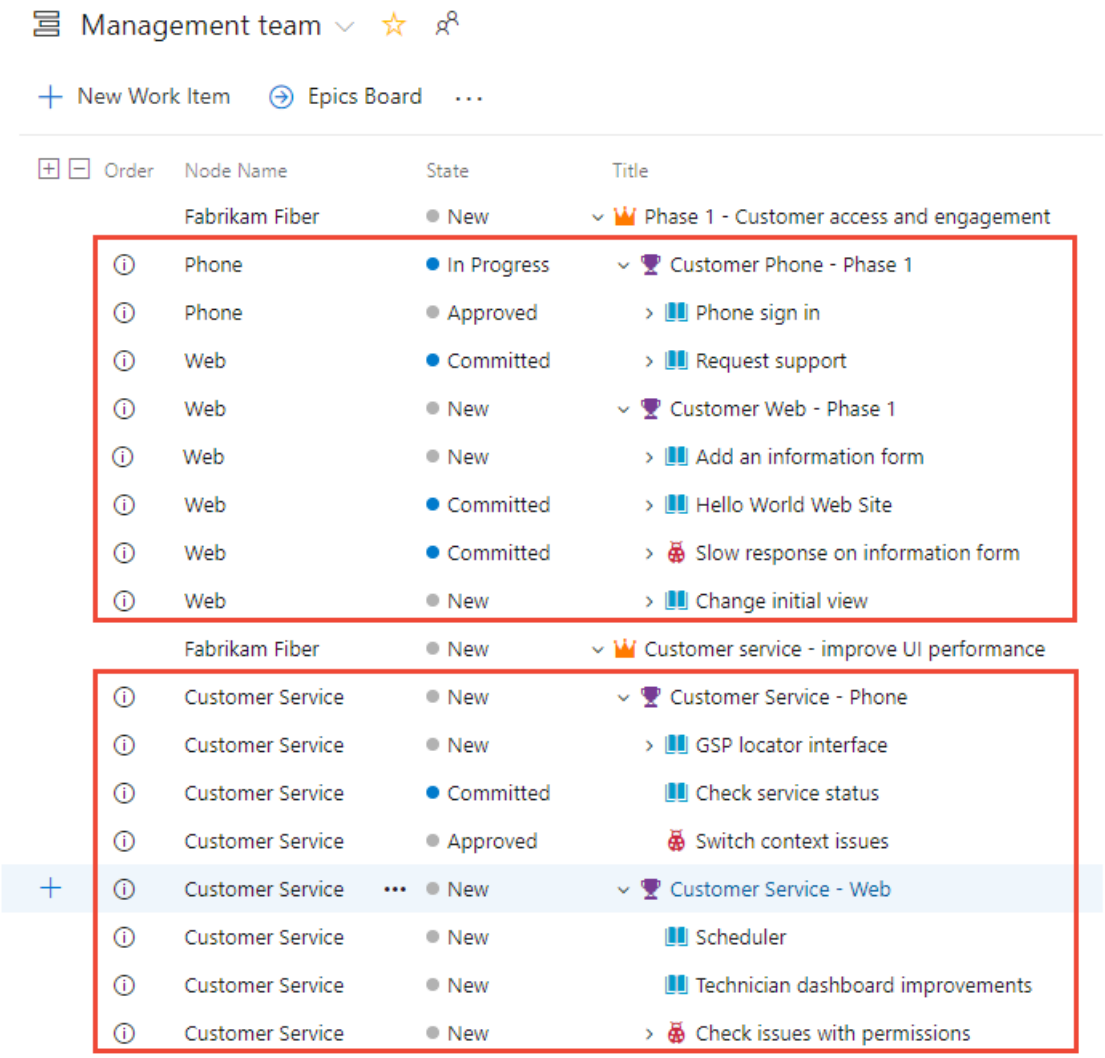
> **NOTE**
> The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to Azure DevOps Services. However, the basic functionality available to you remains the same unless explicitly mentioned.

NOTE

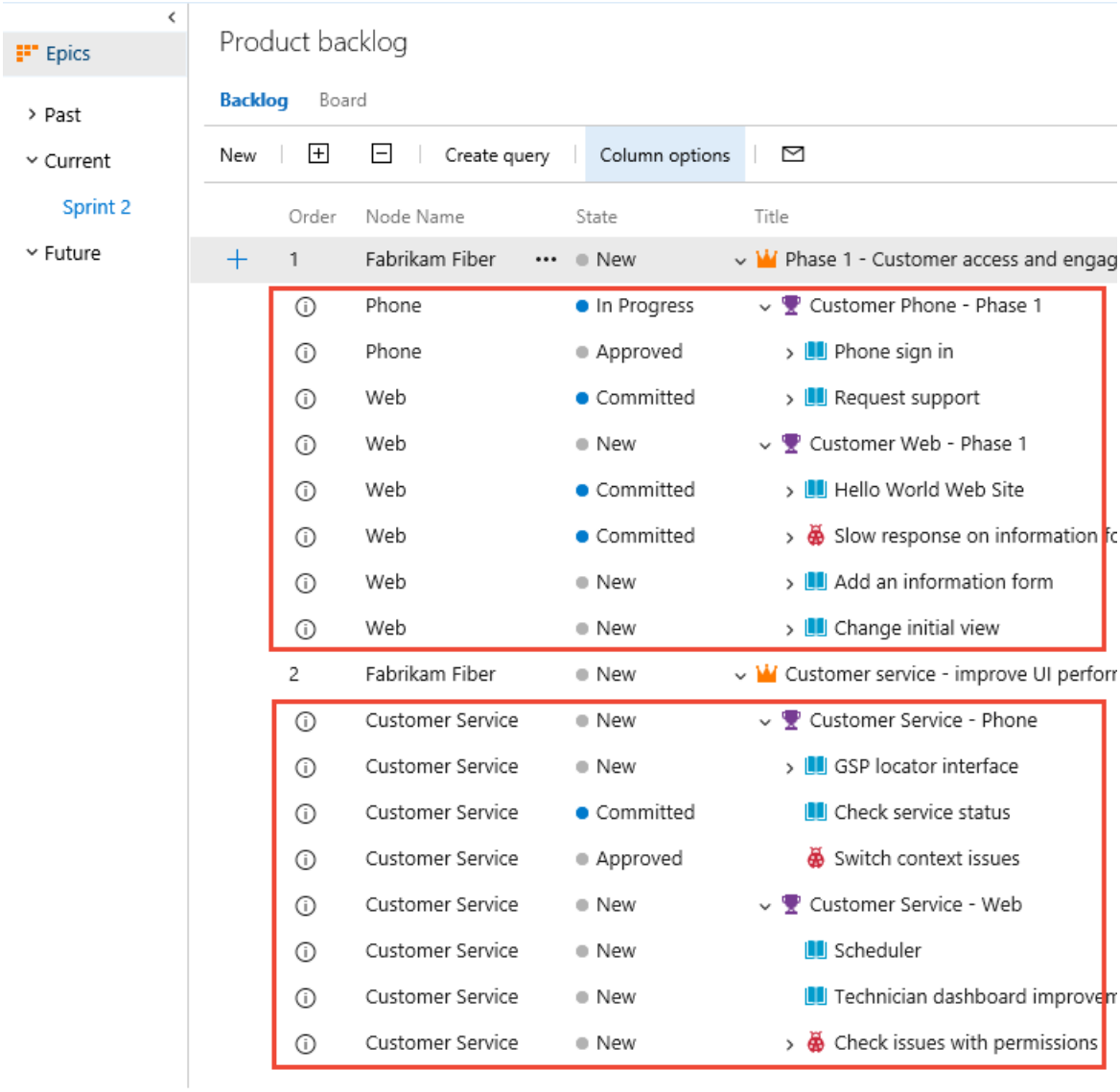
The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to your on-premises Azure DevOps. However, the basic functionality available to you remains the same unless explicitly mentioned.

Management view of team progress

In this example, we show the **Epics** portfolio backlog for the **Management** team. Drilling down, you can see all the backlog items and features, even though they belong to one of three different teams: Customer Service, Phone, and Web.



In this example, we show the **Epics** portfolio backlog for the **Management** team. Drilling down, you can see all the backlog items and features, even though they belong to one of three different teams: Customer Service, Phone, and Web.



TIP

Program managers can also gain insight into progress across teams using Delivery plans. See also Visibility across teams.

Feature team backlog ownership and view of progress

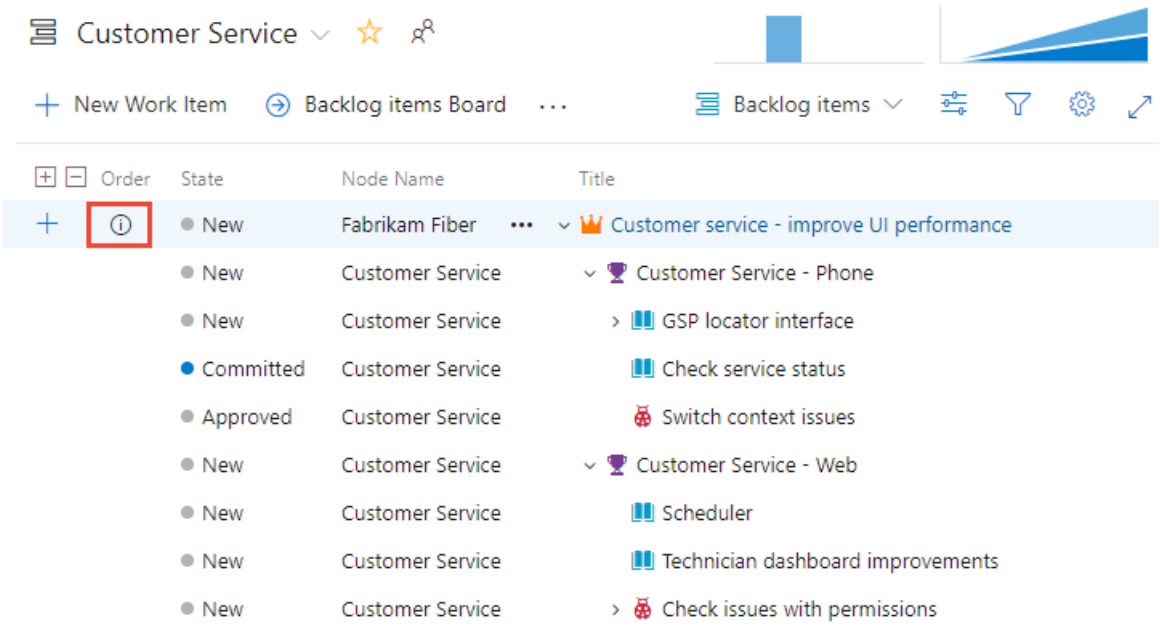
Each feature team has its own team home page or dashboards, product and portfolio backlogs, Kanban boards, and taskboards. These pages only show work relevant to each team. The relevance is based on assignments made to the work item area and iteration paths. For details, see About teams and Agile tools.

TIP

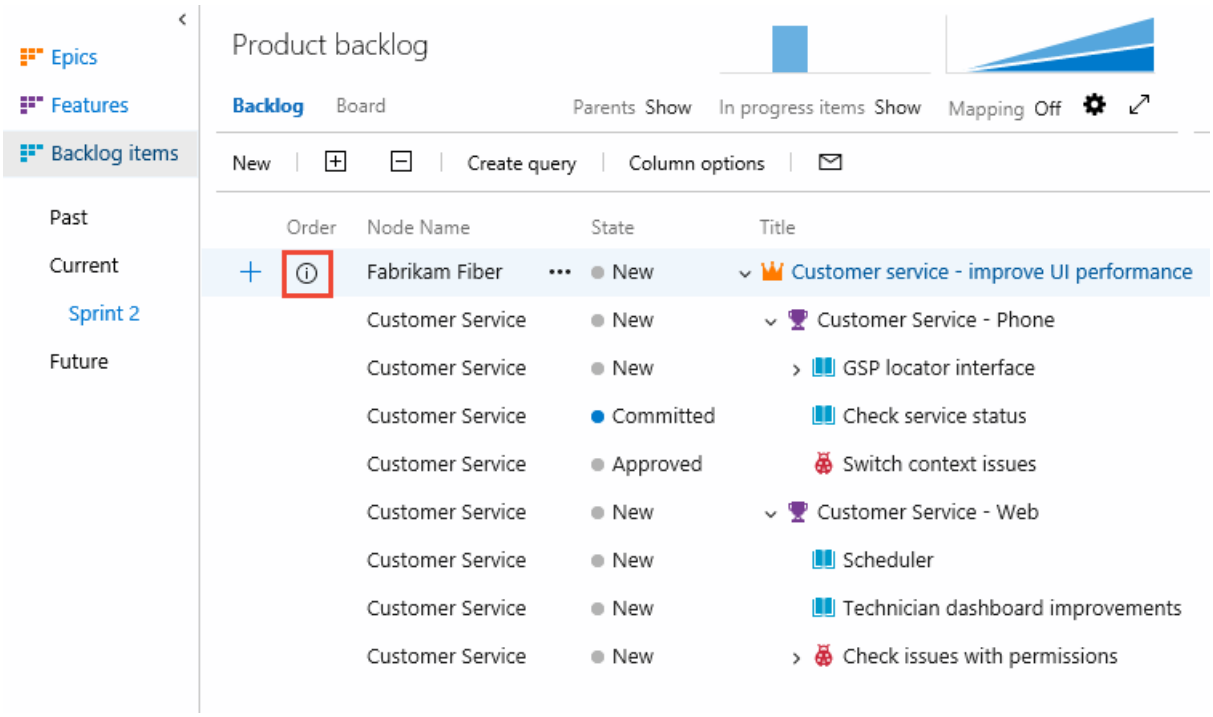
Add **Node Name** to the column options to show the team assigned to the work item.

The Customer Service feature team's view of the backlog only includes those work items assigned to their area path, **Fabrikam Fiber/Customer Service**. Here we show parents that provide a few of the features and epics to which the backlog items belong. Items that are owned by other teams appear with hollow-filled bars. For example, Mobile feedback and Text alerts belong to the Account Management team.

Items that are owned by other teams appear with an information icon, ⓘ .



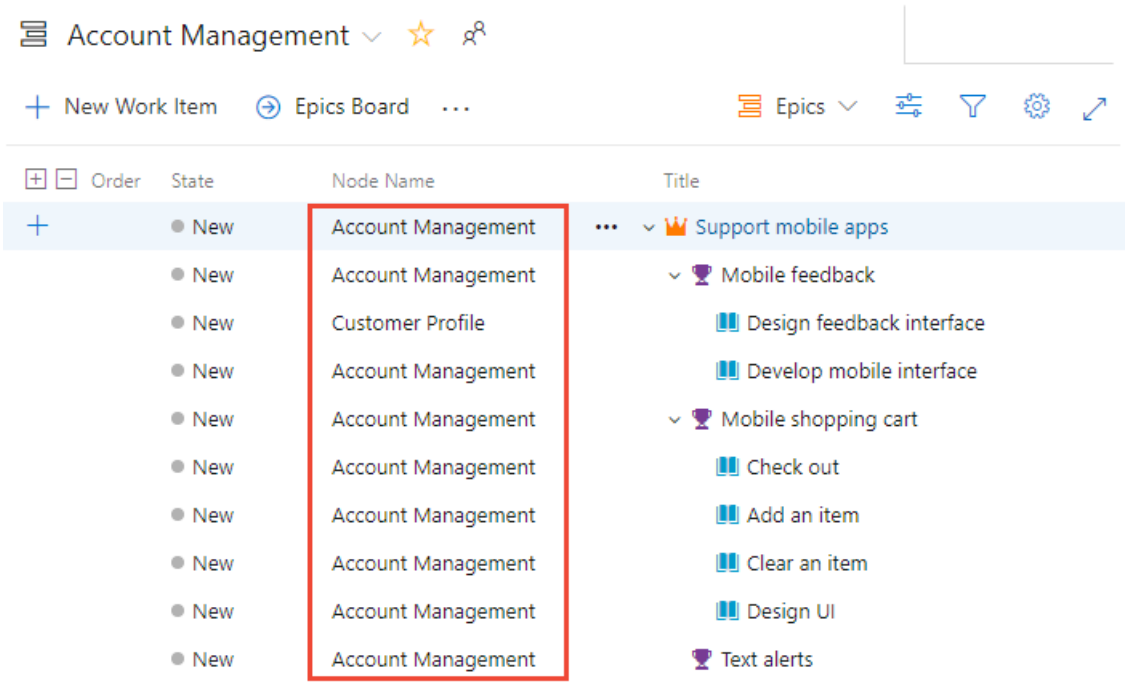
Items that are owned by other teams appear with an information icon, ⓘ .



Assign work from a common backlog

While the hierarchical team and backlog structure works well to support autonomous teams to take ownership of their backlog, it also supports assigning work to teams from a common backlog. During a sprint or product planning meeting, product owners and development leads can review the backlog. Teams can also assign select items to various teams by assigning them to the feature team Area Path.

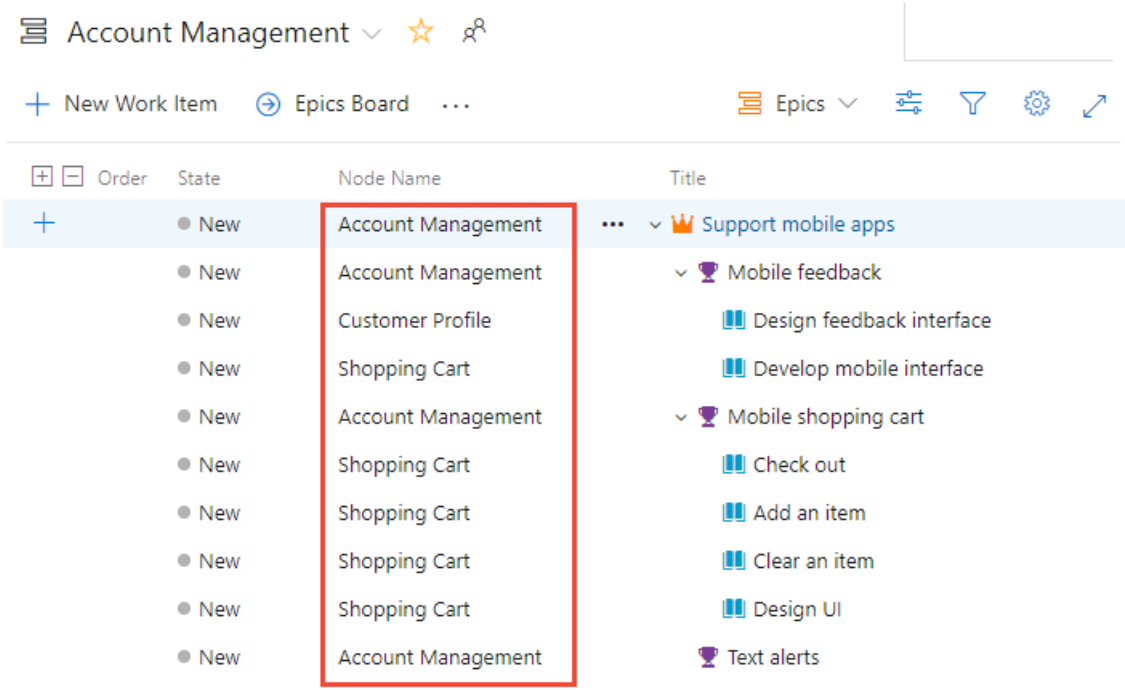
In this view of the Account Management backlog, all items still assigned to **Account Management** have yet to be assigned.



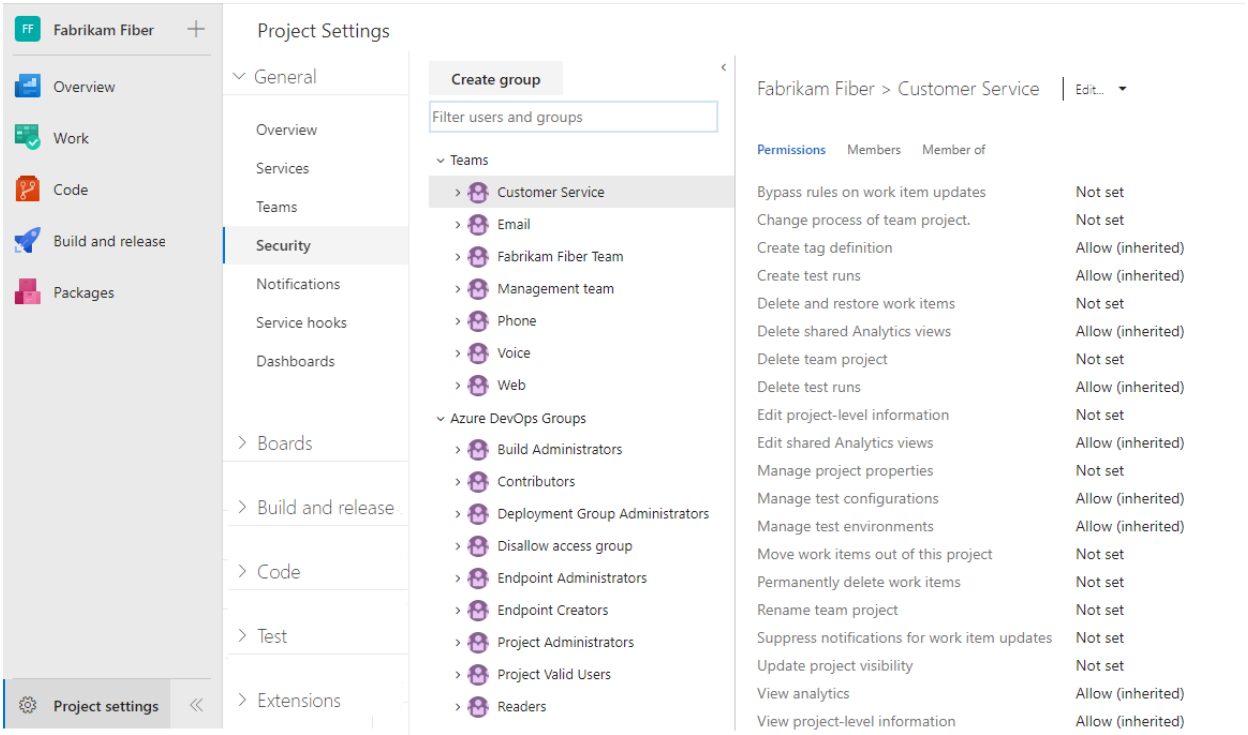
During the planning meeting, you can open each item, make notes, and assign the item to the team to work on it.

> **TIP**
> You can multi-select work items and perform a bulk edit of the area path. See Bulk modify work items.

Here, all backlog items have been assigned to feature teams while all features and epics remain owned by Account Management.



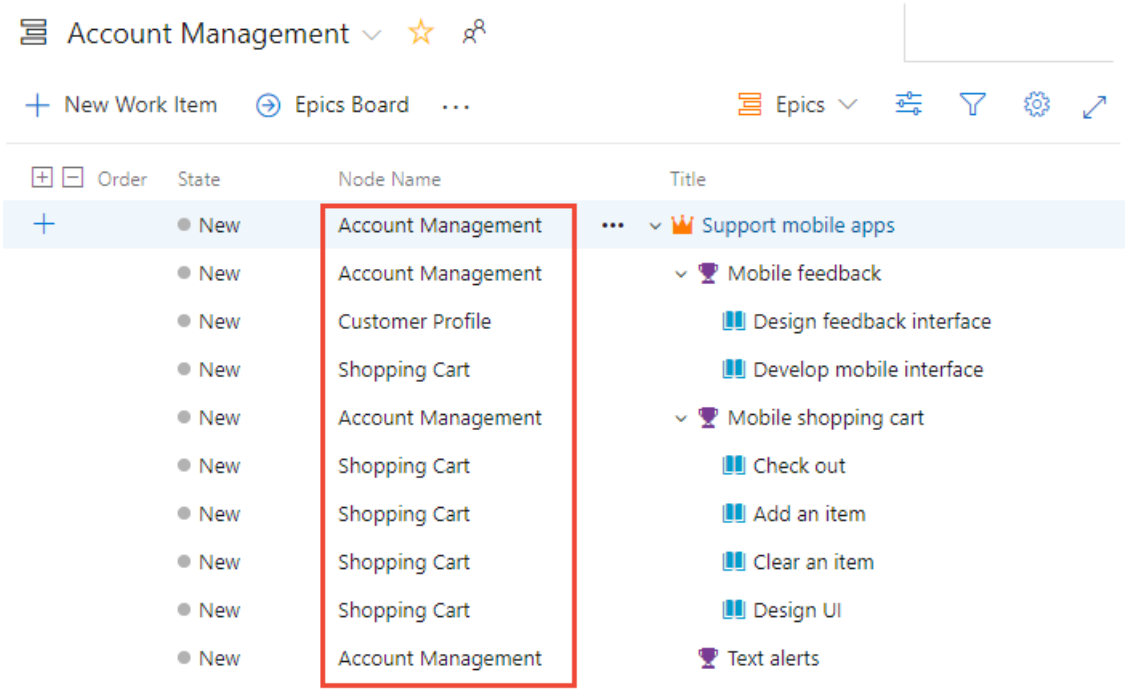
In this view of the Account Management backlog, all items still assigned to **Account Management** have yet to be assigned.



During the planning meeting, you can open each item, make notes, and assign the item to the team to work on it.

> **TIP**
> You can multi-select work items and perform a bulk edit of the area path. See Bulk modify work items.

Here, all backlog items have been assigned to feature teams while all features and epics remain owned by Account Management.



Add portfolio backlogs

If you need more than three backlog levels, you can add more. To learn how, see Customize your backlogs or

boards for a process.

If you need more than three backlog levels, you can add more. To learn how, see Add portfolio backlogs.

Track dependencies across teams

The simplest way to track dependencies across teams is to link work items using the **Related** link type. If they're dependent in time, then you can use the **Predecessor/Successor** link types. You can then create queries that find work items containing these relationships. See Manage dependencies, link work items to support traceability to learn more.

Using Delivery Plans, you can track dependencies across projects within an organization. To learn more, see Track dependencies using Delivery Plans.

Portfolio feature progress

To view feature progress based on linked requirements, you can add a rollup column or view the Feature Timeline. To learn more, see Display rollup and View portfolio progress with the Feature Timeline.

To view feature progress based on linked requirements, you can view the Feature Timeline. To learn more, see View portfolio progress with the Feature Timeline.

Try this next

Configure a hierarchy of teams

Related articles

- Create your backlog
- Kanban quickstart
- Assign work to sprints
- Organize your backlog
- Limitations of multi-team Kanban board views

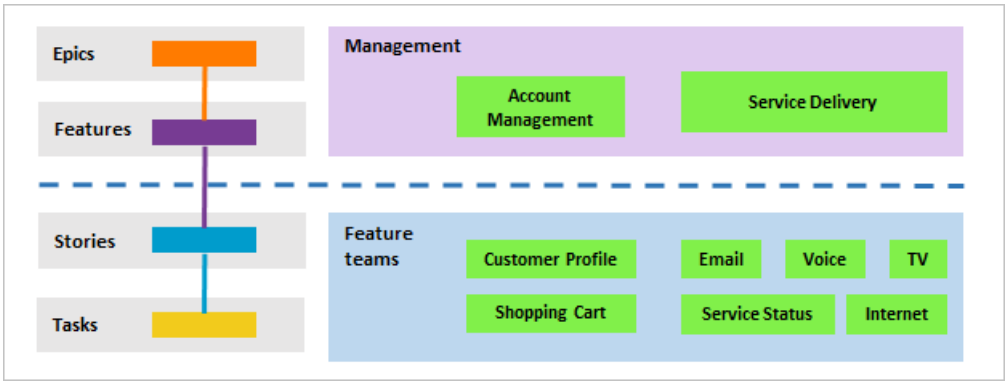
Configure a hierarchy of teams

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

In Portfolio management, we showed how management teams and feature teams can use their backlogs to focus on the work that's most important to them. In this article, we show how to configure teams that best support the different backlog views of management and feature teams.

Specifically, we'll show you how to configure a team structure like the one shown in the image below.



In this article you'll learn how to:

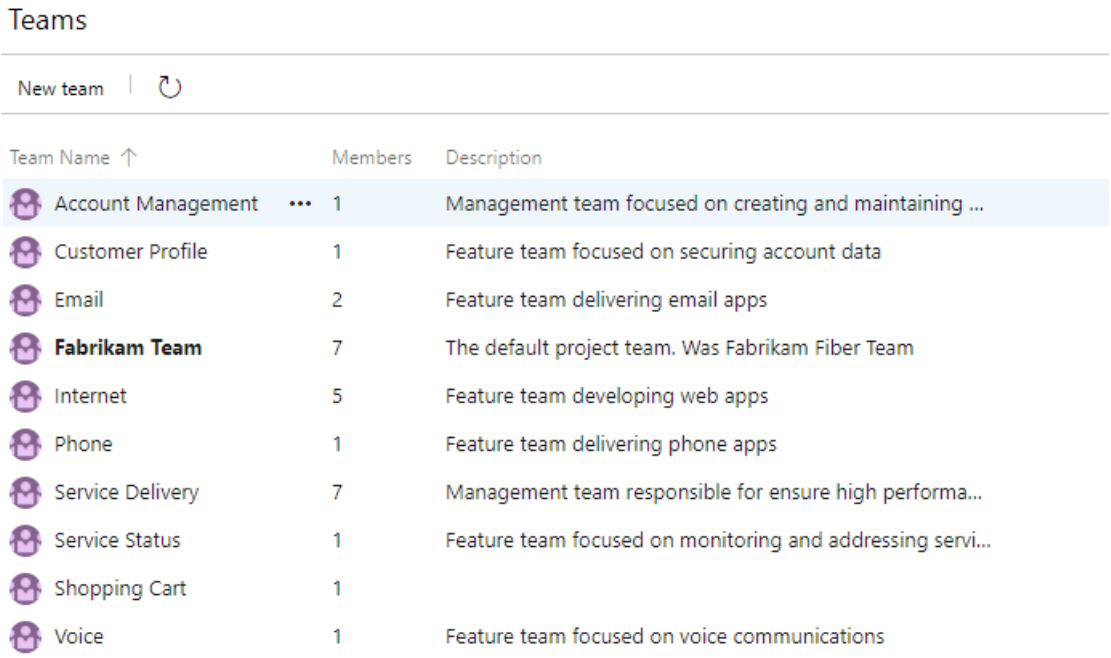
- Set up a hierarchical set of teams and backlogs
- Define a single sprint cadence for all teams
- Review which area paths are assigned to teams

Prerequisites

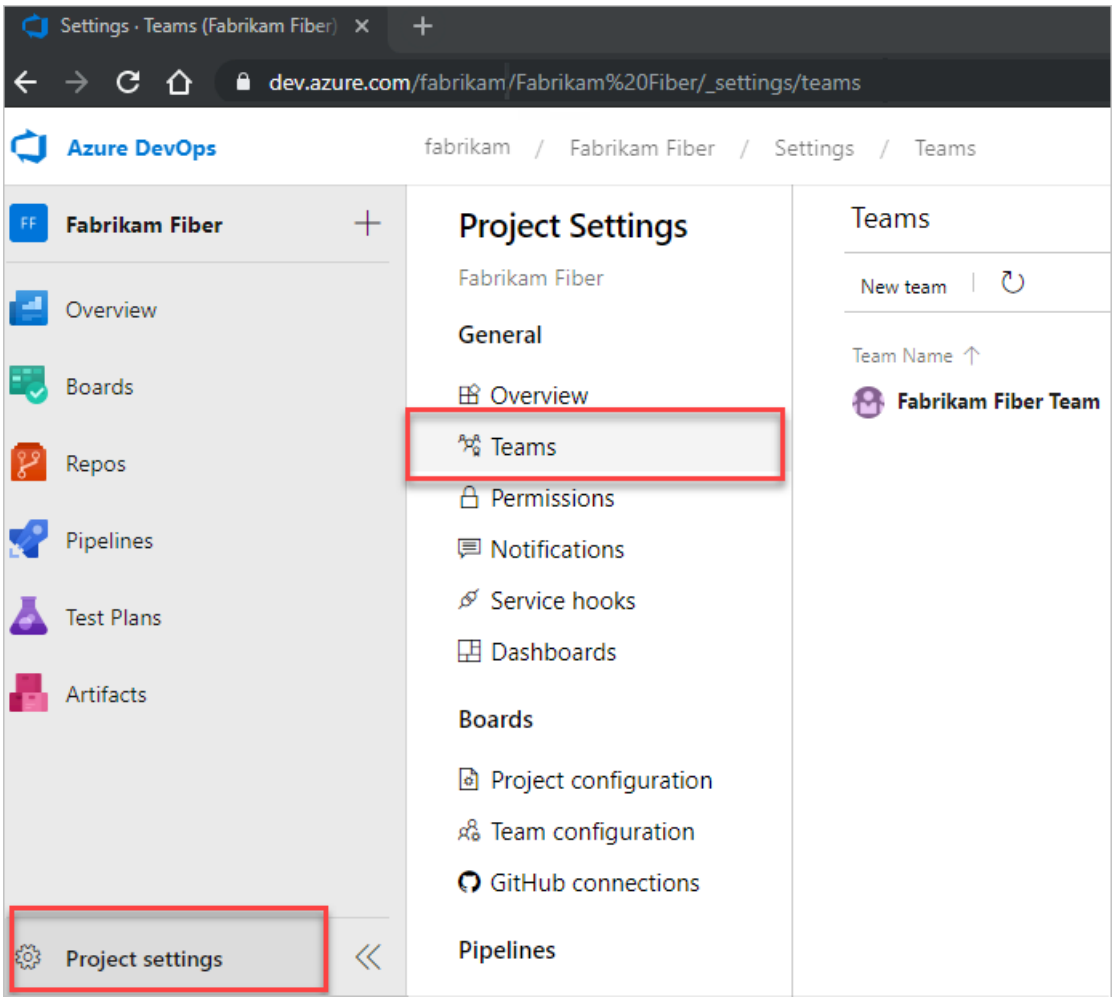
- If you don't have a project yet, create one.
- To add teams, you must be a member of the **Project Administrators** group. To get added to this group, see Change project-level permissions.

Add a team for each management area

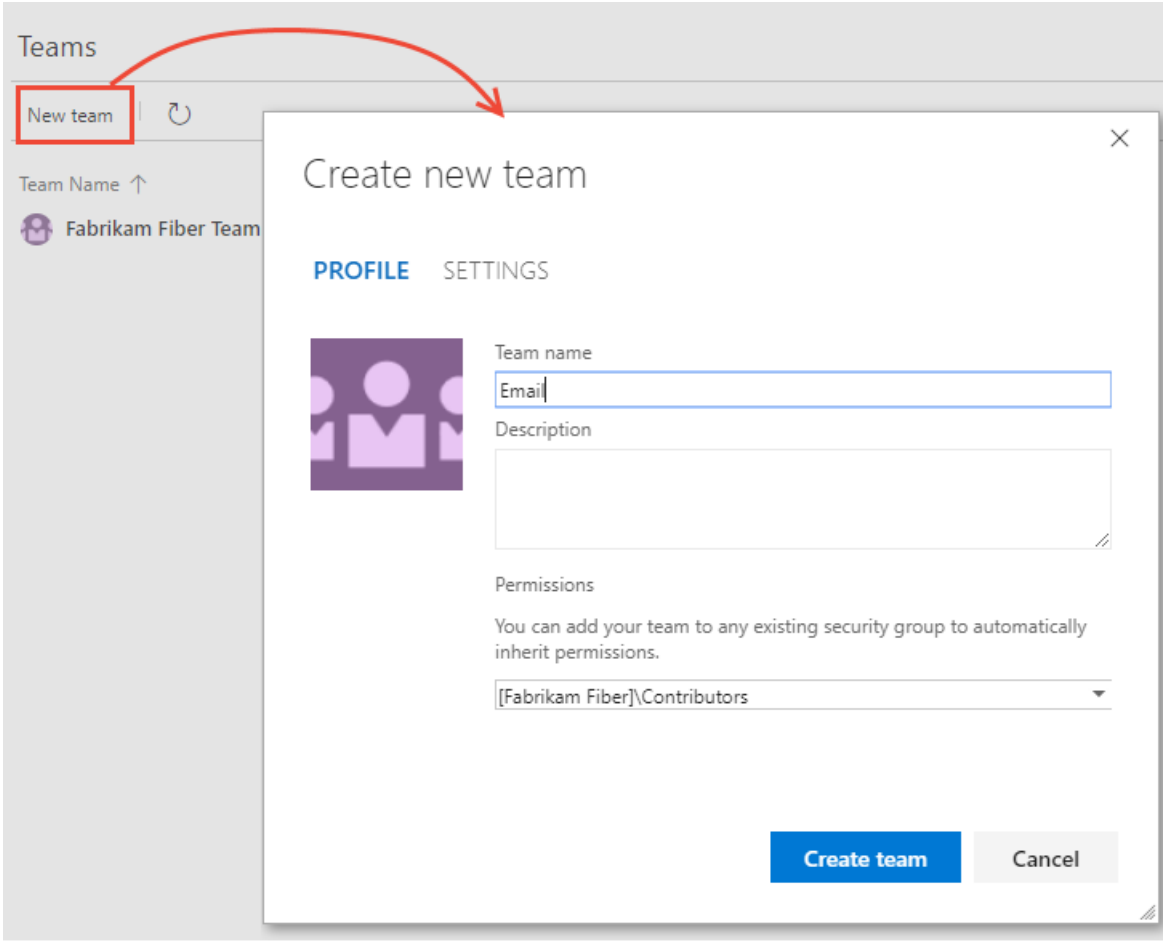
The first step is to add a team for each feature team and management area. You can also rename teams that you've already added. When you finish, you'll have a set of teams similar to the ones shown.



1. From the web portal, choose **Project settings** and open **Teams**.

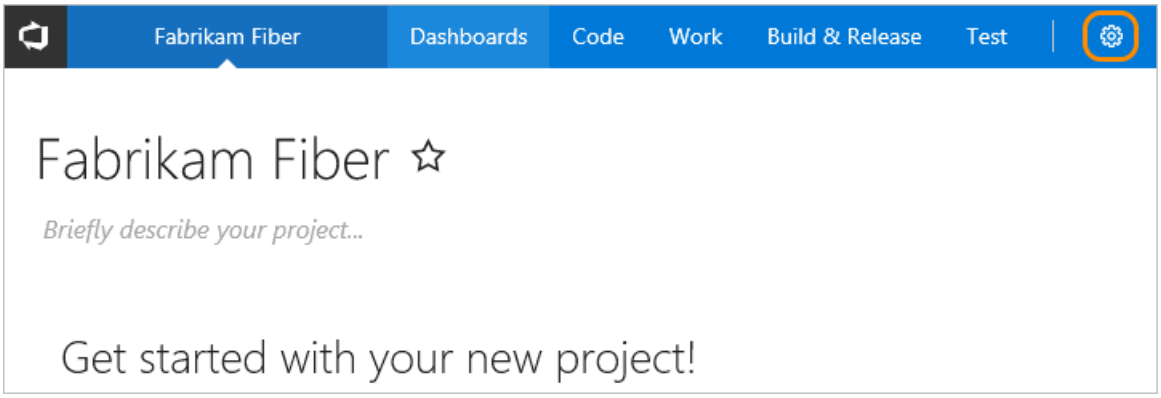


2. Choose **New team**. Give the team a name, and optionally a description.

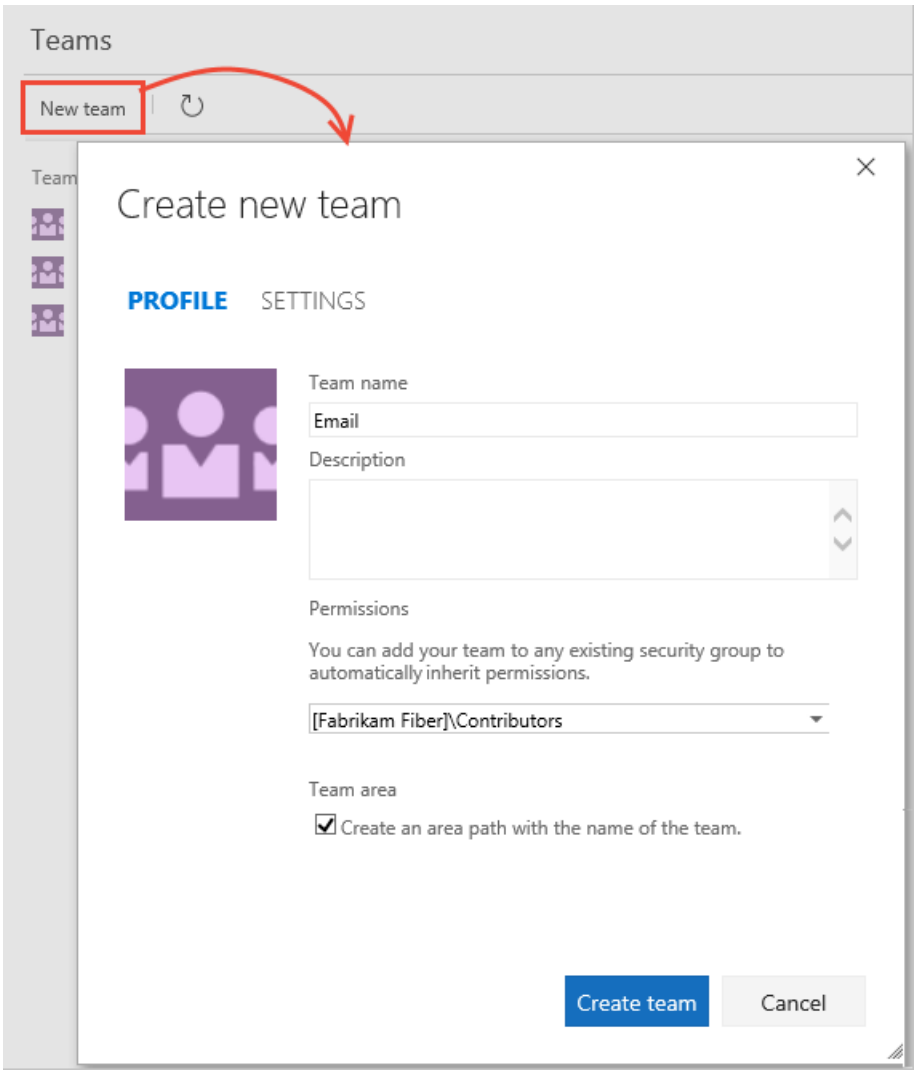


Repeat this step for all feature and management teams you want to create.

1. From the web portal, choose the ⚙ gear settings icon to open the **Project settings** page for the project.



2. Choose **New team**. Give the team a name, and make sure to select **Create an area path with the name of the team**. If you don't select this option, you'll have to set the default area path for the team once you create it. You can choose an existing area path or create a new one at that time. Team tools aren't available until the team's default area path is set.



Repeat this step for all feature and management teams you want to create.

Move area paths into a hierarchical structure

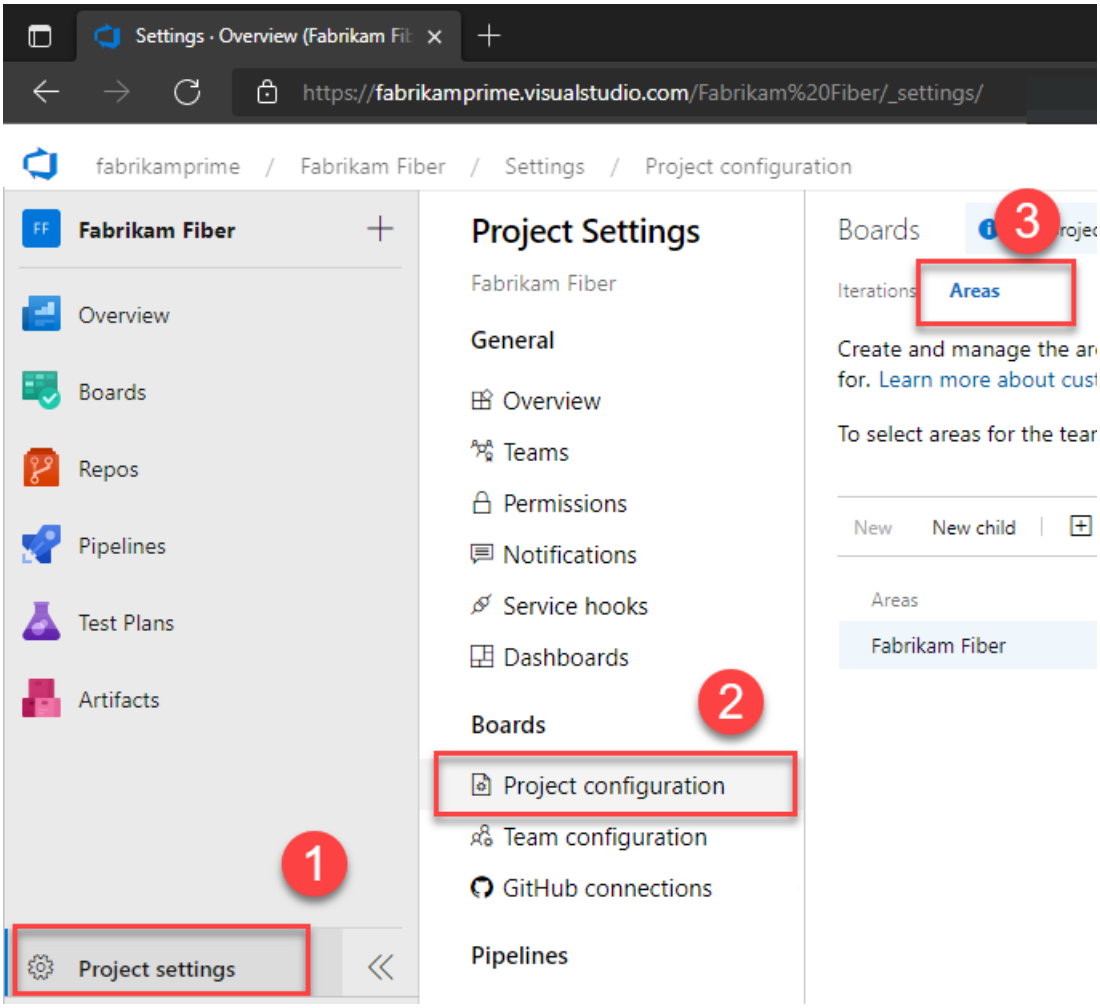
In this step, you want to move the areas paths associated with feature teams from a flat structure to a hierarchical structure.

FLAT AREA STRUCTURE	HIERARCHICAL AREA STRUCTURE
Areas	Areas
Fabrikam Fiber	Fabrikam Fiber
Account Management	Account Management
Customer Profile	Customer Profile
Customer Service	Shopping Cart
Email	Service Delivery
Service Status	Email
Shopping Cart	Internet
TV	Service Status
Voice	TV
Web	Voice

You do this by opening each area path associated with a feature team and changing its location to be under the

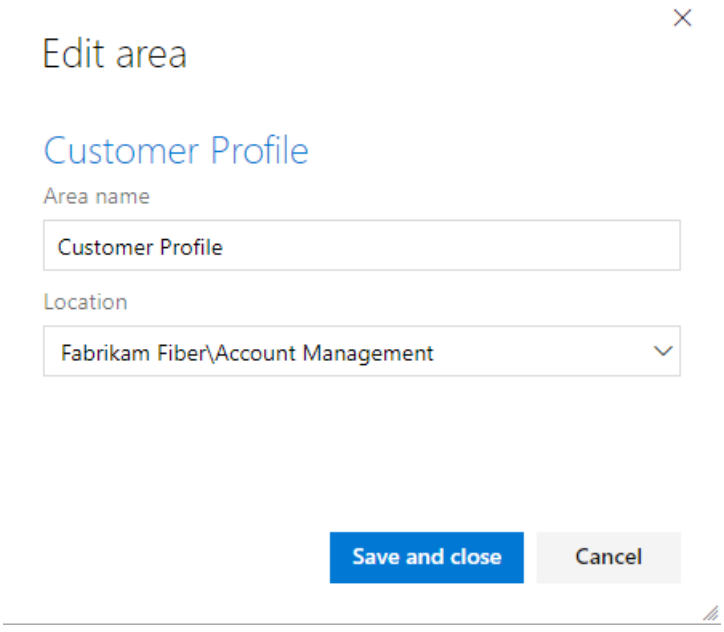
management area path.

1. Choose (1) **Project Settings**, expand **Work** if needed, and choose (2) **Project configuration** and then (3) **Areas**.



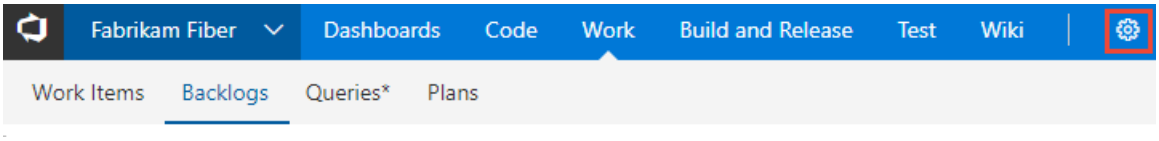
2. Next, choose the ··· actions icon for one of the area paths associated with a feature team and select **Edit**. Then change the **Location** to move it under its corresponding management team area path.

   For example, here we move the Customer Profile to under Account Management.

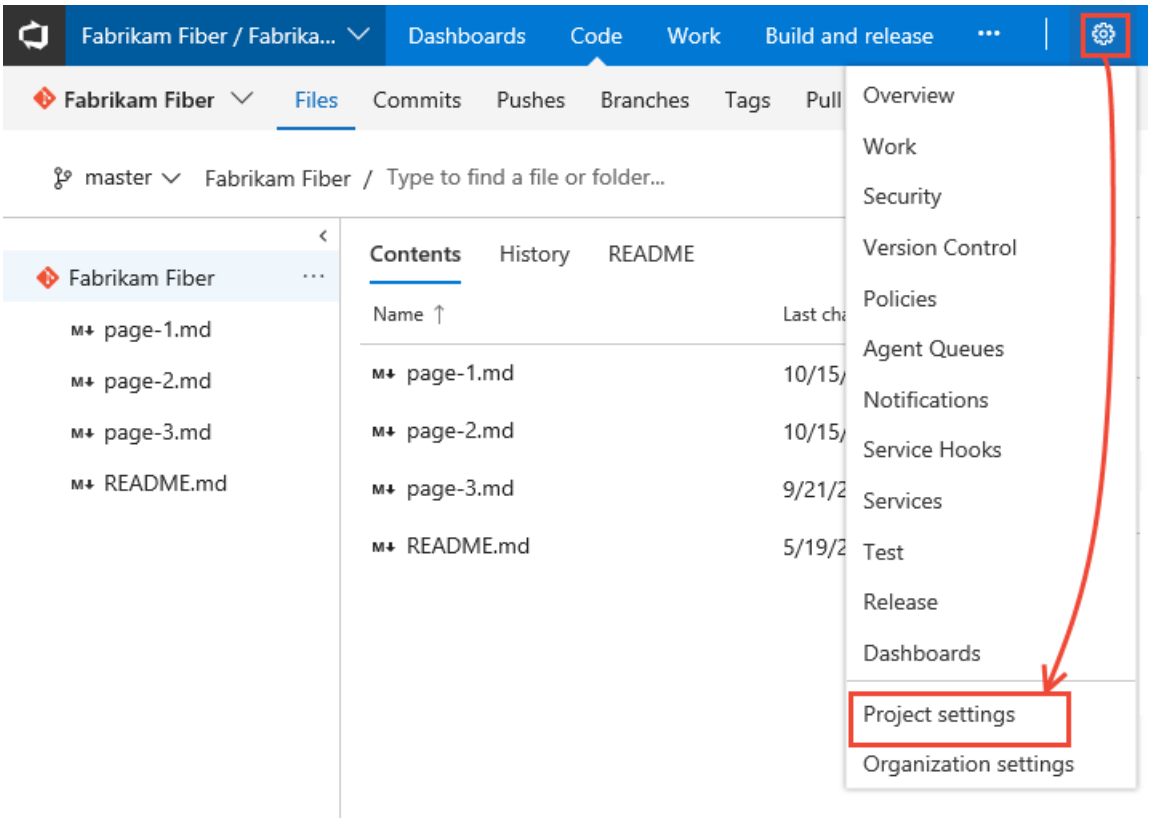


   Repeat this step for all feature team area paths.

1. From the web portal for the project, choose the gear icon.



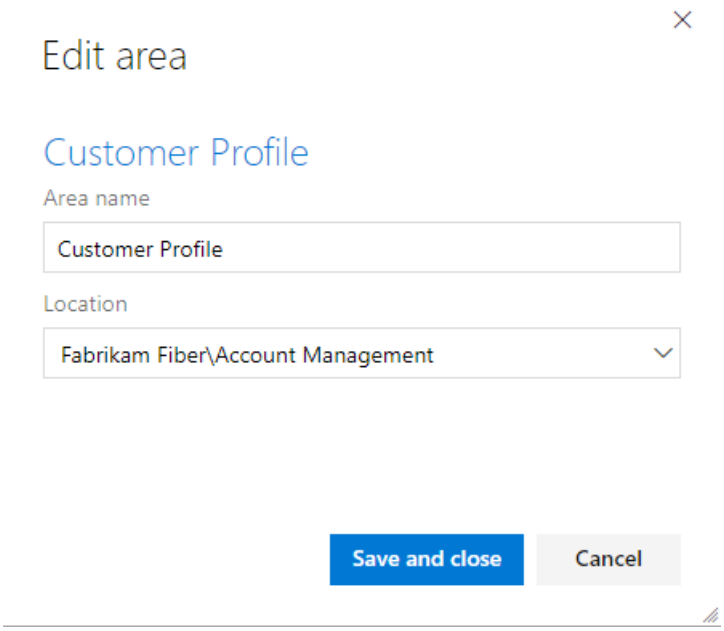
If you're currently working from a team context, then hover over the and choose **Project settings**.



2. Choose **Work**.

3. Next, choose the actions icon for one of the area paths associated with a feature team and select **Edit**. Then change the **Location** to move it under its corresponding management team area path.

   For example, here we move the Customer Profile to under Account Management.



   Repeat this step for all feature team area paths.

Include subarea paths for management teams

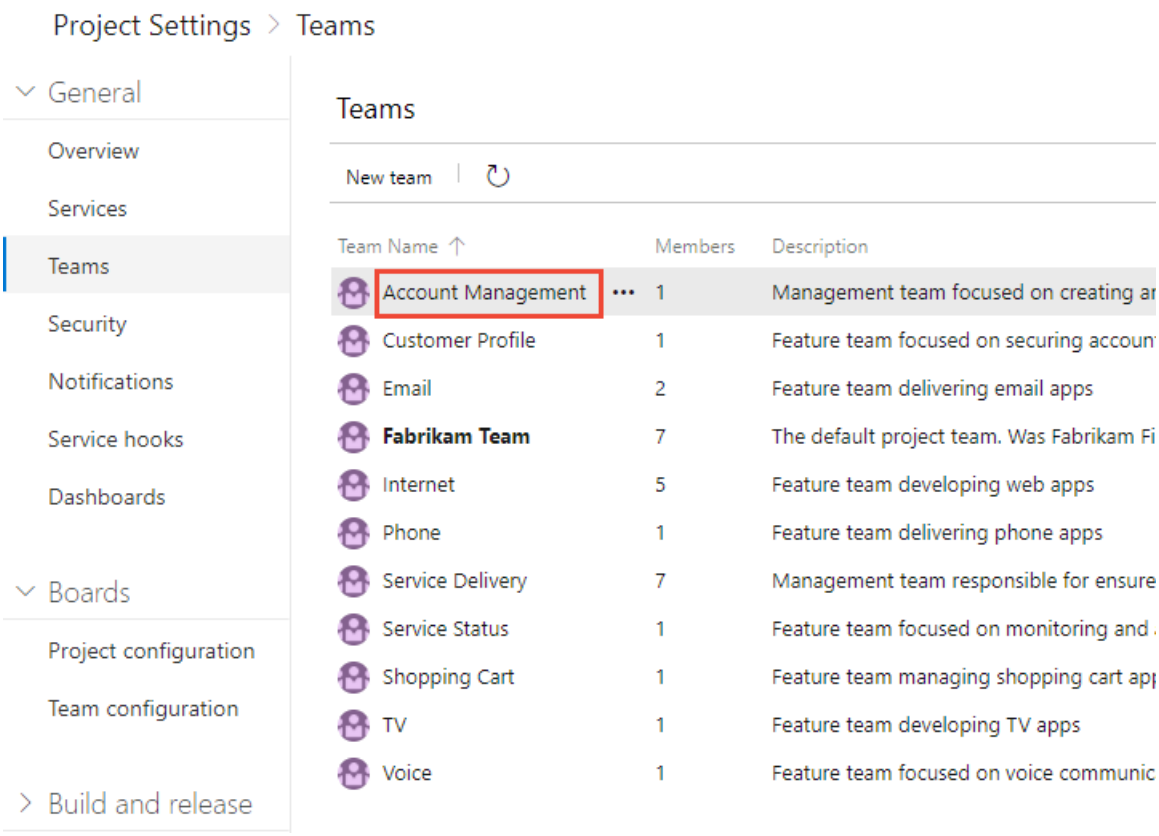
By including subarea paths for the management teams, you automatically include the backlog items of their feature teams onto the management team's backlog. The default setting for all teams is to exclude subarea paths.

> **NOTE**
> Sub-area paths may break a team's ability to reorder or reparent items on the backlog. Also, it can introduce uncertainties with regards to assignments made to the Kanban Board Column, Done, and Lane fields. To learn more, see Exercising select features with shared area paths later in this article.

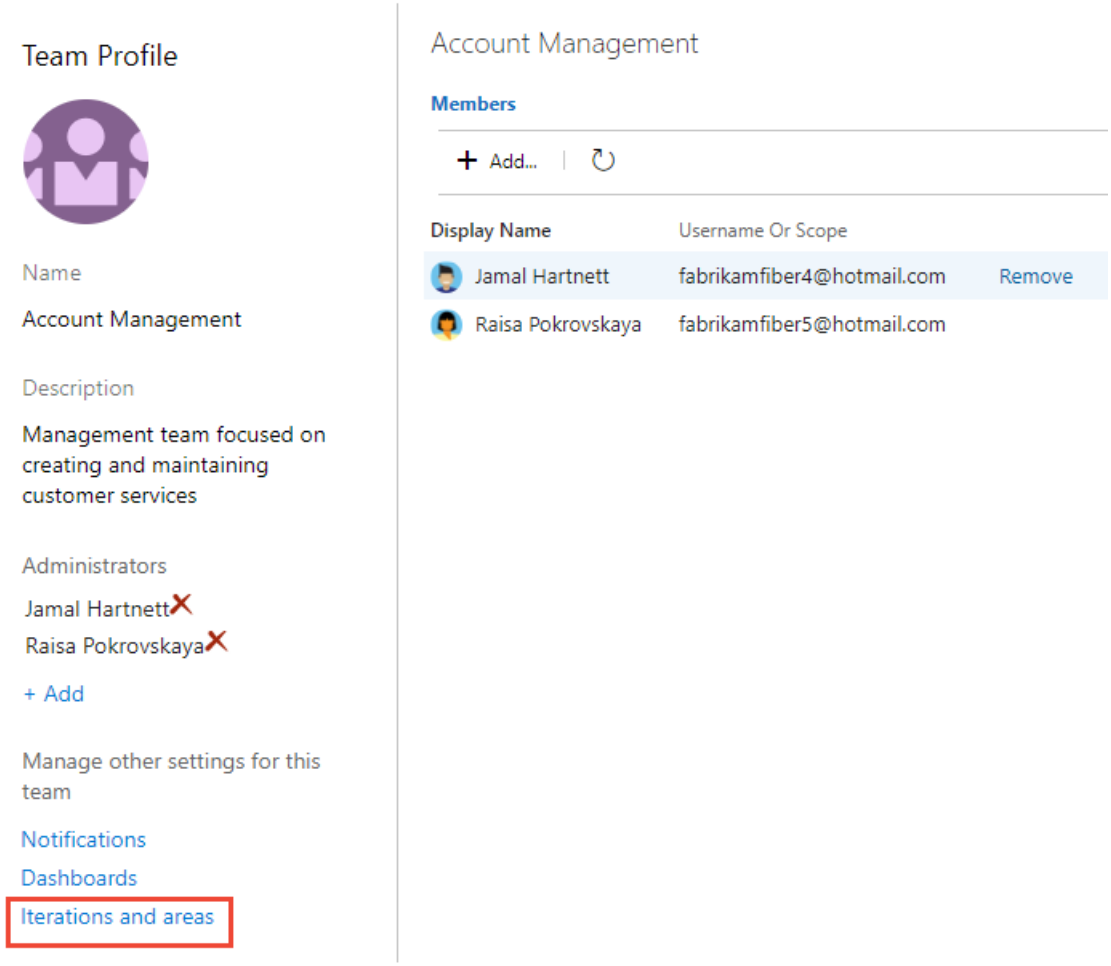
You define both areas and iterations from **Project Settings > Boards > Team configuration**. You can quickly navigate to it from **Teams**.

1. From **Project Settings**, choose **Teams**, and then choose the team whose settings you want to modify.

   Here we open the Account Management team.

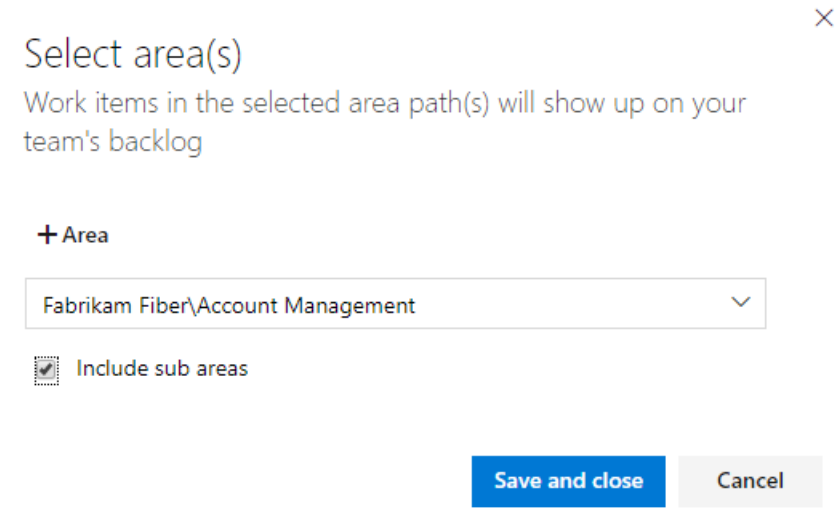


2. Choose **Iterations and areas** and then **Areas**.

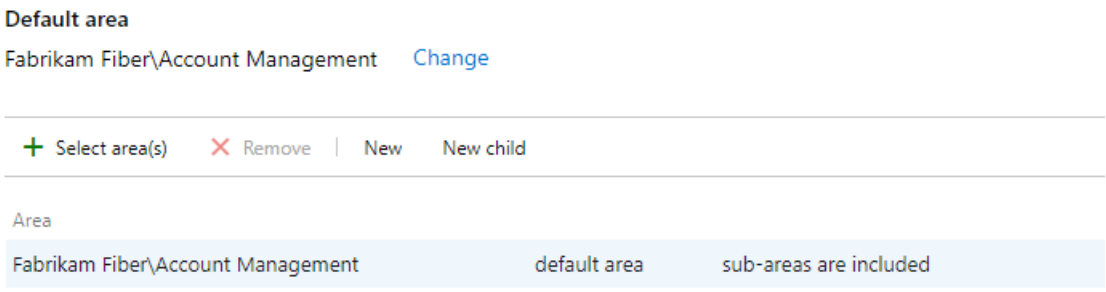


If you need to switch the team context, use the team selector within the breadcrumbs.

3. Choose **Select area(s)**, and select the area path for **Account Management** and check the **Include sub areas** checkbox.

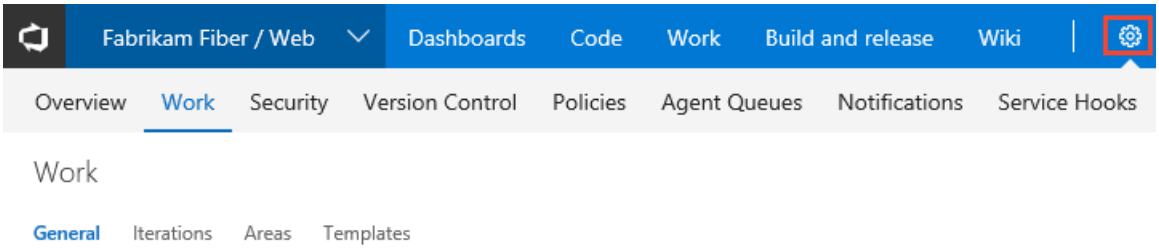


Verify that only this area path is selected for the team and is the default area path. Remove any other area paths that may have been previously selected.

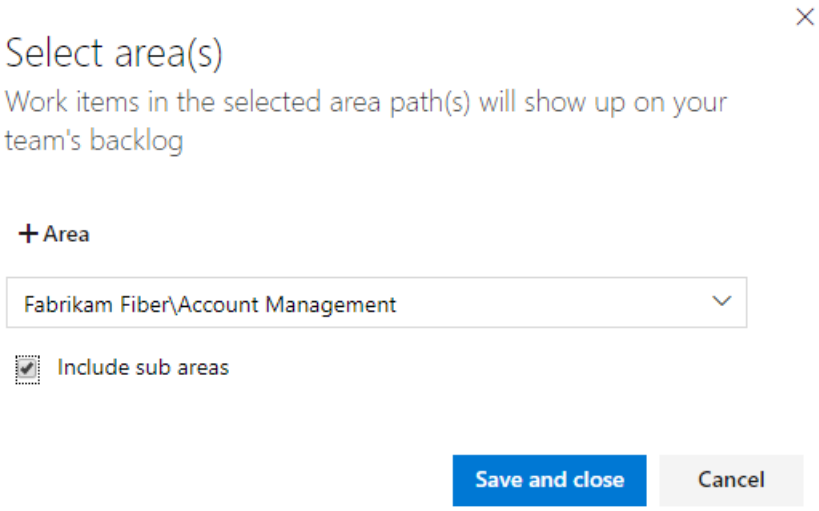


Repeat this step for all your management areas. If you want to enable rollup across all feature teams and management areas to the top-level area, repeat this step for the default team. In our example, that corresponds to Fabrikam Fiber.

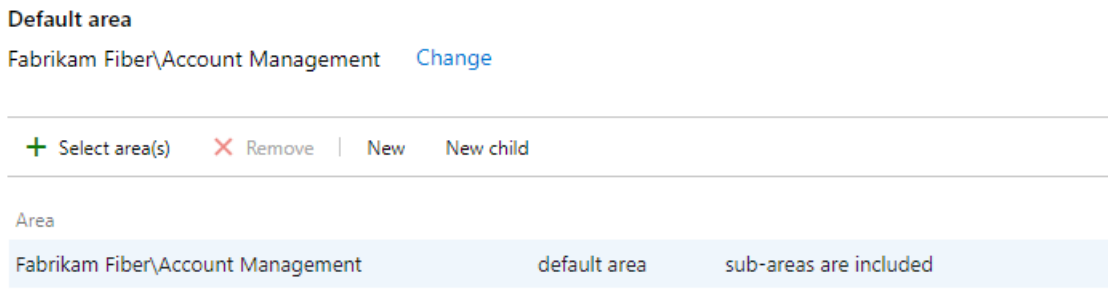
1. You open team settings from the top navigation bar. Select the team you want and then choose the gear icon. To learn more about switching your team focus, see Switch project, repository, team



2. Choose **Work**, and then **Areas**.
3. Choose **Select area(s)**, and select the area path for **Account Management** and check the **Include sub areas** checkbox.



Verify that only this area path is selected for the team and is the default area path. Remove any other area paths that may have been previously selected.

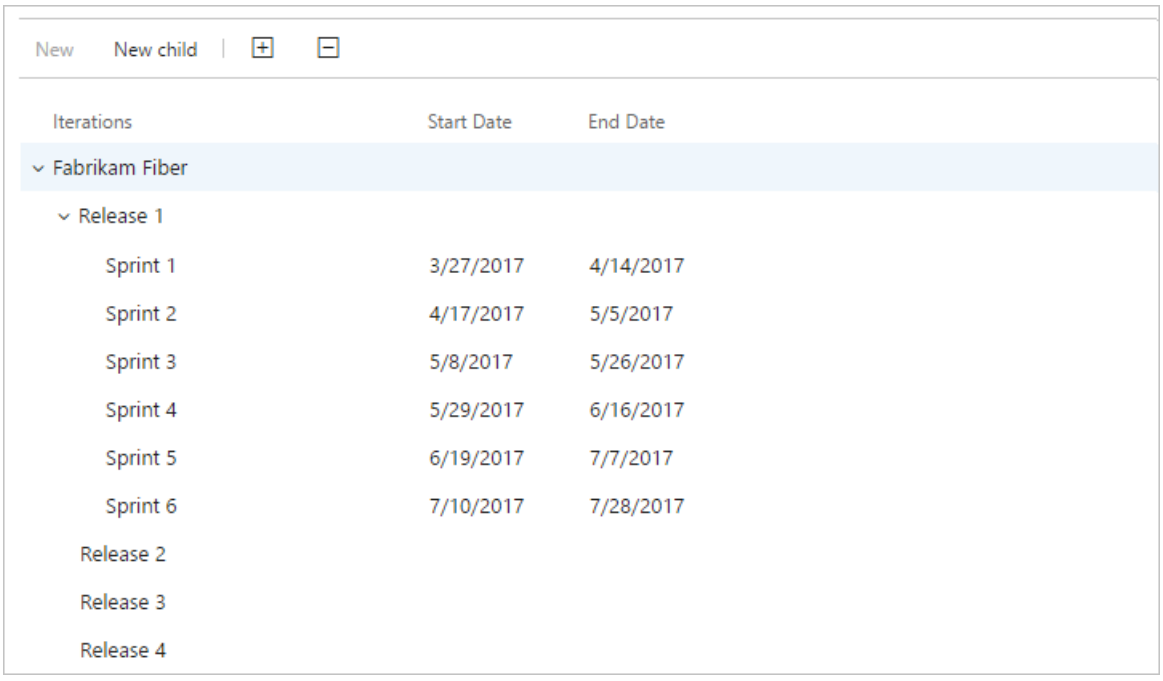


Repeat this step for all your management areas. If you want to enable rollup across all feature teams and management areas to the top-level area, repeat this step for the default team. In our example, that corresponds to Fabrikam Fiber.

Define a single sprint cadence for all teams

If your feature teams use Scrum or use sprints to assign their work, you'll want to set up a series of sprints that all teams can use. By default, you'll see a set of predefined sprints. Add more sprints and set their sprint dates from **Project Settings** as described in Add iterations and set iteration dates. You can rename and edit the default sprints as needed.

> **NOTE**
> While maintaining a single sprint cadence simplifies project administration, you can create different cadences as needed. For example, some teams may follow a monthly cadence while others follow a 3-week cadence. Simply define a node under the top project node for each cadence, and then define the sprints under those nodes. For example:
>
> - Fabrikam Fiber/CY2019
> - Fabrikam Fiber/3Week Sprints
>
> Here we define the start and end dates of the first 6 sprints corresponding to a 3-week cadence.

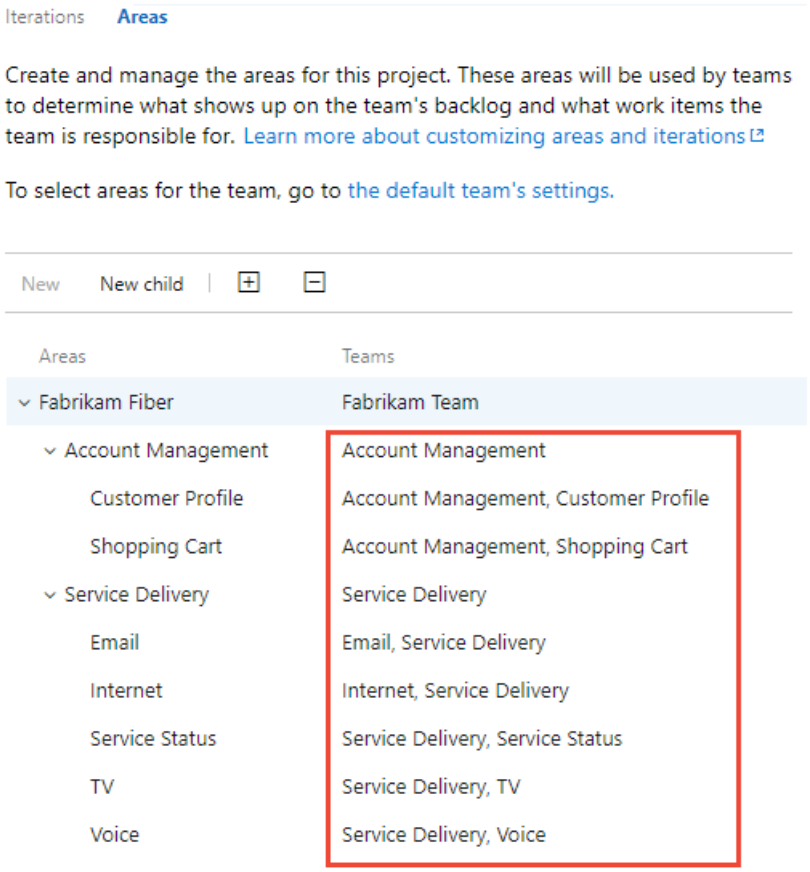


Configure other team settings

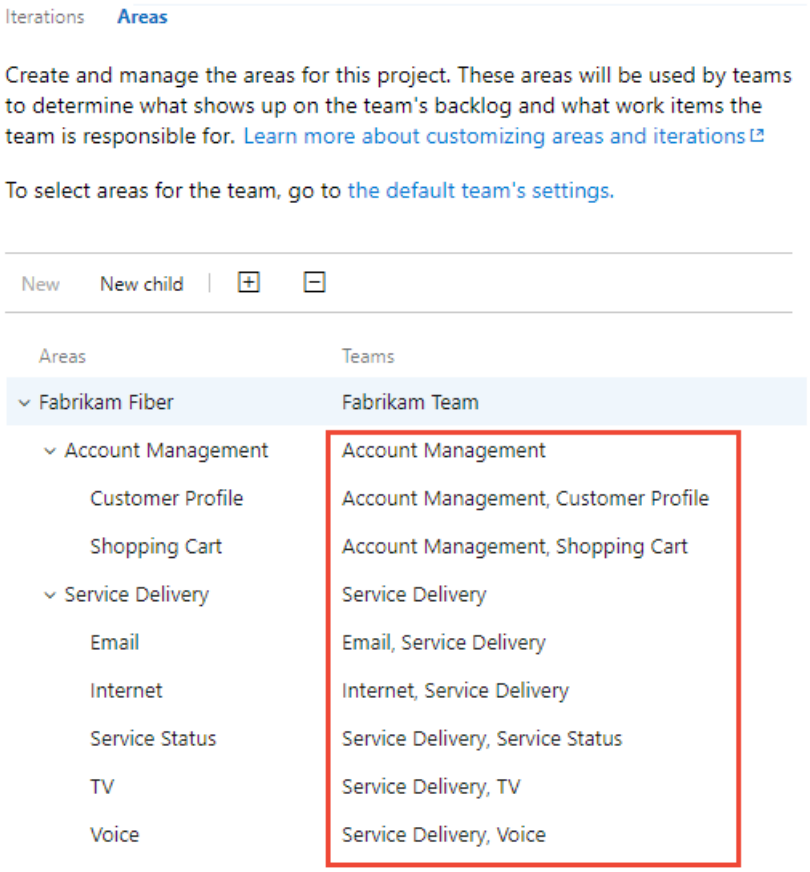
For teams to be well defined, you'll want to add team administrator(s) and have them verify or configure other team settings. To learn more, see Manage and configure team tools.

Review area paths assigned to teams

From **Project Settings > Project configuration > Areas**, you can review which **Area Paths** have been assigned to which teams. To modify the assignments, choose the team and change the team's area path assignments.



From **Project Settings > Work > Areas**, you can review which **Area Paths** have been assigned to which teams. To modify the assignments, choose the team and change the team's area path assignments.



Exercising select features with shared area paths

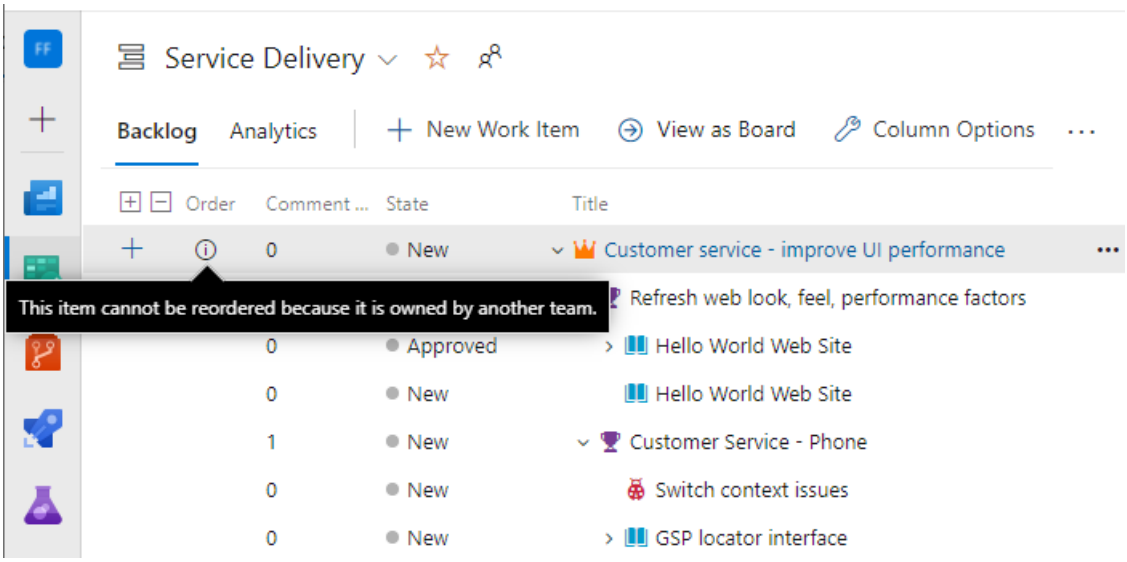
When you share area paths across two or more teams, you'll want to understand how Azure Boards manages conflicts that can arise when exercising these features:

- Reordering or reparenting work items on a backlog or board
- Updates made to Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields when dragging items to a different column

Reordering and reparenting work items

All backlogs and boards support drag-and-drop to reorder and reparent work items. Updates made to one team backlogs and boards are reflected in other team backlogs and boards that share the same area path. You may need to refresh the page to view the changes.

You can only use drag-and-drop to reorder or reparent work items assigned to area paths selected for your team. When the **Parents** view option is enabled, work items may appear on your backlog that your team doesn't own. Anything that appears with the ⓘ information icon can't be reordered nor reparented as it's owned by another team.



Kanban board column updates

Because each team can customize the Kanban board columns and swimlanes, the values assigned to Kanban board fields may differ from what you expect when another team updates the work item from a different board. Even if the management team and the feature teams configure their Feature Kanban board columns with identical workflow mapping, updating work items on one team's Kanban board won't be reflected on another team's Kanban board. Only when the work item moves to a column that maps to a workflow state does the card column reflect the same on all boards.

By design, the team with the longest area path wins the conflict and determines the values for the Kanban **Board Column**, **Board Column Done**, and **Board Lane** fields. If the shared area paths are of equal depth, the results are non-deterministic.

The primary work-around for this issue is to maintain single ownership of work items by Defining area paths and assign to a team. Another option is to add custom workflow states that all teams can use. For details, see Customize the workflow (Inheritance process).

The primary work-around for this issue is to maintain single ownership of work items by Defining area paths and assign to a team. Another option is to add custom workflow states that all teams can use. For details, see Change the workflow for a work item type.

Next steps

Review team Delivery Plans

Related articles

- Create your backlog
- Kanban quickstart
- Organize your backlog
- Work with multi-team ownership of backlog items
- Fix display, reordering, and nesting issues

Review team delivery plans in Azure Boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022

Use the visualization options provided by the Delivery Plans feature of Azure Boards to review the schedule of stories or features that your teams plan to deliver. A delivery plan shows the scheduled work items by sprint (iteration path) of selected teams against a calendar view.

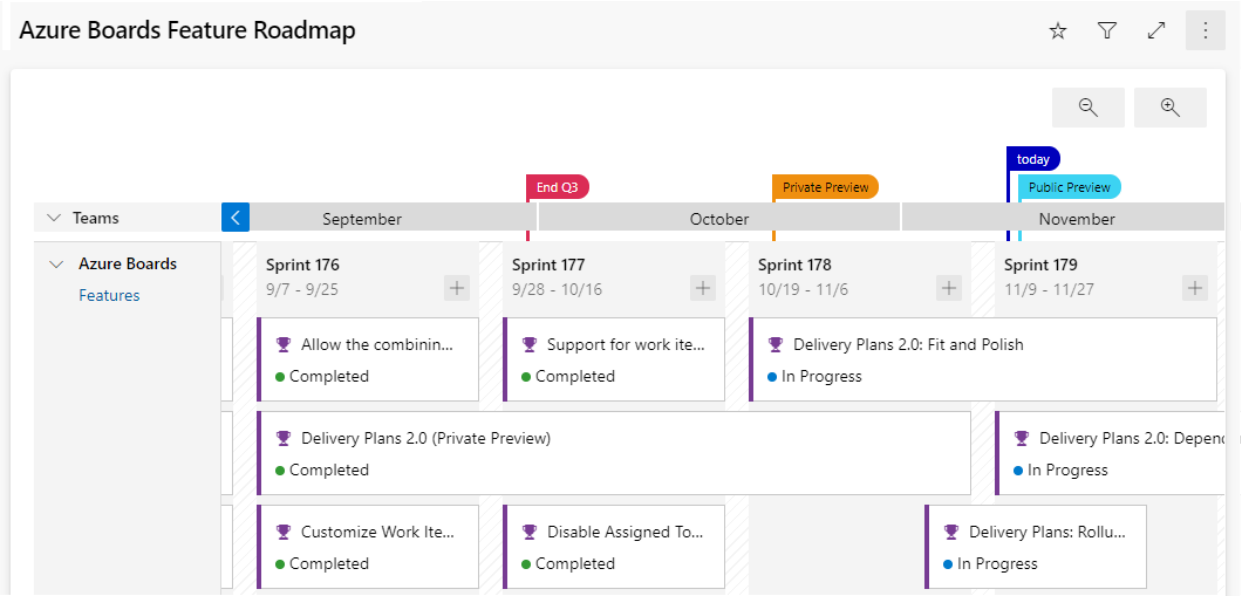
> **NOTE**
> This article describes using Delivery Plans 2.0, which is available for Azure DevOps Services and Azure DevOps Server 2022. For information on the Delivery Plans Marketplace extension that supports Azure DevOps Server 2020 and earlier versions, see Delivery Plans 1.0.

Use the Delivery Plans feature to ensure that your teams are aligned with your organizational goals. You can view multiple backlogs and multiple teams across your whole account. Interact with the plan by using simple drag-and-drop operations to update or modify the schedule, open cards, expand and collapse teams, and more.

Delivery Plans supports these tasks:

- View up to 15 team backlogs, including a mix of backlogs and teams from different projects.
- Add custom portfolio backlogs and epics.
- View work items that span several iterations.
- Reset start date and target date through drag-and-drop borders.
- Add backlog items to a team from a plan.
- View rollup progress of features, epics, and other portfolio items.
- View dependencies that exist between work items.
- Enable stakeholders to view plans.

Any plan that you created with the original Delivery Plans extension works with the Delivery Plans feature. You don't have to migrate any data or reconfigure plan settings. To learn how to add or edit a plan, see Add or edit a delivery plan.



In this article, you'll learn how to:

- Open a plan from the list of defined plans.
- Review a plan with your teams.
- Use the interactive elements of plans and change a plan view.
- View rollups.

For information on working with dependencies, see Track dependencies.

Prerequisites

- To view a delivery plan, you must be a member of the Project Collection Valid Users group. Users granted **Stakeholder** access for a private project can view plans. Users granted **Stakeholder** access for a public project can add and view plans.
- To open or modify a work item or add work items to a plan, you must have **Edit work items in this node** set to **Allow** for the area paths assigned to the work item. For details, see Set permissions and access for work tracking.

For work items and dependency lines to appear on the plan:

- Work items must belong to a team's product backlog or portfolio backlog. Only work items that belong to a category selected for viewing on a team's backlog appear on the plan.
- Team product or portfolio backlog must be enabled.
- Sprints must be selected for each team defined in the plan.
- Start and end dates must be defined for each iteration.
- Iteration paths must be assigned to each work item.
- For dependency icons and lines to show, work items must be linked via the **Predecessor-Successor** link type or other custom dependency link type. (Remote link types are not supported.) You can add custom link types only for on-premises environments.

> **TIP**
> If you edit a plan and the changes that you make don't seem to appear in the plan, refresh your browser. A browser refresh is sometimes needed to trigger the updates.

Review a plan with your teams

It takes multiple autonomous teams to develop large software projects. Autonomous teams manage their own backlog and priority, which contributes to a unified direction for that project. Review Agile culture for a discussion of autonomous teams and organizational alignment.

Regular reviews of the project schedule with these teams help ensure that the teams are working toward common goals. Delivery plans provide the needed multiple-team view of your project schedule.

Questions that you might address during the review include:

- *How confident are the teams in meeting the deliverables scheduled for each sprint?*
- *Are dependencies across teams adequately addressed via the planned deliverables?*
- *Are there gaps in the schedule, where no deliverables are scheduled? What's the cause? Can the issue be mitigated?*

For example, you might use delivery plans internally to share the schedule of features. By seeing the work that many teams have planned for the next three sprints, you can easily see if a plan has the right priorities and spot dependencies.

In this way, a delivery plan is a driver of alignment while letting each team remain autonomous. Individual teams can work to different sprint cadences, if needed, and manage different work item types (stories, features, or epics). Their work is all visible with the same plan view. Teams can even be part of different projects if they use different processes. Customize the card fields so that you see only the data fields that interest you and that apply for each work item type.

Best practices for using a delivery plan

- Determine how you want to use the delivery plan. Some ideas are:
  - Review quarterly plans for features to be delivered.
  - Sync up monthly with several teams that have dependencies.
  - Review cross-project deliverables and identify dependencies.
- Use a consistent sprint schedule across your project teams and organization when possible. Although the plan can accommodate a variety of sprint schedules, it adds to visual clutter. Use the same sprints for backlogs, features, and epics. Don't create specific sprints for epics or other portfolio backlogs.
- Use **Start Date** and **Iteration** to specify the time frame for a work item. Or, use **Start Date** and **Target Date**. However, don't specify both **Iteration** and **Target Date** for a work item. **Target Date** will always override the **Iteration** end date on the plan.
- Minimize the number of fields that you choose to display on your cards.
- Eliminate cross-team ownership of area paths. Cross-team ownership of area paths can lead to undesirable edge cases.
- Keep your work items up to date. When changes occur, update the target dates or iteration paths.

Note the following items:

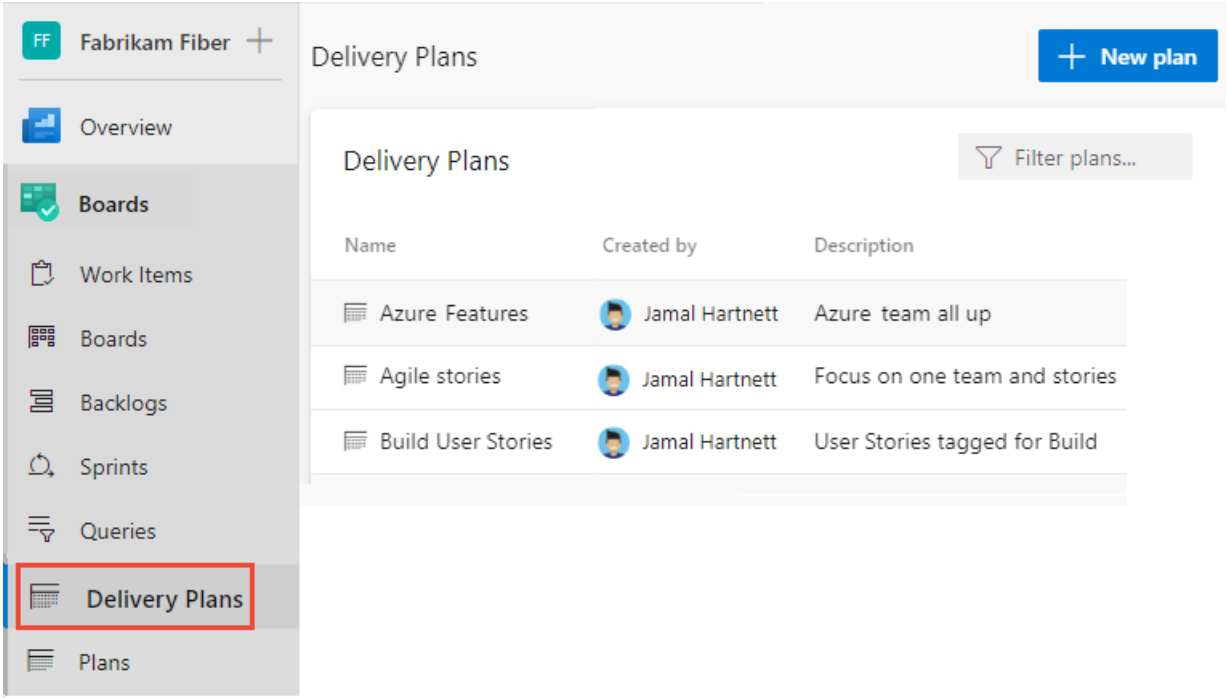
- Plan views display the set of months that correspond to the iteration paths selected by the teams whose backlogs appear in the plan.
- Plan views are limited to a maximum of 15 teams or backlogs.
- Zooming out can cause fields and tags to disappear from the cards. The farther you zoom out, the harder it is to fit items on a card. Certain items might be hidden, depending on the zoom level.
- Rollup isn't supported for child work items that belong to a different project than that of the originating parent work item.
- If **Start Date** or **Target Date** is missing from a work item, you can add it to the custom process defined for the project, as discussed in Add and manage fields (inheritance process).

Open a plan

After you've defined a few plans, you'll see them listed on the **Plans** page under **All**. Or you'll see the ones you've favorited under **Favorites**. You can see their title, description, and most recent creator/editor.

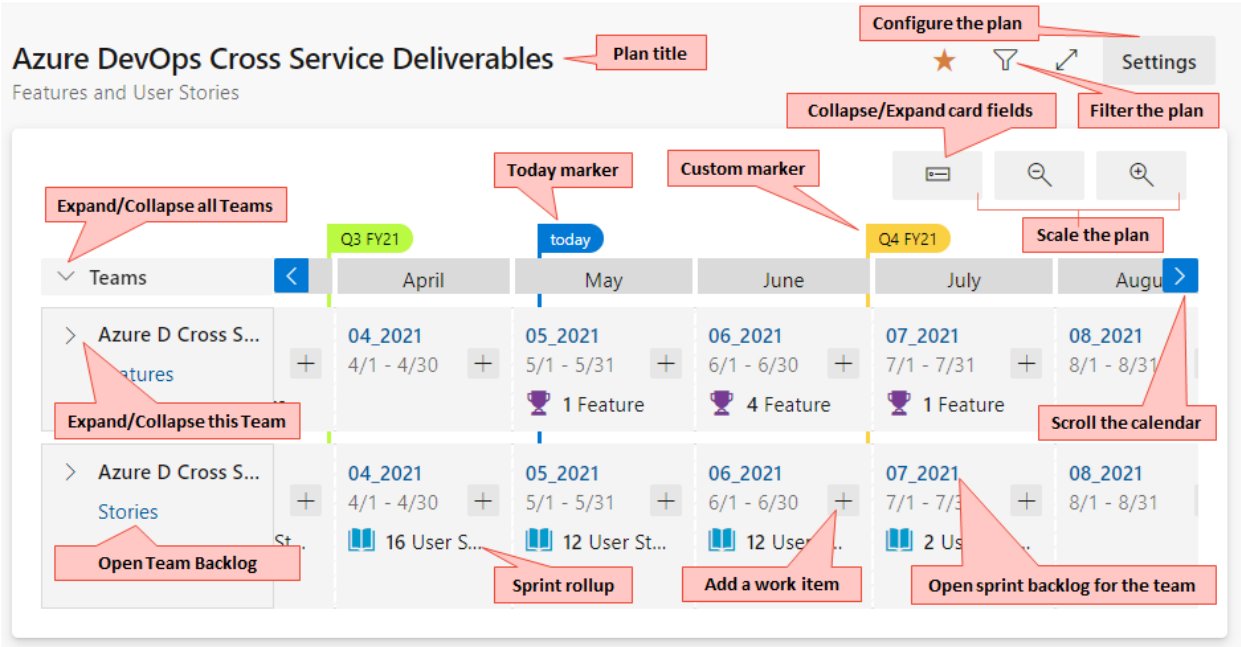
Use **Add to favorites** ☆ to favorite a plan so that you can quickly return to that plan. You can also search for other plans in the project.

To open a plan, open **Boards** > **Delivery Plans** and choose the plan name. You can sort by any of the columns: **Name**, **Created By**, **Description**, **Last configured**, or **Favorites**.



Interact with a plan

Each team's backlog specified in a delivery plan appears as a row within the plan view. When a row is collapsed, a rollup of the backlog items appears. When a row is expanded, a card for each backlog item appears, organized by its assigned iteration.



TIP

Work items appear in the prioritized order listed for the sprint backlog, which inherits the priority from the product backlog.

You can interact with the plan in the following ways:

- Filter the plan by selecting **Filter** . You can filter on any field that you include in the plan. Settings are based on the keyword or text filter. For more information, see Interactively filter your backlogs, boards, and plans.

- Scale the size of the cards and calendar by selecting **Zoom out** or **Zoom in** .
- To view previous or future months, select **Scroll calendar left** or **Scroll calendar right** . You can also scroll through the plan by selecting the plan and dragging your mouse horizontally.
- To view details for a team, expand the team row by selecting **Expand team row** or by clicking once on a sprint within a team row.
- To expand and collapse all team rows, select **Expand all team rows** or **Collapse all team rows** next to **Teams**.
- Scroll the view vertically to view teams that appear lower within the plan view.
- To view titles only, select **Collapsed card fields** . To view all fields, select **Expand card fields** .
- Select a card title to open the backlog item and view details. Close the work item to return to the plan.
- To add a work item to a sprint, select **Add item** within the sprint and team that you want to add it to.
- To change the fields displayed on the cards, select **More actions** .

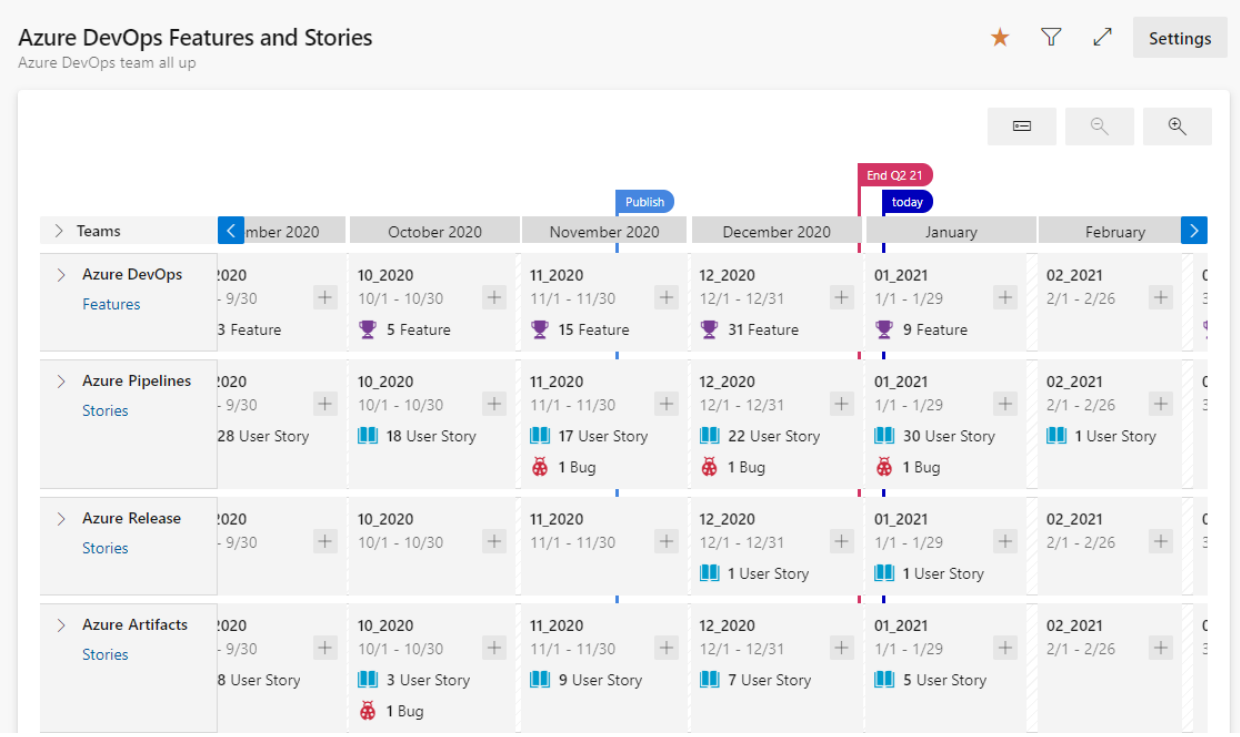
Collapse teams for summary information

A benefit of Delivery Plans is the ability to view multiple teams across the projects that you care about. There are two main ways to view more teams within the plan view:

- Collapse all teams to focus on summary data.
- Minimize the number of fields displayed on cards.

To gain a summary view of work that's scheduled, collapse all teams. You can then more easily look for gaps in the forecast.

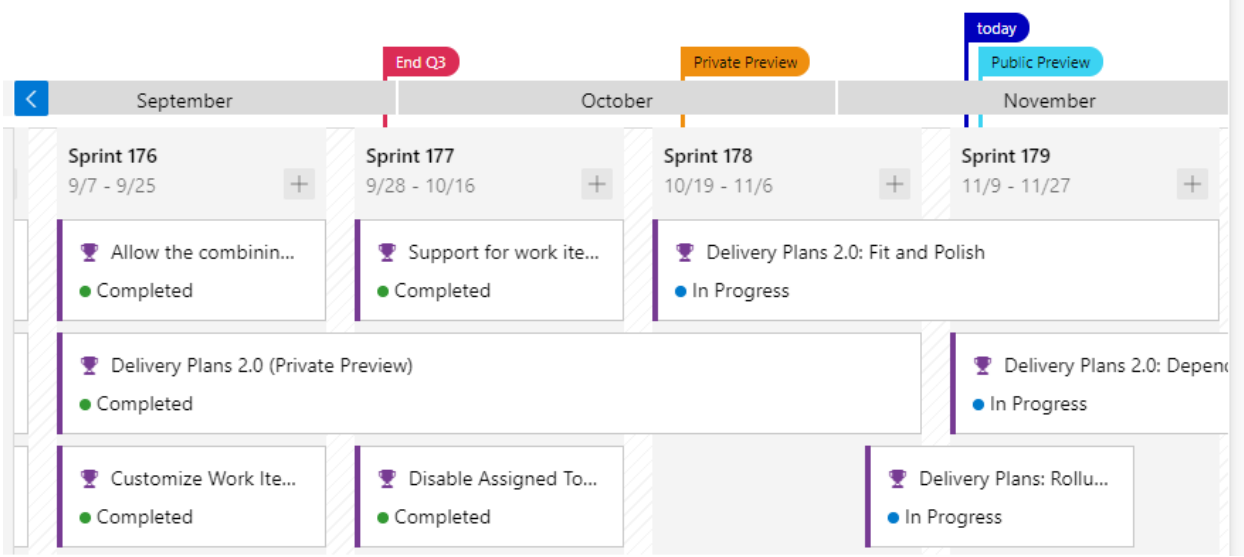
Expand or collapse each team row by selecting **Expand team row** or **Collapse team row** next to the team name.



Show work that spans one or more iterations

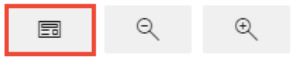
For work items that span one or more iterations, define **Start Date** and **Target Date**. The plan displays cards

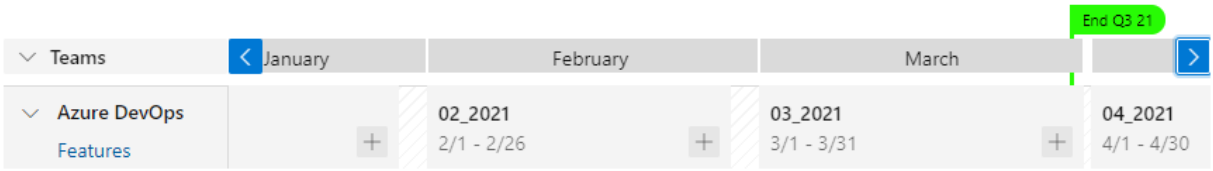
that start and end according to the dates that you set, as shown in the following image. You can also grab the left or right border of a work item and drag it to a new start date or target date.



View titles only in the collapsed card view

The collapsed card view allows you to quickly switch back and forth between cards that show titles only and cards that show all fields configured for the plan. To view titles only, select **Collapsed card fields**  . To view all fields, select **Expand card fields**  .

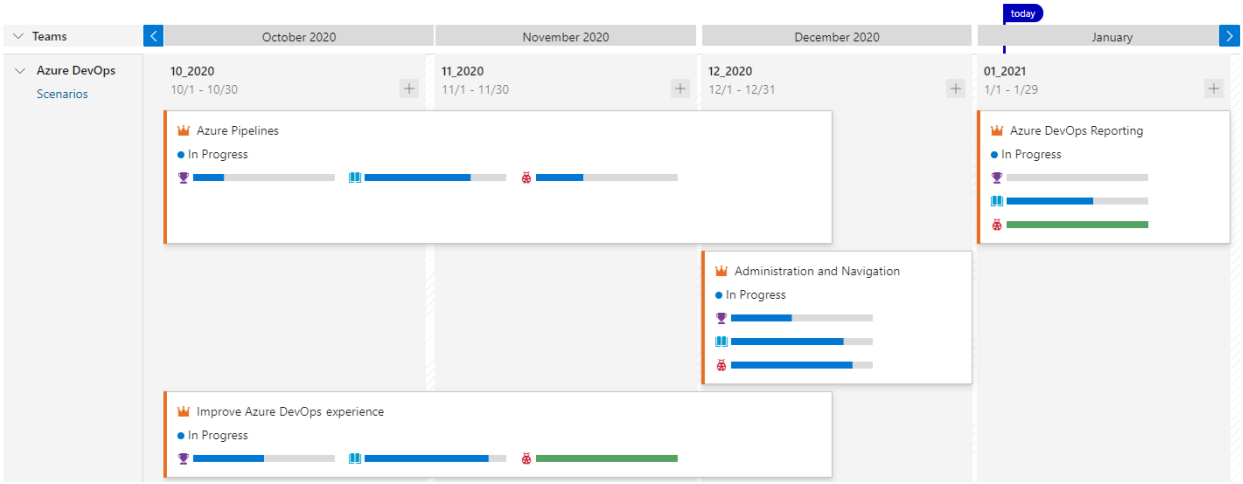




View the rollup of features and epics

A rollup displays a fuller picture of the underlying work directly on the cards in your delivery plan. Rollup views are available for features, epics, or any portfolio backlog you've added to your project. To enable rollups, open your plan settings, select **Fields**, and then select **Show child rollup data**.

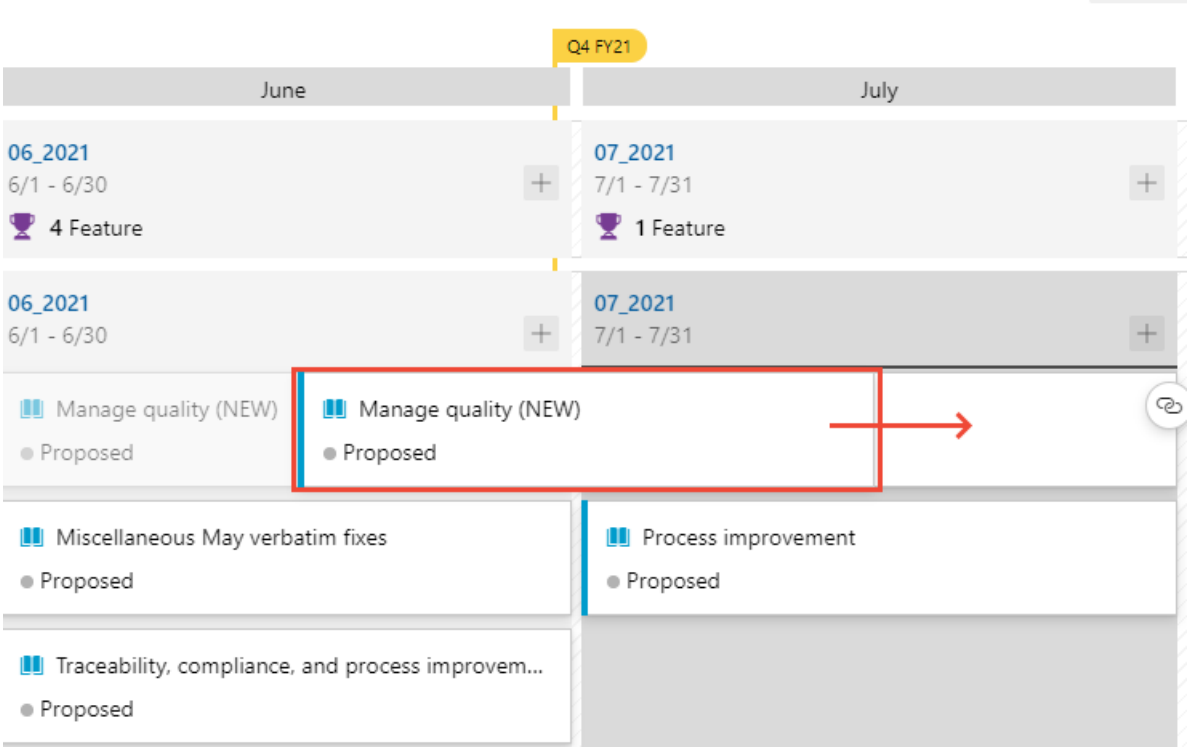
For example, here's a plan view of four scenarios with a rollup of the child features, user stories, and bugs for a single team.



You can also view rollups from a backlog view, as described in Display rollup progress or totals.

Update the iteration for a backlog item

As changes occur to the schedule, you can update the iteration for a backlog item. Move a card to a different iteration. This change helps drive alignment across your organization.



Print a delivery plan

You can print all or a portion of your delivery plan, depending on the view that you want to capture and share. You can print only one page at a time by using your browser's **Print** feature.

Here are some tips for printing portions of a plan:

- Select **Full screen mode** .
- Expand or collapse one or more teams and zoom in or out to get the view that you want.
- Take a screenshot of the plan view, or select the **Print** function of your browser.

Related articles

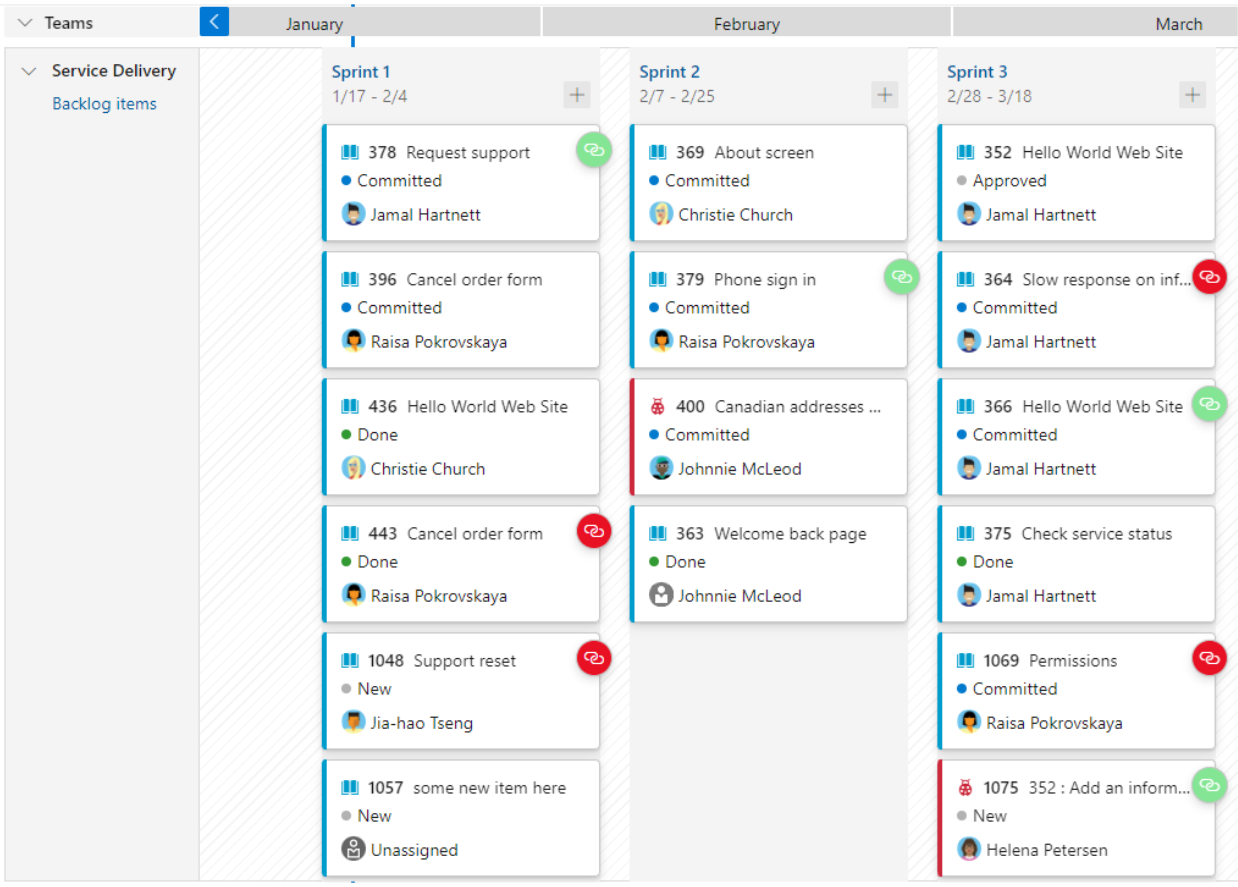
- Add or edit a delivery plan
- Track dependencies by using Delivery Plans
- Interactively filter your backlogs, boards, and plans
- Backlogs, boards, and plans
- Add teams
- Portfolio management
- Manage teams and configure team tools

Track dependencies by using Delivery Plans

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022

With Delivery Plans, you can track dependencies that have been added to work items. Dependency tracking supports the **Predecessor/Successor** link type between work items. The following image shows several work items that have dependencies. Those cards with a green icon indicate there are no dependency issues. Those cards with a red icon indicate there are issues with one or more dependencies. Dependency issues arise when a predecessor work item is scheduled to finish after a successor work item.



To view dependencies, you must first define the Delivery Plan and dependencies between work items. To learn how, see Add or edit a Delivery Plans and Link user stories, issues, bugs, and other work items.

> **TIP**
> You can create dependencies between work items in different projects and different teams within the same organization, but not in projects in different organizations. You can open a work item and add a dependency through the links tab.

Prerequisites

- To view a Delivery Plan, you must be a member of the Project Collection Valid Users group. Users granted **Stakeholder** access for a private project can view plans. Users granted **Stakeholder** access for a public project can add and view plans.
- To open or modify a work item or add work items, you must have the **Edit work items in this node** set to

Allow for the Area Paths assigned to the work item.

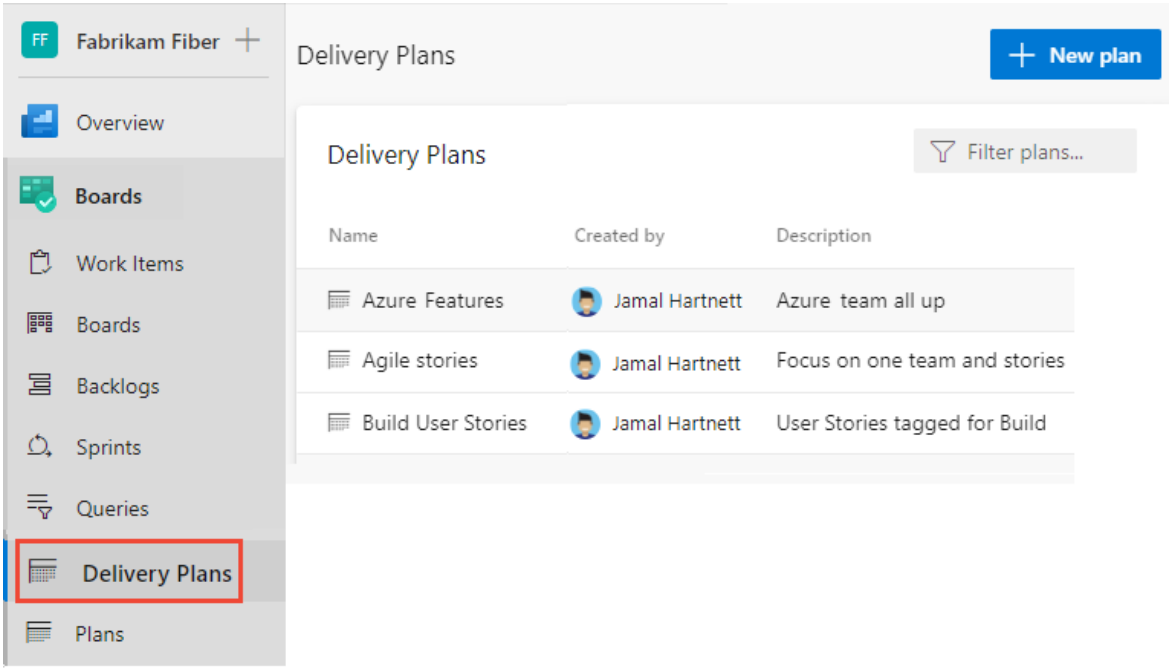
For work items and dependency lines to appear on the plan

- Team product or portfolio backlog must be enabled in order to select it for a plan.
- Work items must belong to a team's product backlog or portfolio backlog. Only work items belonging to a category selected for viewing on a team's backlog and meet any field criteria defined for the plan appear on the plan.
- Sprints must be selected for each team defined in the plan.
- Start and end dates must be defined for each project iteration.
- **Iteration Paths** or **Start Date/Target Date** must be assigned to each work item. When defined, **Start Date/Target Date** overrides the sprint assigned to a work item.
- For dependency icons and lines to show, work items must be linked using **Predecessor-Successor** link type.
- Team or teams must be expanded to view dependency icons and dependency lines.

> **TIP**
> If you edit a plan and don't see the changes you made appear in the plan, refresh your browser. A browser refresh is needed some times to trigger the updates.

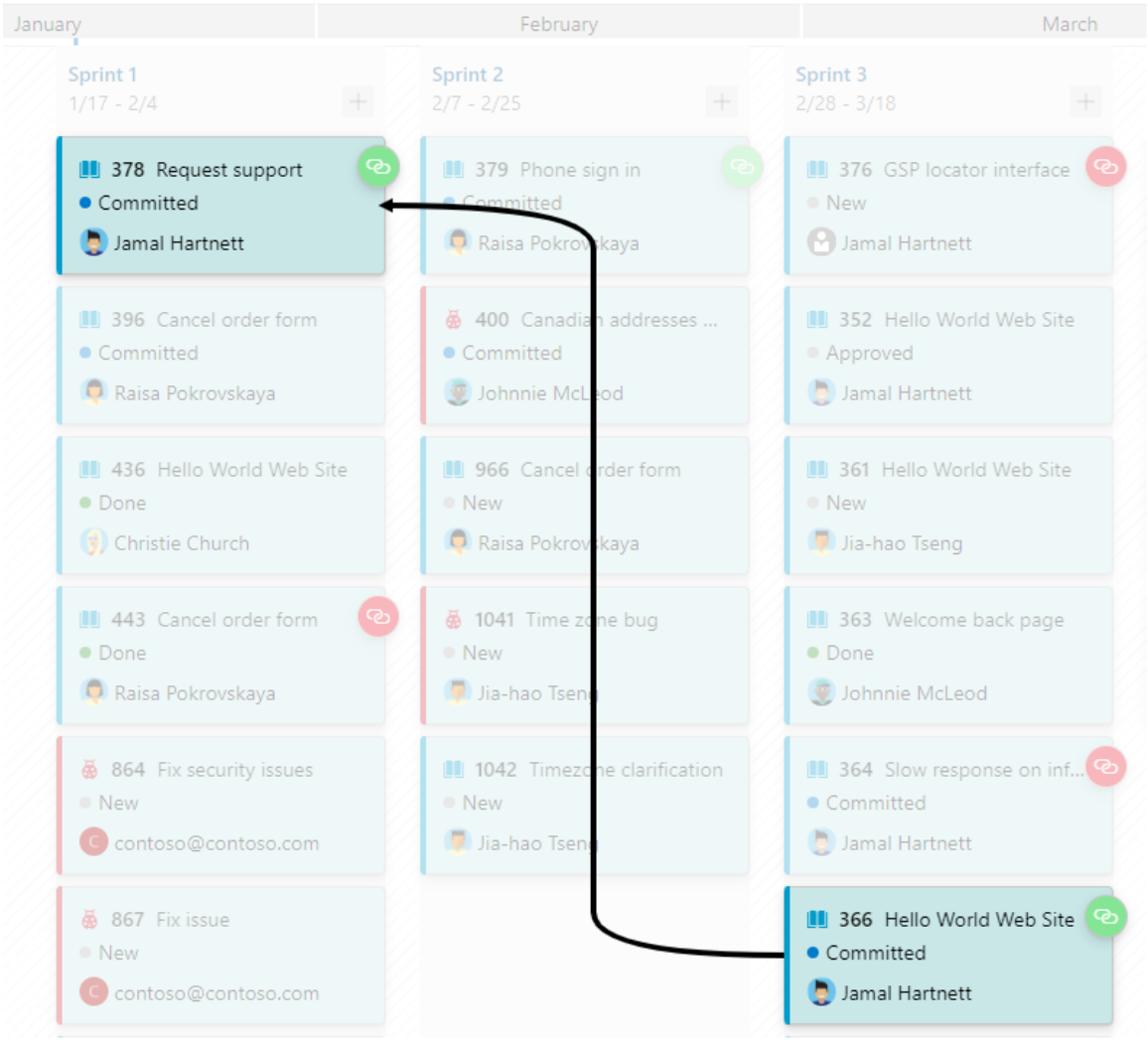
Show dependency lines for a work item

1. Open the Delivery Plan from **Boards > Delivery Plans**.



2. To view dependency lines for a work item, select the top or bottom of its card. To dismiss the lines, select the top or bottom of the card again, or anywhere else within the plan.

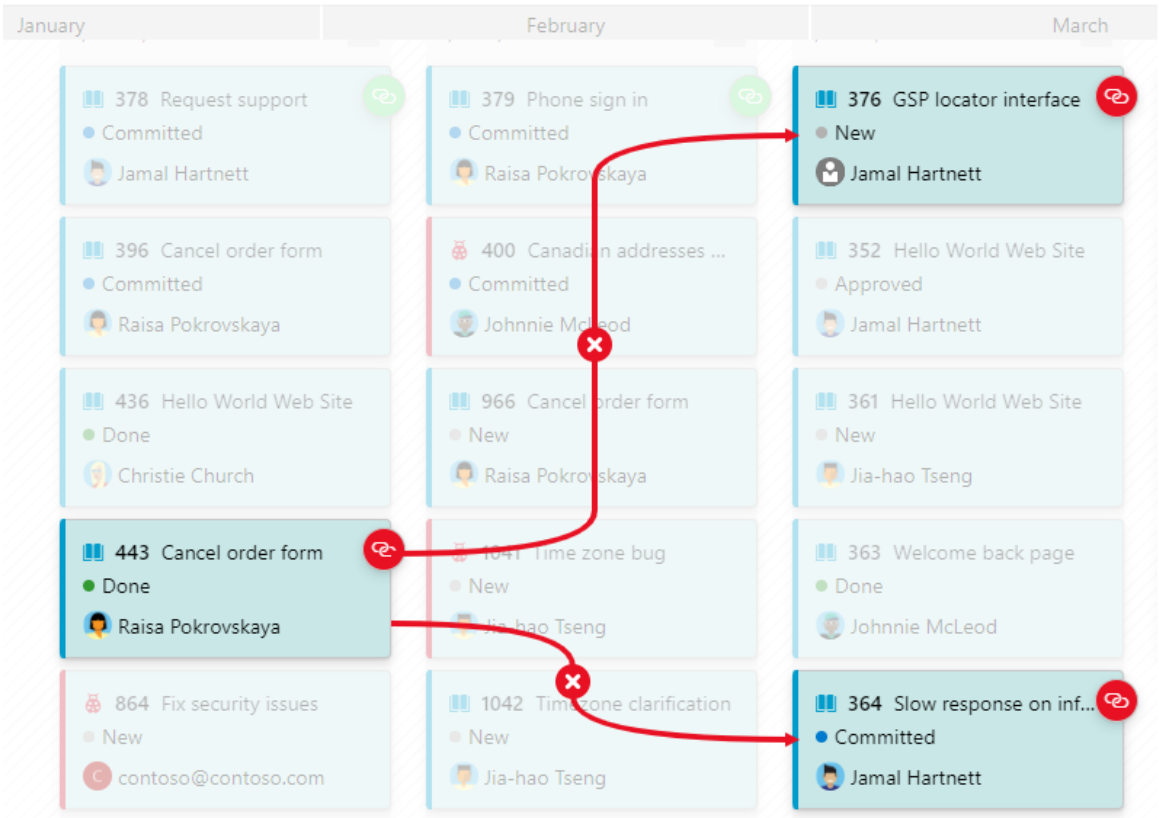
   Dependency lines that have no issues show up as black lines.



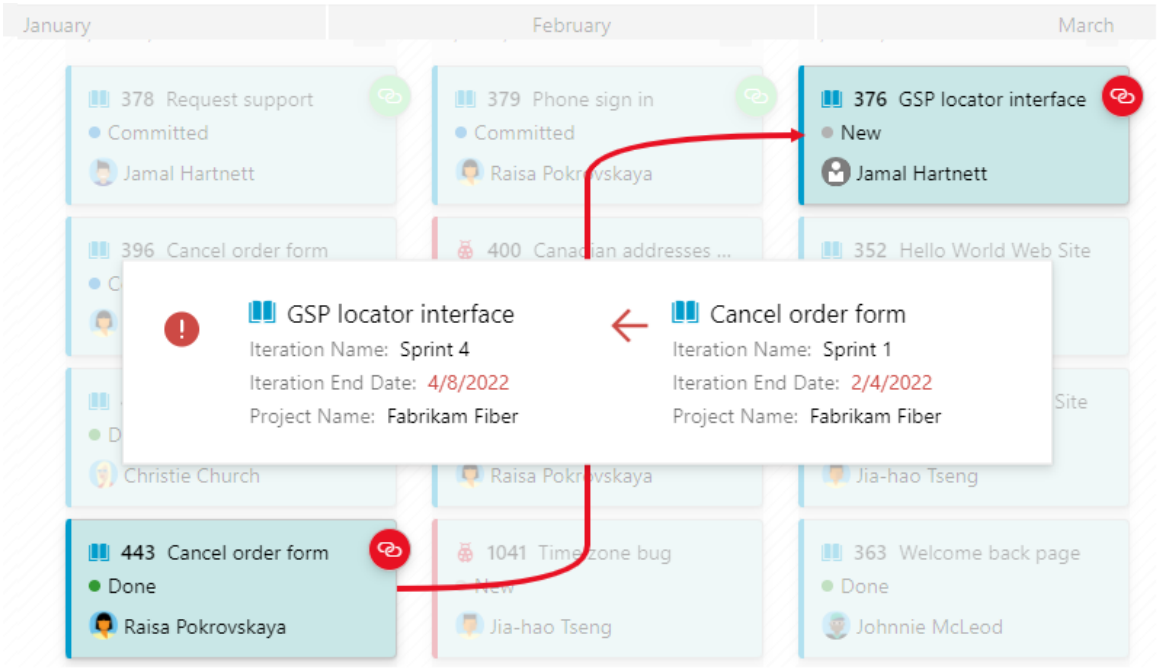
TIP

To view dependency lines across team backlogs, make sure to expand both teams.

Dependency lines that have issues, show up with red lines. The issues indicate that the successor work item is scheduled to end before the predecessor work item is completed.



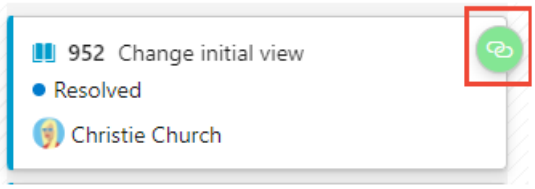
3. To view the issue, choose the icon.



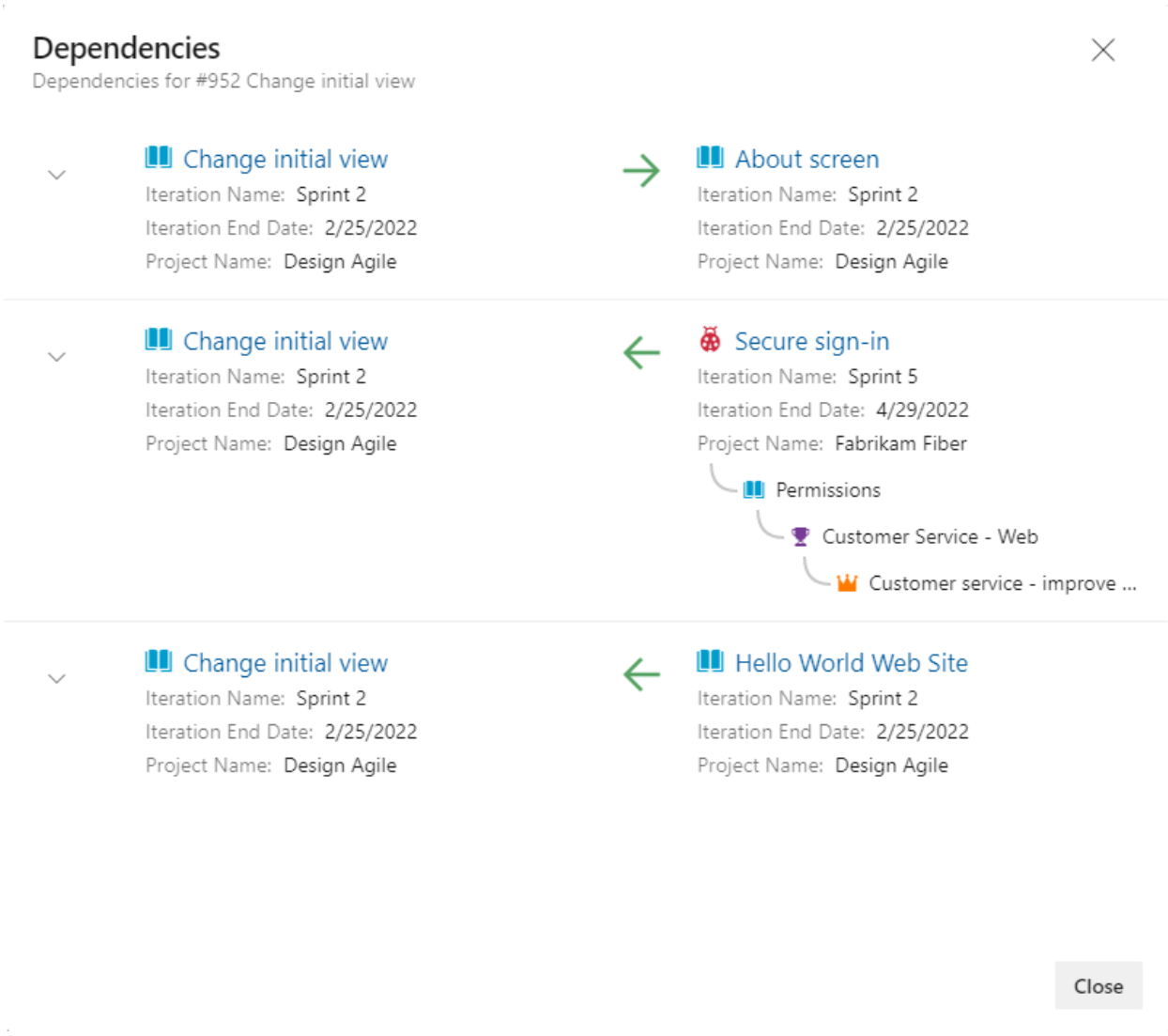
Open the dependency summary for a work item

To drill down into specific dependencies, open the Dependencies dialog for the work item. Choose the icon that indicates the work item has dependencies, either the green or red icon.

For example, here we choose the link icon for a work item with dependencies to several work items within the same project and another project.



The Dependencies dialog indicates that the work item has three predecessors and no issues.

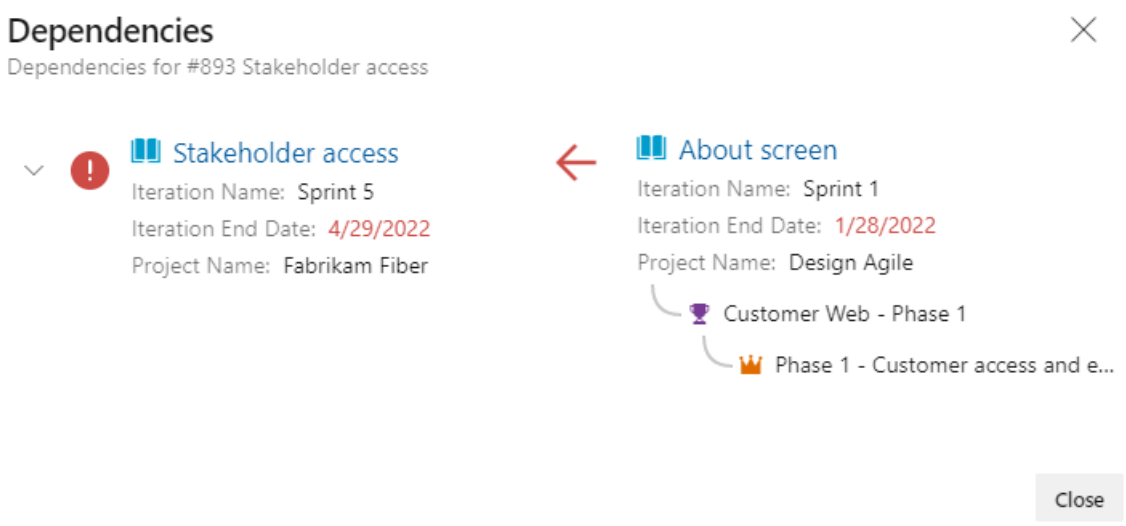


Identify dependency issues

When issues exist, they're highlighted in red. The issue always has to do with an end date for a successor work item occurring before the end date of the predecessor work item. Determine the end date by using either the **Target Date** for the work item or the **End Date** of the work item's assigned **Iteration Path**.

For example, the following Dependencies dialog indicates that two predecessor work items are scheduled to complete before the successor work item is scheduled to complete. A red exclamation mark and red colored arrows indicate there's an issue with the dependency.

When the dependency is to a work item in another project, the project information is shown as are other link relationships.



Related articles

- Add or edit a Delivery Plan
- Review team Delivery Plans
- Interactively filter your backlogs, boards, and plans
- Backlogs, boards, and plans

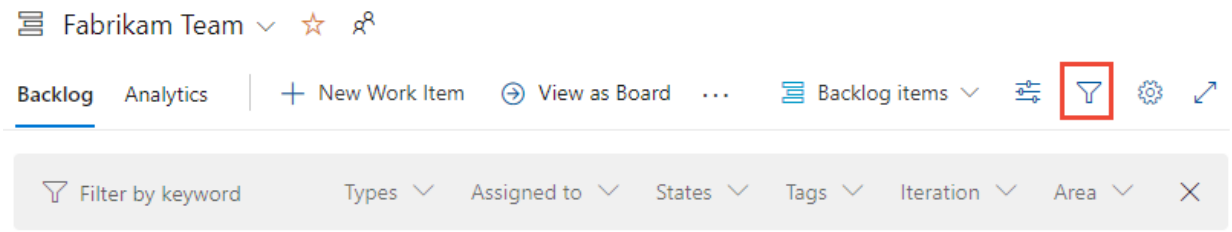
Interactively filter backlogs, boards, queries, and plans in Azure Boards

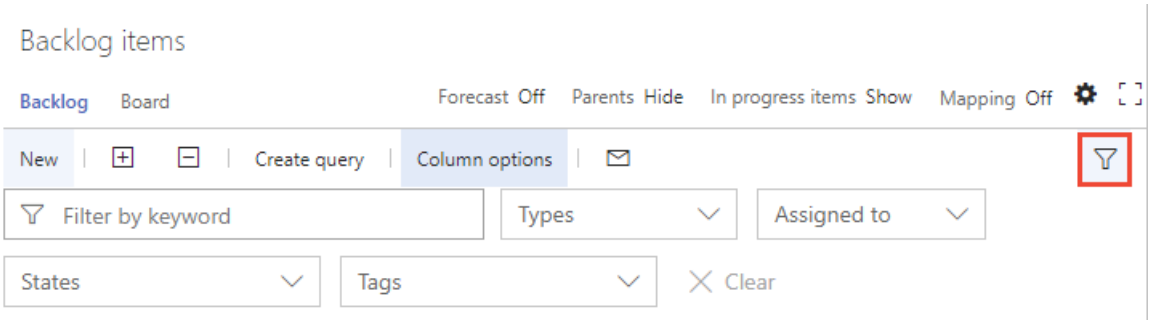
10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With filter functions, you can interactively apply one or more filters to an Azure Boards tool. Each tool is already filtered to show a subset of work items according to the tool function. For example, Backlogs and Boards display work items based on the selected **Area Paths** and **Iteration Paths** for the team. **Query Results** list work items based on the query clauses you've defined.

You enable the filter feature by choosing **Filter**.





From these tools, you may still have a large number of work items listed or displayed. Interactive filtering supports your ability to focus on a subset of them. You can apply one or more filter functions to each of the Azure Boards tools.

Use filters to complete these tasks:

- In daily scrum meetings, filter the Kanban board to focus on assigned work for a specific sprint.
- Or, if your team uses the Sprints Taskboard, filter for a team member's completed assigned work.
- To focus on a group of work items, filter based on the **Parent Work Item**, by **Area Path**, or **Tags**.
- To triage work items, create a query and filter to focus on similar work grouped by Area Path or Tags.

Supported filter functions

Filter functions are available from all Azure Boards tools: Work items, Boards, Backlogs, Sprint Backlogs and Taskboards, Queries, and Delivery Plans. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

The following table indicates the supported options based on the tool indicated with a ✔ or is listed.

Backlogs and boards are subject to filters defined for the team as described in Set up your Backlogs and Boards. Other tools have predefined filters based on the view, query filter clauses, or settings you select.

Filter functions are available from all Azure Boards tools: Boards, Backlogs, Sprint Backlogs, Taskboards, and Queries. The set of features supported depends on the tool and Azure DevOps version. *(Use the content selector to view the filters available for your version.)*

Supported options are indicated with a ✔ or listed.

Tool

Keywords or **ID**

Fields

Parent
Work Item

Tags

Work items

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Boards

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

✔

Backlogs

✔

- Assigned To
- Work Item Type
- States
- Area Path

Note 1

✔

Sprints (Backlogs

& Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔ (Note 2)

✔

Sprints (Backlogs & Taskboards)

✔

- Assigned To
- Work Item Type
- States
- Area Path

✔

Query Results

✔

- Work Item Types
- Assigned To
- States
- Tags

Note 1

✔

Delivery Plans

✔

- Work Item Types
- Assigned To
- States
- Area Path
- Iteration Path
- Tags

✔

✔

Plans

✔

- Work Item Types

- Assigned To
- States
- Tags

✔

Semantic search, Work Items

✔

- Projects
- Area Paths
- Assigned To
- Work Item Types
- States

✔

Notes

1. While the **Parent Work Item** isn't a filter function for Backlogs or Query Results, you can add the **Parent** field as a column and then do a keyword/phrase search on the Parent title to effectively filter on parent work items. The Parent field is supported for Azure DevOps Server 2020 and later versions. See also the Parent field and Parent Work Item section later in this article.
2. The **Parent Work Item** filter is supported for Sprint Backlogs and Taskboards for Azure DevOps Server 2020 and later versions.

Additional filter, sort, group, reorder, and rollup functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have more filters you can apply, sort, group, reorder, and rollup functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Rollup

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

✔

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Delivery Plans

✔ (Note 6)

✔

Semantic search, Work Items

✔ (Note 7)

Other filter, sort, group, and reorder functions

Along with the standard filter functions summarized in the previous table, the following table indicates which tools have other filters you can apply, sort, group, and reorder functions. Some functions, such as reorder, don't work when the filter function is enabled.

Tool

Filter settings

Sort

Group

Reorder

Work items

✔ (Note 1)
Completed Work Items

✔

Boards

✔ (Note 1)

✔

Backlogs

✔ (Note 1)
In Progress items
Completed Child items

✔ (Note 2)

✔ (Note 3)

Sprints, Backlogs

✔ (Note 1)

✔ (Note 2)

✔ (Note 3)

Sprints, Taskboards

✔ (Note 1)
Person

✔ (Note 4)

✔

Query Results

✔

✔ (Note 2)

Plans

✔ (Note 6)

Semantic search, Work Items

✔ (Note 7)

Notes

1. The Work items page is subject to filters based on the view selected. Boards and backlogs are subject to filters defined for the team as described in Set up your Backlogs and Boards. Completed and In Progress work items are determined based on the state categories assigned to the workflow state as described in How workflow states and state categories are used in Backlogs and Boards.

2. Grouping is supported through portfolio backlogs and boards, parent-child links, and tree hierarchy. Tree hierarchies are flattened when filtering is applied and reinstated when filtering is cleared.
3. Backlogs and Sprint Backlogs support reordering. However, when filtering is enabled, reordering isn't supported.
4. Taskboards provides a **Group by** function based on **People** or **Stories**.
5. Query Results supports multi-column sort.
6. Work items appear in the order defined for the team Sprint backlog, which it inherits from the team product backlog.
7. Semantic search supports sorting search results by the following fields—**Assigned To**, **Changed Date**, **Created Date**, **ID**, **State**, **Tags**, **Title**, and **Work Item Type**—and Relevance.

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- Display rollup progress or totals
- About backlogs, Work with multi-team ownership of backlog items

To learn more about these other functions, see the following articles:

- Reorder cards (Kanban Boards)
- About backlogs, Work with multi-team ownership of backlog items

Parent Work Item filter and Parent field

The **Parent Work Item** filter enables you to focus on one or more select features or epics. This filter function was added in July 2016 and made available in Azure DevOps Server 2017 and later versions.

The **Parent** field was added to Azure Boards in July of 2019 and then made available with the release of Azure DevOps Server 2020. You can add the **Parent** field to a list through the **Column Options** dialog, except for the **Work items** tool. You can also add the **Parent** field to cards on the Kanban Boards and Taskboards.

Persistence and saving filter options

Once you set the filter options for a specific view, your settings persist until you change them. There's no save button or other action you need to take.

> **NOTE**
> You can't set default filter options, nor set filter options for other members in your team.

Prerequisites

- All project members can exercise filter functions.
- All filter functions are set only for the current user until the user clears them.
- To filter using fields, first add the field as a column or to the card. For example, to filter by **Assign To**, **Iteration Path**, or **Work Item Type**—or the contents of any other field—add those fields to show on the cards, backlog, plan, or list.

To add columns or fields, see the following articles:

- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards
- For Taskboards, see Customize a sprint Taskboard

- For Plans, see Review team delivery plans.
- For Backlogs and Queries, see Change column options
- For Boards, see Customize cards.

Open and clear filter functions

1. From the Azure Boards tool, choose the view you want. For example:

   - For Work items, choose the **Assigned to me**, **Following**, **Mentioned**, or other view
   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

   - For Backlogs and Boards, choose the backlog level you want, such as **Stories**, **Features**, or **Epics**
   - For sprint Backlogs and Taskboards, choose the iteration
   - For queries, define the query filter criteria of interest.

2. Choose any other view settings available for your view. For example:

   - For Work items, from the **View options** menu, enable/disable **Completed Work Items**
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items** or **Completed Child items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

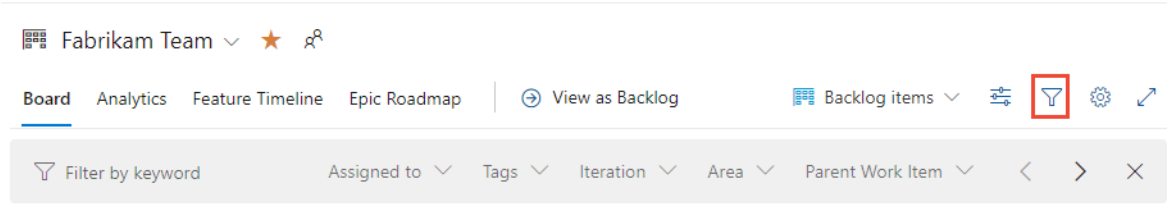
   - For Backlogs, from the **View options** menu, enable/disable **In Progress items**
   - For Taskboards, from the **Person** menu, choose **All**, **Unassigned**, or a specific team member.

3. For list views, add columns to display fields containing text you want to filter on or possibly sort on. For card views, add fields to display on cards containing text you want to filter on.

4. Open the filter function.

   Choose **Filter** . Or, enter the **Ctrl+Shift+f** keyboard shortcut.

   For example, here we open the filter toolbar for the Kanban board, **Backlog items**.

   Choose **Filter** .

5. Choose the filters of interest.

   The filter icon changes to a solid icon, **Filter** , to indicate filtering is applied.

   The page refreshes to show only those work items that meet all the selected filter criteria.

Inactive functions

When filtering is applied, the following functions are disabled or altered.

- For backlogs, the add-a-backlog-item panel, reordering (stack ranking), and forecasting tools are disabled.
- For backlogs set to **Show Parents**, the tree hierarchy is flattened.

Clear or dismiss filtering

To clear and dismiss filtering, choose **Clear and dismiss filtering** .

Filters remain in place until you explicitly clear them. When you refresh your backlog, board, or other tool, or sign in from another browser, filters remain set to your previous values.

Once the board is filtered, you can choose the filter icon to hide the drop downs and view the applied filters on the board. The filter icon turns opaque to signify a filtered board.

Filter logic and Boolean operators

Applying Boolean operators to filters is only supported for tags, as described in Filter based on tags later in this article. All other filters are applied with an implicit **AND** operator.

Apply keyword and ID filters

The keyword filter function filters lists or cards based on the fields displayed via **Column Options** or board settings. Also, you can enter a value for an ID, even the ID field is visible. As such, when filtering, consider what fields contain the keyword text or tags you want to filter on and make sure it's displayed.

Filtering is case-insensitive.

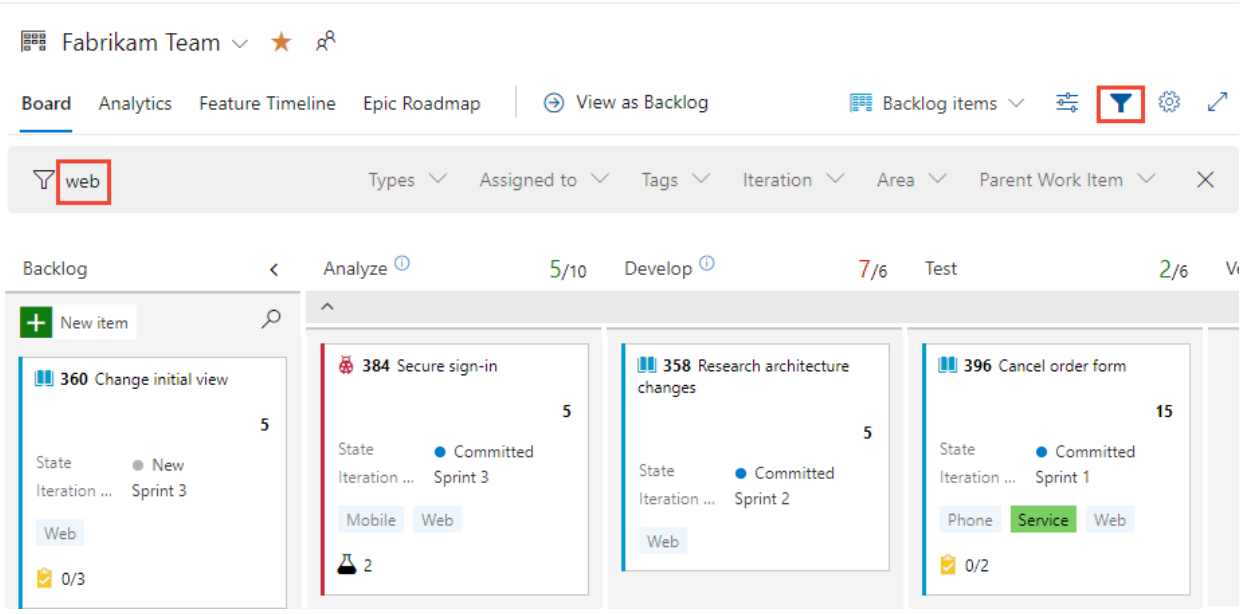
Ignore characters by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

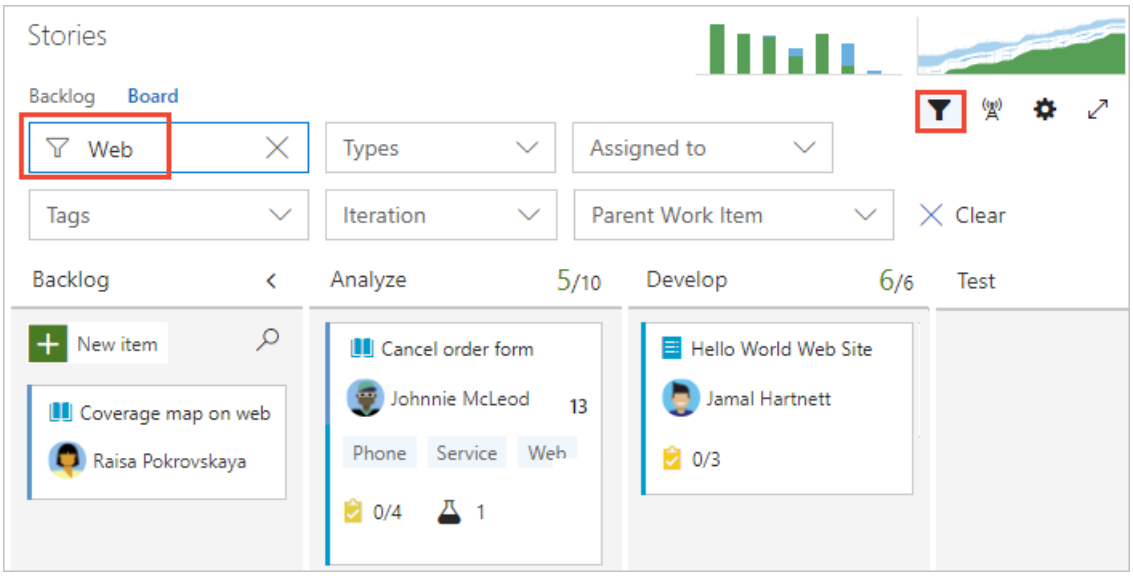
The filter criteria ignores the following characters when the field value starts with the character:
`{, (, [, !, @, #, $, %, ^, &, *, ~, `, ', "`.

Filter a board using a keyword

Here we filter the Kanban board to only show those cards that include 'web', either in the title, tag, or field.

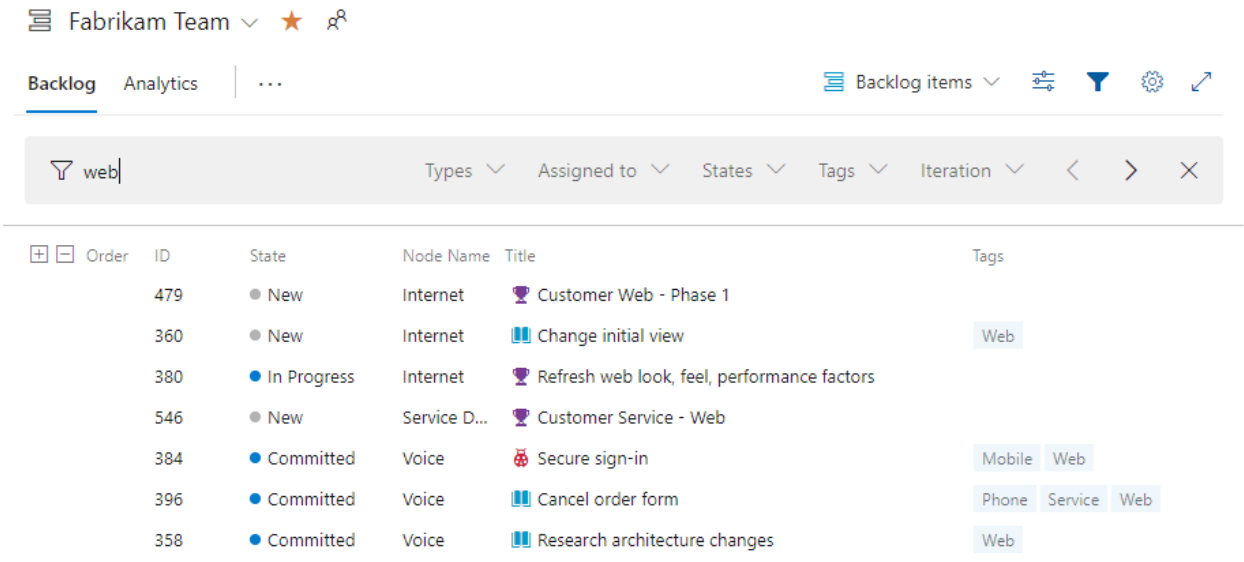


Here we filter the Kanban board to only show those cards that include 'Web', either in the title, tag, or displayed field.



Filter a backlog by using a keyword

Here we filter the Backlog with **Show Parents** enabled, to only show work items that include 'web'.



The filtered set is always a flat list, even if you've selected to show parents.

Filter based on a field

With filtering turned on, choose one or more values from the multi-select drop-down menu for each field available to you. The values for these fields are populated as follows:

- **Area**: The Node Name, which specifies the last node of an Area Path, of valid Area Paths and for which there are work items assigned to that Area Path
- **Assigned To**: All users who are currently assigned to work items on the board plus Unassigned
- **Iteration**: All Iteration Paths selected for the current team and for which there are work items assigned to that iteration
- **Work item type**: Work item types defined for the Requirements Category (product backlog) or Features or Epic categories (feature or epic portfolio backlogs), subject to work items being assigned to the work item types
- **Tags**: All tags assigned to work items on the board
- **Parent Work Items**: All features defined for the team, or all epics defined for the team when viewing the

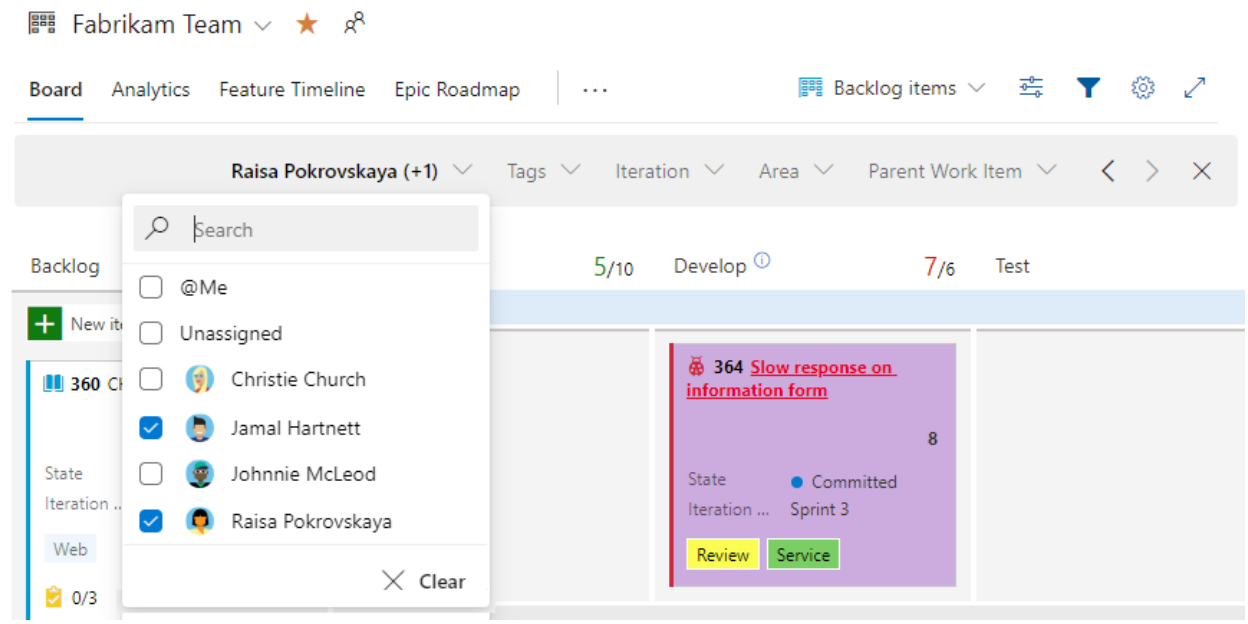
Features board

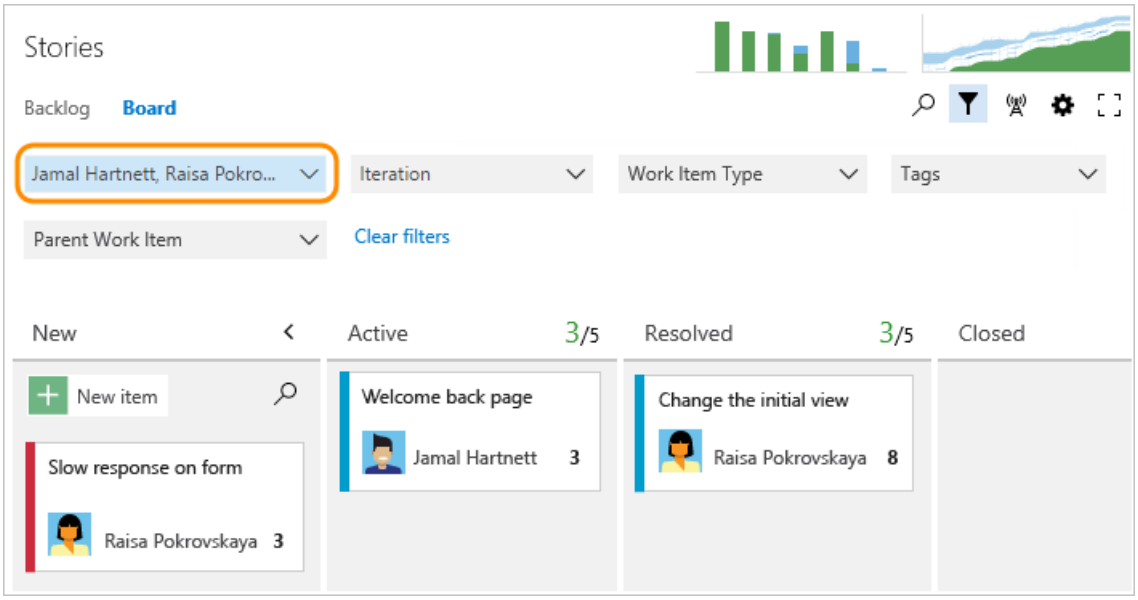
> **NOTE**
> Filter options are dependent on the work items that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Filter a Kanban board by using select field values

You can filter by select field values using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features or Epics).

For example, here we filter for all items assigned to Jamal and Raisa.





Kanban board filter logic

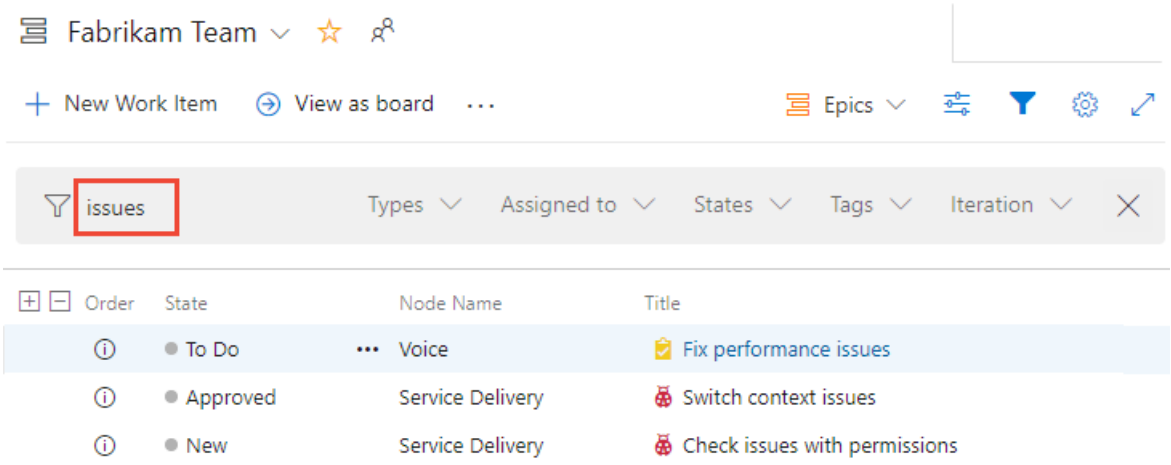
Cards are filtered based on the assignments made in the following order and logic:

1. **Assigned to**: Show all cards that are assigned to user 1 `OR` user 2 `AND`
2. **Iteration**: Show all cards that are assigned to Iteration 1 `OR` Iteration 2 `AND`
3. **Work Item type**: Show all cards that are work item type 1 `OR` work item type 2 `AND`
4. **Tags**: Show all cards that have tag 1 `AND` or `OR` tags 2, based on your selection of `AND | OR` . `AND`

5. **Parent Work Items**: Show all cards that have Parent Work Item 1 `OR` Parent Work Item 2.

Filter a backlog by using fields

Here we show a filtered backlog based on the keyword "issues". Filtered pages show the filtered icon. The filtered set is always a flat list, even if you've selected to show a hierarchical backlog view.



Filter based on the Parent Work Item

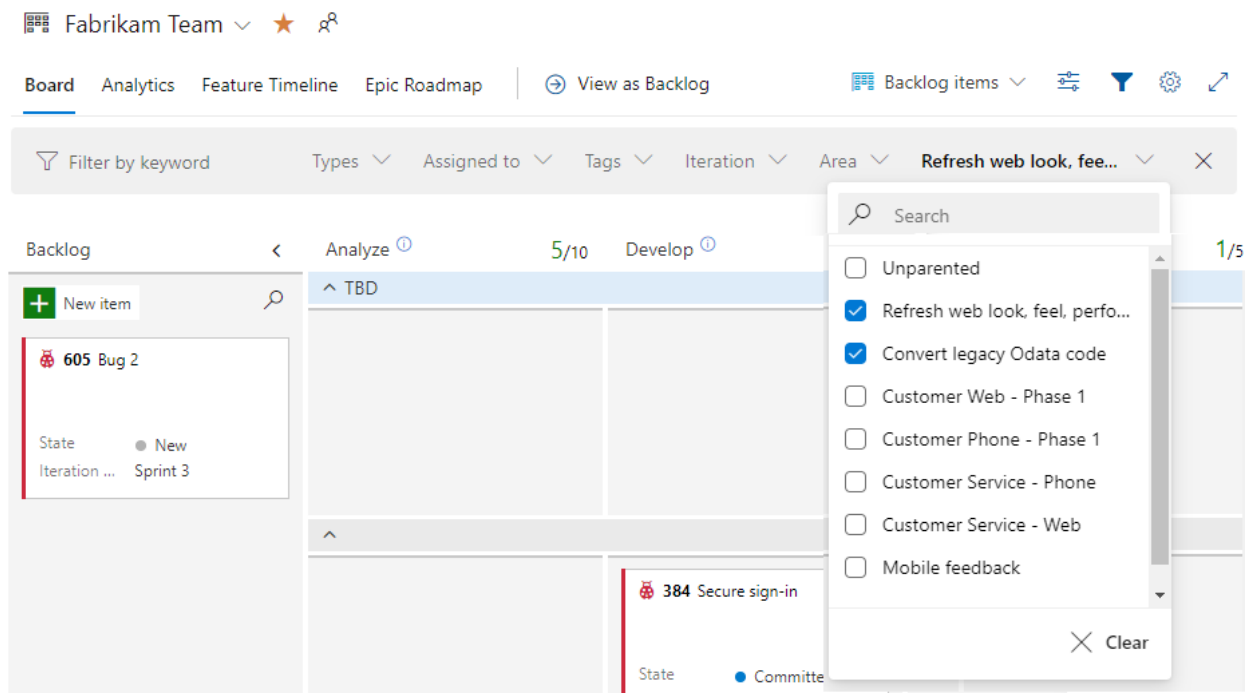
You can use the **Filter by parent** feature to filter by select parent work items using the Kanban board for your product backlog (Stories, Product Backlog Items, or Requirements) or a portfolio backlog (Features).

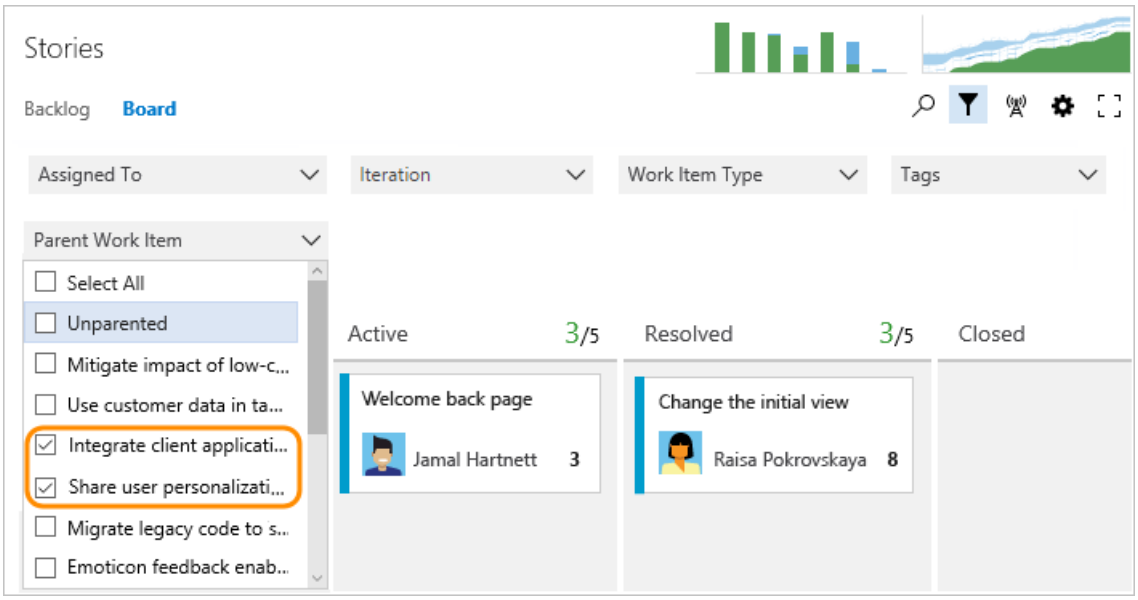
You can use this feature only when you've created features or epics and linked them to user stories or features, respectively. A quick and easy way to create the links is to map them using drag-and-drop. Mapping creates parent-child links between the work items.

> **NOTE**
> The **Filter by parent** feature doesn't support filtering of parent work items of the same work item type. For example, you can't filter the Stories backlog by specifying user stories that are parents of nested user stories.

To start filtering, choose **Filter** . Choose one or more values from the multi-select drop-down menu for the Parent Work Item. These values are derived from the Features you've defined.

Here, we choose two features on which to filter the board.



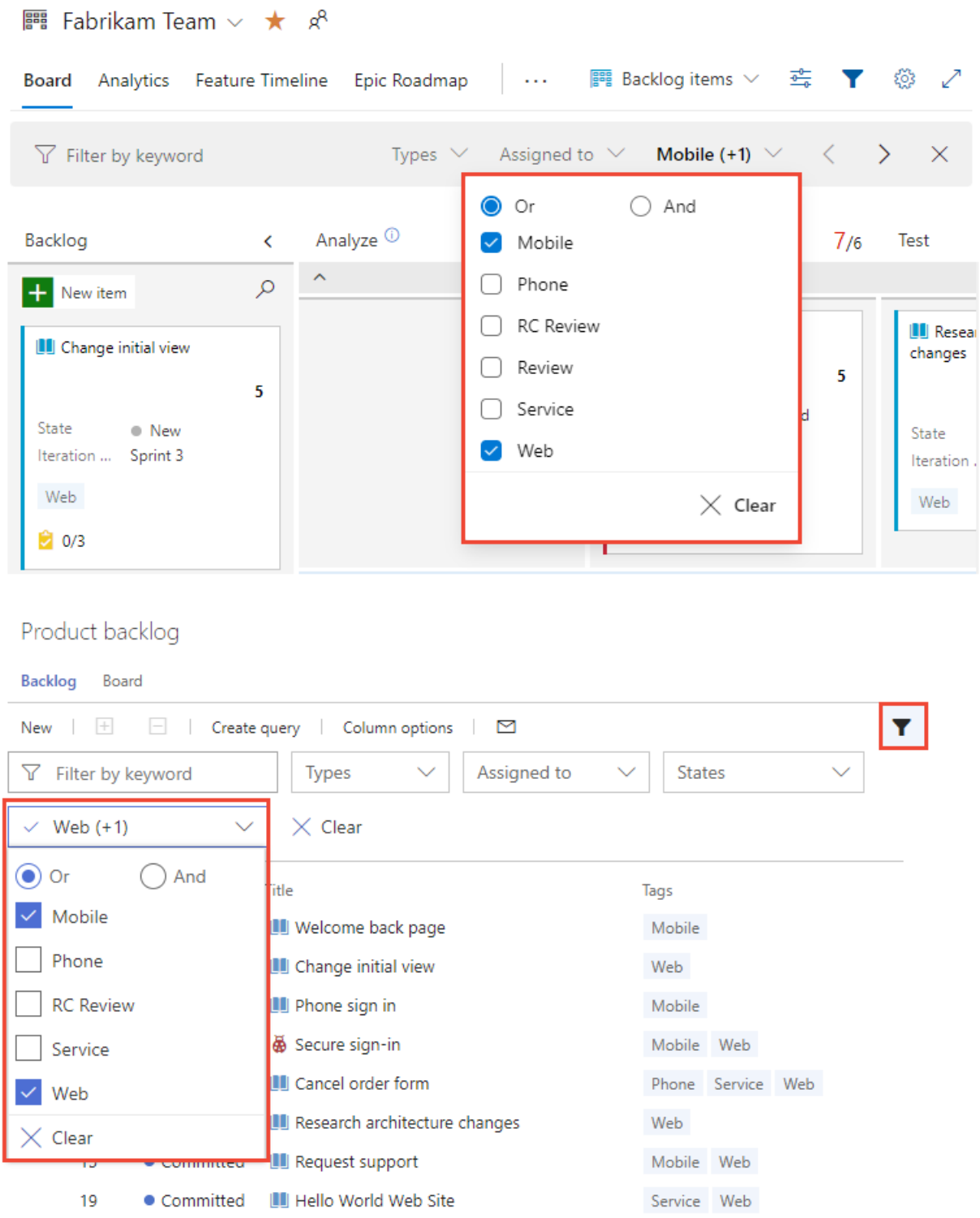


The final board displays just those stories linked as child work items to the selected features.

Filter based on tags

If you've added tags to your work items, you can filter your work using one or more tags. For backlogs and query results, add **Tags** as a column option before filtering on tags.

Check the boxes of those tags that you want to filter on. Keep the **OR** selection to do a logical OR for all the tags you selected. Or, choose the **AND** option to do a logical AND on all the selected tags.

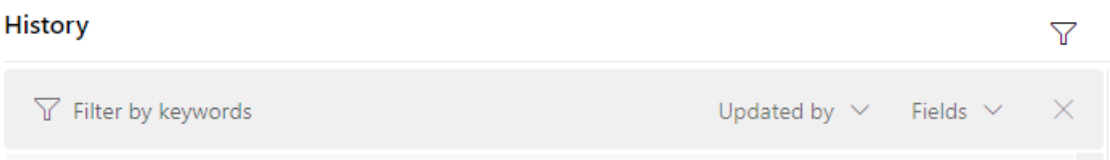


To learn more about tags, see Add tags to work items to categorize and filter lists and boards.

Filter the history view within a work item form

In addition to all the filter features described earlier in this article, you can also filter the history view within a work item form.

To quickly find revisions made that contain a keyword, or made by specific people or to a specific field, enable the filter feature by choosing **Toggle filter**.



To learn more, see Query work item history and discussion fields.

Related articles

- Set up your Backlogs and Boards
- About backlogs
- Change column options
- Display rollup progress or totals
- Customize cards
- Customize a sprint Taskboard
- Tags
- Query work items that you're following
- Reorder cards (Kanban Boards)

Add or edit a Delivery Plan

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022

Delivery Plans provide a highly interactive calendar view of multiple team backlogs. For the use cases, benefits, and interactions you can do, see Review team Delivery Plans.

Use this article to learn how to complete these tasks:

- Open a plan from the list of defined plans
- Add and edit a plan
- Add field criteria, customize cards, and add markers

> **NOTE**
> This article describes how to add or edit Delivery Plans 2.0 which is available for Azure DevOps Services and Azure DevOps Server 2022 and later versions. For information on the Delivery Plans Marketplace extension which supports Azure DevOps Server 2020 and earlier versions, see Delivery Plans 1.0.

Prerequisites

- To add or edit a Delivery Plan, you must be a member of the Contributors group for the project where you add the plan.
- To add team backlogs to a plan, you must have view permissions to those projects.
- To view a Delivery Plan, you must be a member of the Project Collection Valid Users group. Users granted **Stakeholder** access for a private project can view plans. Users granted **Stakeholder** access for a public project can add and view plans.
- To manage permissions for a Delivery Plan or edit or delete a plan, you must be the creator of the plan, a member of the Project Administrators or Project Collection Administrators group, or granted explicit permission through the plan's Security dialog. For details, see Edit or manage Delivery Plan permissions.

Before you define a plan

To add and configure a Delivery Plan, the following elements must be configured:

- Teams and team backlogs.
- Team product or portfolio backlog must be enabled.
- Area paths and team area paths assignments.
- Iteration (sprint) paths and team iterations
  - **Iteration Paths** must be assigned **Start** and **End Dates** or they won't appear on the plan.
  - Also, **Iteration Paths** must be selected for the team whose backlogs you select, or work items associated with those **Iteration Paths** won't appear on the plan.
- Teams have defined product backlog items, or portfolio backlogs and assigned those items to either a **Start Date** and **End Date** or an **Iteration Path**.

TIP

If you edit a plan and don't see the changes you made appear in the plan, refresh your browser. A browser refresh is needed some times to trigger the updates.

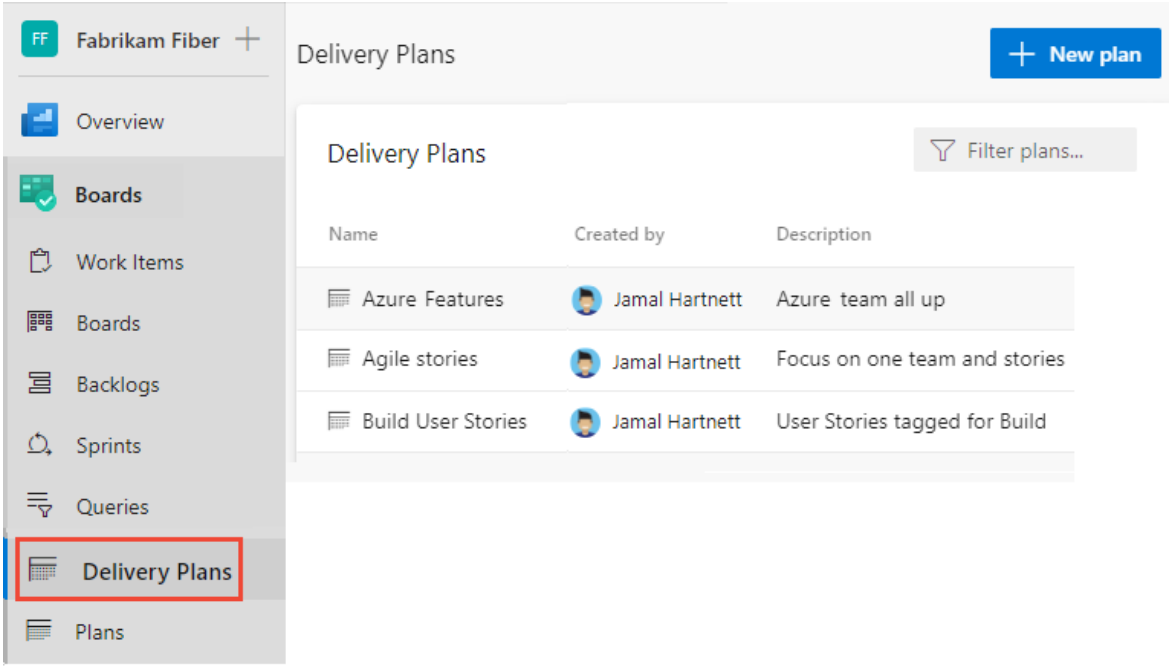
Plan customization options

Once you open the Plan settings dialog, choose one of the following tabs to set your customization options.

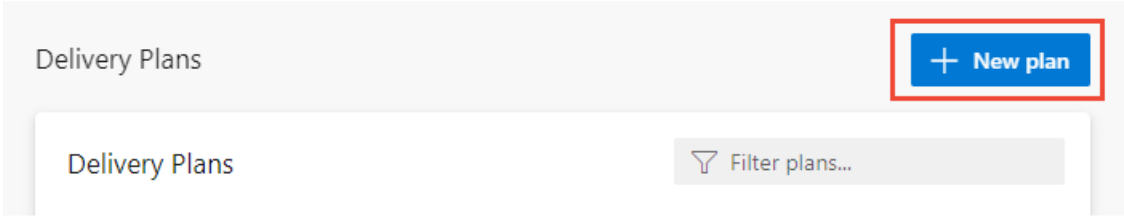
PAGE	USE TO...
Overview	Edit the plan **Name** or **Description**.
Teams	Add or remove a team backlog. You can add up to 20 backlog levels for Azure DevOps Services or 15 backlog levels for Azure DevOps Server 2022. You can add a mix of backlog levels and teams from any project defined for the organization.
Field criteria	Specify field criteria to filter work item types displayed on the plan. All criteria is evaluated as an AND statement. If no fields are specified, then all work item types that appear on the teams backlog level appear on the delivery plan.
Markers	Add up to 30 milestone markers to the plan. Specify a label and select a color.
Fields	Add or remove fields from cards to display on the plan, similar to how you customize them for your Kanban board. You can't add rich-text (HTML) fields, such as the Description field, to a card even if it appears in the list. These field types represent too many challenges to format on a card.
Styles	Add styling rules to change card color based on field criteria.
Tag colors	Add tags and specify a tag color. Optionally enable or disable a tag color.

Add a plan

1. Open **Boards > Delivery Plans**.

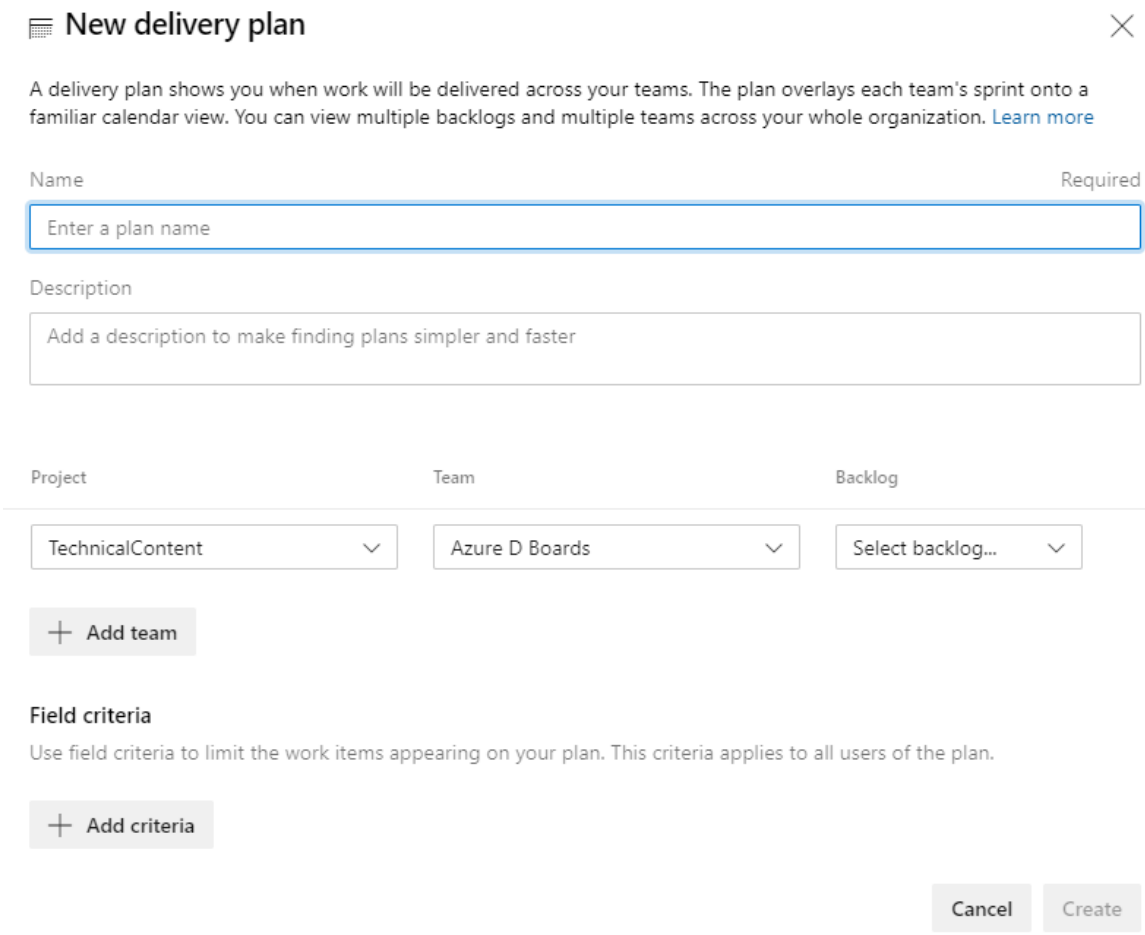


2. To add a plan, choose **New Plan**.



All users have permissions to create a plan and manage the plans they create.

3. Fill in the form to name, describe, and specify the team backlogs that you want to appear within your plan. You can add up to 15 teams to a plan.



When defining a plan, note the following information:

- Use the name and description field to clearly identify your plan within the project
- You can choose one or more teams from any project defined in the organization or collection. There can be up to a maximum of 15 teams
- You can choose one or more active backlogs for a team

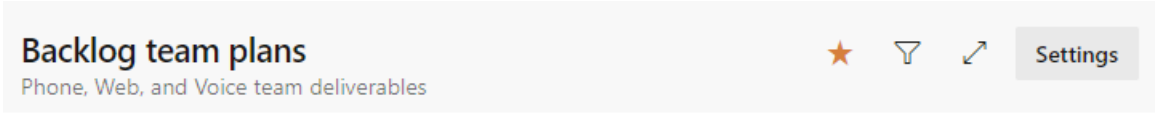
> **NOTE**
> If you aren't able to select a backlog level, check the Team Backlog settings to ensure the backlog level is enabled for the team. To learn more, see Select backlog navigation levels for your team.

- You can reorder the team backlogs by dragging and dropping them into the sequence you want
- To filter for specific work items, specify the field criteria. For example, to exclude bugs from the view, add the following criteria: `Work Item Type <> Bug`.

Edit a plan

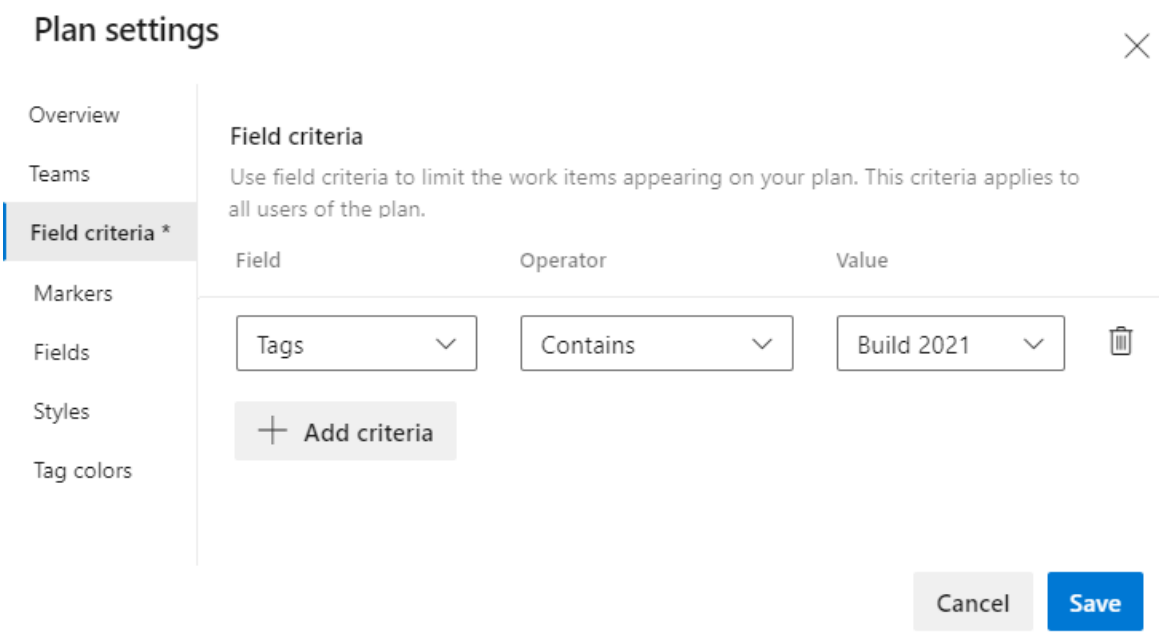
Once you've defined a plan, you can modify it or further customize it.

1. Choose the **Settings** to open the **Plans settings** dialog.



2. Then, choose the page you want to edit based on the customizations you want to make. Here, we add the **Tags** to the **Field criteria**. Only work items that contain the *RC Review* tag will appear in the Delivery

Plan.



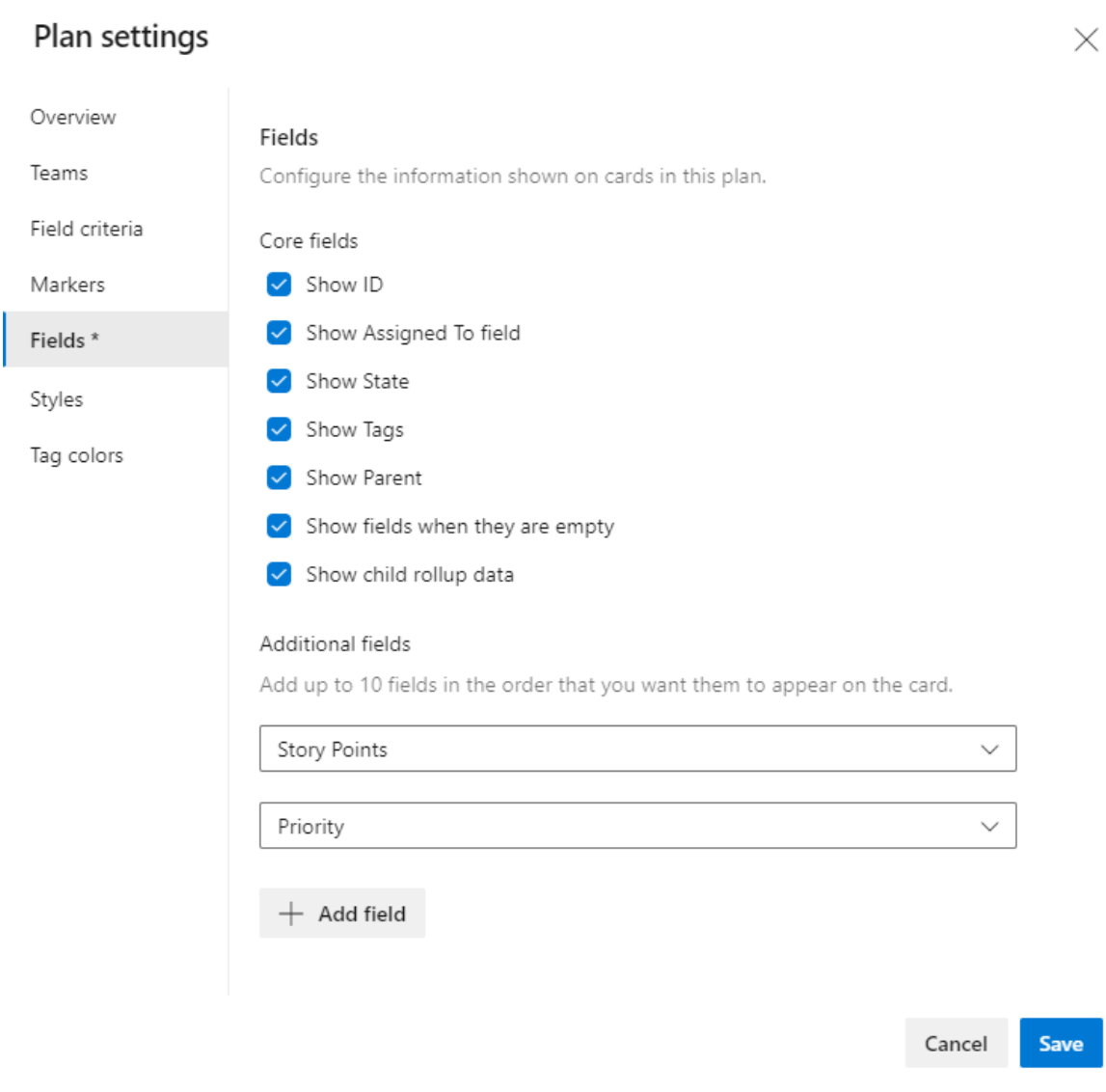
Choose fields to appear on cards

Show those fields that are useful for your review purposes or if they contain keywords that you may want to use to filter your plan. Unlike the Kanban board, you can't change the field displayed on the card. First, open the work item to make field changes.

> **TIP**
> To add a custom field, you must first add it to the process used to customize the project.

1. From the Plan settings dialog, choose the **Fields** tab. Place a check mark in the check box for those fields you want to have appear on the board.
2. To add a field, choose the + plus icon and enter the name of a field you want to add. You can add both default and custom fields, including Boolean fields. The only fields you can't add are rich-text or HTML fields.

   Here we select all standard fields and add the **Story Points** and **Priority** fields to display on cards.



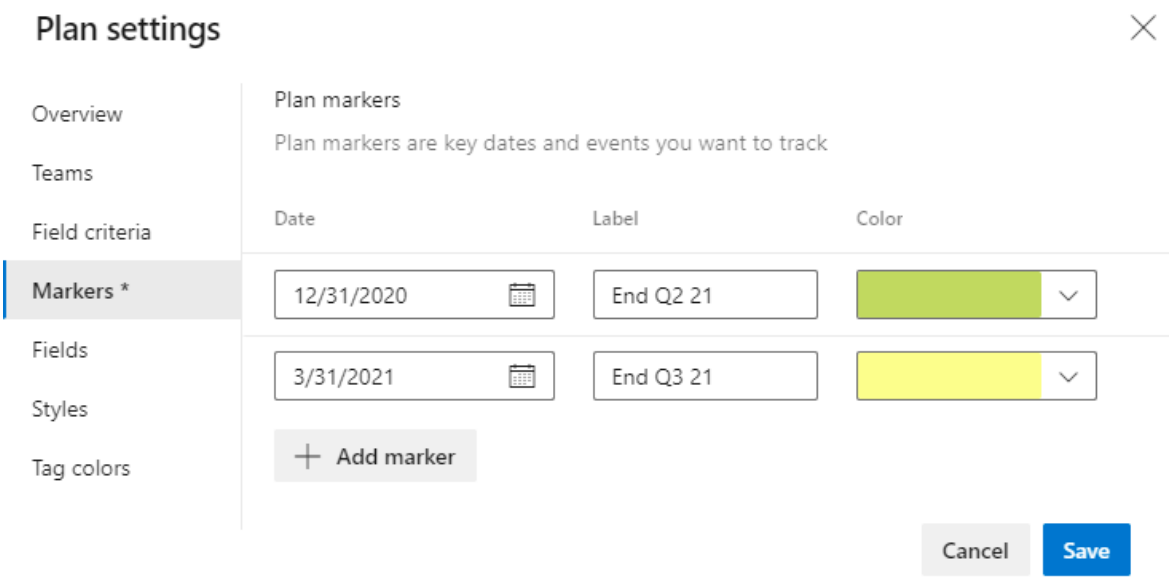
TIP

To show the **Title** of the parent work item, choose the **Parent** field. Choosing the **Parent** title from a card opens the parent work item. To change the parent work item, open the child work item and remove the link and add a different parent work item. You can filter your plan based on parent work items, whether the **Parent** field is added to cards or not.

3. To remove a field, choose the delete icon next to the field.
4. When done with your changes, choose **Save**.

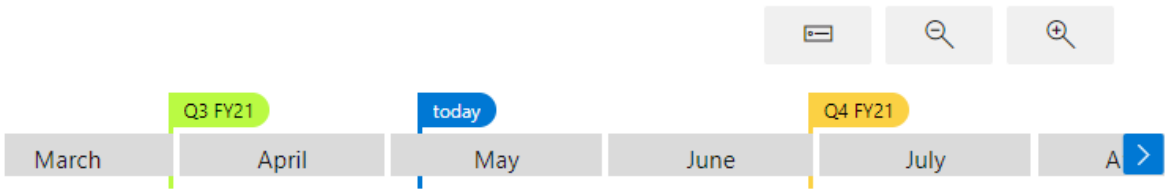
Add milestone markers

1. To set a marker, open **Markers**, specify a date and specify a hexadecimal color, or choose the color palette icon to change to a new color selected by the system.



2. To add additional markers, choose **+ Add marker**. You can add up to 30 markers. The **+ Add marker** button becomes disabled after 30 markers have been added.
3. Choose **Save** when done.

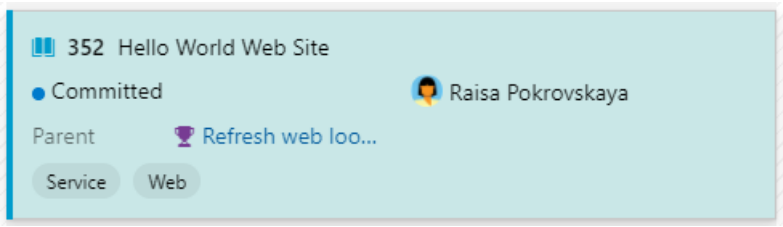
   Markers appear on the plan as shown.



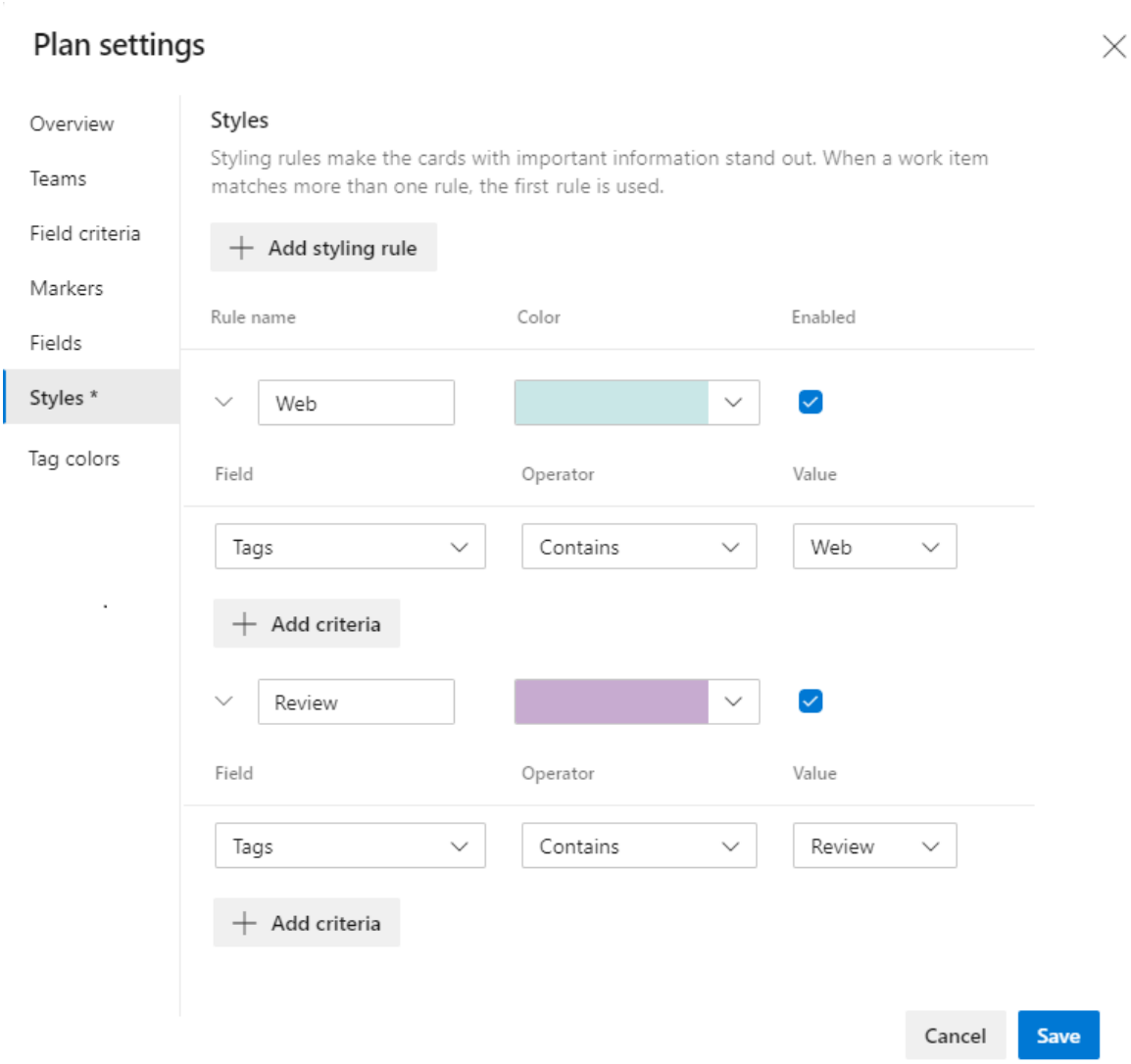
4. When done with your changes, choose **Save**.

Change card color

With styling rules, you can cause cards to change color when their corresponding work items meet the field criteria that you set. This feature is similar to the one you can define for Kanban boards as described in Customize cards. Here we highlight the card based on its **Tags** assignment.

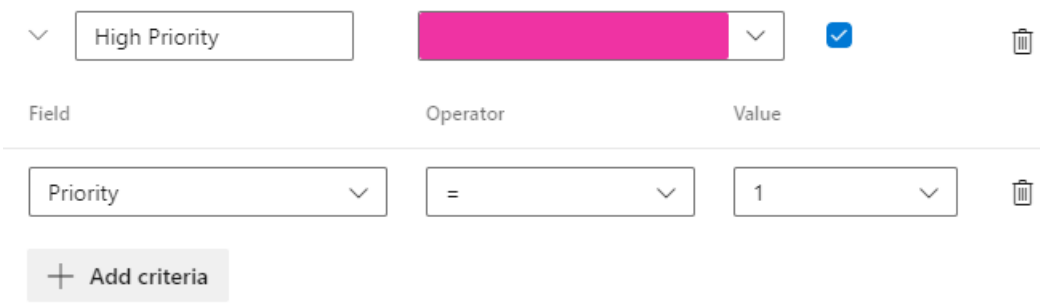


1. To change the card color, open the **Styles** tab. You can specify up to 10 styles. There are some limits to the fields you choose.



2. Choose **+Add styling rule**. Enter a name for the style and choose the color from the color picker. Then specify the field criteria. You can add multiple field values. For style purposes, they're all evaluated as a logical AND. Choose the field and the value for the field.

   For example, here we choose to highlight cards with a **Priority=1**.

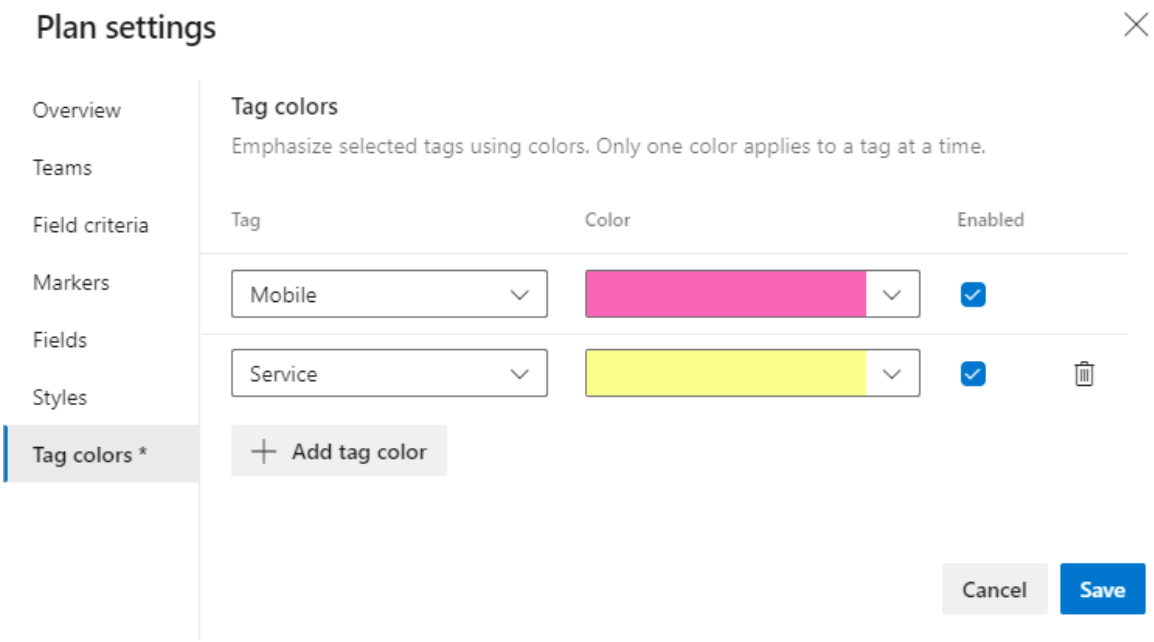


> **NOTE**
>
> Some fields aren't supported for selection, such as the **Title** field, **Description** and other rich-text fields, **Assigned To** and other identity fields. Also, you may be able to select a field but not be able to specify a value or the value you want. For example, you can't specify **Tags** that are *Empty* or *Not Empty*.

Set tag colors

Before setting tag colors, first add tags to backlog items that you want to highlight with color.

1. From the Plan settings dialog, choose **Tag colors** and then choose **Add tag color**. Then, select the tag and the color you want to appear on the cards.



2. To enable or disable a tag color, select the **Enabled** checkbox.
3. When done with your changes, choose **Save**.

> **TIP**
> If tags don't display on the cards, choose **Fields** and make sure that you've checked **Show Tags**.

Programmatically manage Delivery Plans

You can manage plans using the REST API, Plans.

Related articles

- Review team plans
- Interactively filter backlogs, boards, queries, and plans
- Edit Delivery Plan permissions
- Backlogs, boards, and plans
- Add teams
- Portfolio management
- Manage teams and configure team tools

Manage Delivery Plan permissions in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can control who has access to a Delivery Plan by setting its permissions. You can grant or restrict access to users and groups to delete, edit, view, or manage permissions of delivery plans.

By default all members of an organization or project collection can view Delivery Plans, including users with Stakeholder access for private projects. The plan creator and the project and collection administrators, can edit or delete a plan, or change the plan's permissions. To learn more about Delivery Plans, see Review team delivery plans.

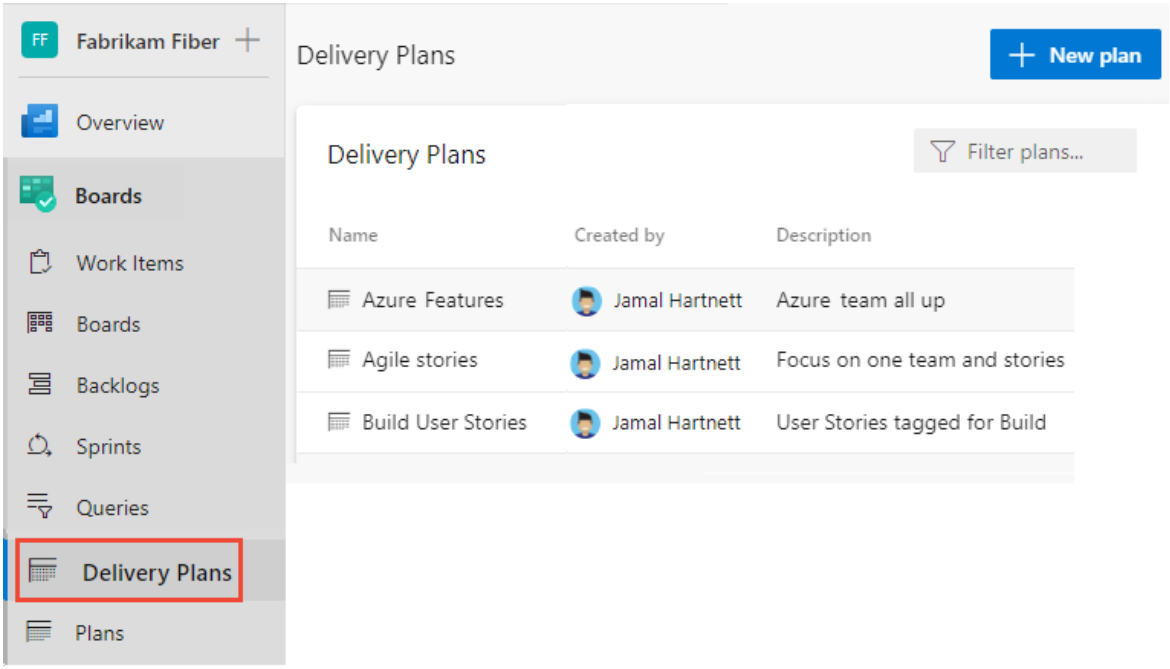
By default all members of an organization or project collection can view Delivery Plans, except users with Stakeholder access for private projects. The plan creator and project and collection administrators, can edit or delete a plan, or change the plan's permissions. To learn more about Delivery Plans, see Delivery Plans 1.0.

Prerequisites

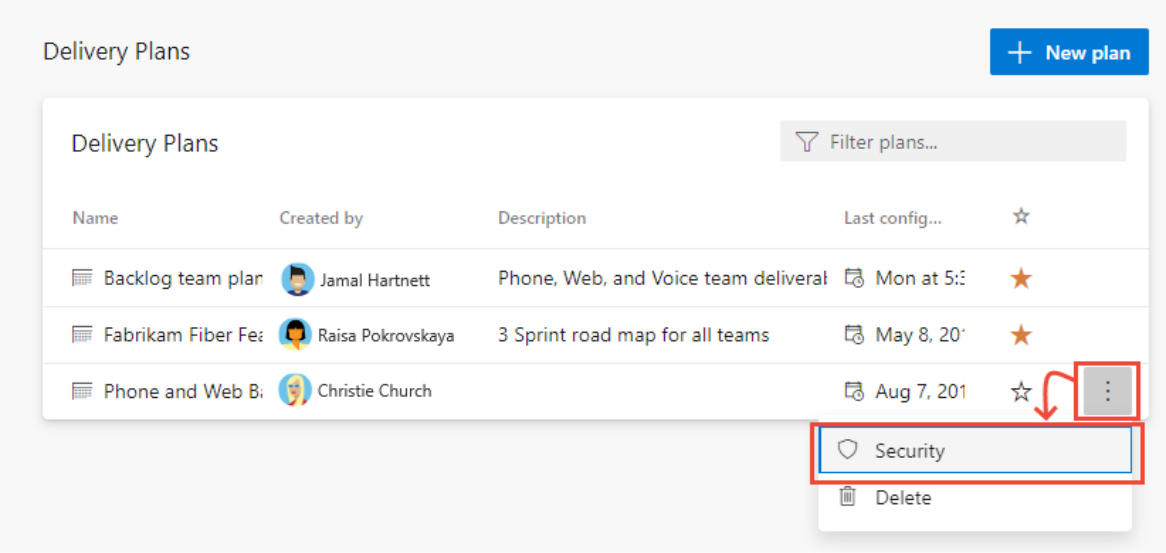
- To edit the permissions for a Delivery Plan, you must be the creator of the plan, a member of the Project Administrators or Project Collection Administrators group, or granted explicit permission through the plan's **Security** dialog.

Edit permissions for a Delivery Plan

1. Open **Boards > Delivery Plans**.



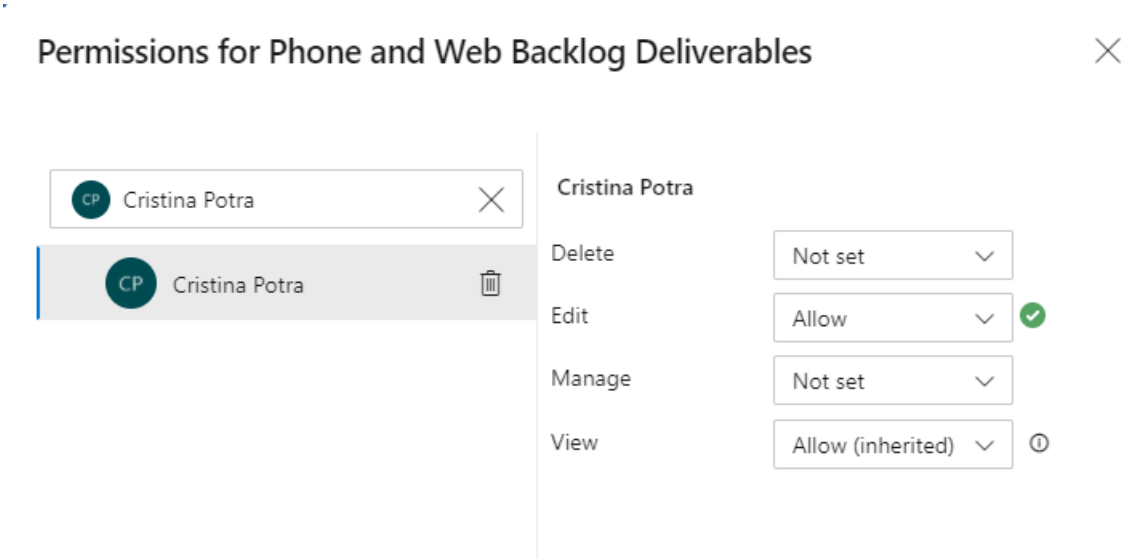
2. To grant permissions to a group or user to manage or edit a specific plan, choose ⋮ **More options** to open the **Security** dialog for the plan.



3. Add a user, team group, or other security group who you want to grant permissions to or restrict access. For details, see Change project-level permissions. By default, non-administrators can't delete or edit a plan.

4. With the user or group selected, set the permission you want them to have to **Allow**. **Manage** set to **Allow** enables the user to manage permissions for the plan.

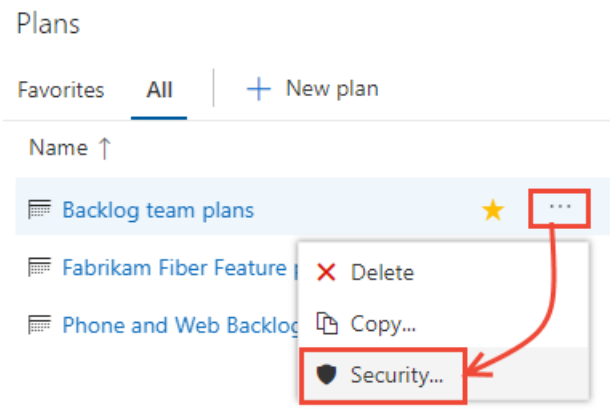
   For example, here we grant permission to Cristina to edit the plan.



5. When done, close the dialog. The changes are automatically saved.

1. Open **Boards > Plans**. For details, see Review team delivery plans.

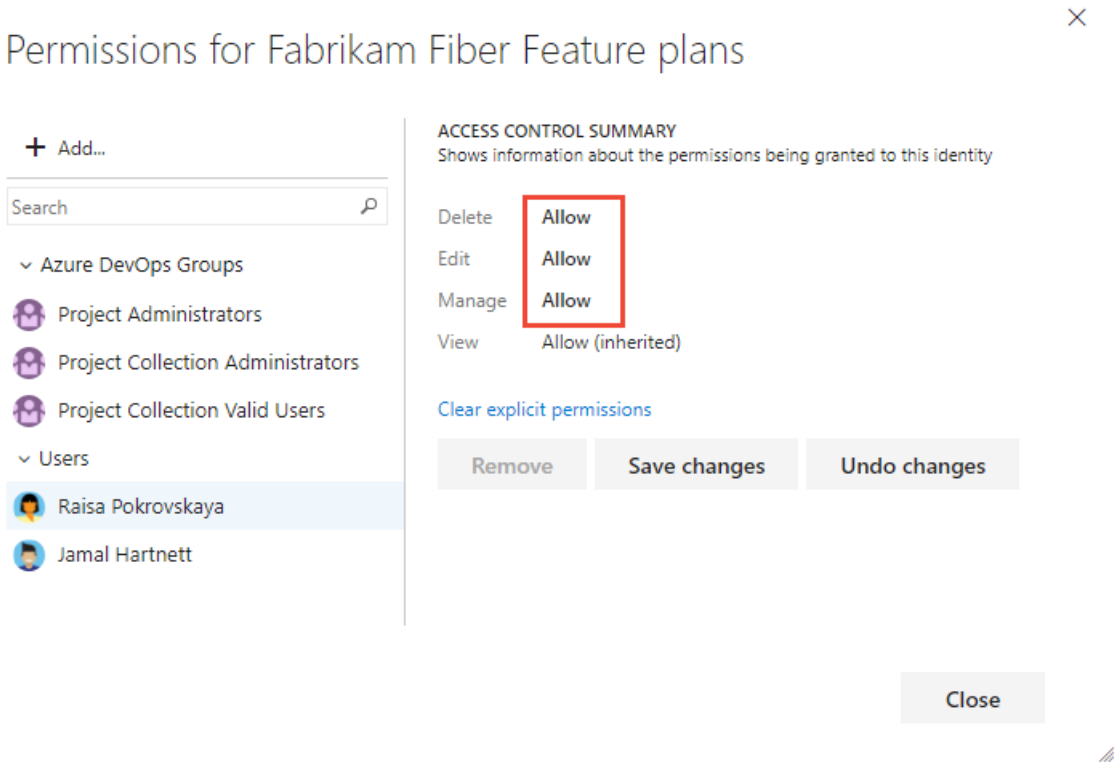
2. To grant permissions to a group or user to manage or edit a specific plan, choose the ··· actions icon to open the **Security** dialog for the plan.



3. Add a user, team group, or other security group who you want to grant permissions to or restrict access. For details, see Change project-level permissions. By default, non-administrators can't delete or edit a plan.

4. With the user or group selected, set the permission you want them to have to **Allow**. **Manage** set to **Allow** enables the user to manage permissions for the plan.

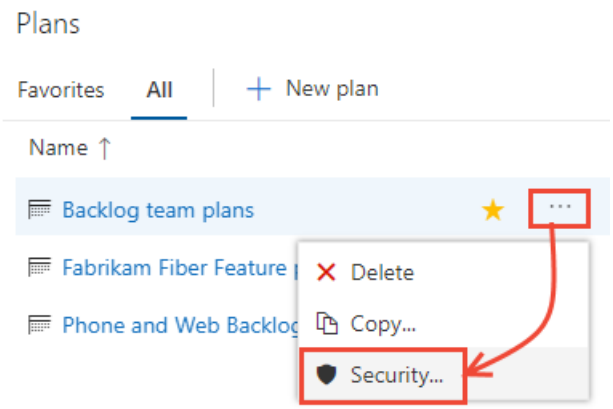
   For example, here we grant permission to Raisa to edit the plan.



5. Choose **Save changes** when done.

1. Open **Work>Plans**. For details, see Review team delivery plans.

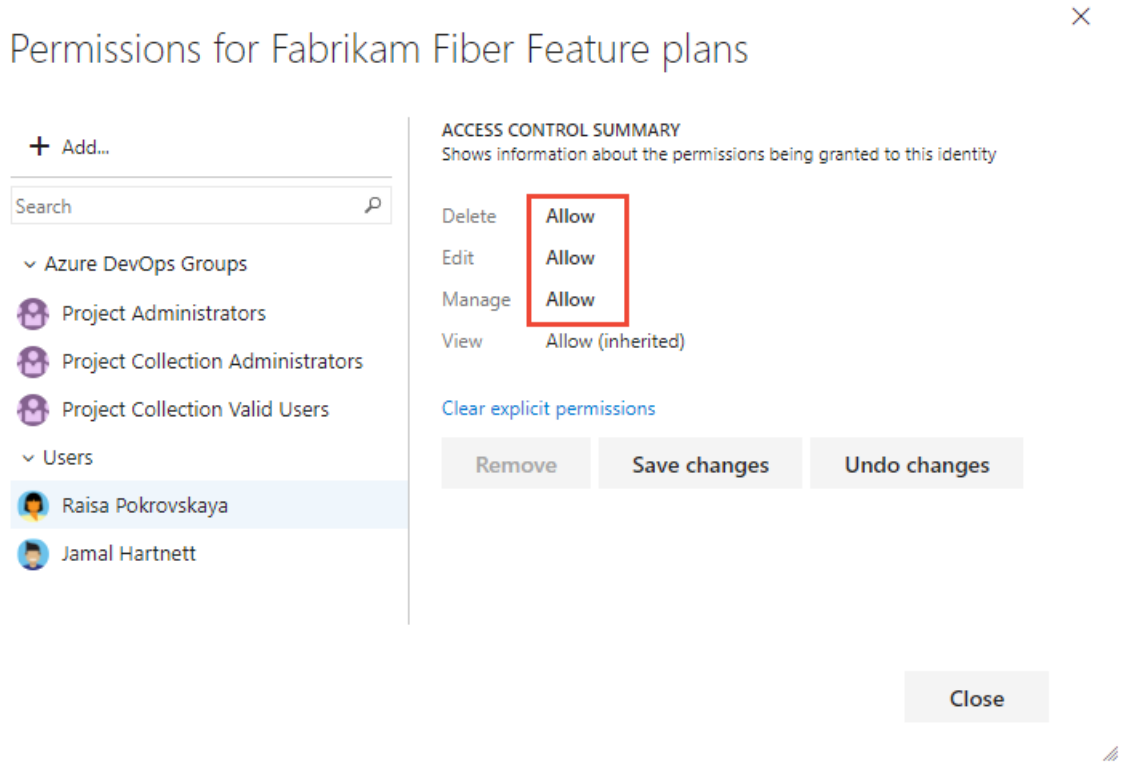
2. To grant permissions to a group or user to manage or edit a specific plan, choose the ··· actions icon to open the **Security** dialog for the plan.



3. Add a user, team group, or other security group who you want to grant permissions to or restrict access. (For details, see Change project-level permissions). By default, non-administrators can't delete or edit a plan.

4. With the user or group selected, set the permission you want them to have to **Allow**.

   For example, here we grant permission to Raisa to edit the plan.



5. Choose **Save changes** when done.

Related articles

- Review team delivery plans
- About permissions and inheritance
- Stakeholder access quick reference

Implement Scaled Agile Framework® in Azure Boards

10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Many enterprises benefit from individual Agile teams. Greater interest grows to scale Agile practices as the organization grows. The need for enterprises to view progress of many Agile teams and across a portfolio continues to increase. To address these needs, many businesses have adopted the Scaled Agile Framework® (SAFe®).

If you're familiar with Scrum but not familiar with SAFe®, these videos at Scaled Agile are a good way to orient yourself.

Azure Boards supports SAFe® practices through its autonomous teams, backlogs, boards, reports, and metrics. This article introduces you to how Azure Boards artifacts support SAFe practices and artifacts.

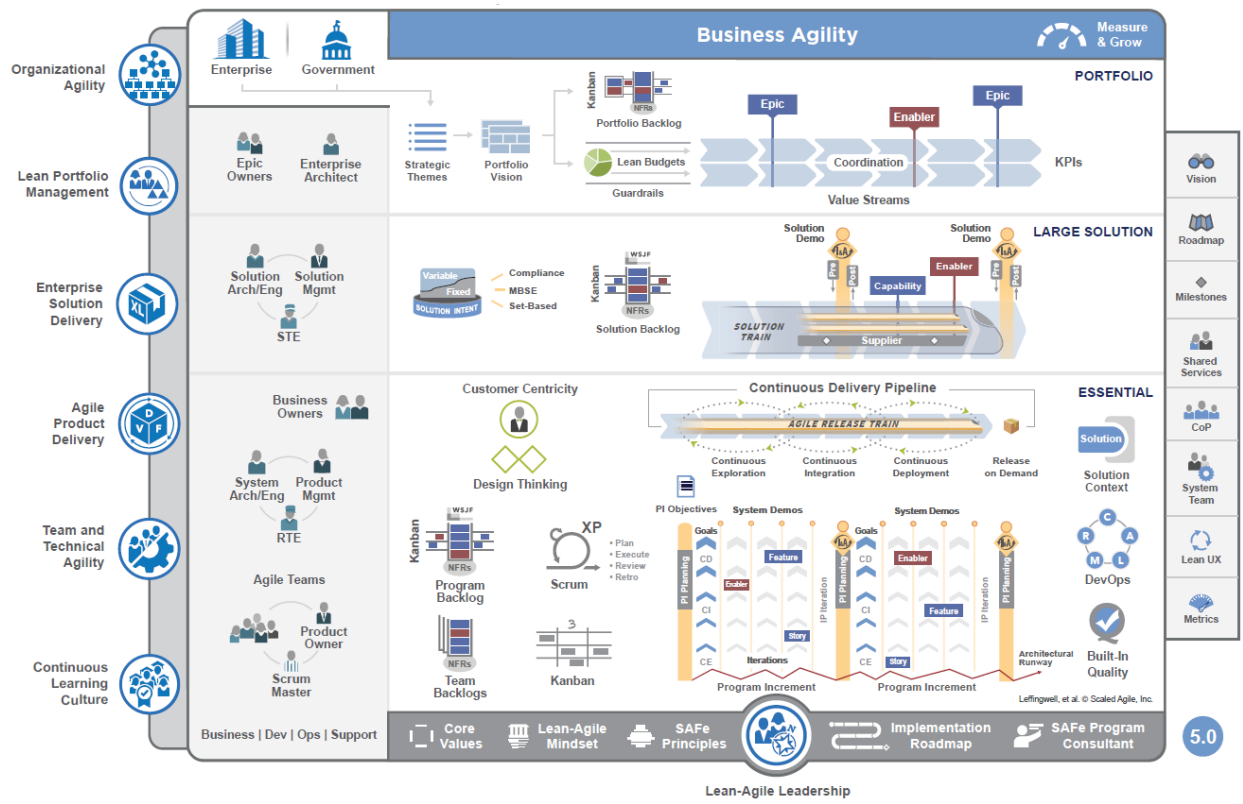
- The Scaled Agile Framework®
- Essential SAFe®
- Portfolio SAFe®
- Large Solution SAFe®
- Quick reference mapping
- Azure Boards implementation of SAFe®

> **NOTE**
> This article is one of a set of Scaled Agile Framework® tutorials that applies to Azure Boards and Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features and procedures are specific to the cloud or the latest version of Azure DevOps Server.

Scaled Agile Framework®

The SAFe® addresses how a portfolio vision is met by a hierarchy of teams, all concerned with specific goals. This framework breaks down Epics into Features and Stories. Teams work on these items in Sprints and deliver through Program Increments (PIs) and Release Trains. Furthermore, the portfolio backlog can track deliverables that map to value streams and associated budgets.

SAFe® architectural overview version 5.0

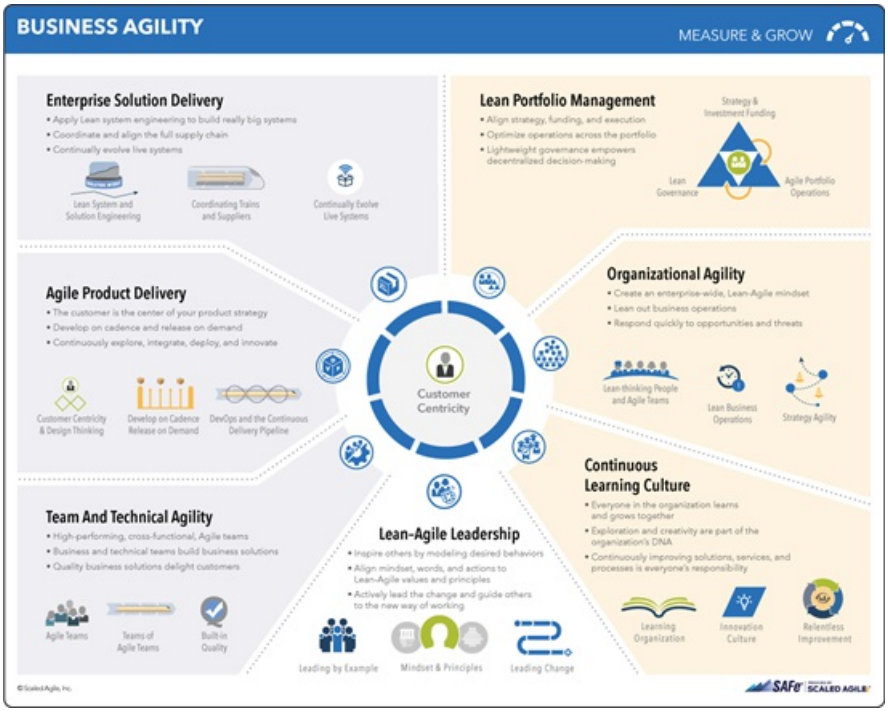


Reproduced with permission from © 2011-2020 Scaled Agile Inc.. All rights reserved.

SAFe® and Scaled Agile Framework are registered trademarks of Scaled Agile Inc.

SAFe® 5.0 Business Agility

Many SAFe® practices include growing a culture that supports agility, alignment, and autonomy, all while being customer-centric.



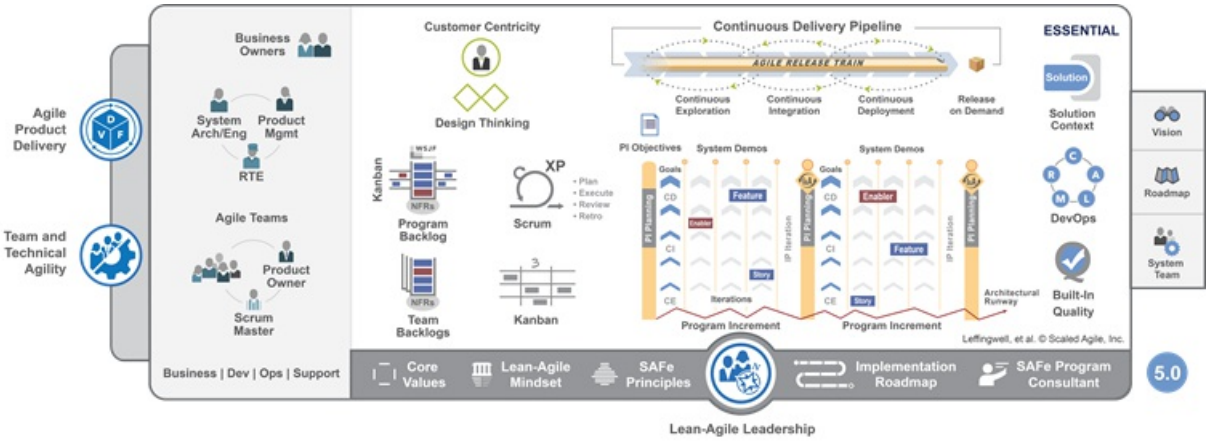
Reproduced with permission from © 2011-2020 Scaled Agile Inc.. All rights reserved.

Some of the ways Azure Boards supports business agility and agile culture are discussed in the following articles:

- Agile culture
- Practices that scale

Essential SAFe®

Essential SAFe® requires support for the artifacts and practices illustrated in the following poster.



Reproduced with permission from © 2011-2020 Scaled Agile Inc.. All rights reserved.

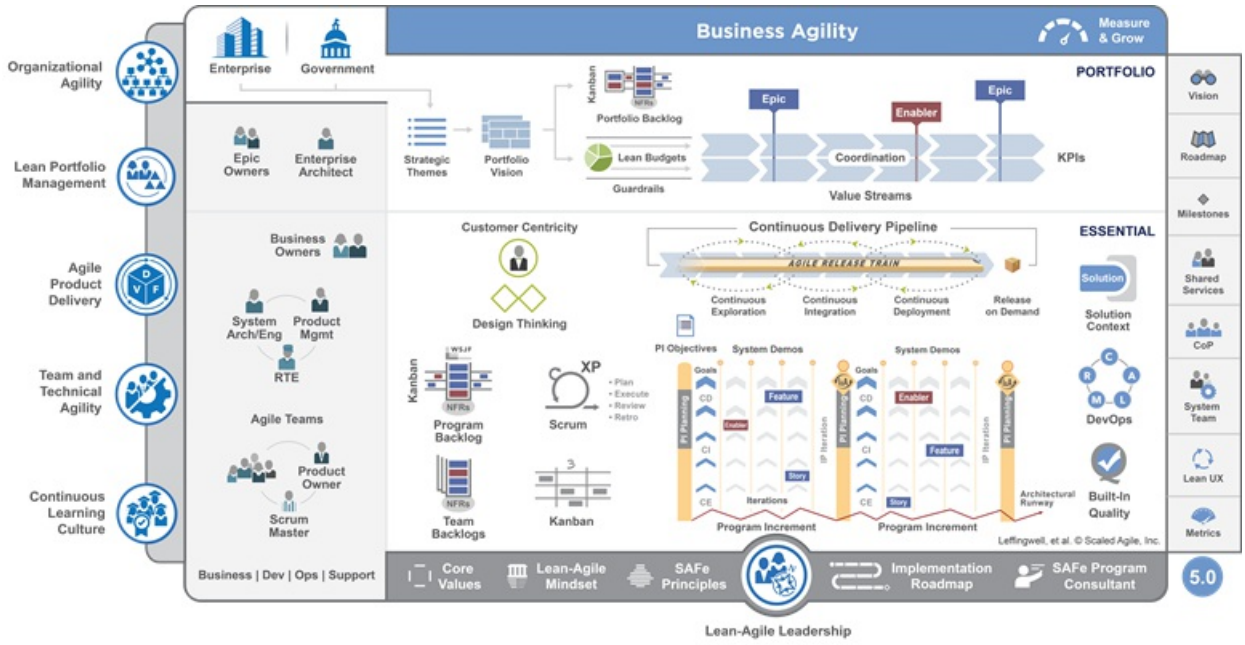
All of these artifacts and practices are supported by Azure Boards.

- **Stories, Features**, and **Enablers**: Implemented as work items that capture information and status of work. These work items automatically appear on team backlogs and Kanban boards.
- **Team Backlogs** and **Program Backlogs**: Implemented as team backlogs that filter work items assigned to a team and support prioritizing and grouping of work.
- **Scrum** and **Kanban**: Practices that are fully supported using Kanban boards, Sprint backlogs and Taskboards, teams, and sprint cadences.
- **Iterations**, **Innovation and Planning (IP) Iteration**, **Program Increments (PI)**, **Milestones**, and **Release Trains**: Implemented via a flat-list or a hierarchical configuration of Iteration Paths.
- **Agile Release Train**: Implemented by a set of Agile teams and Program teams configured to support specific team and program views.
- **PI Objectives**, **Team Goals**, and **Solution context**: Teams can use the built-in project wiki to share objectives, goals, customer information, and solution requirements.

For an overview of how Azure Boards implements Scrum and Kanban, see About Sprints, Scrum, and project management and About Boards and Kanban.

Portfolio SAFe®

Portfolio SAFe® adds support for managing portfolios through epics, enablers, and value streams.



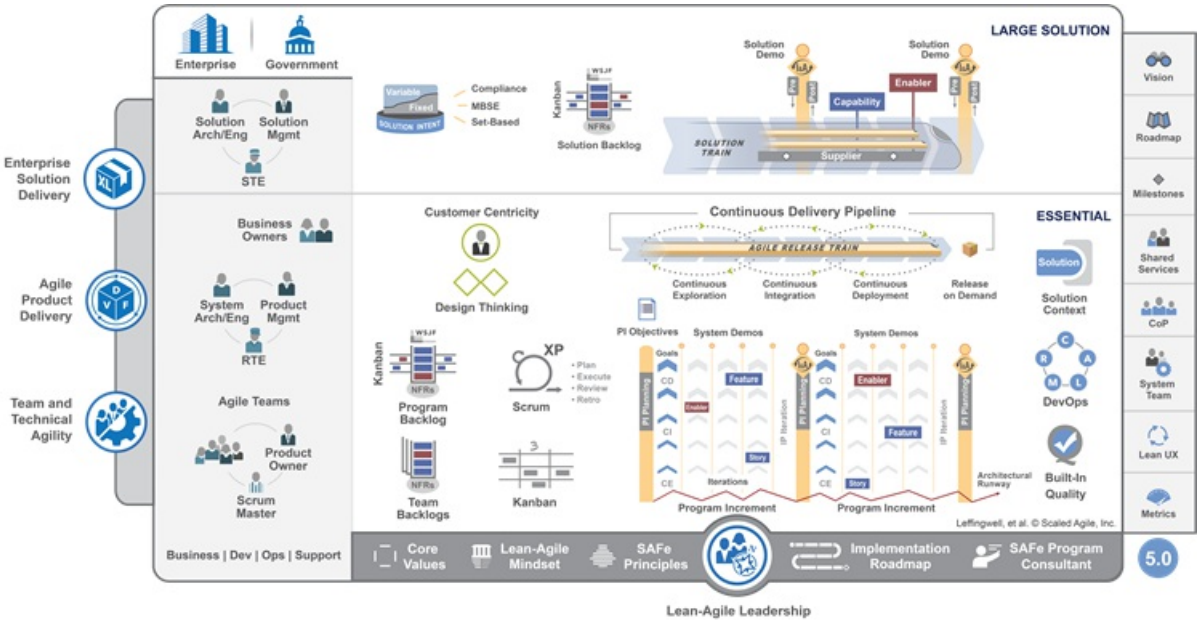
Reproduced with permission from © 2011-2020 Scaled Agile Inc.. All rights reserved.

Azure Boards provides supports for the following portfolio components:

- **Epics**: Map to the Epic work item type and allow tracking, grouping, and rollup of child items.
- **Portfolio backlogs**: Implemented as a portfolio backlog that supports filtering of work based on review of business needs.
- **Portfolio Vision** and **Strategic Themes**: Business owners and portfolio managers can use the built-in project wiki to share their vision, objectives, and goals.
- **Value Streams**: Value streams can be tracked using tags or custom fields.
- **Lean budgets**: Budget information can be captured in custom fields and rolled up to gain visibility to the Feature and Epic levels.
- **KPIs**: Several reports and dashboard widgets provide out-of-the box metrics. Power BI and the Analytics service provide support to create custom reports quickly.

Large Solution SAFe®

Large Solutions SAFe® includes support for a Solution Backlog, Solution Trains, and Capabilities.

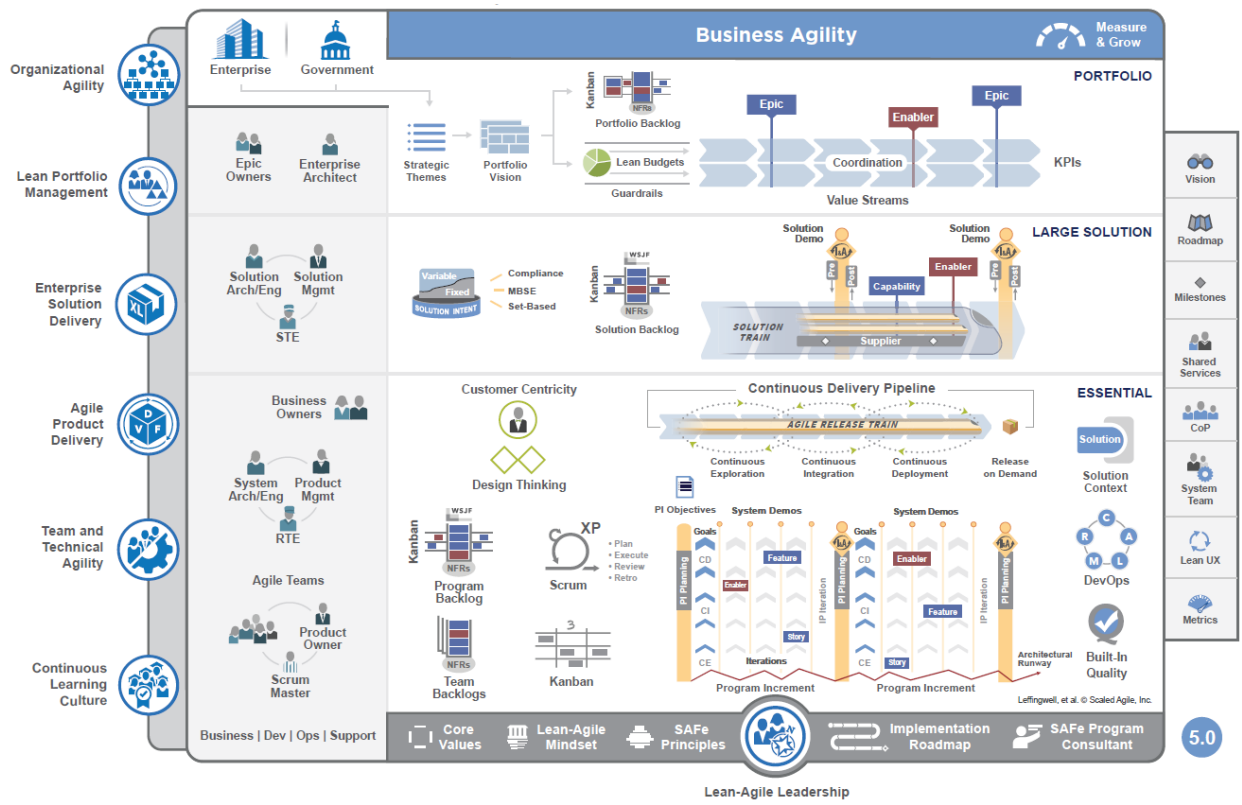


Reproduced with permission from © 2011-2020 Scaled Agile Inc.. All rights reserved.

You can implement large solutions in much the same way as you implement Portfolio SAFe®. However, you can also add custom work item types and custom backlogs to support other solution requirements.

Full SAFe®

Full SAFe® includes the three levels of Essential SAFe®, Large Solution SAFe®, and Portfolio SAFe®.



How SAFe® artifacts map to Azure Boards

The following table maps SAFe® terms or artifacts to the equivalent Azure Boards term or artifact. Choose the link to learn about implementation details.

SAFe® term or artifact	Azure Boards term or artifact
Agile teams	**Teams**. You define a hierarchy of teams to meet the needs of feature or development teams, program and portfolio teams, or solution train teams.
Agile Release Train (ART)	**Teams**. Agile teams manage the work of deliverables for a set of Features. Each Agile team has a set of Agile tools to support the flow of work and review progress and deliverables.
Budgets	**Tags, Value Area**. You can use tags or the *Value Area* field to track work associated with a specific budget or value stream.
Capabilities	**Work item**. You define, plan, and track Capabilities similar to Epics and Features. You capture them in work items and within various team backlogs.
Enablers	

Work item. You define, plan, and track Enablers similar to Epics, Features, and Stories. You capture them in work items and within various team backlogs.

Epics

Epic work item. You define an Epic using the Epic work item type. Epics are at the top of the work item hierarchy of Epics, Features, and Stories.

Features

Feature work item. You define a Feature using the Feature work item type. Features are a container for many Stories and are represented in their own portfolio backlog.

Innovation and Planning (IP) Iteration

Iteration Path. You define Iteration Paths for a project and set their start and end dates. Each team subscribes to the iterations they work with.

Iteration

Iteration Path. You define Iteration Paths for a project and set their start and end dates. Each team subscribes to the iterations they work with.

Milestones

Milestones and key events. Milestones occur at the end of each iteration. Custom fields and tags can also be used to associate work with milestones and key events.

Portfolio Backlog

Portfolio backlog. A portfolio backlog lists the Epics associated with a portfolio with the option to expand and display the child Features and Stories.

Portfolio Kanban

Portfolio Epics board. The Portfolio team's board displays the Epics backlog as cards in an interactive, configurable, and filterable Kanban board.

Portfolio Vision

Wiki. Use the project wiki to share broadly within the organization information related to strategy, solutions, and how teams will collaborate to produce portfolio and program deliverables.

Program Backlog

Feature backlog. A Feature backlog lists the Features associated with a program with the option to expand and display the child Stories.

Program Kanban

Program Features board. The Program board displays the Features backlog as cards in an interactive, configurable, and filterable Kanban board.

Program Increment (PI) Iteration Path

Iteration Path. Iteration Paths define a time box for a project with start and end dates. Iteration Paths can be defined from one week to 12 weeks or longer.

Retrospectives and reviews

Retrospectives. Each team can add a board to capture, prioritize, and create action items to support their improvement processes.

Roadmap

Delivery Plans, Feature Timeline. Azure Boards provides configurable and interactive views to review roadmaps and team deliverables.

Shared Services

Shared services team structure: Resources that are shared across teams can be represented through their own Agile feature team. Each can manage their backlog while having their work also appear in the backlogs of the teams they support.

Solutions

Solutions: Solutions can be represented through a custom Solution work item type.

Solution Backlog

Solution portfolio backlog. You can define a custom work item type and portfolio backlog to capture special business requirements of large solutions, or use Epics and Epic portfolio backlogs to capture solutions.

Strategic Themes

Wiki. Strategic Themes, similar to Portfolio Vision, can be captured in a project wiki.

Stories

User Story work item. User Stories capture the functionality you want to be delivered. They're typically sized so as to be completed with a single iteration.

Team Backlog

Stories backlog. The Stories backlog lists the User Stories assigned to the area path associated with the team.

Team Kanban

Stories board. The Stories board displays the Stories backlog as cards in an interactive, configurable, and filterable Kanban board.

Value Streams

Tags, Value Area. You can use tags or the Value Area field to track work associated with a specific budget or value stream.

Azure Boards implementation of SAFe®

Each of the following articles within this suite of tutorials provide details on how you can configure, customize, and use Azure Boards to implement your SAFe® programs and projects.

- How SAFe® concepts map to Azure Boards artifacts
- Configure Azure Boards to support SAFe®
- Customize Azure Boards to support SAFe®
- Plan and track SAFe® programs and portfolios
- View SAFe® progress, roadmaps, and metrics

Try this next

How SAFe® concepts map to Azure Boards artifacts

Related articles

- Scale Agile to Large Teams
- Agile culture
- Practices that scale
- About Sprints, Scrum and project management
- About Boards and Kanban
- Scaled Agile Framework: SAFe® resource site.
- SAFe® in 7 minutes: video from Inbar Oren, the Lean Samurai.
- Scaling Agile and SAFe® Metrics with TFS: Blog post that illustrates a SQL Server report developed by InCycle to illustrate how TFS can be used to support scaled agile or SAFe.

About the authors

Many thanks to the following contributors for their review and feedback to the current content.

- Phillip Eng is a Senior Architect at Microsoft, Digital Pursuit and Guidance.
- Hosam Kamel is a technology solution professional for Microsoft and ALM Ranger.
- Willy-Peter Schaub is a former program manager with the Visual Studio ALM Rangers at the Microsoft Canada Development Center. You can follow Willy-Peter on Twitter at twitter.com/wpschaub.

The articles in this series were updated from a previous white paper developed in collaboration with the following authors:

- Gordon Beeming is a Software Developer at Derivco in the sunny city of Durban, South Africa. He spends most his time hacking away at the keyboard in Visual Studio or with his family relaxing. His blog is at gordonbeeming.xyz and you can follow him on Twitter at twitter.com/gordonbeeming.
- Brian Blackman is a principal consultant with Microsoft Premier Developer, focusing on affecting ISV partners and Enterprises success in engineering and the marketplace. He has an MBA, and is a CSM, CSP, MCSD (C++), and MCTS and is a Visual Studio ALM Ranger. When he's not Ruck Mastering and contributing to Visual Studio ALM Ranger projects, he spends his time writing code, creating and delivering workshops, and consulting in various concentrations, especially helping organizations in their quest for business agility.
- Gregg Boer is a principal program manager at Microsoft. Gregg is the product owner for the Agile management experience provided by Azure DevOps and on-premises TFS.
- Kathryn Elliott is a senior technical writer at Microsoft.
- Susan Ferrell is a senior technical writer and a Visual Studio ALM Ranger.
- Willy-Peter Schaub is a former program manager with the Visual Studio ALM Rangers at the Microsoft Canada Development Center. Since the mid-'80s, he has been striving for simplicity and maintainability in software engineering. You can follow him on Twitter at twitter.com/wpschaub.
- Special thanks to the following technical experts for reviewing this article: Mike Douglas (independent consultant, ALM Ranger), Richard Hundhausen (independent consultant, ALM Ranger) and Bill Heys (independent consultant, ALM Ranger).

How SAFe® concepts map to Azure Boards artifacts

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

If you're interested in using Scaled Agile Framework (SAFe®), you can configure your Azure Boards project to track SAFe® deliverables. Just as Azure Boards supports Scrum and Agile practices, it can support SAFe® and large numbers of teams to work together on Epics that span releases.

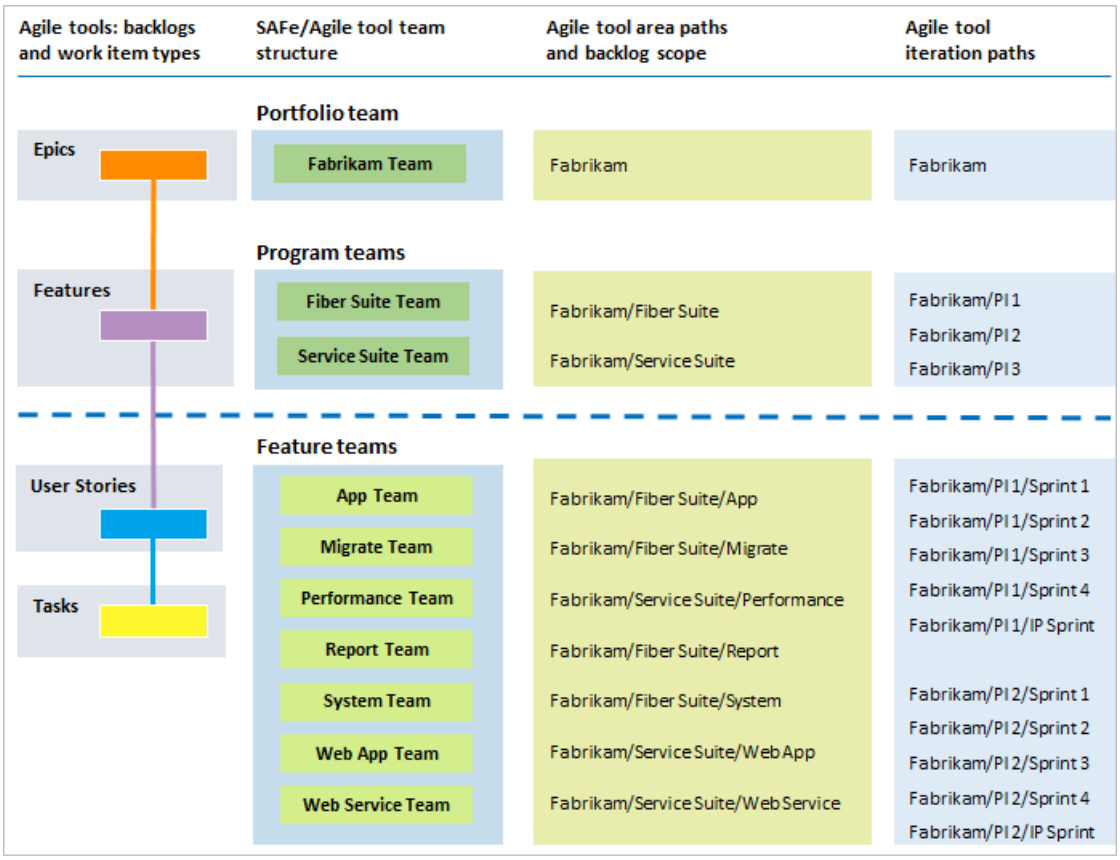
This tutorial illustrates how the following SAFe® artifacts map to specific Azure Boards artifacts.

- SAFe® Agile, program, and portfolio teams
- SAFe® deliverables such as epics, features, and stories
- SAFe® Product, program, and portfolio views
- SAFe® Release trains, sprints, and other timeboxes
- SAFe® Iteration goals and objectives
- SAFe® Value streams and budgets
- SAFe® Portfolio Vision and Strategic Themes
- SAFe® Roadmaps
- SAFe® Milestones and events
- SAFe® Retrospectives and reviews

For an overview of how Azure Boards implements Scrum and Kanban, see About Sprints, Scrum, and project management and About Boards and Kanban.

> **NOTE**
> This article is one of a set of Scaled Agile Framework® tutorials that applies to Azure Boards and Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features and procedures are specific to the cloud or the latest version of Azure DevOps Server.

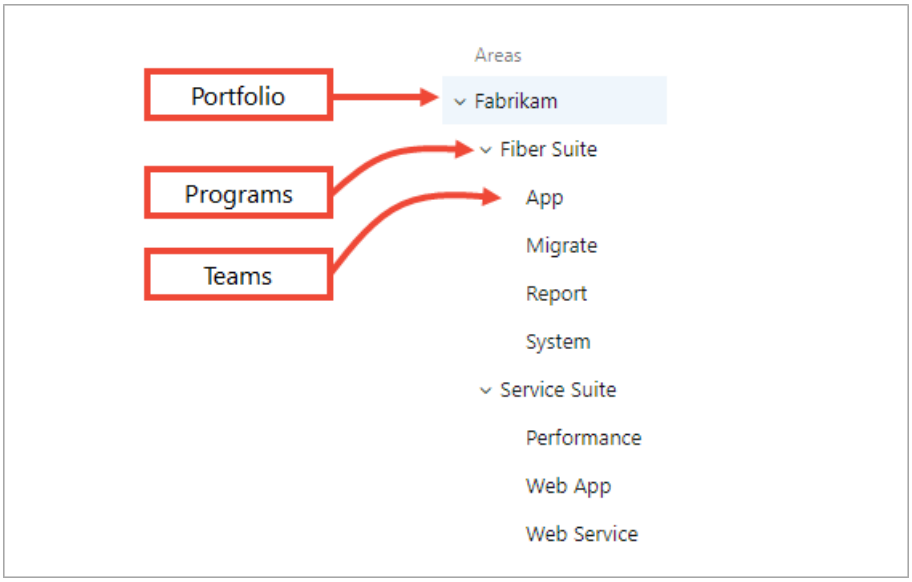
The following image illustrates how you can configure Azure Boards to support a three-level team hierarchy and map teams to their respective area and iteration paths. The examples build from the Agile process, However, the changes can be applied to any project and process hosted on Azure Boards.



Examples provided below illustrate how a three-level team hierarchy is configured using hierarchical area paths. The examples build from the Agile process, However, you can apply these changes to any project hosted on Azure Boards.

Agile feature, program, and portfolio teams

Azure Boards supports each team to have its own view their work. By configuring a hierarchical team structure, each team can focus on their work, and have their work roll up to the next level within the team hierarchy.



To support SAFe® teams, you reconfigure the default team as the Portfolio team to manage your epics. You then create subteams for program-level work and team-level work. Work can be tracked across teams and throughout each of the levels.

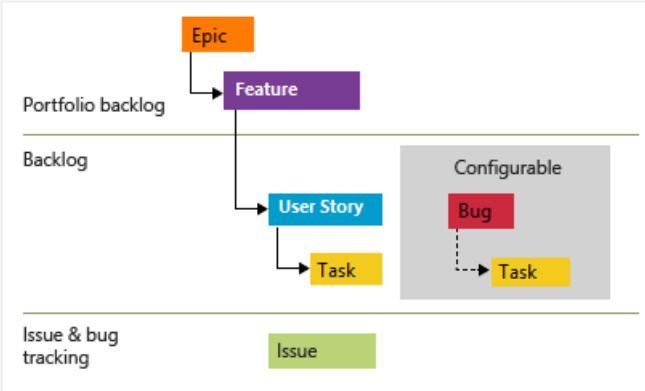
Stories, Features, Epics, Enablers, and Capabilities

All work and deliverables are captured in work items. Each work item is associated with a specific work item type with a predefined workflow. Each Azure Boards process provides support for specific work item types that you can use to track any of the SAFe® deliverables.

The work item types available to you're based on the process used when your project was created—Agile, Basic, Scrum, or CMMI—as illustrated in the following images.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

The items in your backlog may be called *User Stories* (Agile) *Issues* (Basic), *Product backlog items* (Scrum), or *Requirements* (CMMI). All four are similar: they describe the customer value to be delivered and the work to be completed.

You can track Enablers using User Stories or Features, and Capabilities using Features or Epics. Or, you if you have specific tracking and reporting needs, you can add custom work item types to track these types of deliverables. For more information, see Customize Azure Boards, Add custom work item types.

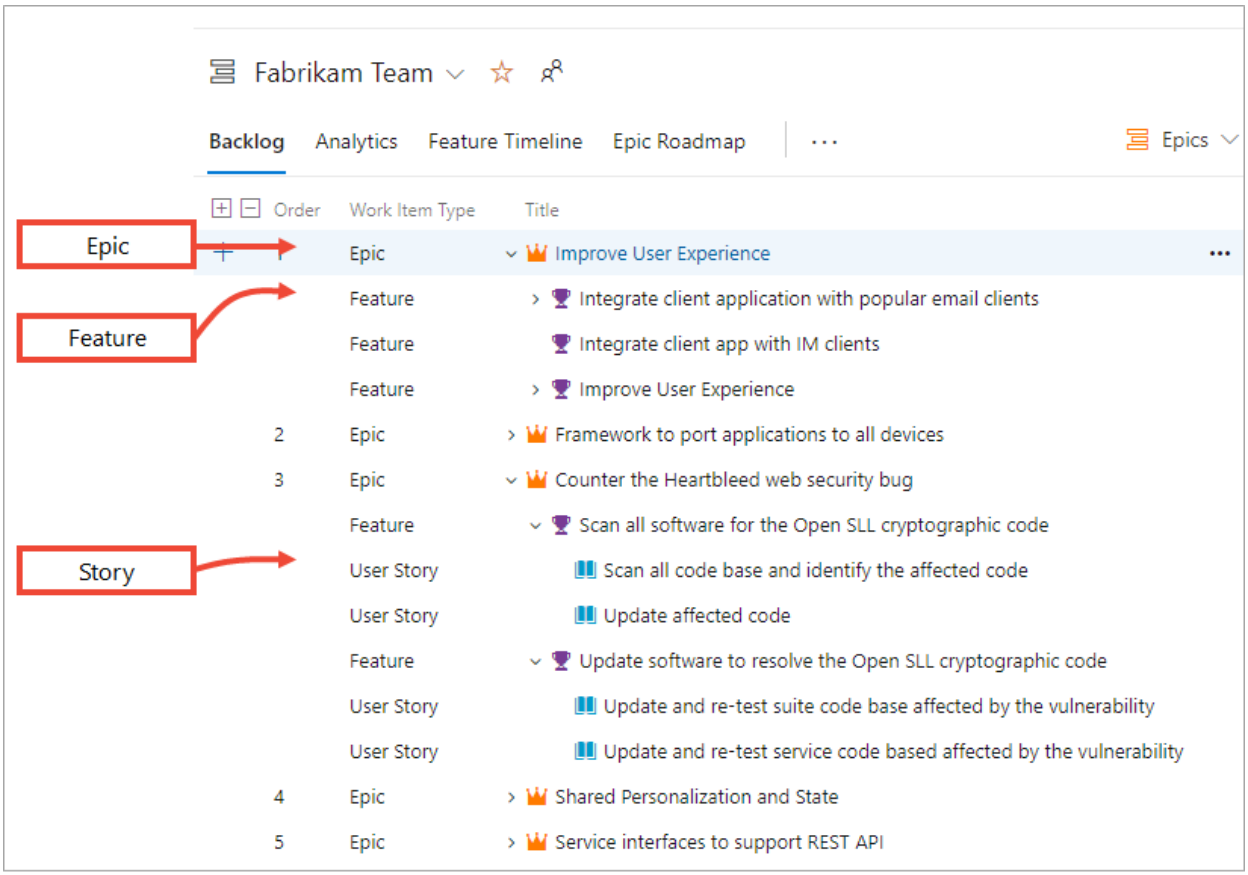
Work items provide support for the following tasks:

- Add description and acceptance criteria
- Assign to a team or area path and to a project member
- Update status and assign to an iteration or sprint
- Link work items, attach files, add tags
- Add comments and view a discussion thread

Product and portfolio backlogs enable teams to quickly add and prioritize their User Stories, Features, and Epics. For more information about work items and work item types, see Track work with user stories, issues, bugs, features, and epics.

Team backlogs and boards

SAFe® backlogs map to team, program, and portfolio backlogs. Out of the box, the Agile process supports User Story, Feature, and Epic backlog levels. The hierarchical backlog structure shows work done to support Features and User Stories in the progress of an Epic.

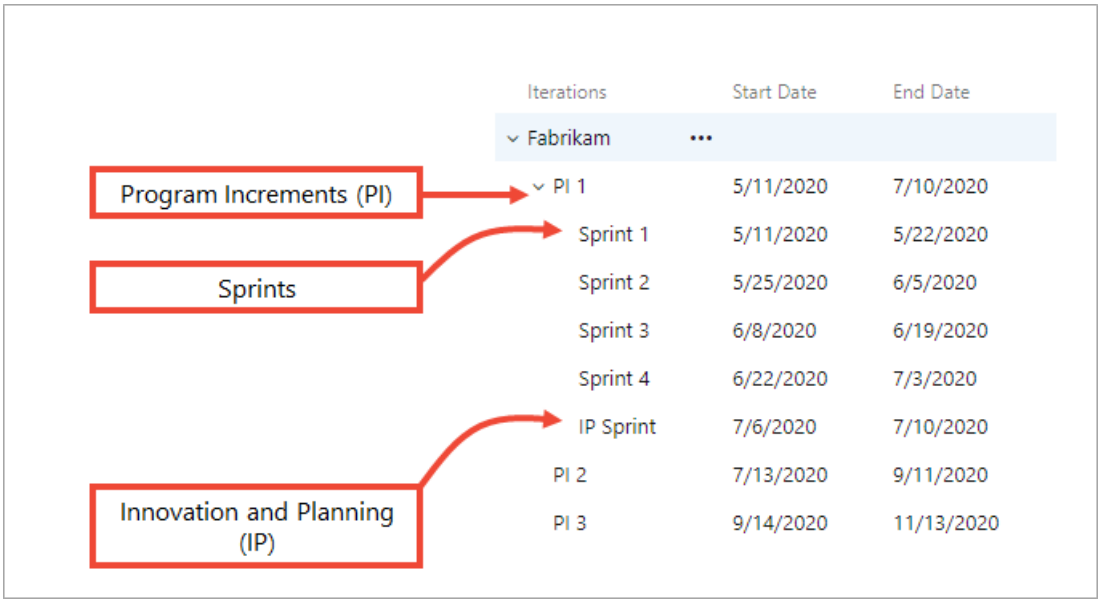


You can customize your backlog and boards, even adding portfolio backlogs, as described in Customize Azure Boards, Customize backlogs.

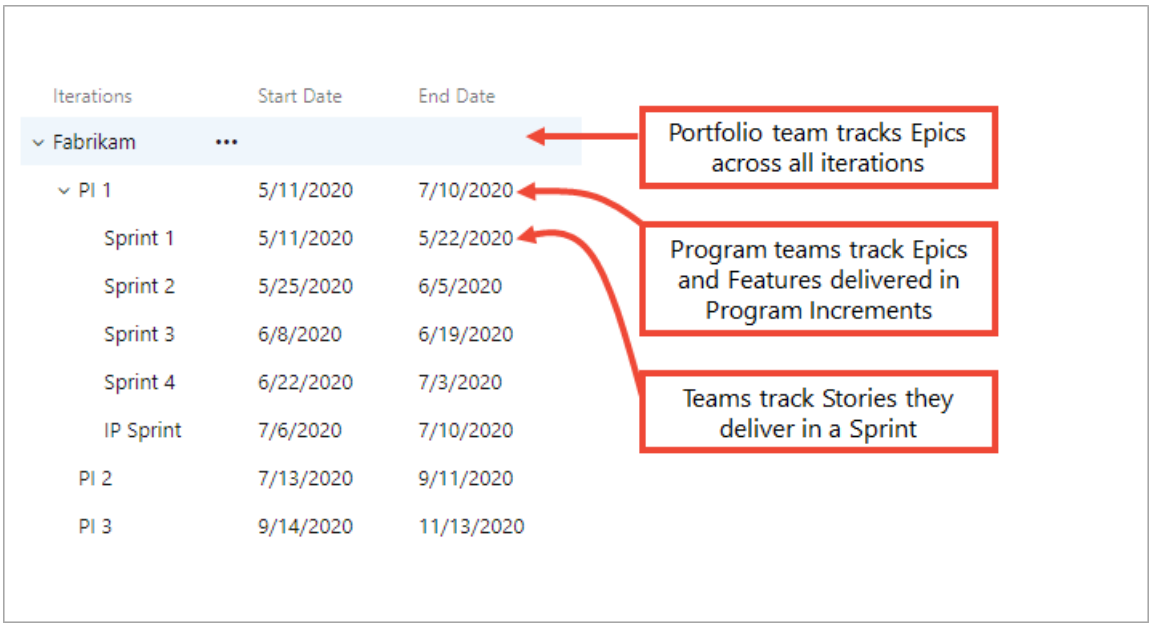
The Kanban board view of each backlog is configurable by each team.

Program Increments, releases, and sprints

SAFe® Release Trains, Releases, Iterations, Program Increments (PIs), and Sprints map easily to your iteration paths. By sharing iterations across the team hierarchy, you manage the releases in a cohesive manner.



Because epics can span several release trains, the Portfolio team isn't associated with any specific iterations. Program teams track their Feature deliverables, which ship with a PI. And Feature teams work in Sprints to complete several stories. Each team chooses which iterations will support them to track their focused set of deliverables.



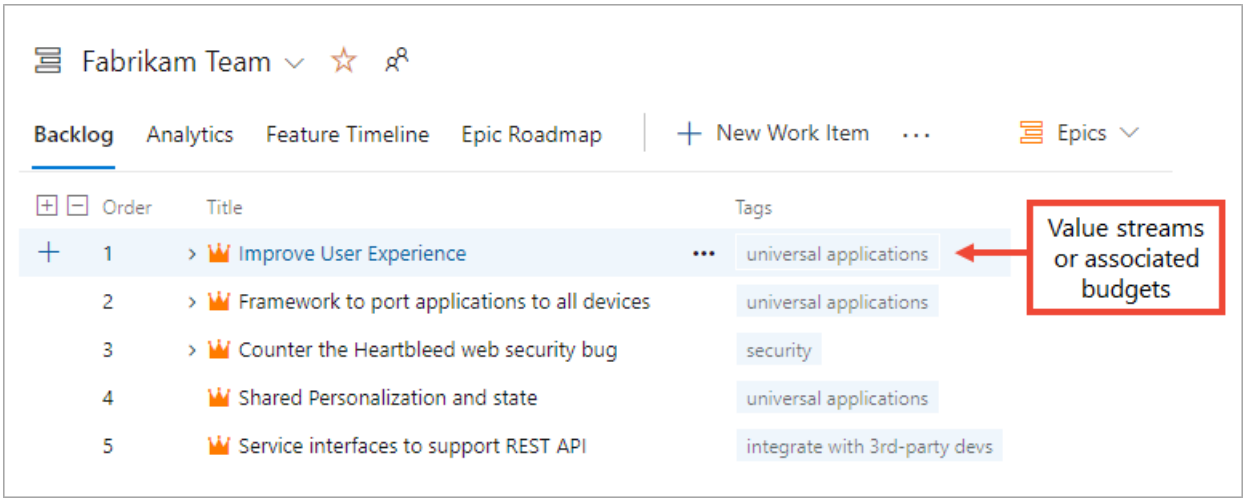
Iteration goals and objectives

SAFe® practices include Agile release teams defining their iteration goals and objectives. We recommend using the project wiki or team dashboards to capture team information. The project wiki and team dashboards both support Markdown to add and format information.

To learn more, see Share information later in this article.

Value streams and budgets

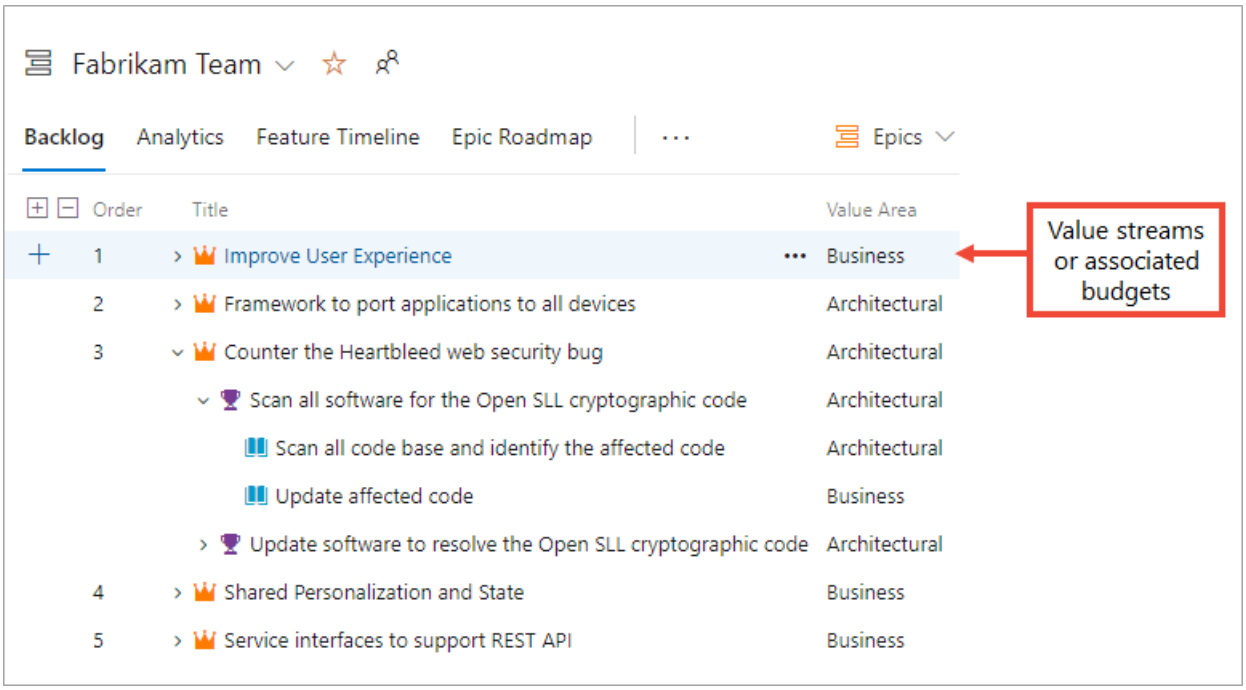
You can use tags for a quick and easy way to map Features and Epics to their Value Streams, Strategic Themes, and associated Budgets. You can add custom fields to capture budget estimates for Features that can then roll up to Epics.



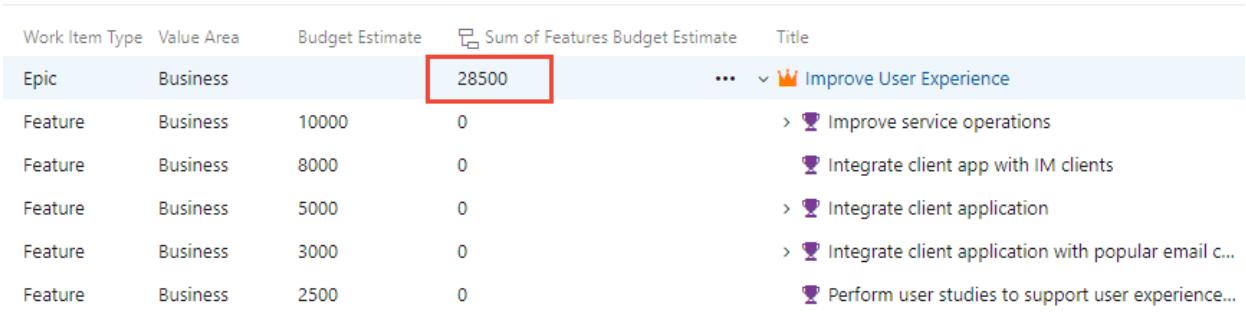
With tags that you add to work items, you can:

- Filter any backlog or Kanban board
- Create queries based on tags, and filter query results by tags
- Create progress and trend charts or reports based on tags

For a more robust mapping of work to architecture or business features, you can specify the Value Area for each Epic, Feature, or Story.



With rollup, you can get Budget Estimates for Epics from a rollup of the estimates defined to their child Features, as shown in the following image.



To add custom fields, see Customize Azure Boards, Add a custom field.

Use the project wiki to support your portfolio vision and strategic themes

Information can be widely shared with an organization using the Azure DevOps project wiki. The wiki is a similar to a git repository that supports adding and editing pages using Markdown and a WYSIWYG editor. It versions each page so that it's easy to track who made changes and recover past versions.

Use your project wiki to support sharing the following SAFe® artifacts:

- Portfolio Vision
- Strategic Themes
- Taxonomy
- Goals
- Objectives
- Customer-centric practices

To learn more about the project wiki, see Share information later in this article.

Milestones and key events

The end of each Program Increment, Sprint, Release Train, or Innovation and Planning (IP) Iteration represent natural SAFe® milestones. Many milestones are associated with specific ceremonies or practices, such as conducting retrospectives or demonstrating working software.

In Azure Boards, you can track other types of milestones or key events in the following ways.

- Custom field, such as Milestone or Release field with predefined picklist
- As a tag added to work items
- As a work item that specifies a target date
- As a one-day Iteration Path

With custom fields and tags, you can quickly filter backlogs, boards, and queries based on a specific milestone.

Shared services team structure

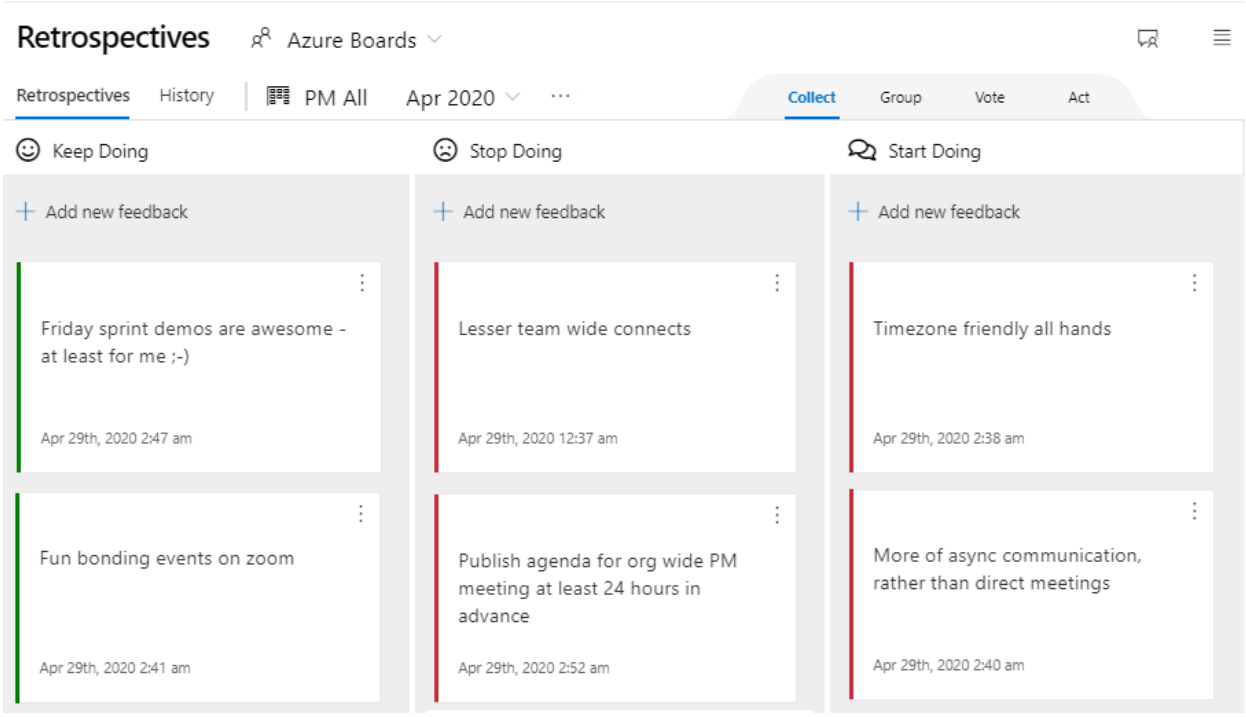
Resources that are shared across teams can be represented through their own Agile feature team, such as a UX Design team or a Security Compliance team. They can manage their backlog while having their work also appear in the backlogs of the teams they support.

Here we show how area paths are assigned to the UX Design team, and then selective subarea paths to other Agile teams. Work items that appear on shared area paths appear on the backlogs and boards of the associated teams.

UX Design	UX Design
App	App, UX Design
Report	Report, UX Design
Web App	UX Design, Web App
Web Service	UX Design, Web Service

Retrospectives and reviews

To support teams doing retrospectives and reviews, we recommend using the Retrospectives extension by Microsoft DevLabs.



This extension allows teams to create their own retrospective boards and capture the following tasks:

- Collect feedback on project milestones
- Organize and prioritize that feedback
- Create and track actionable tasks to help each team in their improvement processes.

Share information

Azure Boards provides many ways to share information.

- Work item forms provide rich-text fields to capture descriptions, acceptance criteria and more. File attachments can be added to work items or links to network file shares.
- Project and team dashboards can be used to share information along with status and progress charts and widgets. To learn more, see Add Markdown to a dashboard.
- The Project wiki provides a central repository with versioning control built-in to share information with all project members. Other wikis can be created as needed. To learn more, see About Wikis, READMEs, and Markdown.

For details on supported Markdown features, see the following articles.

- Syntax guidance for Markdown usage in Wiki
- Syntax guidance for basic Markdown usage

Try this next

Configure Azure Boards to support SAFe®

Related articles

- Agile process
- Scrum process
- CMMI
- View SAFe® progress, roadmaps, and metrics

Culture and scale

- Agile culture
- Practices that scale
- Scale Agile to Large Teams

Configure Azure Boards to support SAFe® programs and portfolios

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This tutorial walks you through the steps to convert a new project with a single team to one that is configured to support Scaled Agile Framework (SAFe®) programs and portfolios. Specifically, you'll learn how to configure Azure Boards to support SAFe® programs and portfolios by completing the following tasks:

- Define Agile, program, and portfolio teams
- Configure a hierarchy of Area Paths to support your teams
- Define Iteration Paths to support SAFe® release trains, PIs, sprints, and IPs
- Configure each team to support SAFe®

You'll need to be a member of the Project Administrators group to make these configurations.

Once you've completed these core configurations, you can then consider customizing your project to support specific business needs. Customization options are addressed in Customize Azure Boards to support SAFe® .

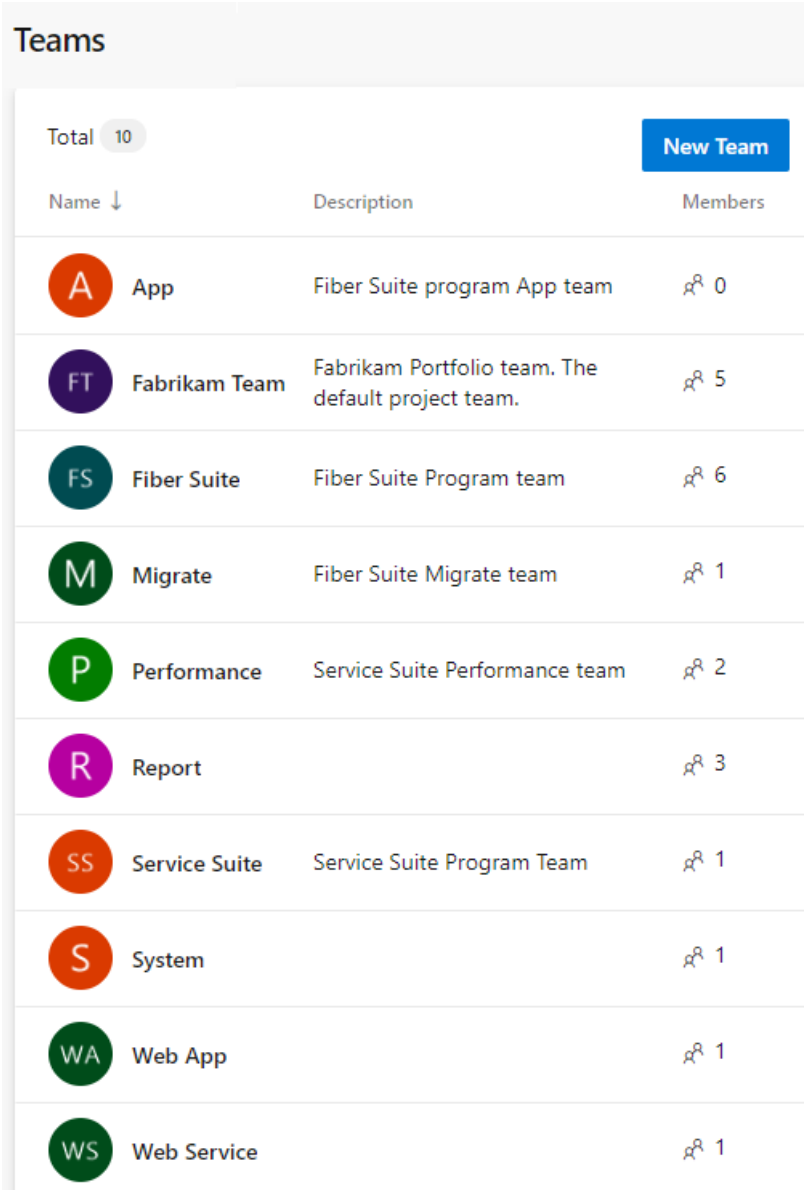
> **TIP**
> If you plan to add custom work item types, portfolio backlogs, or workflows; you may want to make those customizations first and then define and configure your teams.

If you're new to Azure Boards, we recommend that you review About teams and Agile tools and About area and iteration (sprint) paths before adding and configuring your teams. Also, two excellent articles to review around team structure and Agile culture are Introduction to planning efficient workloads with DevOps and Building productive, customer focused teams.

> **NOTE**
> This article is one of a set of Scaled Agile Framework® tutorials that applies to Azure Boards and Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features and procedures are specific to the cloud or the latest version of Azure DevOps Server.

Understand team hierarchy

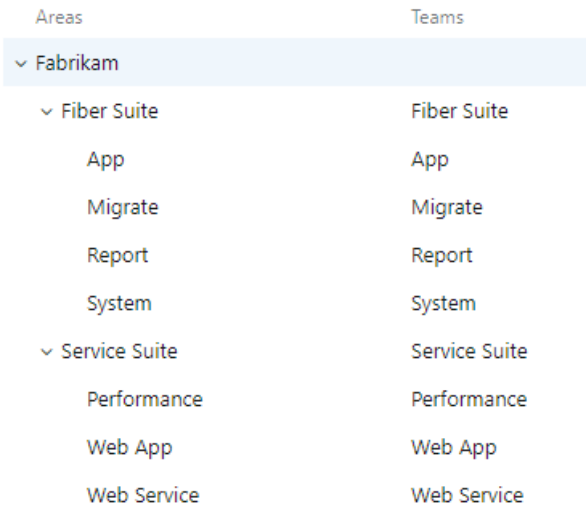
In this article, we'll go from having one project and one team, both named "Fabrikam", to the following set of nine teams.



NOTE

Azure Boards doesn't support a hierarchy of teams. However, by configuring the Area Paths as indicated in this article, you effectively create a type of team hierarchy. The hierarchy is defined through the structure of Area Paths.

We'll then configure the area path to the following hierarchy and configuring each team's area path. This configuration supports each team's backlog view and rollup of views within the hierarchy.



> **TIP**
> If you have a large number of teams, area paths, and iterations that you need to add, you may want to use command line or programmatic tools. See the Command line and programmatic tools provided later in this article.

All teams can manage their own workload and priorities while clearly understanding how their work supports those epics managed in the portfolio team's backlog. At the same time, the portfolio team can monitor progress of its backlog on their own Kanban board, prioritize the items on their backlog, and view progress across release trains.

While the above may sound complicated, it actually takes little configuration to set up the teams and get started. To go from one project with one default team, we'll first define each team while automatically creating a default area path for that team. Then we'll reconfigure the flat set of area paths to a hierarchical structure. Next, will define the iteration paths to support the release structure we want and the program and Agile teams to use. Lastly, we'll configure each team and populate the membership of teams.

Define your teams

To start, we'll add each team, creating a default area path for each. Later in this article, we'll configure those area paths into the necessary hierarchy. This structure maps the following SAFe® teams to Azure Boards teams:

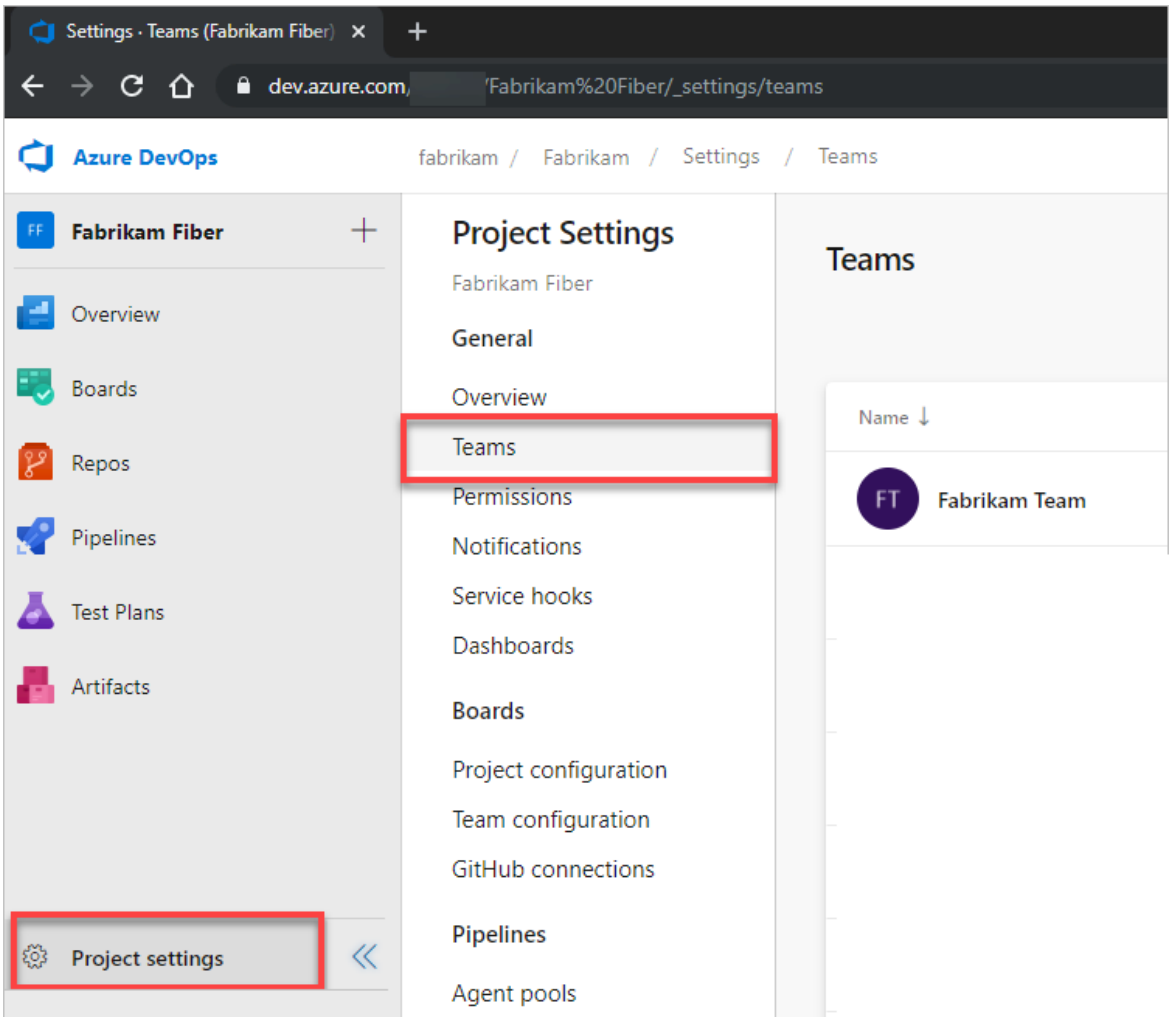
- Portfolio team -> default top-level team, the Fabrikam team (already defined)
- Program teams -> secondary-level teams, Fiber Suite, and Service Suite
- Agile teams -> tertiary-level teams defined under Fiber Suite and Service Suite.

You'll need to be a project administrator to complete these steps. If you need more-detailed guidance, see Portfolio management.

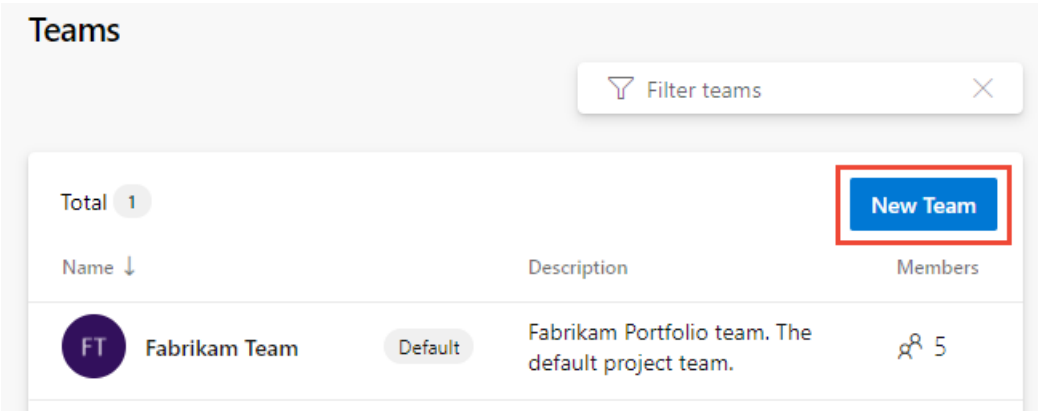
Add each team, one by one.

> **NOTE**
> The following procedure uses the **New Teams Page** user interface that is in preview. To enable this feature, see Manage or enable features.

1. From the web portal, choose **Project settings** and open **Teams**.

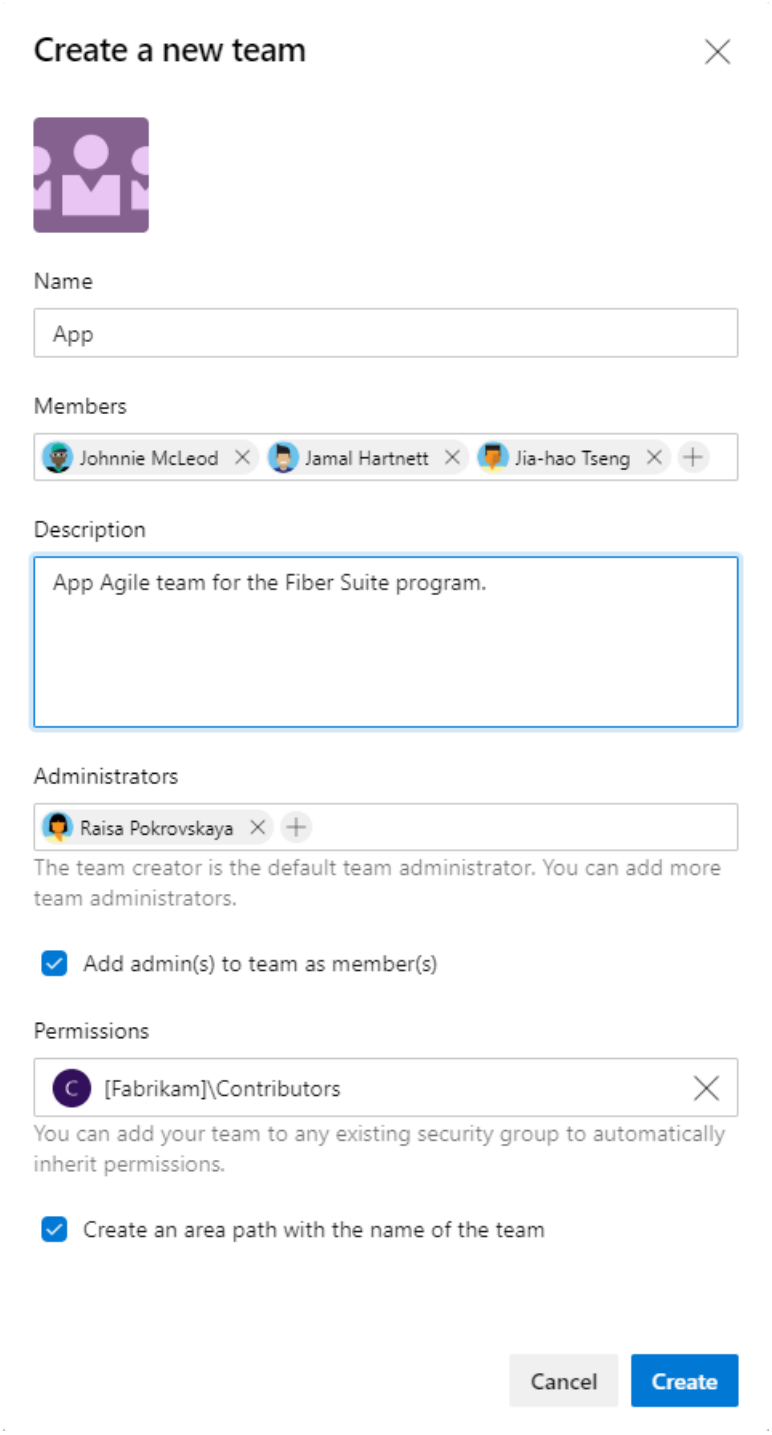


2. Choose **New team**.



3. Give the team a name, and optionally a description.

   Here we add the *App* team. Choose the team administrator and ensure the **Create an area path with the name of the team** checkbox is checked. Optionally add team members.



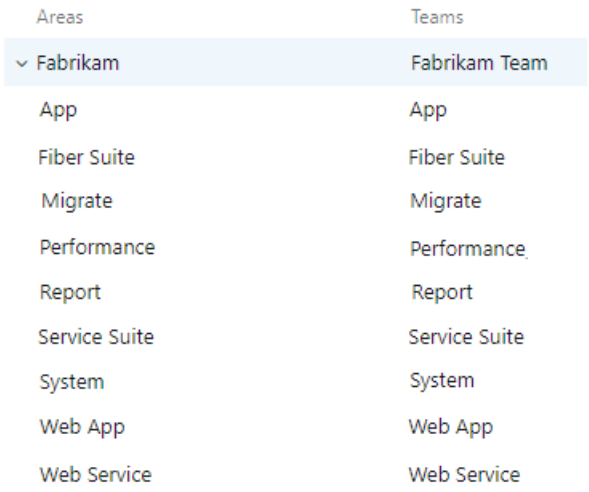
Assign the team's Scrum Master, Program Manager, or Portfolio Manager as the team administrator. As team administrators, they can configure their team's tools to support their Agile practices and business needs.

4. Repeat steps 2 and 3 to define all teams.
5. Optional. If you have two or more Portfolio teams, create a team for each of them.

Configure Area Paths

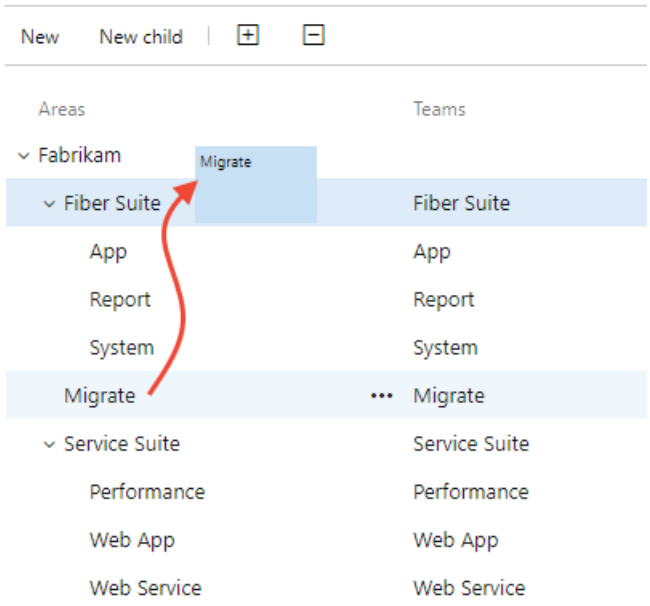
To support your team hierarchy, you'll now configure the area paths created in the first step of defining teams into a hierarchy.

1. From the **Project Settings** page, choose **Project configuration** and then **Areas**. You should see a flat list of Area Paths.



2. You'll want to choose each feature team's Area Path under the top Area Path and move it under the Area Path hierarchy to which it belongs.

   You can drag and drop each area path under the parent node where it belongs. For example, here we drag the Migrate node to the Fiber Suite node.



   Instead, you can open the context menu for the Area Path, choose Edit, and select the node where you want to move it.

3. Repeat step 2 and 3 for the remaining Agile team area paths.

   If you've defined two or more portfolio teams, you'll need to change the move each program team's area path under their corresponding portfolio team's area path.

4. When finished, your area path structure should appear similar to the following image.

> **IMPORTANT**
> This structure shows that area paths are owned by Agile teams, program teams, and the portfolio team. We'll correct this structure later in this article when we configure each team to be the sole owner of its area path.



Areas	Teams
Fabrikam	Fabrikam Team
Fiber Suite	Fabrikam Team, Fiber Suite
App	App, Fabrikam Team, Fiber Suite
Migrate	Fabrikam Team, Fiber Suite, Migrate
Report	Fabrikam Team, Fiber Suite, Report
System	Fabrikam Team, Fiber Suite, System
Service Suite	Fabrikam Team, Service Suite
Performance	Fabrikam Team, Performance, Service Suite
Web App	Fabrikam Team, Service Suite, Web App
Web Service	Fabrikam Team, Service Suite, Web Service

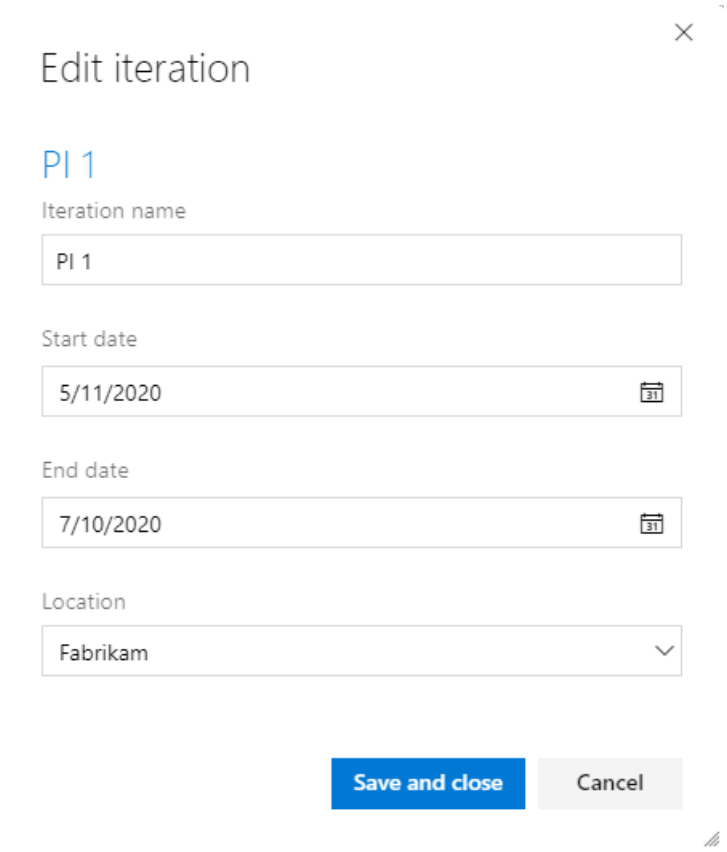
Define Iteration Paths

To track progress towards Releases, create your iteration path structure. Unlike area paths, multiple teams can share the same iteration path structure. Sharing the iteration structure lets multiple teams work in the same sprint cadence towards the same release trains.

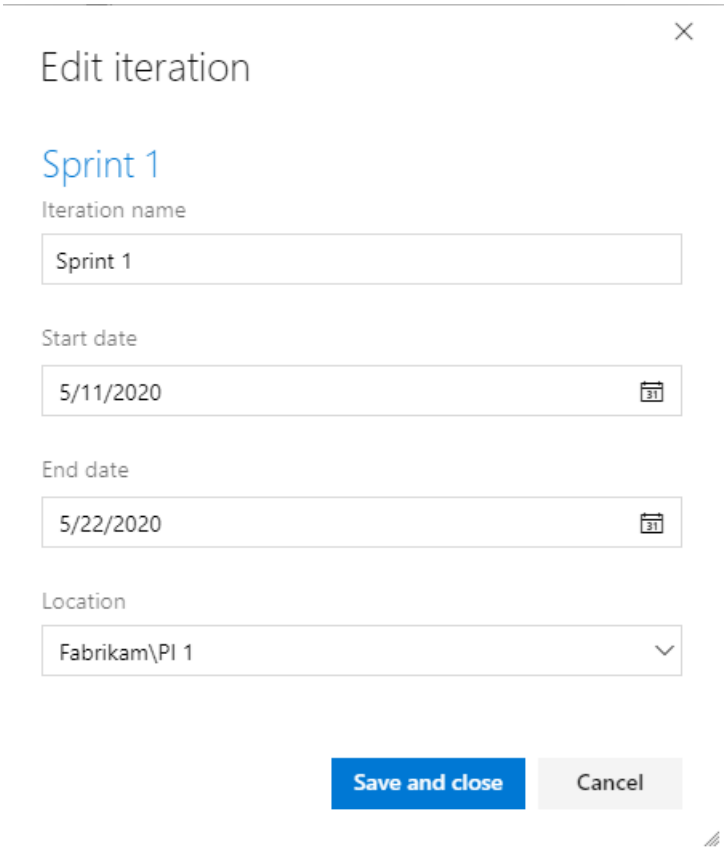
> **IMPORTANT**
> Deleting, renaming, or moving iteration paths causes a loss of associated historical data.

If you already have iterations for your default team, you can rename them. You'll want to create an iteration structure that supports your entire team structure, not just one team.

1. From the **Project Settings** page, choose **Project configuration** and then **Iterations**.
2. Under the default iteration, which shares the same name as the project, create a child iteration that will represent your first program increment (PI). Optionally, add a start and end date for the PI, but keep in mind that the iteration will be broken down further into sprints.

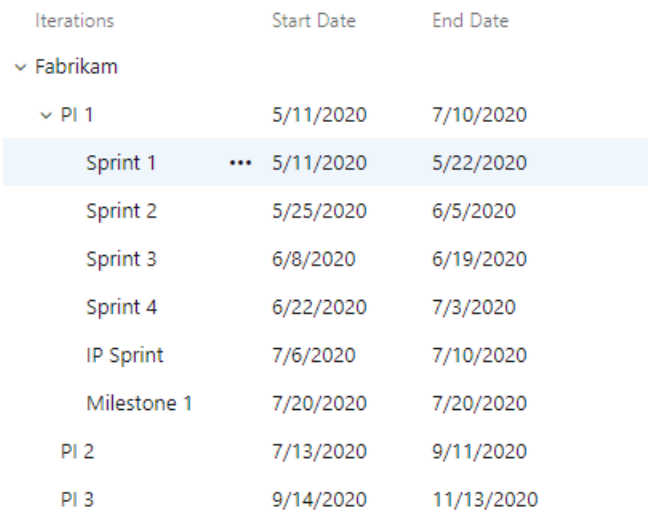


3. Next, create a child iteration for each Sprint within the PI. Set dates for these sprints to correspond your Agile teams' cadences.



4. Continue to add as many iterations as needed to meet the time box cadence structure for all your teams.

   When finished, you should have a structure similar to the following image.



Iterations	Start Date	End Date
Fabrikam		
PI 1	5/11/2020	7/10/2020
Sprint 1	5/11/2020	5/22/2020
Sprint 2	5/25/2020	6/5/2020
Sprint 3	6/8/2020	6/19/2020
Sprint 4	6/22/2020	7/3/2020
IP Sprint	7/6/2020	7/10/2020
Milestone 1	7/20/2020	7/20/2020
PI 2	7/13/2020	9/11/2020
PI 3	9/14/2020	11/13/2020

TIP

You can drag and drop Iteration Paths to structure your iterations, similar to as shown in Step 2 under Configure Area Paths. Azure Boards always lists the iteration paths in order of their dates under each parent node.

Configure your teams

Now that your teams, Area Paths, and Iteration Paths are defined, the next step is to configure each team. You'll want to configure the following settings for each team.

- Active backlogs
- Working with bugs
- Set default Iteration Path
- Select team Iteration Paths

The following table lists the recommended settings to make based on the team level.

Configure

Agile feature team

Program team

Portfolio team

Backlog navigation levels

Features, Stories

Features, Stories

Epics

Working with bugs

Bugs are managed with requirements

Bugs are not managed on backlogs and boards

Bugs are not managed on backlogs and boards

Default Iteration

@CurrentIteration

@CurrentIteration

@CurrentIteration

Backlog Iteration

Fabrikam

Fabrikam

Fabrikam

Selected iterations

Sprint 1 thru Sprint 4, IP Sprint

PI 1, PI 2, PI 3

None

Areas

Include sub areas

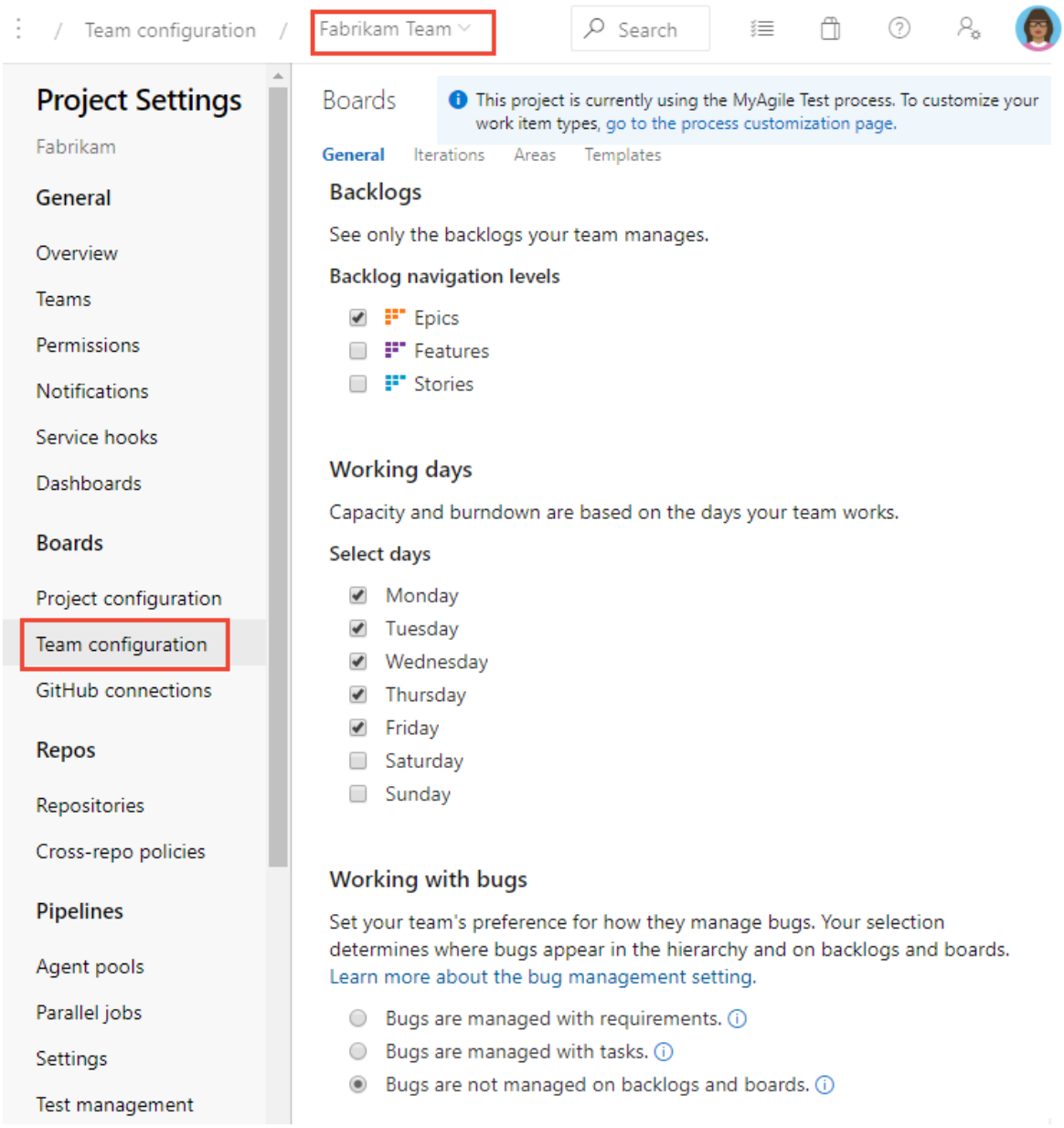
Exclude sub areas

Exclude sub areas

> **NOTE**
> By setting the Default Iteration to **@CurrentIteration**, all work items created from the team's backlog or board are assigned to the current iteration based on the current date. By setting the Backlog Iteration to the root, **Fabrikam**, indicates that only the Area Path acts as a filter for work items to appear on the team backlogs and boards.

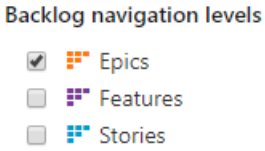
1. From the **Project Settings** page, choose **Team configuration**.

   Choose the team you want to configure from the Team selector.

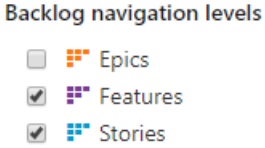


2. On the **General** page, uncheck backlogs you don't want to be active.

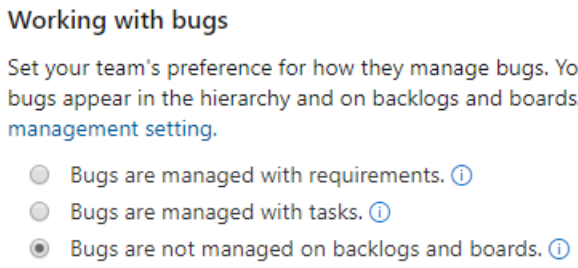
   For example, for the Portfolio team, only check the **Epics** checkbox.

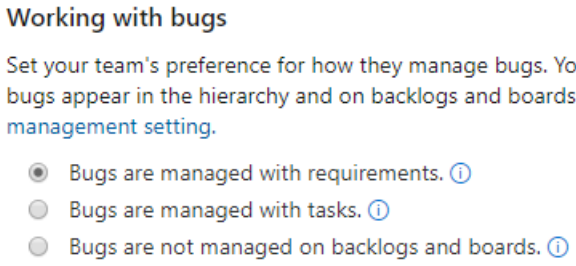
   For program and Agile teams, uncheck the **Epics** checkbox.

3. For program and portfolio teams, choose the **Working with bugs** radio button as shown.

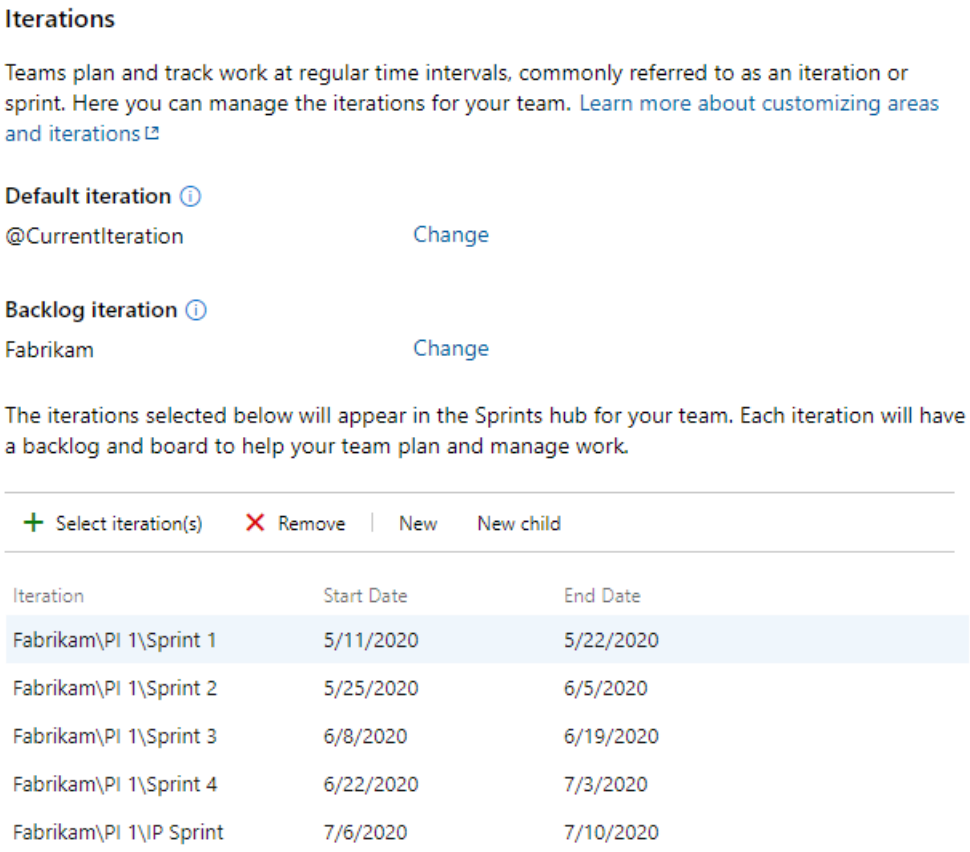


And, for Agile teams, choose the **Working with bugs** option to track bugs along with requirements.

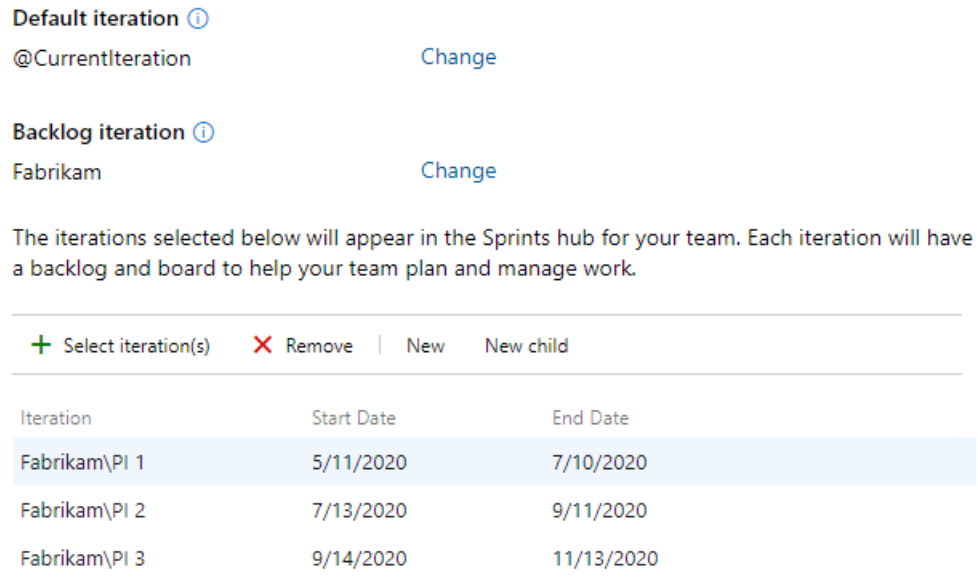


4. Choose the **Iterations** tab to configure the team's iterations.

   For Agile teams, configure the settings as shown.

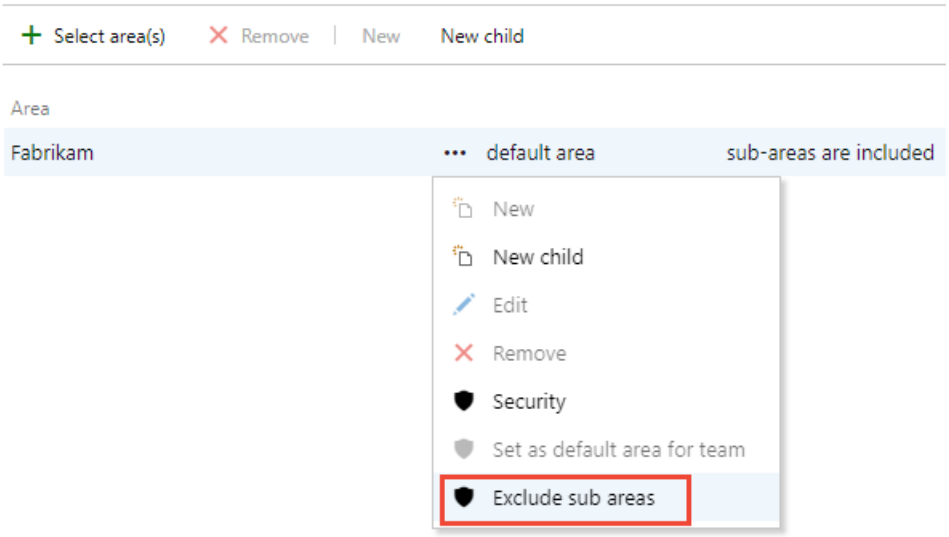


   For program teams, choose only the PI iterations.



5. For program and portfolio teams, choose the **Areas** tab to change the default setting from **Include sub areas** to **Exclude sub areas**.

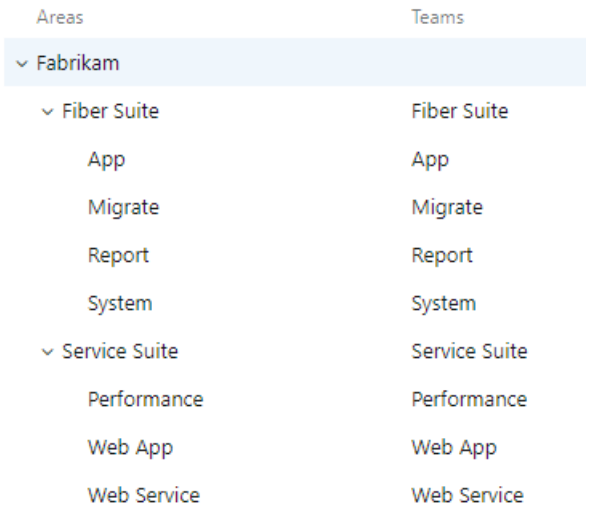
   Open the ··· context menu, and choose **Exclude sub areas**.



> **NOTE**
> Because we created each team with the **Create an area path with the name of the team** checked, each team is already preconfigured with their default area path. This Area Path acts as the main filter for work items that appear on each team's backlogs and boards.

6. Repeat steps 2 through 5 as needed for each team you need to configure.

7. After you've completed step 5 for all teams, verify the Area Path-Team structure. Choose **Project configuration** and **Areas**. The Area Path and team structure should now appear as shown, where each team owns their Area Path and doesn't share it with any other team.

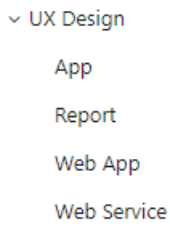


Configure teams to support Shared Services

For teams that support several other teams, such as a UX Design team, configure your teams as described in the following steps.

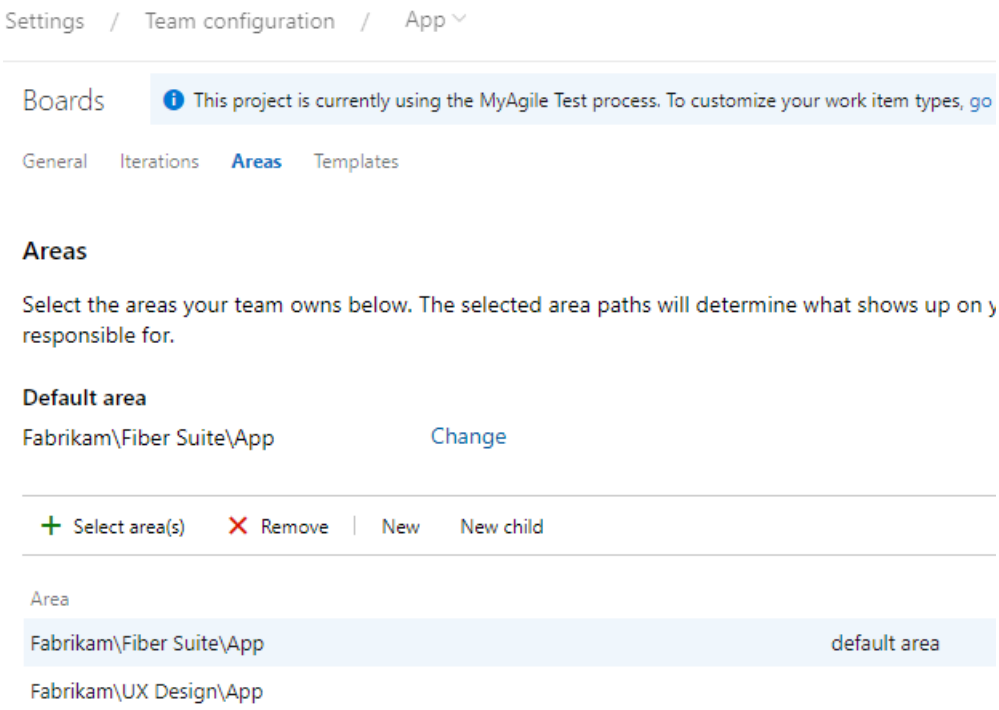
1. Add a team for each Shared Services team. Refer to Define your teams for details.
2. Return to the **Project configuration > Area Paths** page and under each shared services area path, add sub area paths for each Agile team supported by the shared services. For details, see Configure Area Paths provided earlier in this article.

   For example, here we add four sub area paths under the UX Design area path, one for each Agile team supported by the UX Design team.

3. Configure each Shared Services team as an Agile feature team as described in Configure your teams.
4. For each Agile team, open the **Team configuration > Areas** page as shown in Step 5 of Configure your teams. Choose **Select areas** and add the sub area path for that team.

   Here we add the **UX Design\App** sub area path to the App feature team.



5. Return to the **Project configuration > Area Paths** page and verify that the Area Path structure appears as expected for each Shared Services area path.

   For the UX Design team, the structure should appear as shown.

UX Design	UX Design
App	App, UX Design
Report	Report, UX Design
Web App	UX Design, Web App
Web Service	UX Design, Web Service

   Work items that appear on shared area paths appear on the backlogs and boards of the associated teams.

Command-line and programmatic tools

You can use Azure DevOps command-line tools to add or update the following artifacts:

- **Teams**: Azure DevOps team create
- **Area Paths**: Azure DevOps area project create
- **Iteration Paths**: Azure DevOps iteration project create

Use programmatic tools

You can use Azure DevOps REST APIs to add or update the following artifacts:

- **Teams**: Teams (REST API)
- **Area Paths**: Classification nodes (REST API)
- **Iteration Paths**: Classification nodes (REST API)

Try this next

Customize Azure Boards to support SAFe®

Related articles

- Add teams
- Manage teams and configure team tools
- Define area paths and assign to a team
- Define iteration paths and configure team iterations
- Azure DevOps CLI
- Teams (REST API)

Customize Azure Boards to support SAFe® practices

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The main reason to customize your process is to support tracking and monitoring progress, reporting on key metrics, and meeting specific business requirements. In this article, you'll learn about select process customizations you can make and why you might want to make them support your Scaled Agile Framework (SAFe®) practices. Most of these customizations are optional.

Specifically, you'll learn how Azure Boards supports SAFe® practices through the following operations.

- Customize work item types or add custom work item types
- Add a custom field or customizing existing fields
- Customize the workflow
- Add custom rules to a work item type
- Add custom controls or custom extensions
- Customize your backlogs or add a custom portfolio backlog

> **NOTE**
> This article is one of a set of Scaled Agile Framework® tutorials that applies to Azure Boards and Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features and procedures are specific to the cloud or the latest version of Azure DevOps Server.

About customization and the inherited process

Azure Boards provides a graphical user interface to support customization of your projects. This feature is called the Inherited process. All projects that use an inherited process are automatically updated when customizations are made to that process.
For an overview of all the customizations you can make to an inherited process, see About process customization and inherited processes.

Customize work item types

Each work item type defines the fields that capture and store information. You can customize existing work item types in the following ways to support specific SAFe® tracking requirements.

- Add a custom field to support tracking budget costs, value streams, or customer-centric information
- Customize existing fields, such as modifying the picklist or changing a field label
- Add custom rules to make select fields required or specify actions to take under select conditions
- Change the workflow to reflect your team's Kanban workflow process
- Add a custom control or extension to support custom functions such as a calculated field.

For details on customizing a work item type, see Add and manage work item types.

Add a custom field

You add a custom field to support tracking data requirements that aren't met with the existing set of fields. Some fields to consider adding to one or more work item types include those items listed in the following table.

Field name	Work Item Types	Notes
Budget cost	Feature, Epic	Use to capture estimated costs. Can use rollup to capture the total estimated cost of an Epic's Features.
Category or Group	Feature, Epic, User Story	Use to specify a picklist to indicate the work item is cataloged as one of the following SAFe® categories: *Feature*, *Capability*, *Enabler*, or *Solution*.
Milestone	Feature, Epic, User Story	Use to specify a picklist of Milestone of Events which a story, feature, or epic should meet.
Value Stream	Feature, Epic, User Story	Use to specify a picklist to support a taxonomy of value streams you want to associate with work.

For details on adding a custom field, see Add a custom field to a work item type.

Field versus tags usage

You can capture a value stream using a field or tags. Tags represent a more informal and adhoc method for categorizing work. A specific field, particularly one with preset items, is more formal. When determining how you want to use tags and fields, consider the following statements:

- You can make a field required through custom rules, however, you can't require tags be added to a work item.
- You can create query charts based on custom fields, however, you can't specify a tag for use in query charts.
- You can filter backlogs, boards, and queries based on fields or tags.
- The number of tags created can quickly grow as anyone can add new tags as long as they have the correct permissions.

Customize existing fields

You customize existing fields to support one or more of the following actions:

- Relabel the name of the field.
- Change where the field appears on the work item form or remove it from the form.
- Add or change a picklist (drop-down menu). For example, the *Value Area* provides two options, *Business* and *Architectural*. You can add to this picklist of values.
- Change the default assignment made to a field.
- Make a field required.

- Add a rule to a field as described in the next section.

For an index of existing fields, see Work item field index. For details on customizing a field, see Add and manage fields for an inherited process.

Add rules to a field

Field rules provide support for many business use cases. Rules specify an action to take based on a selected condition. For example, you can make a field required based on the value assigned to another field. You can add several rules to a field.

The following images show the supported conditions and actions you can select from.

SUPPORTED CONDITIONS	SUPPORTED ACTIONS
A work item is created ...	Clear the value of ...
A work item state changes to ...	Copy the value from ...
A work item state is not changed ...	Hide the field ...
A work item state changes from ...	Make read-only ...
A work item state is ...	Make required ...
A work item state is not ...	Restrict the transition to state ...
A work item state moved from ...	Set the value of ...
= The value of ... (equals)	Use the current time to set the value of ...
≠ The value of ... (not equals)	Use the current user to set the value of ...
A value is defined for ...	
No value is defined for ...	
A change was made to the value of ...	
No change was made to the value of ...	
Current user is member of group ...	
Current user is not member of group ...	

SUPPORTED CONDITIONS	SUPPORTED ACTIONS
A work item is created ... A work item state changes to ... A work item state is not changed ... A work item state changes from ... A work item state is ... A work item state is not ... = The value of ... (equals) ≠ The value of ... (not equals) A value is defined for ... No value is defined for ... A change was made to the value of ... No change was made to the value of ... when current user is member of group ... when current user is not member of group ...	Clear the value of ... Copy the value from ... Make read-only ... Make required ... Set the value of ... Use the current time to set the value of ... Use the current user to set the value of ...

SUPPORTED CONDITIONS	SUPPORTED ACTIONS
A work item is created ... A work item state changes to ... A work item state is not changed ... A work item state changes from ... A work item state is ... A work item state is not ... = The value of ... (equals) ≠ The value of ... (not equals) A value is defined for ... No value is defined for ... A change was made to the value of ... No change was made to the value of ...	Clear the value of ... Copy the value from ... Make read-only ... Make required ... Set the value of ... Use the current time to set the value of ... Use the current user to set the value of ...

For details on setting field rules, see Add a rule to a work item type (Inheritance process).

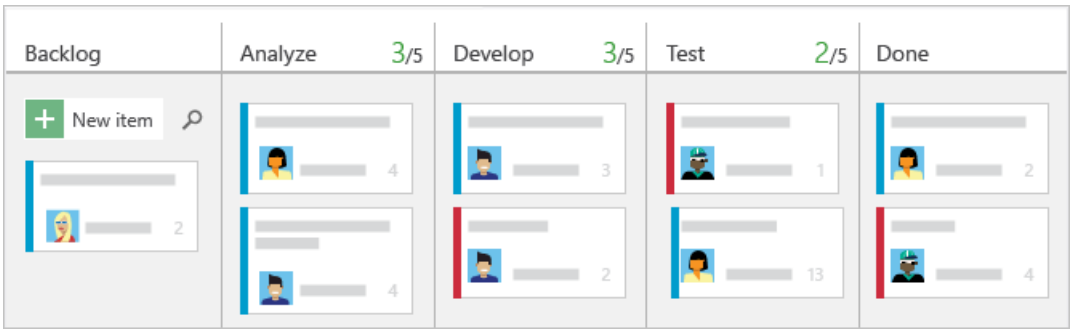
Customize the workflow

You may want to customize the workflow for User Stories, Features, and Epics so that it matches your workflow process. By customizing the workflow early, you minimize the Kanban board configuration teams must do.

The default workflow for the Agile process includes *New*, *Active*, *Resolved*, and *Closed* states. While each team can add workflow columns to their Kanban board, you might want to customize the workflow to track these columns instead. That way the Kanban boards for all teams are set up to use the same workflow states.

For example, you can add and rename workflow States to match the Kanban columns shown in the following

image—*Backlog*, *Analyze*, *Develop*, *Test*, and *Done*.



Review with your team's what workflow states will most support their Agile practices. For more details, review the following articles:

- Customize the workflow (Inheritance process)
- Add columns to your Kanban board
- Definition of Done

Custom controls

With custom controls, you can add rich functionality to a work item form. A custom control is an extension that's been added to the Marketplace Extensions for Azure DevOps.

You can add controls from the Marketplace or create your own.

- WorkBoard OKRs Integrates WorkBoard helps organizations align, localize, and measure Objectives and Key Results (OKRs) across the business. With this integration, teams can view and update their OKRs from within Azure DevOps.

Add custom work item types

The User Story, Feature, and Epic work item types are meant to support product planning and tracking. However, other work item types might be useful to support your SAFe® organization's customer-centric focus. Specifically, you might want to add work items to capture customer feedback, customer requests, and more.

When defining a new work item type, think through the following items:

- Information you want to capture, track, and report on
- How work will be captured
- The workflow to support tracking the work

To keep things simple, however, it's always best to minimize the amount of customizations you make. So, if you can get by with existing work item types, you might consider adding custom field(s) as needed to track specific information.

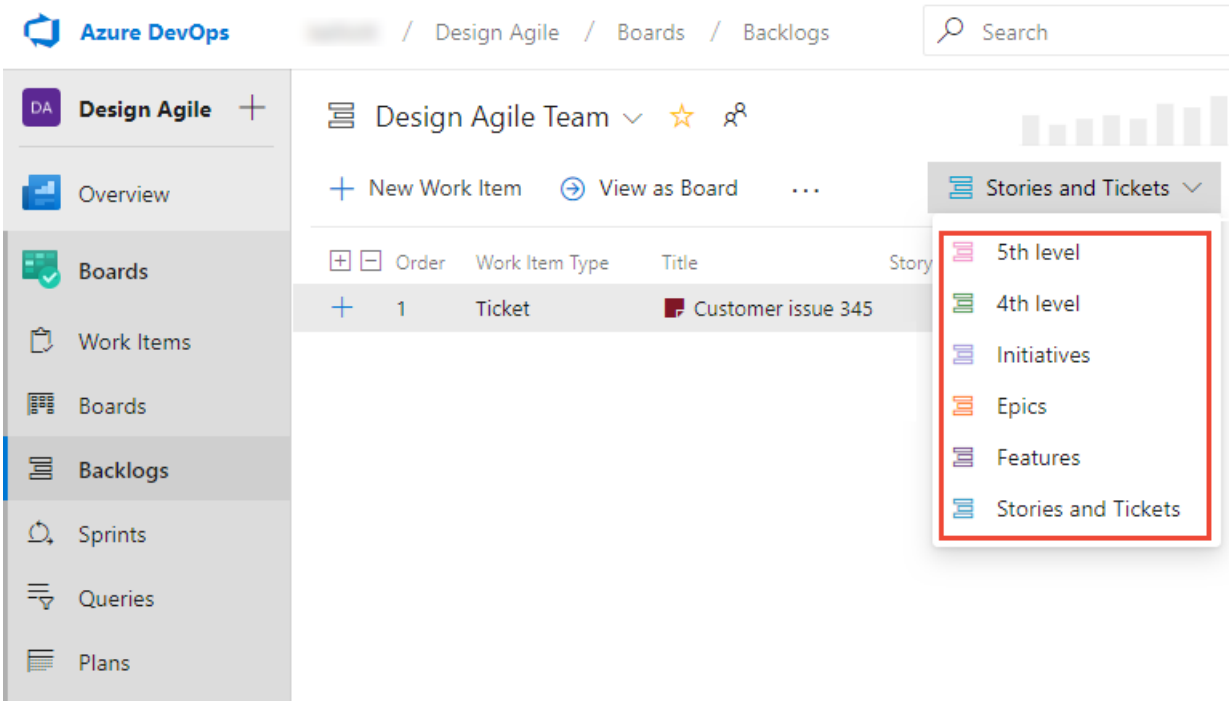
Customize your backlogs

Each team's backlog and board is designed to support specific work item types. For the Agile process, the work item types are as listed.

- **Agile Release Teams**: User Stories and, optionally, Bugs
- **Program Teams**: Features
- **Portfolio Teams**: Epics

However, you can add other work item types, existing or custom, to these backlogs. Each team can subscribe to the set of backlogs that they need to track.

You can also add up to three more portfolio backlogs as shown in the following illustration. Portfolio backlogs are designed to be hierarchical. For SAFe, you may want to add a Solution Backlog that appears as a parent to the Epic backlog.



For details on customizing backlogs, see Customize your backlogs or boards (Inheritance process).

Add even more functionality

You add the following Marketplace extensions to get access to many rich features that support SAFe.

- Delivery Plans
- Feature Timeline and Epic Roadmap
- Dependency Tracker
- Retrospectives

Try this next

Plan and track SAFe® programs and portfolios

Related articles

Before customizing your project, we recommend you read the Configure and customize Azure Boards. It provides detailed information on administrating a project for several teams and supporting various business objectives.

See also:

- Grant or restrict access
- Develop a web extension for Azure DevOps Services
- About projects and scaling your organization
- Plan your organizational structure

Plan and track SAFe® programs and portfolios in Azure Boards

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Once you've configured your Agile tools to support SAFeⓇ, trace relationships can be created from Stories all the way up to Epics. Additionally, you can view progress from the portfolio, program, and feature team levels.

This article walks you through some of the basic tools you'll use to plan and track your SAFeⓇ programs and portfolios. Specifically, you'll learn how to quickly complete these tasks:

- Define epics, features, and stories
- Group or map stories to features, and features to epics
- Assign value streams
- Plan a sprint
- Review feature team progress
- Review program features
- Review portfolio epics

> **NOTE**
> This article is one of a set of Scaled Agile FrameworkⓇ tutorials that applies to Azure Boards and Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features and procedures are specific to the cloud or the latest version of Azure DevOps Server.

Define portfolios and epics

To get started, each team needs to add work items for each deliverable that supports the product vision. Each team can do that by working from their backlog and then mapping their work to create a hierarchy of work.

Or, they can develop a list of work items and import them using Excel. By specifying each team's area path for each work item, they create the mapped hierarchy.

The following sections indicate how to work from Excel or through the web portal.

Import a top-down plan by using Excel

You can import work items using Excel. As shown in the following image, Epics, Features, and User Stories are defined in a tree-list with their Value Area, Area Path, and a few tags defined. The State value is New.

	A	B	C	D	E	F	G	H	I	J
1	Project: Fabrikam Server: https://dev.azure.com/fabrikam Query: [None] List type: Tree									
2	ID	Work Item Type	State	Value Area	Title 1	Title 2	Title 3		Area Path	Tags
3		Epic	New	Business	Improve User Experience				\	universal applications
4		Feature	New	Business		Improve service operations			\Fiber Suite	
5		User Story	New	Business			Update and retest service code		\Fiber Suite\System	
6		User Story	New	Business			Cancel order form		\Fiber Suite\App	
7		User Story	New	Business			Review architecture framework		\Fiber Suite\App	
8		User Story	New	Business			Establish design criteria		\Fiber Suite\App	
9		User Story	New	Business			Review framework findings with service teams		\Fiber Suite\App	
10		Feature	New	Business		Integrate client app with IM clients			\Fiber Suite	
11		Feature	New	Business		Integrate client application			\Fiber Suite	
12		User Story	New	Business			As a <user>, I can select a number of elements		\Fiber Suite\System	
13		Feature	New	Business		Integrate client application with popular email clients			\Fiber Suite	
14		User Story	New	Business			Implement a factory which abstracts the email		\Fiber Suite\System	
15		User Story	New	Business			As a user, I can select a number of support case		\Fiber Suite\App	
16		Feature	New	Business		Perform user studies to support user experience imroveme			\Service Suite	enabler
17		Epic	New	Architectural	Framework to port applications to all devices				\	universal applications
18		Feature	New	Business		Emoticon feedback enabled in client application			\Service Suite	
19		User Story	New	Business			As a user, I can select an emoticon and add a sl		\Service Suite\Web App	
20		User Story	New	Business			Add animated emoticons		\Service Suite\Web Service	
21		Feature	New	Business		Migrate legacy code to portable frameworks			\Fiber Suite	
22		User Story	New	Business			Implement a factory that migrates legacy to por		\Fiber Suite\Migrate	
23		User Story	New	Business			As a developer, I can analyze a code base to det		\Fiber Suite\Migrate	
24		Epic	New	Business	Shared Personalization and State				\	universal applications
25		Feature	New	Business		Web pages			\Service Suite	
26		User Story	New	Business			Welcome back page		\Service Suite\Performance	
27		User Story	New	Business			Cancel order form		\Service Suite\Performance	
28		User Story	New	Business			Hello World web site		\Service Suite\Performance	
29		Epic	New	Architectural	Counter the Heartbleed web security bug				\	security
30		Feature	New	Architectural		Update software to resolve the Open SLL cryptographic cod			\Fiber Suite	
31		User Story	New	Business			Update and re-test suite code base affected by t		\Fiber Suite\Migrate	
32		User Story	New	Business			Update and re-test service code based affected		\Fiber Suite\Migrate	
33		Feature	New	Architectural		Scan all software for the Open SLL cryptographic code			\Fiber Suite	
34		User Story	New	Architectural			Scan all code base and identify the affected coc		\Fiber Suite\System	
35		User Story	New	Business			Update affected code		\Fiber Suite\System	
36		Feature	New	Architectural		Ensure all compliance requirements are met			\Service Suite	enabler
37		Epic	New	Business	Service interfaces to support REST API				\	
38		Feature	New	Business		Service support			\Service Suite	
39		User Story	New	Business			Lookup service outages		\Service Suite\Web Service	
40		Feature	New	Business		Migrate impact of low coverage areas			\Service Suite	
41		Epic	New	Architectural	Improve DevOps Continuous Pipeline Delivery				\	enabler

Once you publish the list, work item IDs are assigned. Each team can then start working with their work items, adding details, and assigning work to team members and to sprints.

To learn more about bulk update, see the Bulk add or update work items section provided later in this article.

Document Portfolio Vision and Strategic Themes

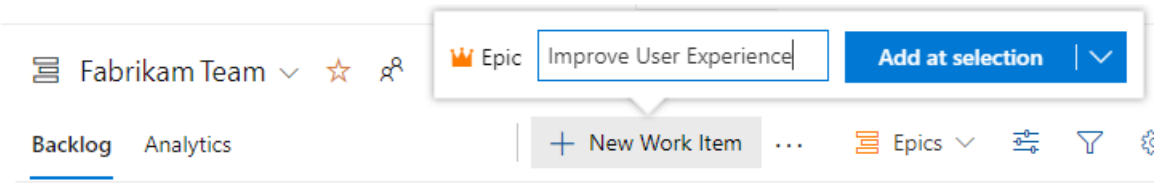
We recommend you use the project wiki to capture and share the Portfolio Vision, Strategic Themes, and other information relevant to your programs. Consider defining and sharing the following information:

- How to use tags or custom fields to specify value streams
- Taxonomy terms as described in Scale Agile to Large Teams
- How release trains and sprints are used
- Key milestones and events
- Customer-centric programs

Information can be easily shared and updated by any member of the team. To learn more about using the project wiki, see About Wikis, READMEs, and Markdown

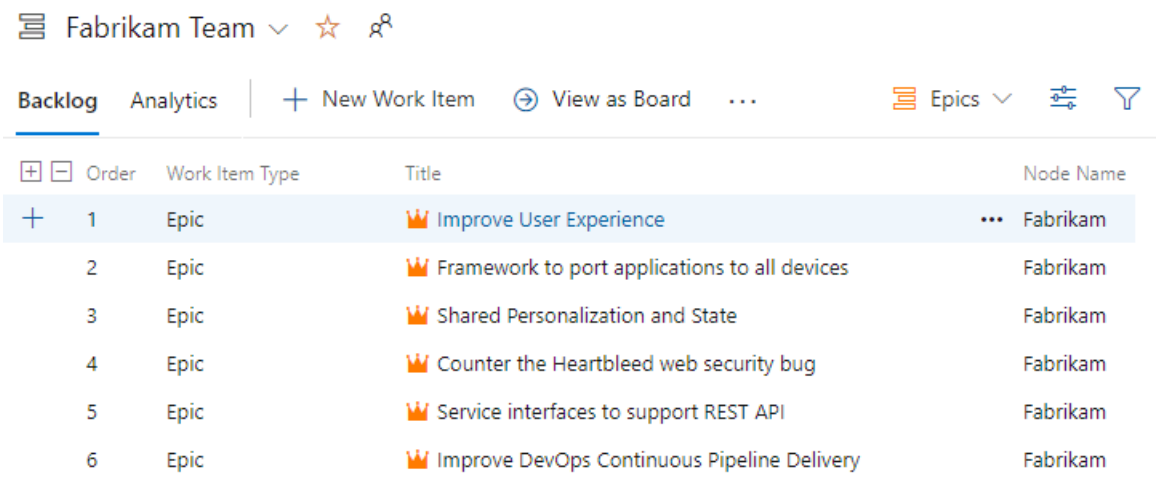
Define and prioritize the Epics portfolio backlog

1. Open the Fabrikam team's Epics portfolio backlog and choose **New Work Item**. Enter the title of the Epic and choose **Add at selection**.



2. Continue adding epics as needed by continuing to type in their titles. You can add details later.

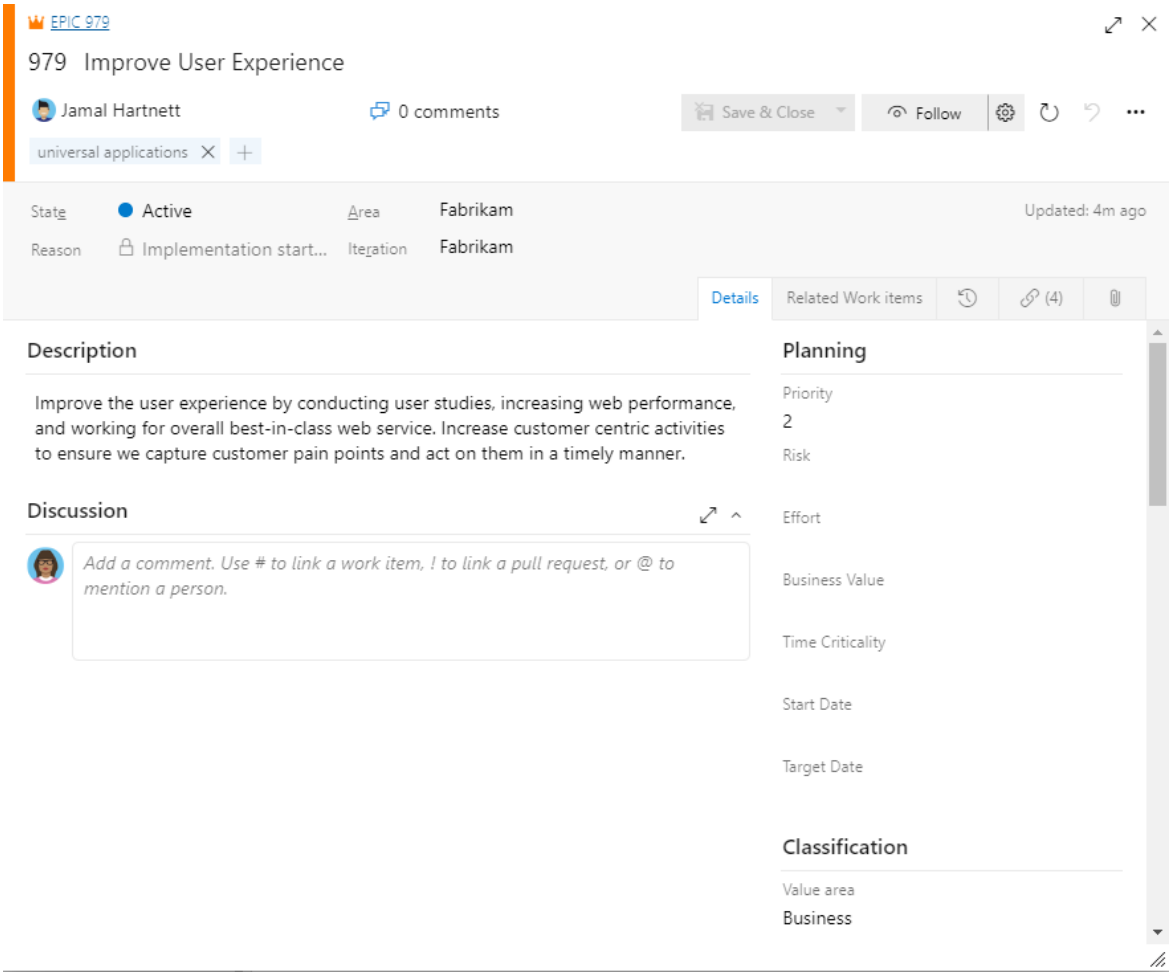
   When finished, you should have a list of epics as shown:



3. As needed, you can drag and drop epics within the list to reflect their priority.

4. Define details. Choose each work item title to open the form. Add information to the Description, assign to an owner, add tags, and choose the Value Area.

   Here we assign the first epic to Jamal, add a Description, and specify Business for the Value Area.



To update several work items in bulk, see Bulk add or update work items section provided later in this article.

NOTE

Since an important aspect of SAFe® is to map work to business or architecture goals, you'll want to make sure to set the **Value Area** = *Architectural* for any Feature mapped to an architecture epic. Because the default choice is *Business*, you don't have to change the field for any item that supports a business epic. You can also add tags to track the investment.

The same principles apply to Stories in progress. You can map them to Features, change the **Value Area** to *Architecture* for work that you do to support architectural epics, and add tags for tracking themes.

Define programs and features

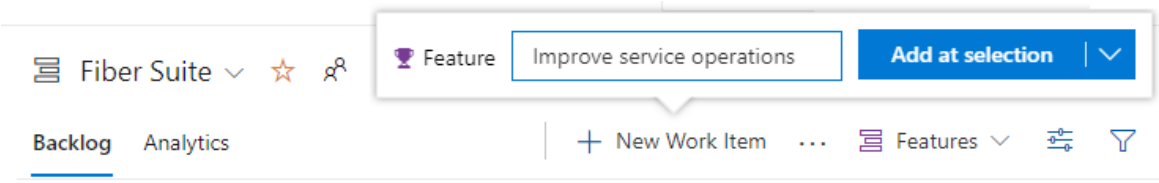
In a similar fashion to portfolio teams, program teams work from their Features backlog and define the features that support the Epics they're tasked with delivering.

Each program manager defines and prioritizes their features. Here, we show the Fiber Suite team add Features.

Define the Features backlog

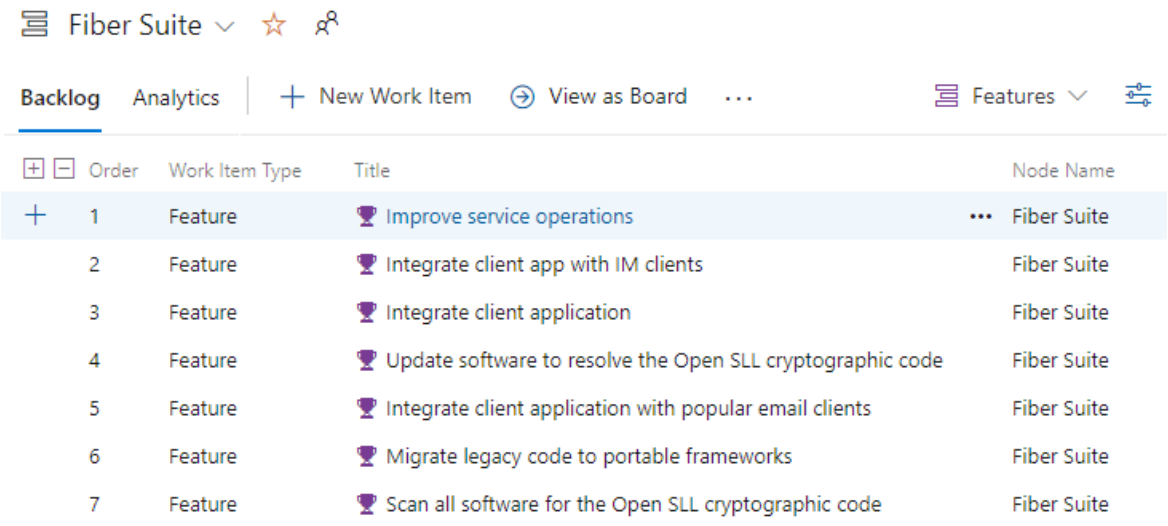
1. Open the Fiber Suite team's Features backlog and choose **New Work Item**.

   Enter the title of the Feature and choose **Add at selection**.



2. Continue adding Features as needed by continuing to type in their titles. You can add details later.

   When finished, you should have a list of Features as shown. The Node Name indicates the last node in the area path specified for the work item. By adding Features from the team's Feature Backlog, the Area Path is automatically assigned the team's Default Area Path. For the Fiber Suite, that is Fabrikam\Fiber Suite.



3. As needed, you can drag and drop Features within the list to reflect their priority.

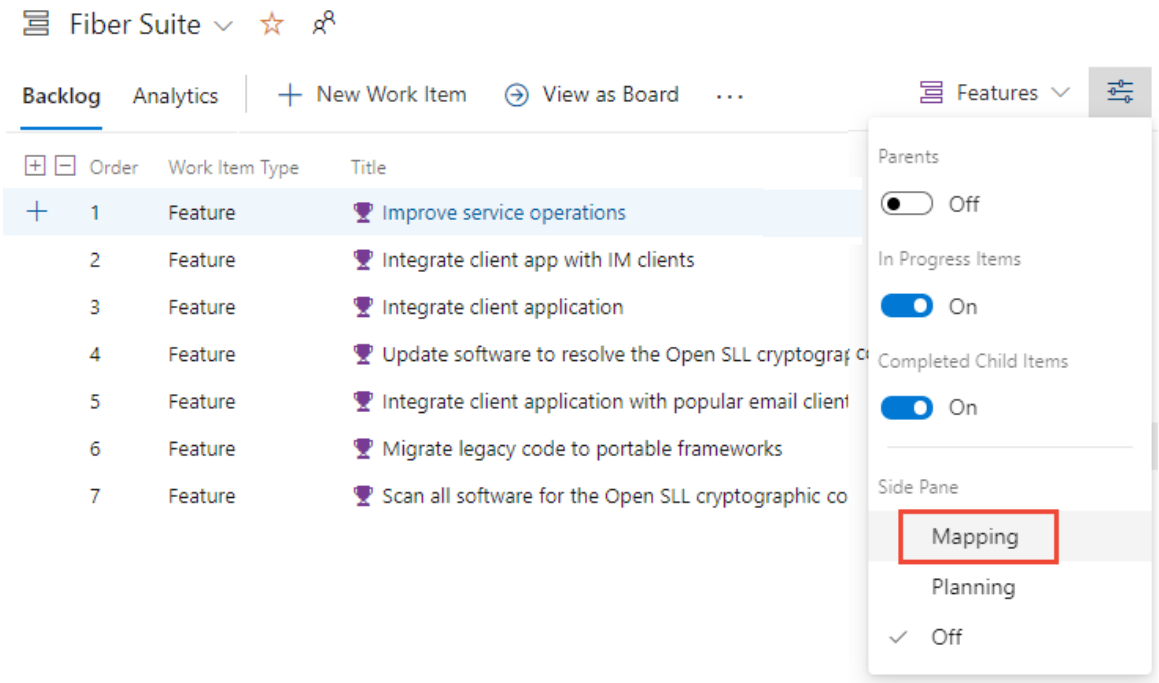
4. Define feature details. Choose each work item title to open the form. Add information to the Description, assign to an owner, add tags, and choose the Value Area.

Map Features to Epics

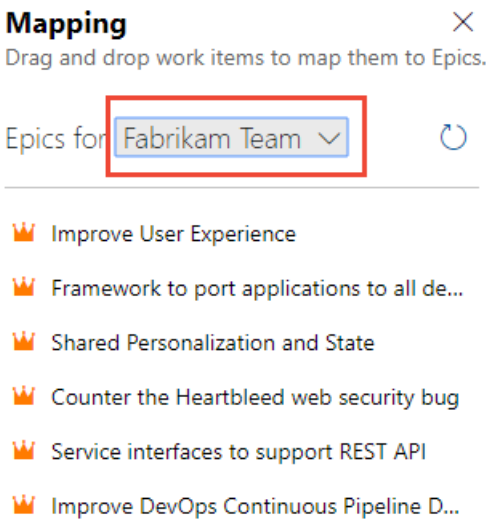
In this next step, you'll map each Feature to its parent Epic. The Mapping tool quickly creates parent-child links

between Epics and Features.

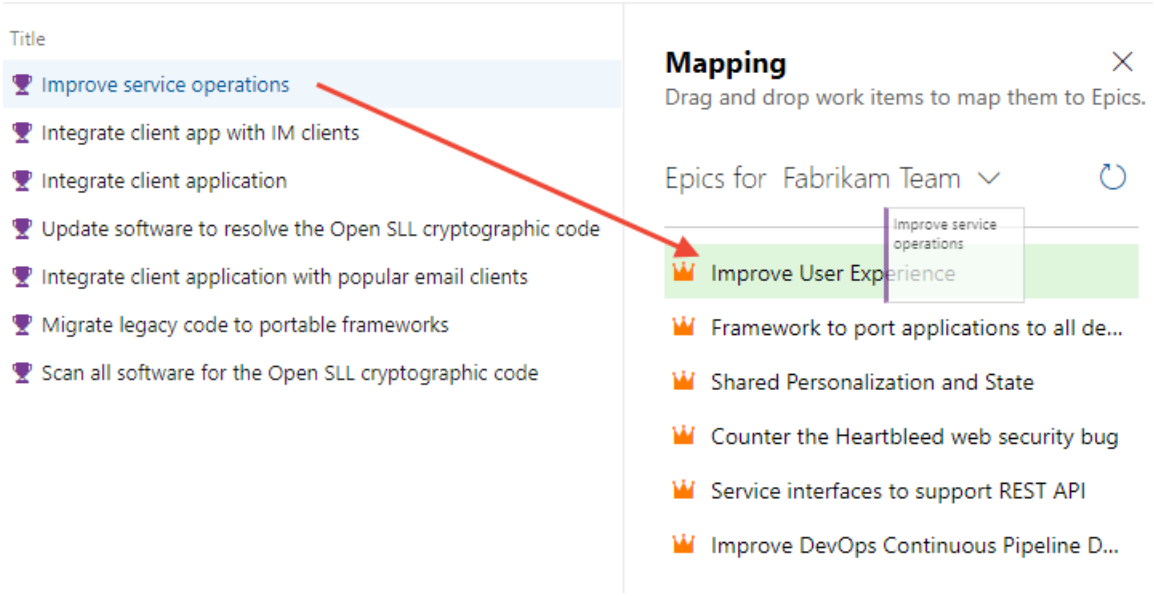
1. From the Features backlog, choose the view options icon and select **Mapping**.



2. Choose the Fabrikam Team's backlog of Epics.

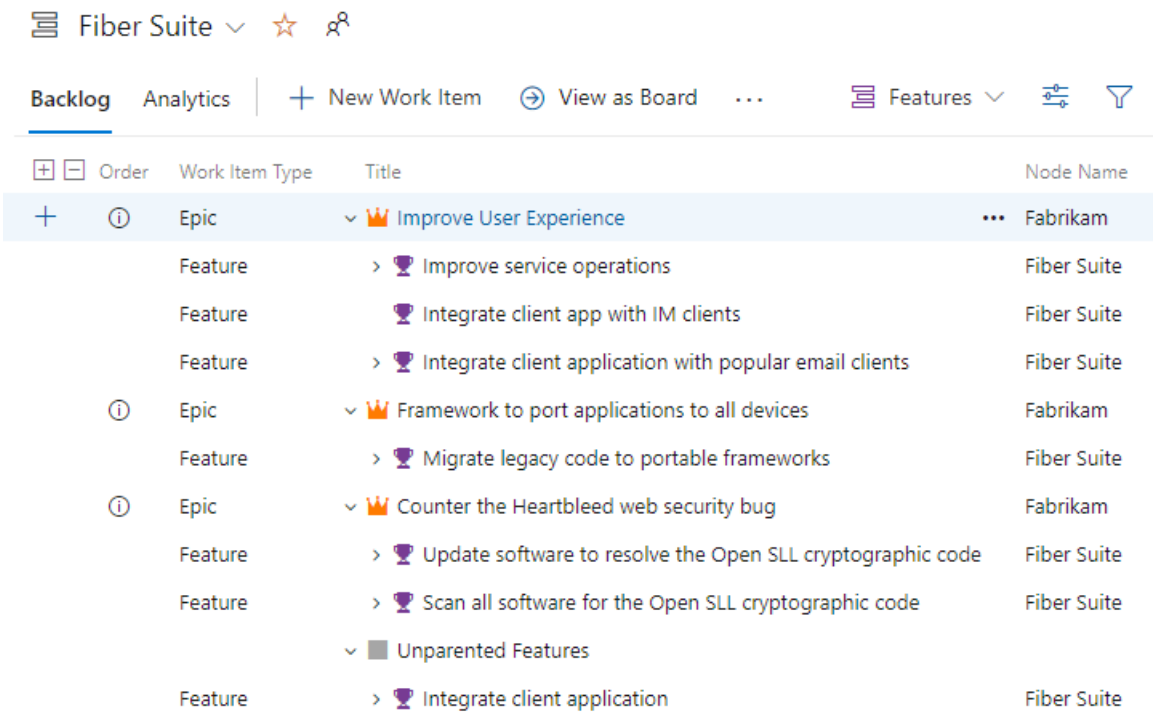


3. One by one, drag each Feature onto its parent Epic.



4. When finished, choose the view options icon and enable **Parents** and turn **Mapping** off.

   Your list should look something like that shown in the following image. The info icon appears next to each Epic, indicating that the work item is owned by another team than the one currently selected.



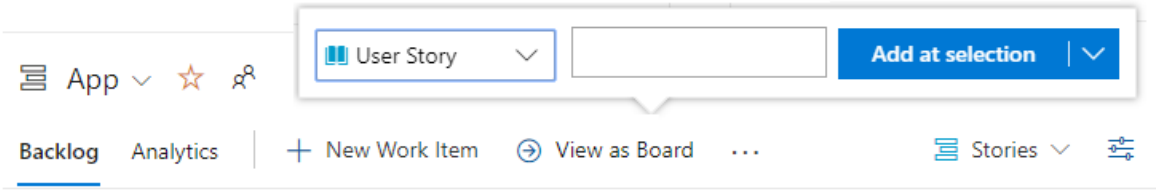
Define Agile feature team deliverables

In a similar manner to portfolio and program teams, each Agile feature team add Stories to their backlog to support the Features assigned to them.

Define the Stories backlog

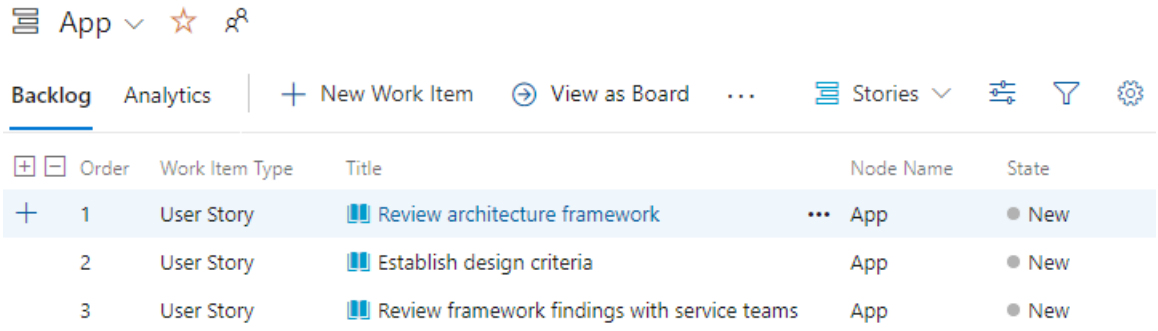
1. Open the App team's Stories backlog and choose **New Work Item**.

   Enter the title of the User Story and choose **Add at selection**.



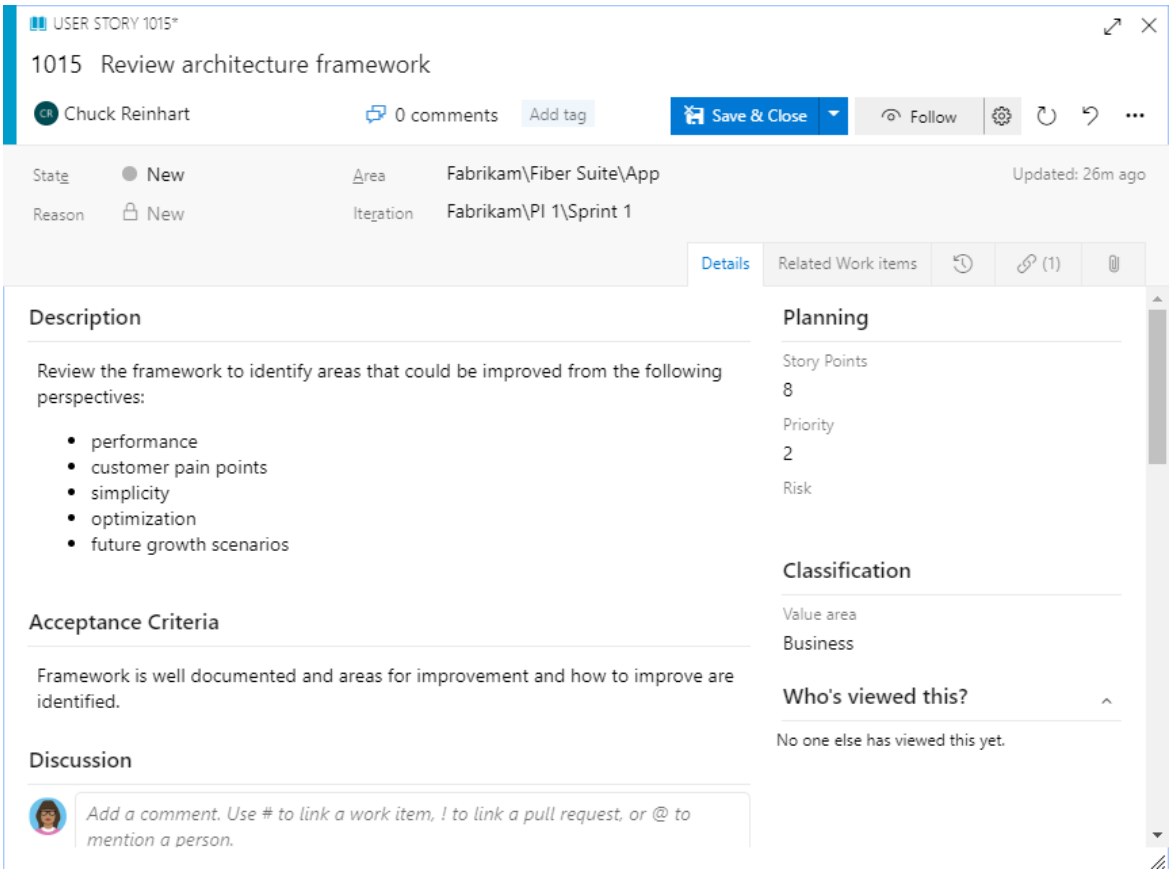
2. Continue adding User Stories as needed by continuing to type in their titles. You can add details later.

   When finished, you should have a list of Stories as shown:



3. As needed, you can drag and drop User Stories within the list to reflect their priority.

4. Define story details. Choose each work item title to open the form. Add information to the Description and Acceptance Criteria, assign to an owner, add tags, specify the Story Points, and choose the Value area.

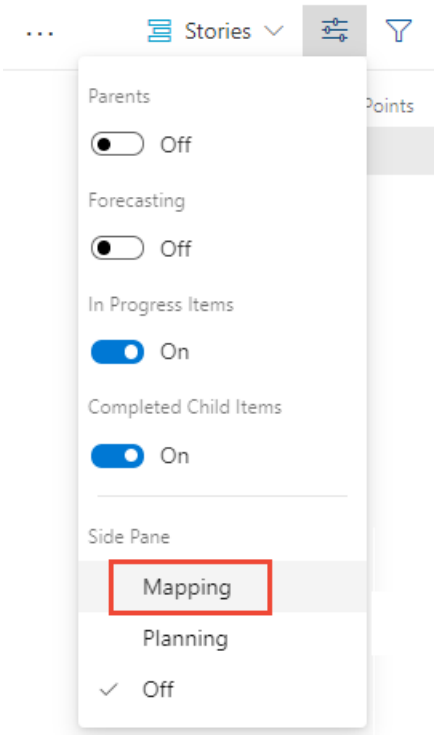


   To update several work items in bulk, see Bulk add or update work items section provided later in this article.

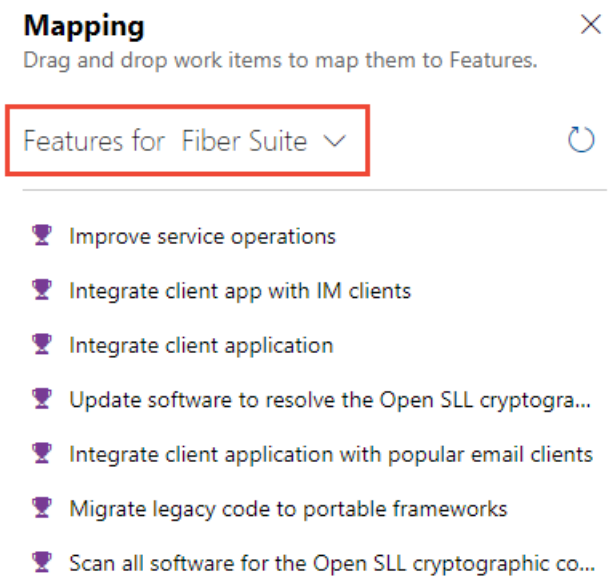
Map Stories to Features

Just as you earlier mapped each Feature to its parent Epic, now you'll map each User Story to its parent Feature. The Mapping tool quickly creates parent-child links between Features and User Stories.

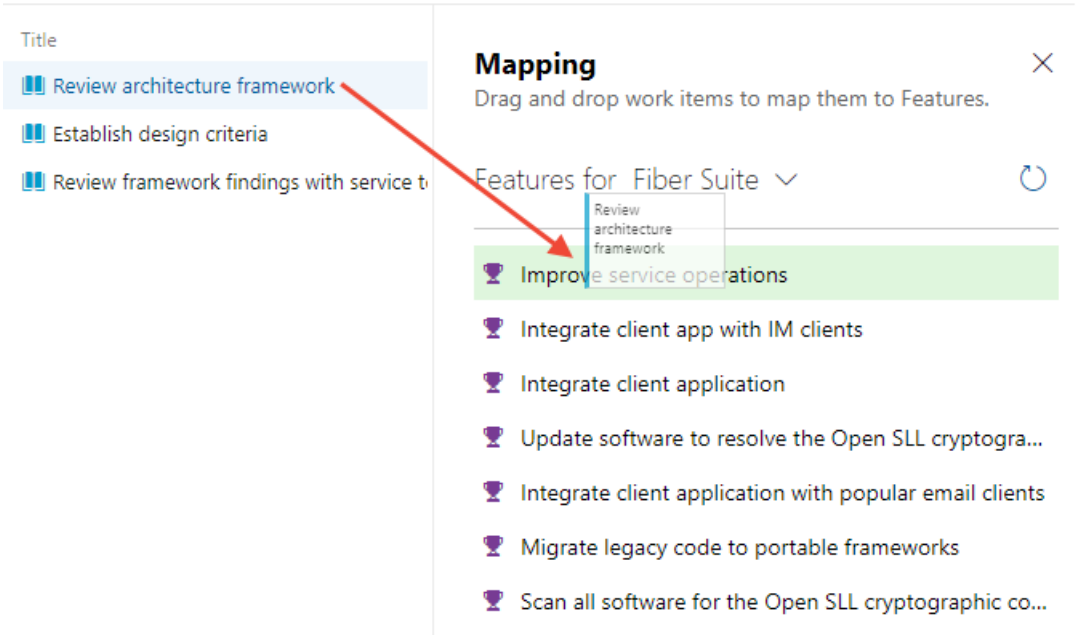
1. From the Stories backlog, choose the view options icon and select **Mapping**.



2. Choose the Fiber Suite's backlog of Features.

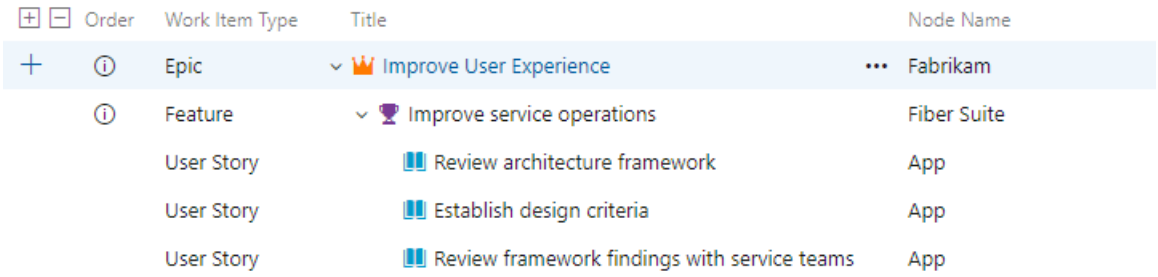


3. One by one, drag each User Story onto its parent Feature.



4. When finished, choose the view options icon and enable **Parents** and turn **Mapping** off.

   Your list should look something like that shown in the following image. The info icon appears next to each Epic and Feature, indicating that the work item is owned by another team than the one currently selected.



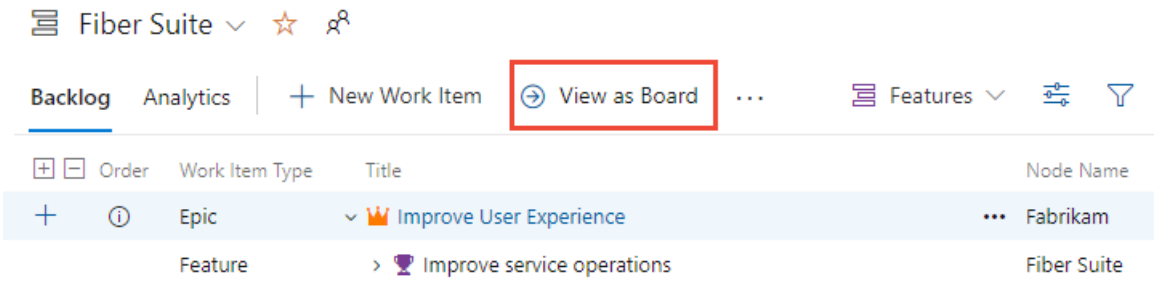
View Kanban boards and update status

Each team's Kanban board provides a visual interactive space for the team to plan and update status. With it, your team can track the critical information they need by seeing which work items are in progress, where the bottlenecks are, who work is assigned to, and more.

For an overview of Kanban board tools and usage, see About Boards and Kanban. Each team can customize the boards columns, card fields, card styles, and more. To learn more, see Customize your boards.

Open a Kanban board

You open the Kanban board from any backlog by choosing the **View as Board** link.

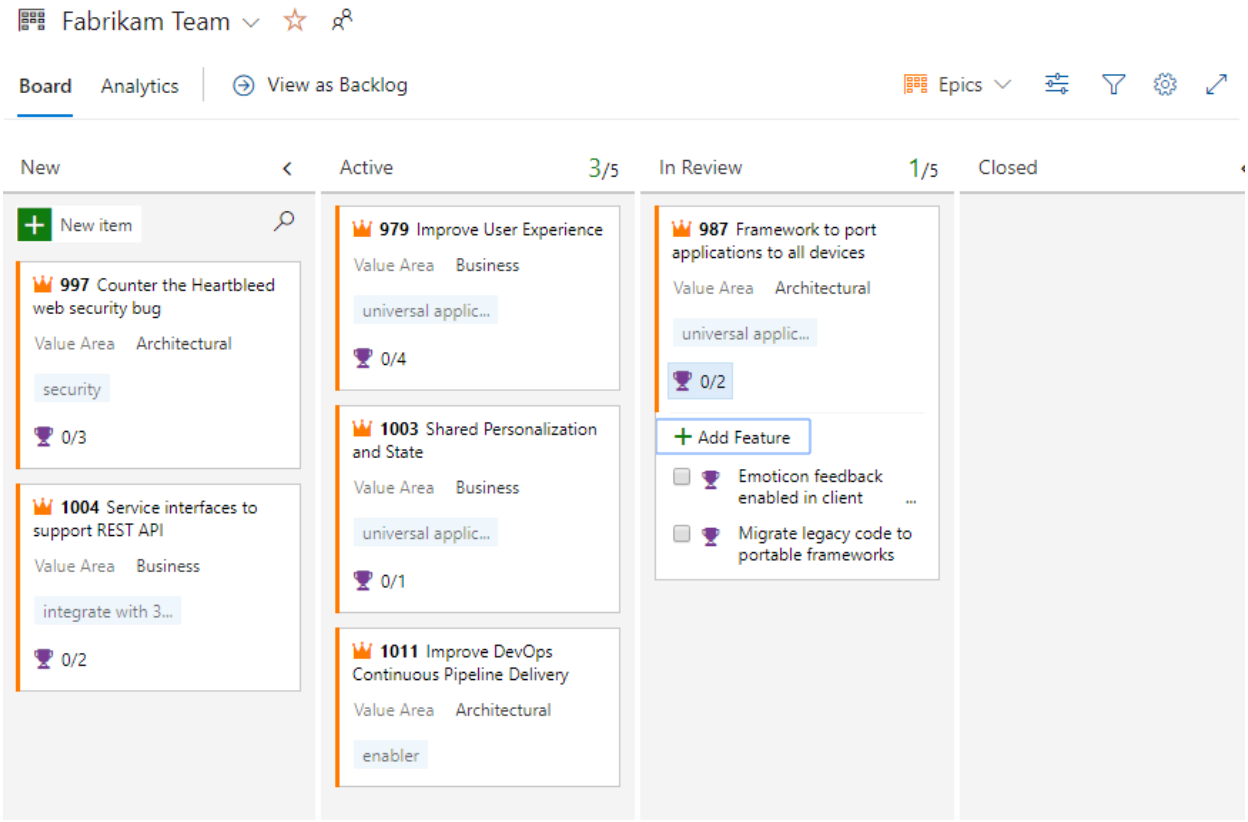


Each board supports the following tasks:

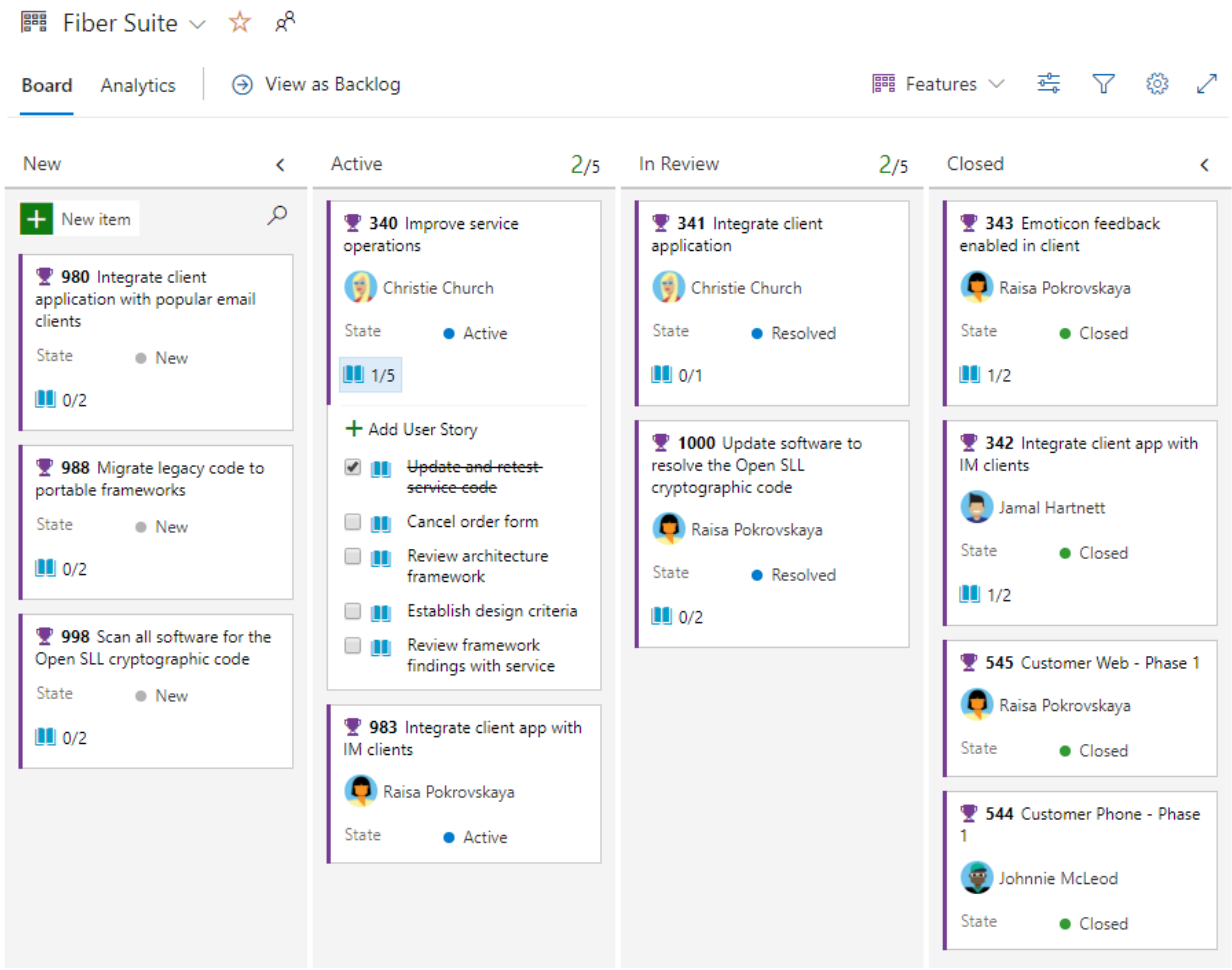
- Add work items and child work items
- Update status through drag-and-drop
- Add and specify the labels of each column
- Configure the display of cards, add tags, fields, and apply rules
- Configure swimlanes and set WIP limits
- Assign values or update fields displayed on cards
- Filter based on keywords and key fields

Portfolio Kanban board

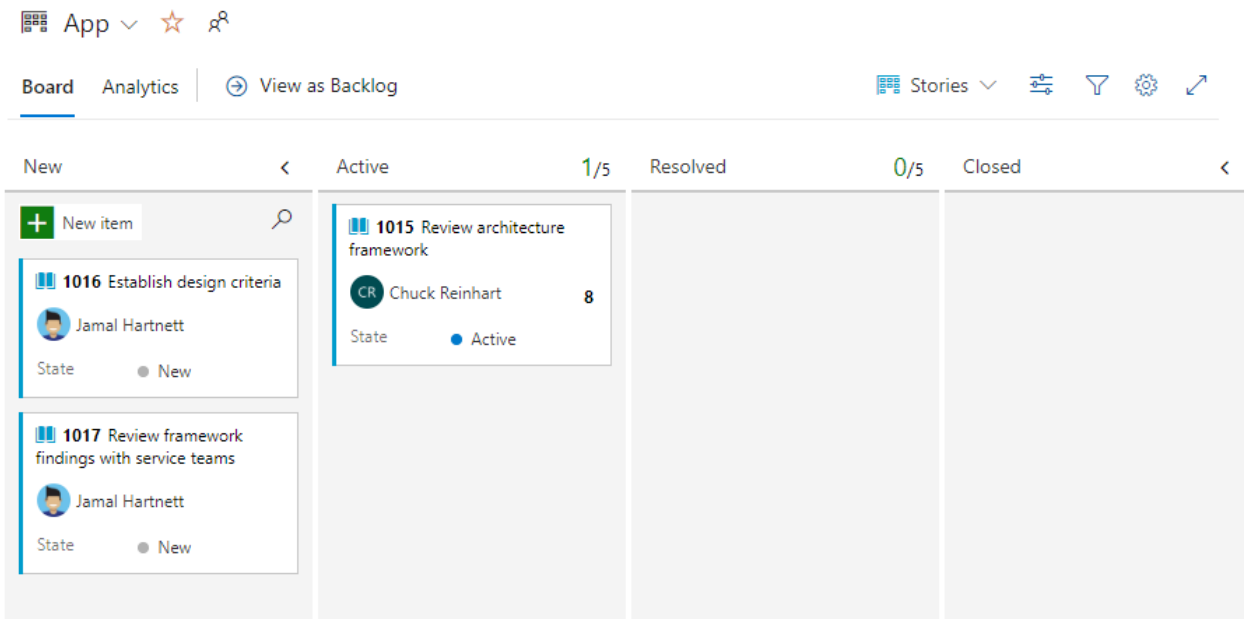
Child items of Epics are listed within each card. You can expand and collapse the list of child items.



Program team Kanban board



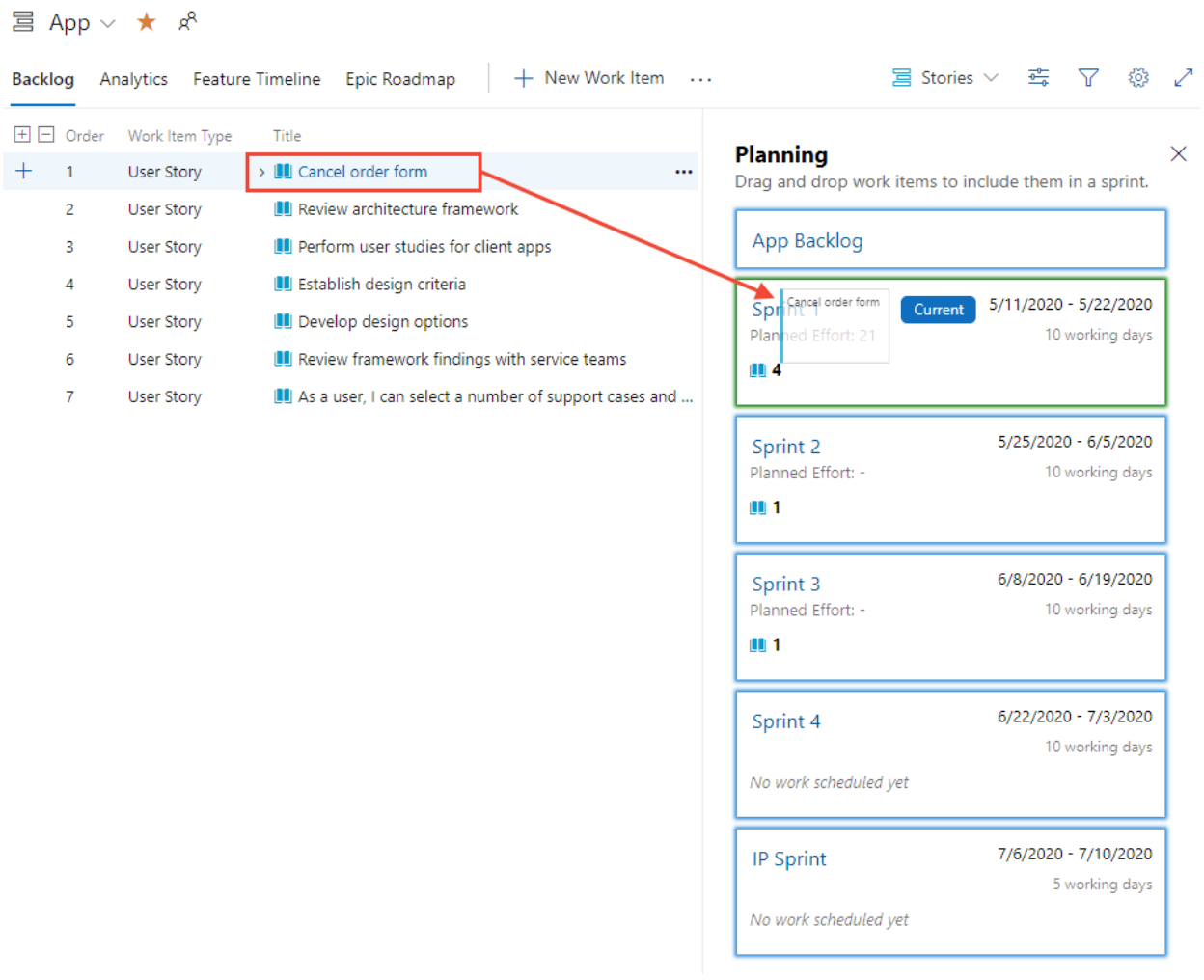
Agile team Kanban board



Plan a sprint

Each Agile release team can plan their sprints using the sprint planning tools. Starting with sprint planning, each team can move their backlog items to a sprint using the Planning tool.

As shown in the following image, the App team plans their sprints.

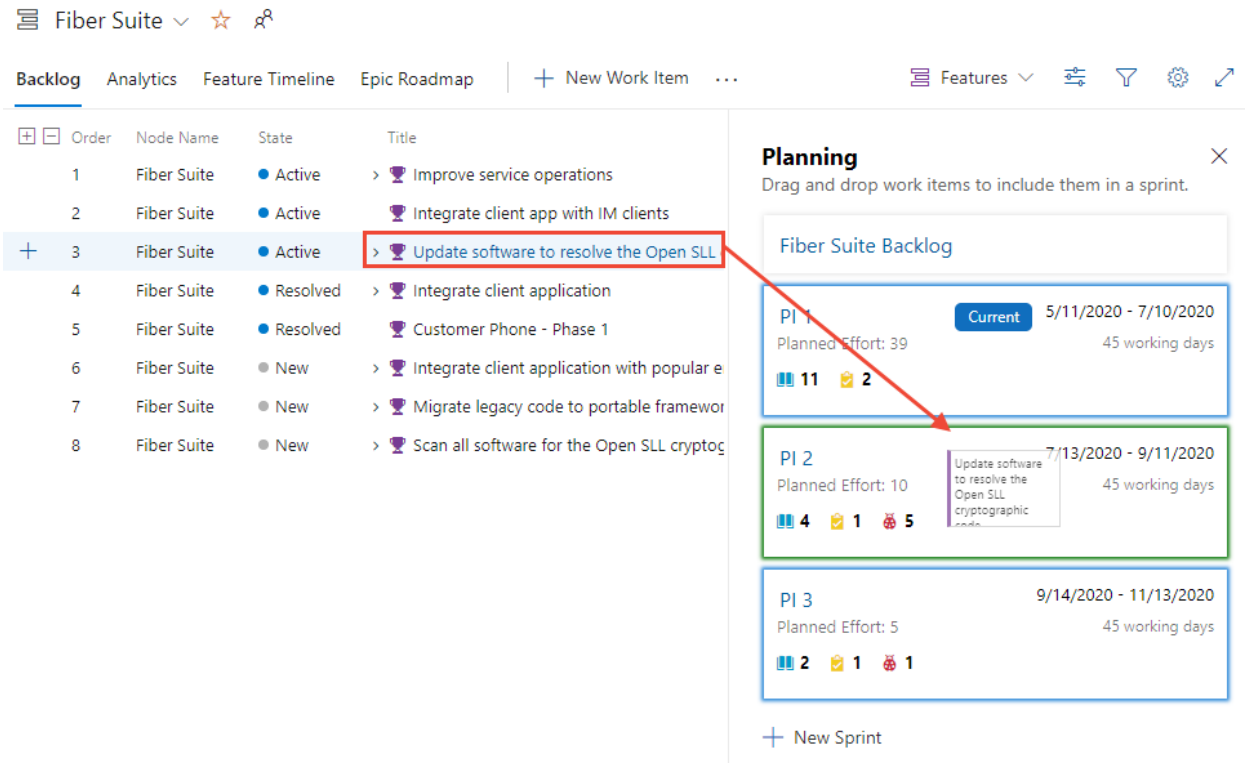


To learn more about planning and conducting sprints, see the tutorials for Plan and work a sprint.

Plan a release train

In a similar manner to Agile release teams sprint planning, each program team can plan a release train.

Here we show the Fiber Suite program team plan the releases for their features.



Tips for working with work items

When creating and updating work items, understand what you can and can't do.

- You can only assign a work item to one team member; if you need to assign similar work to more than one user, copy the work item and make the assignment
- Can only assign a work item to a single Area Path and Iteration Path
- The quickest way to add backlog work items is from the backlog or Kanban board
- You can use work item templates to quickly fill in work item fields

Bulk add or update work items

Common fields you may want to bulk update include:

- Area Path
- Assigned to
- Iteration Path
- Tags (Add or Remove)
- Value Area

You can bulk update and change the work item type from the web portal. Or, if needed, you can remove of delete work items. See the following articles:

- Bulk modify (web)
- Move or change work item type
- Remove or delete work items

Also, you can bulk add or update work items with the following tools:

- Bulk add or modify work items (Excel)
- Bulk import or update work items (CSV)

Try this next

Review progress, delivery plans, and roadmaps

Related articles

- About work items
- About Boards and Kanban
- About About Sprints, Scrum, and project management
- Plan and work a sprint
- Use work item templates
- Track your work when assigned to two or more teams

View SAFe® progress, roadmaps, and metrics

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With team's configured and working backlogs and boards, you're ready to start viewing and monitoring progress.

Azure Boards provides many in-context charts and dashboard widgets that allow you to monitor and report on various SAFe® metrics. Specifically Azure Boards provides access to the following tools to support teams in deriving SAFe® metrics and monitoring and reporting progress.

- Roll up columns on backlogs
- In-context reports
- Managed query charts such as pie, bar, stacked bar, trend, and pivot
- Dashboard widgets
- Team and project dashboards
- Analytic Views to support Power BI reports
- OData queries to use with Power BI reports

For an overview of these tools, see About dashboards, charts, reports, & widgets. Another backlog tool is Forecast which teams can use in their iteration planning.

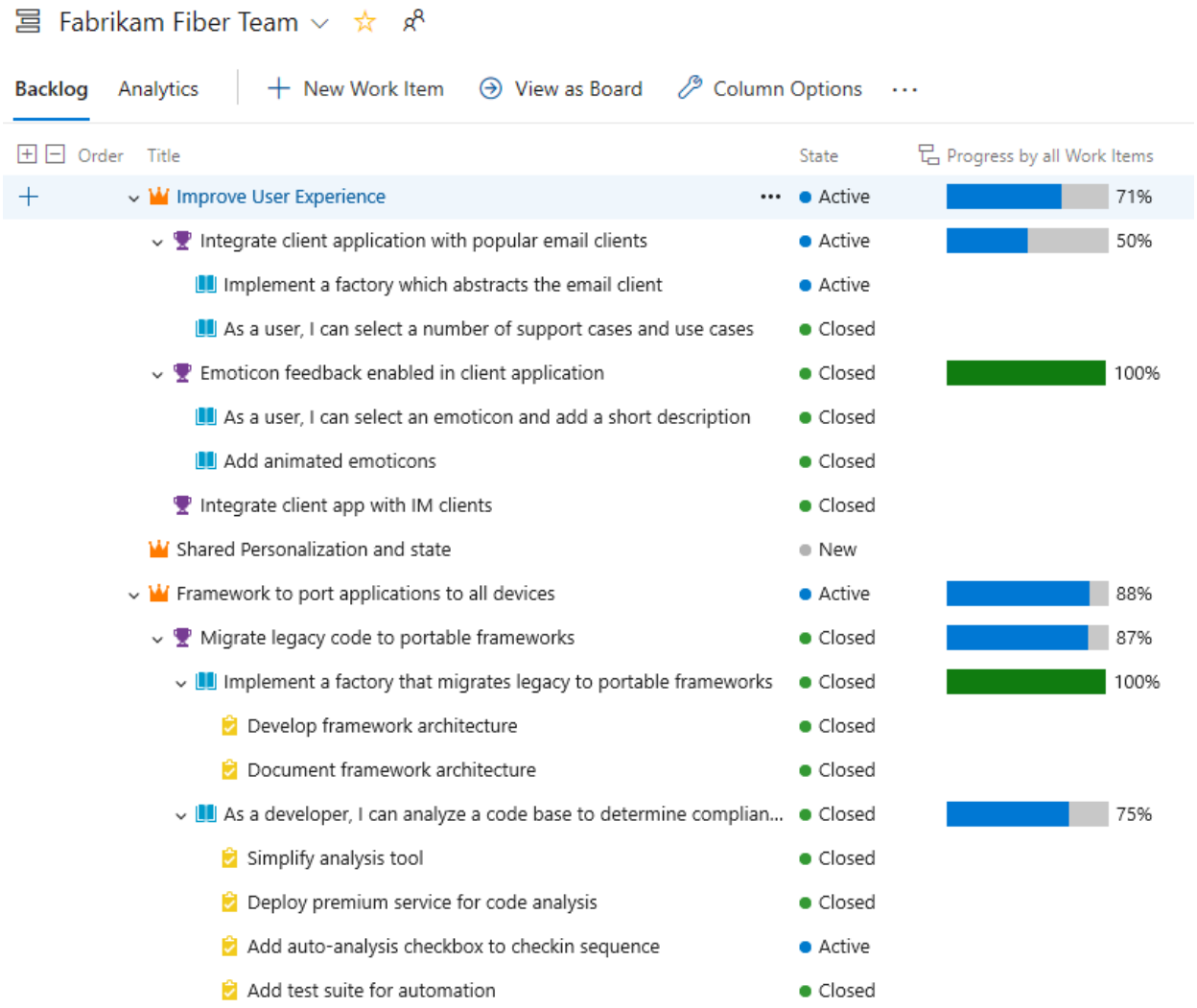
In this tutorial, we illustrate some of the out-of-the-box charts and widgets that you'll have instant access to monitor some of these key SAFe® metrics

- Progress reports
- Cumulative Flow Diagram
- Lead time and cycle time charts
- Iteration planning, team velocity, and forecast

> **NOTE**
> This article is one of a set of Scaled Agile Framework® tutorials that applies to Azure Boards and Azure DevOps Services. Most of the guidance is valid for both the cloud and on-premises versions. However, some of the features and procedures are specific to the cloud or the latest version of Azure DevOps Server.

View progress rollup

Quick progress views are available from each team's backlog through rollup columns. Here's an example that shows progress based on completion of child work items.



Other rollup options include progress by specific work item types, progress by story points, count of work items, or sum of a numeric field. To learn more, see Display rollup progress or totals.

View team velocity

Each team has access to their velocity through the in-context velocity report. These reports show a bar chart count of planned, completed, completed late, and incomplete work items for the last six or more iterations. As shown in the example below, the chart also provides the average velocity calculated for the number of iterations shown.



This average can be used to forecast work by plugging it into the forecast tool.

Use the Forecast tool

By assigning Story Points to each User Story, a team can determine how much work they can complete using the Forecast tool. For details on its usage, see Forecast your product backlog.

Forecasting based on velocity of 20

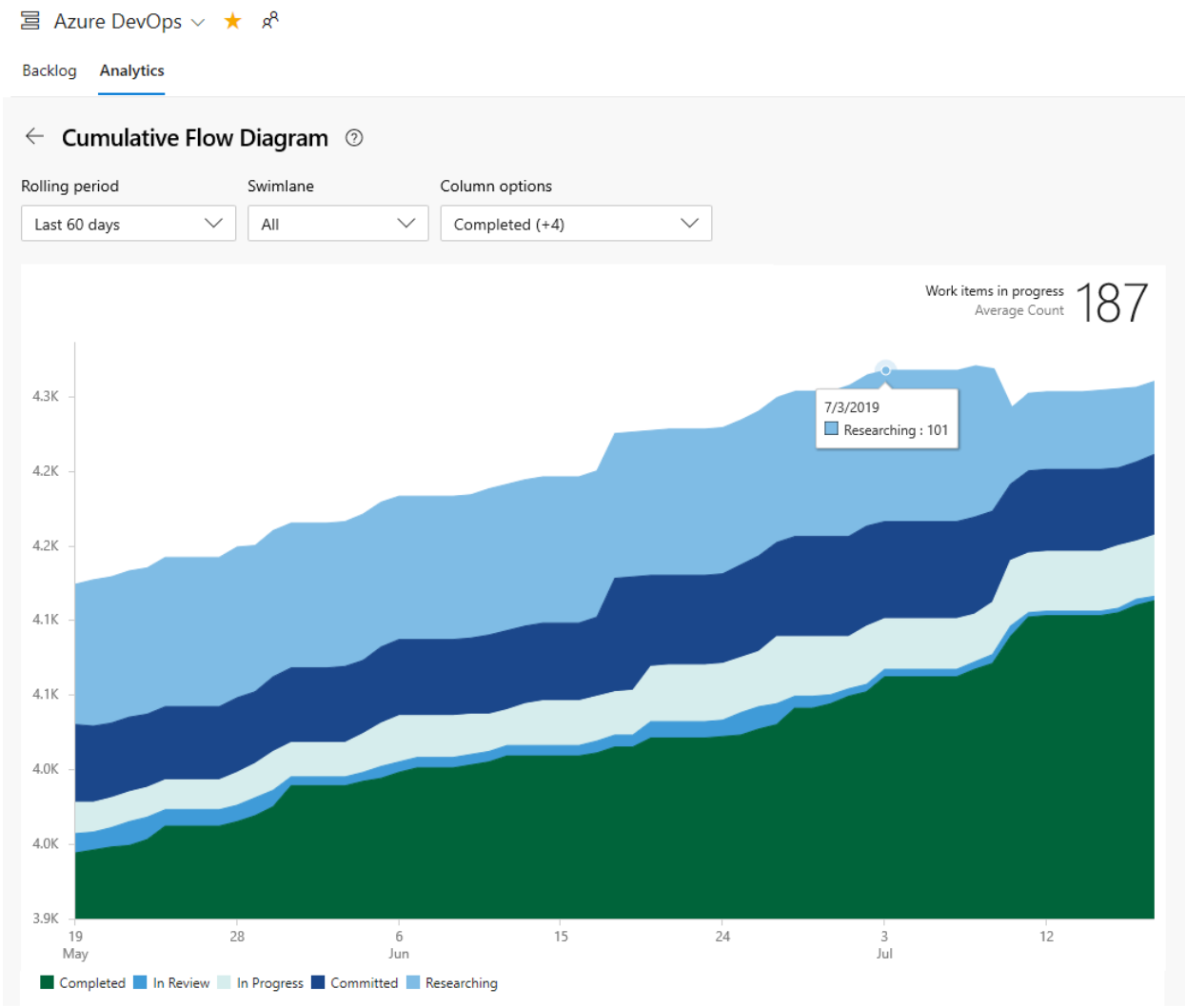
Forecast	Order	Work Item Type	State	Story ...	Title
+	1	User Story	New	8	Hello World Web Site
Iteration 2	2	User Story	New	5	Research architecture changes
	3	User Story	New	8	Phone sign in
	4	User Story	New	8	Interim save on long form
Iteration 3	5	User Story	New	8	Request support
	6	Bug	New	8	Slow response on the form
	7	User Story	New	8	Add info form
	8	User Story	New	5	Change initial view
Iteration 4		Task	New		Design doc
	9	Bug	New	5	Secure sign-in
	10	User Story	New	3	Welcome back
	11	User Story	New	3	Resume
Iteration 5	12	User Story	New	5	Switch context issues
	13	User Story	New	8	Locator interface
	14	User Story	New	8	Check service status
	15	User Story	New	3	Pause
Iteration 6	16	User Story	New	4	Resolve sign-in issues

View the Cumulative Flow Diagram (CFD)

Each Azure Boards backlog and board provide configurable CFD views. So each team at every level of SAFe®

implementation can monitor progress using these built-in charts.

The following image shows an example CFD chart for User Stories with all Kanban columns displayed.



Teams can use their CFD to identify bottlenecks and monitor the batch size of work in their various Kanban states.

In-context CFD charts are quickly accessible from each backlog and board view. Also, CFD charts can be added to team and project dashboards. To learn more, see View/configure a Cumulative Flow Diagram.

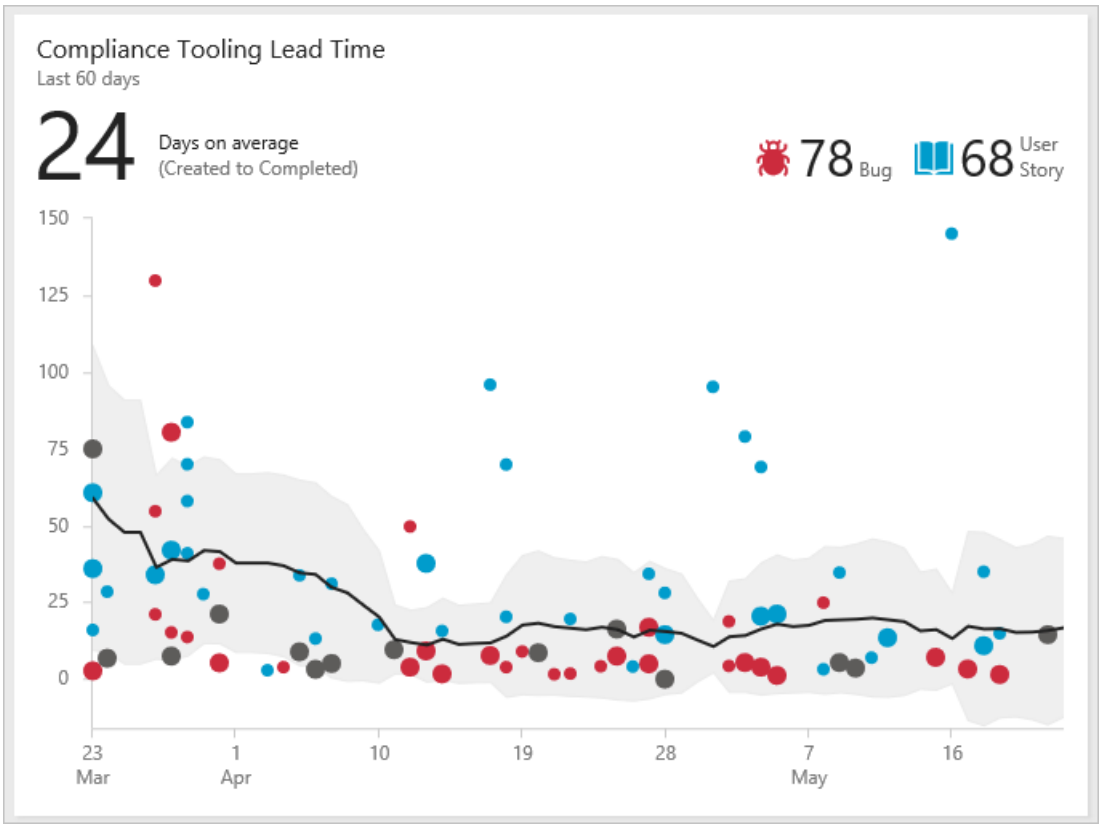
View Lead Time and Cycle Time charts

Other metrics that teams use are derived from the Lead time and cycle time charts. These charts can be added to a team dashboard and monitored to learn the following information:

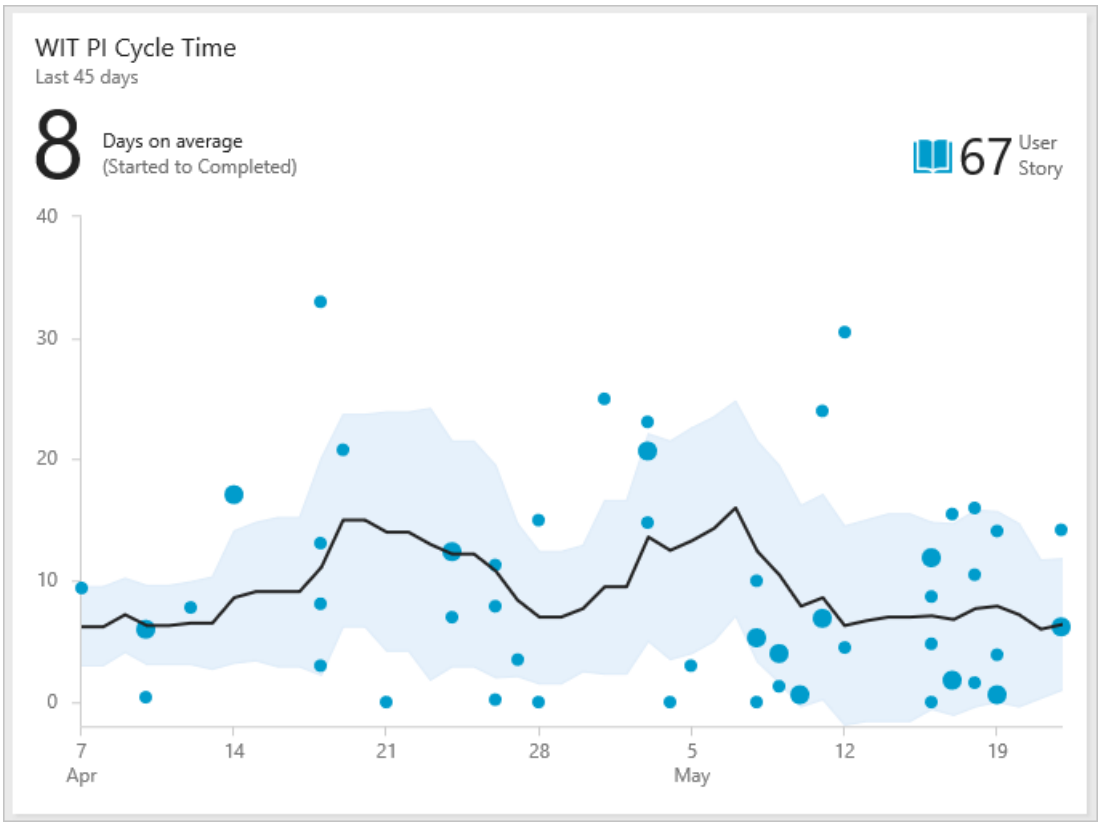
- Lead time: Days on average to complete deliverables from date created
- Cycle time: Days on average to complete deliverables from date work started
- Number of outliers

Both Lead Time and Cycle Time widgets display as scatter-plot control charts. They display summary information and provide several interactive elements. To learn more, see Cumulative flow, lead time, and cycle time guidance.

Example Lead Time widget



Example Cycle Time widget

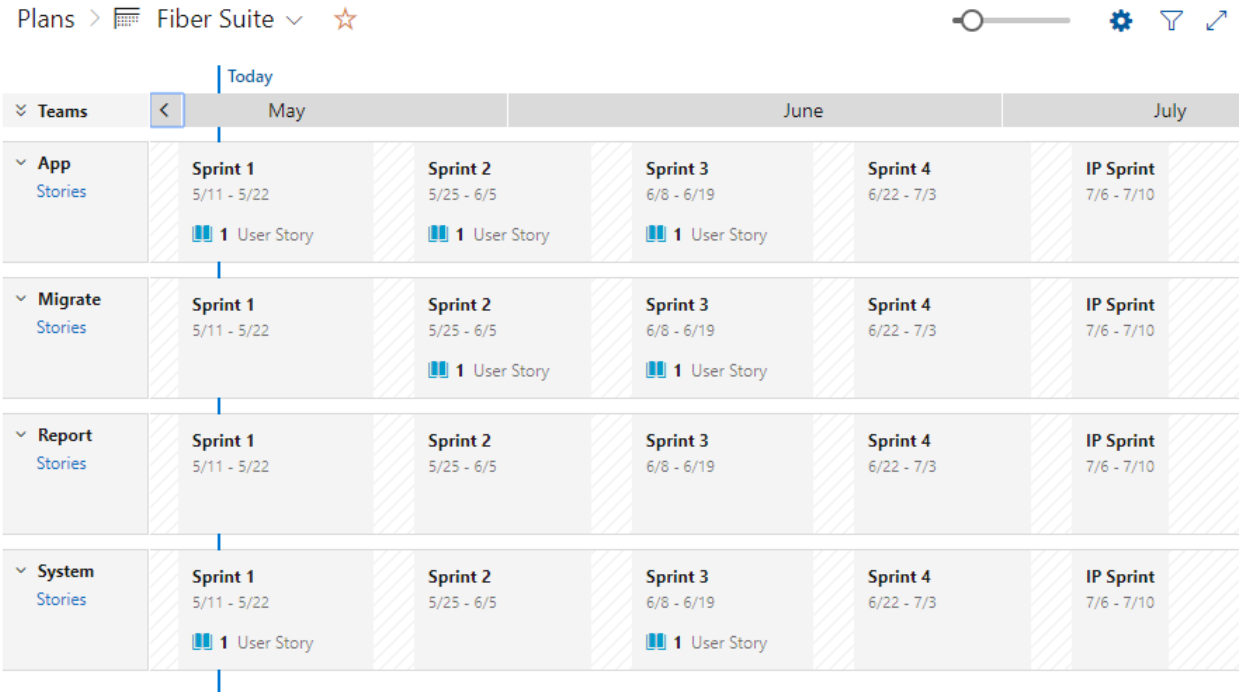


View and update roadmaps

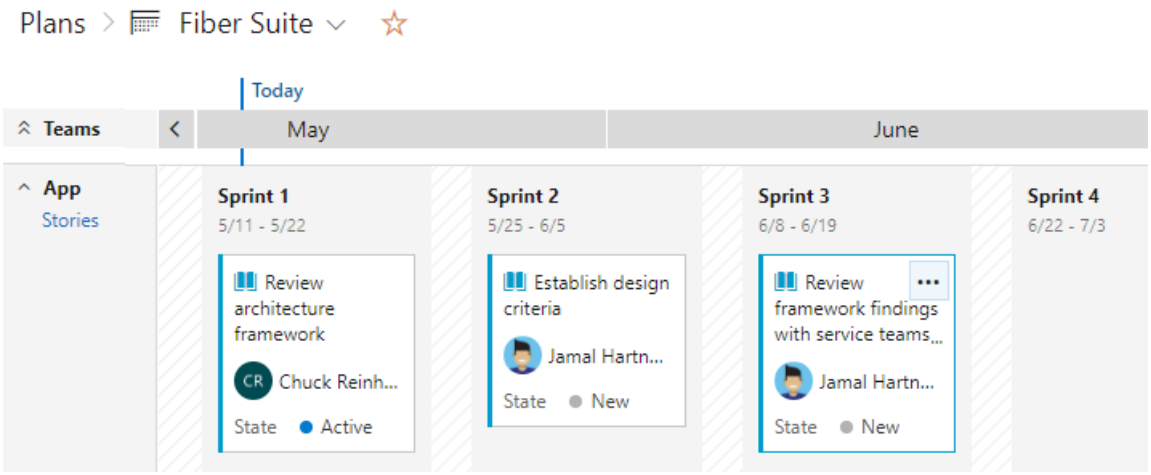
You can review roadmaps of SAFe® deliverables using the Delivery Plans, Feature Time, and Epic Roadmap tools. Delivery Plans are fully configurable to show the teams and work item types of interest.

Review feature team Delivery Plans

Program teams can review roadmaps of the deliverables of their Agile Release Teams. As an example, the following image shows a snapshot of the Fiber Suite teams story deliverables.

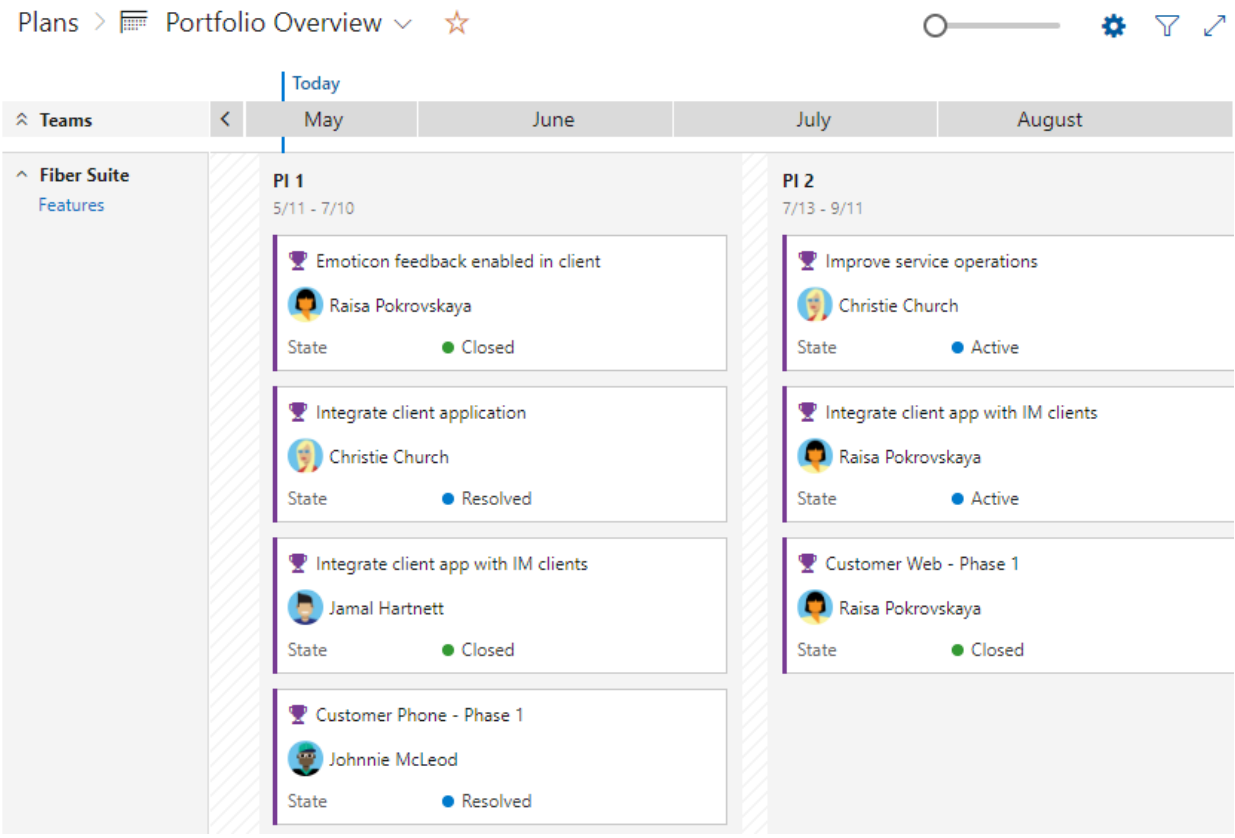


You can expand each feature team to see details. Story deliverables are assigned to the PI 1 sprints. Delivery Plans are fully interactive, allowing you to drag and drop work items to update their sprint assignments, or open work items to update fields, add comments, and other information.



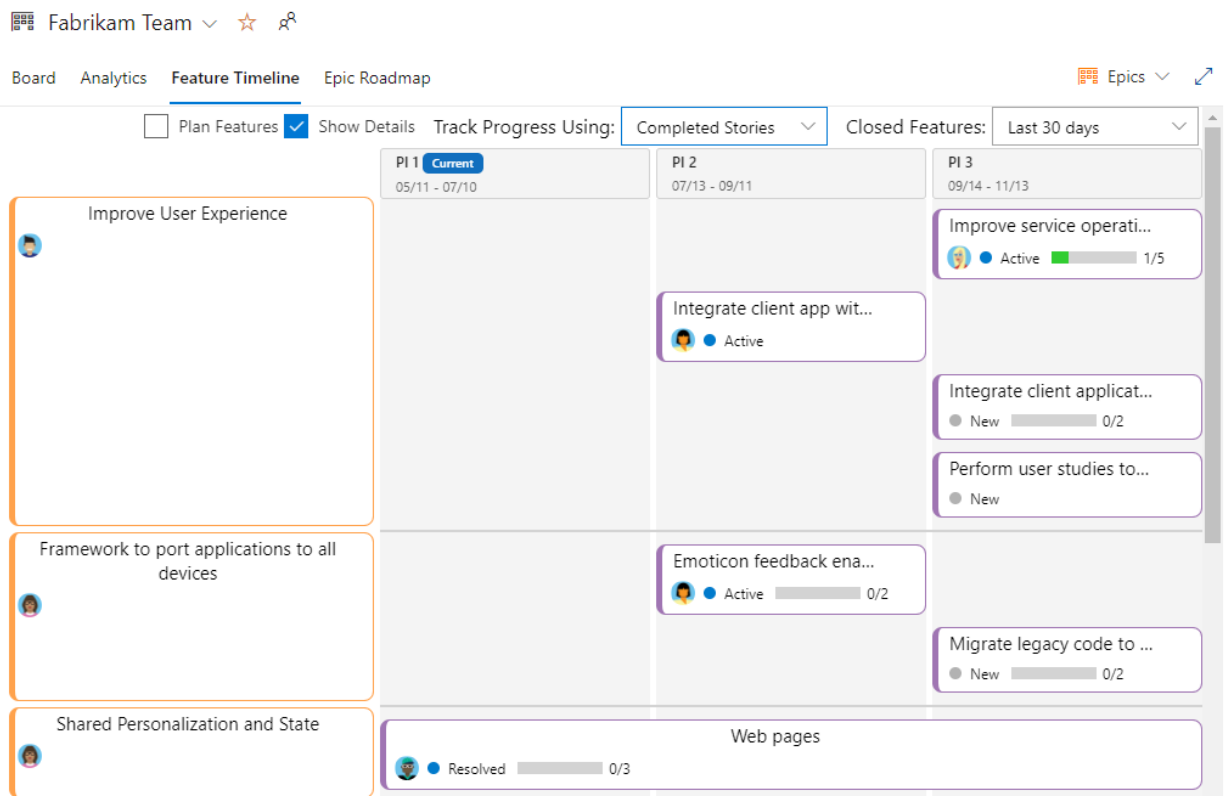
Review the portfolio features deliverable

Portfolio teams can review the Features under development by their program teams. For example, Features under development by the Fiber Suite team are shown in the following delivery plan view. The Features under development show up under the Program Increment timeboxes.



Review feature timeline roadmaps

The feature timeline tool provides another view into progress of deliverables. Here we show the Fabrikam Team's Epics as shown in the Feature Timeline tool. Progress bars are configurable based on completed stories or effort.



Try this next

Sign up for Azure Boards for free

Related articles

- Review team Delivery Plans
- View portfolio progress with the Feature Timeline

Promote an Agile culture within your team

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

> **NOTE**
> Are you new to Agile? Learn more about Agile Culture and Scaling Agile to Large Teams.

As your team grows, you want your tools to grow with it. And, if you're an enterprise adopting Agile methodologies, you want your Agile tools to support the business goals of your enterprise.

However, to successfully scale Agile requires addressing both the culture and tools within your organization.

Alignment enables autonomy

Organizations that aspire to be agile need to consider the twin obligations of creating alignment across the enterprise and supporting team autonomy. Teams need autonomy to be efficient. And, enterprises need alignment across teams and the organization to be efficient.

Too much alignment with insufficient team autonomy leads doesn't support innovation or agility of teams to get things done. Too little alignment with each team running their own program doesn't provide the insight and coordination required to meet business goals.

With the right level of alignment across the organization and team autonomy, individuals are empowered to innovate and inspired to collaborate to meet business goals.

Create alignment and support autonomy

As you plan how you want to grow your Agile tool set, consider the following areas. These areas are key to creating enterprise alignment while developing team autonomy.

Area

Create alignment

Support autonomy

Product vision

The organization defines the goals and roadmap for the organization. Goals can be defined as epics and features that show up on the portfolio backlog.

Teams determine how to best meet the roadmap. Teams break down goals into user stories or product backlog items using their team backlogs.

Team structure

Based on business goals, organizations determine the number and size of teams. Vertically structured feature teams lead to greater autonomy and efficiency.

With teams, there should be some established roles, such as product owner and development leads, but also room to rotate roles. For example, team members can take turns acting as Scrum Master, developing sprint demos, running sprint retrospectives, or crafting sprint emails.

Development cadence

Agile organizations need to release products and feature updates at regular intervals. Establishing regular release and sprint schedules promotes the rhythm of the business.
Each sprint--a time boxed iteration of constant duration between two and four weeks—includes planning, executing, delivering value, reflecting, and engaging in continuous improvement.

All teams manage their work within the set sprint cadence. Teams provide input into the length of sprint that works best for them.
Teams choose the Agile methods that work for them, Scrum, Kanban, or a mix of both. Teams also take ownership of starting and acting on their own set of continuous improvement practices.
It's possible for some teams to execute in shorter sprints. For example, if an organization sets a 2-week sprint cadence, some teams may choose to operate in 1-week sprints, while still aligning with the organizational schedule.

Communication cadence

Just as sprints bring a natural rhythm to the flow of work, so too do regular communications. By setting expectations for the types of communications they want to see to stay aligned and how often they occur, organizations naturally create alignment across teams and the enterprise.
Team sprint emails, bug bar status, and release team feature delivery status are examples of such regular communications.

Teams determine the details that they communicate and who will develop the communication. Their sprint emails may contain a summary of previous sprint accomplishments and next sprint plans or include a demo of recently completed features.

Quality

Each organization needs to set the criteria and standards by which they assess quality and set expectations for quality standards. A few ways they define the criteria is to set exit criteria for new feature development, standards to manage technical debt, and bug caps for teams or individuals.
Also, they can monitor bug status and trends by creating bug dashboards.

Teams choose how they meet the quality standards. They may stage bug bashes for new features or at the end of each sprint. They may choose an individual to act as a bug shield on a rotating basis.

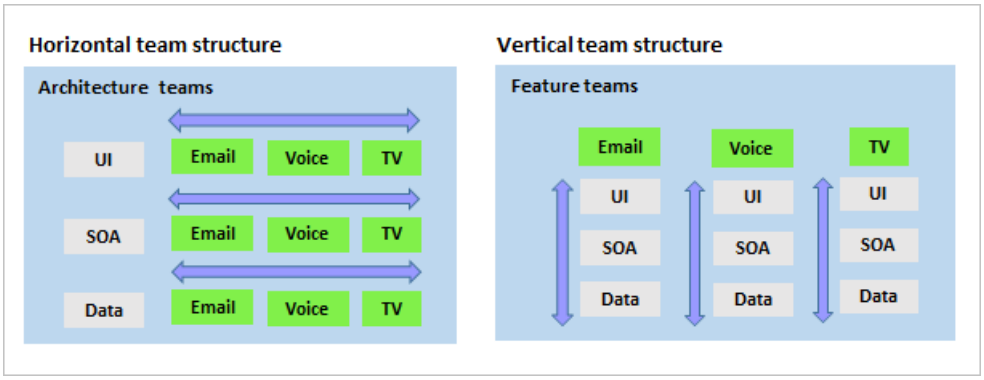
Manage risk, track work

The organization determines how each functional unit will communicate status and risk. They establish a "contract of communication" as to the minimum required information the organization needs.
Also, the organization provides the infrastructure to reduce risks. The organization owes the teams anything they can do to reduce risks that are common across teams.

Beyond meeting the needs set by the organization, teams determine any other details they need to manage and track to reduce their risks. Whether they use a white board with sticky notes or a full Gantt chart, they manage the details.
For example, teams may add a backlog item to track a dependency they have on another team. Or, they may track their risks via a list of issues or impediments. Also, teams regularly contribute to improving the process and infrastructure to support the organizations ability to manage risk and gain insights.

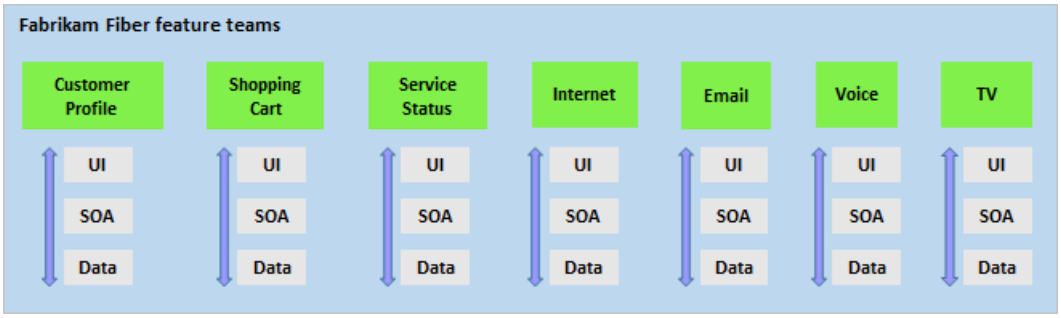
Feature teams

As you scale, one of the most important tasks to consider is how you structure your teams. Traditionally, horizontal team structures divide teams according to the software architecture: user interface, service-oriented architecture, and data teams.



However, with the adoption of Agile practices, vertical team structures that span the architecture have been shown to provide greater team autonomy. Vertical teams can deliver on the features they own by working across the software architecture. They also spread the knowledge needed to work at all architecture levels throughout all the teams.

Configure your teams along the value streams your organization wants to deliver. For example, Fabrikam Fiber, organizes their teams into the following seven feature teams.



Each team plans the features they'll deliver. They have the autonomy to determine how they'll structure the data, architect the services, and design of the web and mobile user interfaces. They plan in adherence with the quality standards set by the organization and to which all teams contribute.

Configure your Agile tools to scale

As your organization grows, you can scale your Agile tools in the following ways.

- **Add teams and filtered backlog views**: You add teams to support team autonomy and provide them with the tools they can configure and manage that supports how they want to work. These tools include product backlogs, Kanban boards, sprint backlogs and taskboards, and many others.

  Also, you can configure teams to support a hierarchy of backlogs and portfolio backlogs so that portfolio managers can review priority and progress across several teams.

- **Set up sprints and releases**: You can structure your iterations to support a flat set of sprints, or a set of sprints embedded within scheduled releases. Each team activates the set of sprints and releases that they need to participate in.

- **Manage portfolios**: by setting up a hierarchy of teams and backlogs and activating portfolio backlogs. Feature teams focused on a subset of the product backlog can remain focused on just their backlog. Portfolio managers who want to view and organize backlogs to track progress and dependencies can manage portfolio backlogs of Features and Epics.

  If other portfolio backlogs are needed--for example, Scenarios or Initiatives--you can add them as well.

- **Configure dashboards**: With team dashboards, you can configure many charts that track progress within a team or across teams. Specifically, you can add status and trend charts based on queries you create.

- **Group or categorize work:** There are several ways to group work that you want to track. Backlogs filter work items based on team area assignments. And, portfolio backlogs allow you to group backlog items under Features and Epics.

  If you want to track and report on work items based on other groupings, you can. You can add tags to work items and then filter backlogs or queries based on tags. Also, you can add subarea paths to represent more granular feature areas.

- **Add folders and use team favorites:** As your teams grow, you'll see a growing list of work item queries, build definitions, and source code folders. By using folders, subfolders, and team favorites, you can manage many of these lists more easily. You can add team favorites for shared queries, source code, and build definitions.

Related articles

Before you can create or work with any of the Agile tools, you need a project. If you don't have one yet, you can create one.

If you're ready to move from one team to two teams, or configure several teams, see Add teams. To add a team administrator or configure team assets, see Manage teams and configure team tools.

For more information, see these articles:

- Practices that scale
- Visibility across teams
- Review team delivery plans
- Implement Scaled Agile Framework® to support epics, release trains, and multiple backlogs.

Agile culture industry resources

- Nexus, the Exoskeleton of Scaled Scrum
- Culture over process
- The Culture Game - Tools for the Agile Manager
- Scaled Agile Framework (SAFe)
- Scaling Agile Software Development, - Disciplined Agility at Scale (White Paper)

Scale with teams and not projects

Often, organizations look at adding a project for each software development project.

Adding teams to scale your tools rather than adding projects is recommended for the following reasons:

- **Visibility:** It's much easier to view progress across all teams
- **Tracking and auditing:** It's easier to link work items and other objects for tracking and auditing purposes
- **Maintainability:** You minimize the maintenance of security groups and process updates.

To learn more, see About projects and scaling your organization.

Implement Agile practices that scale

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Enterprise organizations adopt Agile practices for many reasons. Prime among these reasons include:

- Shorten time-to-market, accelerate product delivery
- Improve organizational effectiveness to manage changing priorities
- Enhance software quality and delivery predictability
- Improve project visibility and reduce project risk

As your organization grows, you'll want to scale your practices to remain agile and meet changing goals. To do that, consider these two guiding principles:

- **What does success look like** to you, your teams, and your organization? What's of most interest: On-time delivery? Product quality? Predictability? Customer satisfaction?
- **Return to first principles**, return to the principles and shared values enumerated in the Agile manifesto As noted by Ken Schwaber, one of the founders of Scrum:
  - "Values and principles scale, but practices are context sensitive."
  - "Keep the values, keep the principles, think for yourself. A core premise of Agile is that the people doing the work are the people who can best figure out how to do it."

Create rhythm and flow

By adopting a shared cadence and set of periodic communications, you create a constant flow of activity throughout the organization. Practices that help create rhythm and flow within larger organizations include:

- **Shared cadence**: Regular sprints and releases establish the rhythm of the business. Having all teams work to a shared cadence helps with all coordination and collaboration activities.
- **Sprint emails**: To keep the organization and all teams informed about the progress and plans of feature teams, each feature team can email a summary of their previous sprint results and current sprint plans.
- **Sprint demos**: A quick--2 to 3 minute--video that illustrates a new feature the team produced. Links to such videos can be included within sprint emails.
- **Showcase meetings**: To inform other teams and ask for feedback about software under development, teams showcase the work they've done. Conduct these meetings at regular intervals throughout the project lifecycle, and open them to all interested parties.
- **Bug summary emails**: To support insight into product quality and to encourage maintaining bug discipline, periodically share quality metrics with the organization. These metrics may include active bugs per feature team, bug trends, and bugs per engineer.
- **Coordination meetings**: Hold meetings that coordinate teams at either regular intervals or as often as needed to address overlapping goals, dependencies, and risks.

Interact with customers

Engaging customers throughout your product lifecycle is a primary Agile principle. Empower each team to interact directly with customers on the feature sets they own.

- **Continuous feedback**: Build in customer feedback loops. These loops can take many forms:
  - **Customer voice**: Make it easy for customers to give feedback, add ideas, and vote on next

generation features. Giving feedback is often done through a dedicated website.
    - **Product feedback**: In-product feedback buttons are another way to ask for feedback about the product experience or specific features.
    - **Customer demos**: Regularly scheduled demos that ask for feedback from your customers can help shape next generation products and keep you on track to build applications your customers want to consume.
- **Early adopter programs**: Such programs should be developed with the idea that all teams may want to participate as some point. Early adopters gain access to early versions of working software, which they then can provide feedback. Often, these programs work by turning on select feature flags for an early adopter list.
- **Data-driven decisions**: Find ways to instrument your product to obtain useful data and that can test various hypotheses. Help drive to an experiment-friendly culture that celebrates learning.

Improve project visibility

The more insight you and your teams have into the goal, vision, and progress of the work being done, the better enabled you'll be to reduce risks and manage dependencies.

- **Team structure**: No matter how large your organization gets, structuring your organization around small teams of 6 to 9 scales. Create vertical, autonomous feature teams grouped under portfolio management areas.
- **Work breakdown structure**: Breaking down large goals, features, or requirements into smaller ones remains a stable of project management. By breaking down work into similar-sized tasks, teams can make better estimates and identify risks and dependencies.
- **Consolidated views**: Use your online tracking tools to aggregate work to gain knowledge across teams. Build dashboards to show progress and trends.
- **Experience reviews**: These meetings, held before development begins on a feature, are used to educate leadership on scenarios and priorities, collect feedback, set expectations, and surface any cross-team issues about the feature.

Empower a productive workforce

Some specific Agile practices that scale well and lead to happier, engaged, and productive employees include:

- **Embedded leadership**: Empower teams and leaders within the organization to self-organize and self-manage as much as possible. Team autonomy increases organizational agility team effectiveness. Ensure teams have the corporate sponsorship needed to succeed.
- **Daily stand-ups**: Or, Scrum meetings help keeps teams focused on what they need to do daily to maximize their ability to meet their sprint commitments. As organizations grow, they should consider staggering these meetings so that cross-team participation can occur as needed.
- **Scrum of scrums**: Daily stand ups of members from different Agile teams meet daily to report work completed, next steps, and issues or blocks occurring within their representative teams.
- **Team communications**: Provide and encourage teams to share their practices and guidance, which they and other teams can access through the corporate network. Common tools used for this purpose include team wikis, OneNotes, or Markdown sites.
- **Collaboration**: Encourage informal team-to-team communications and collaboration within the team. Institutionalizing practices such as code reviews, design reviews, spec reviews not only increase team collaboration but help develop individual and overall corporate competence.

Improve organizational culture

You improve organizational effectiveness by attending to the culture you want to build. Culture changes occur when individuals, teams, and organizations adopt one or more continuous improvement practices. Several

scalable Agile practices include:

- **Retrospectives**: By asking questions such as: "What went well?", "What should we do differently?", and "What should we stop doing?" help teams reflect on how they can improve on their processes and practices. Retrospectives help teams surface what is working well and what needs improvement. Retrospectives can be conducted anytime and anywhere. However, institutionalizing certain retrospectives at a regular cadence help institutionalize continuous improvement practices. For example:
  - **Sprint retrospectives** can help teams identify areas to improve at a regular cadence.
  - **Release retrospectives** can help organizations identify areas to improve communications and internal practices and fuel improvement for the next release.
  - **Operational reviews**: are typically held monthly and include representatives from a whole value stream. Spanning a portfolio of projects and other initiatives and using objective, quantitative data, design these retrospectives to provoke discussions about the dynamics affecting performance between teams.

    See the Agile Retrospective Resource Wiki for ideas, tips, and tools for planning and conducting retrospectives. See also the Marketplace Retrospectives extension.
- **Improvement tracking board**: Good ideas to improve processes can arise from any one at any time. Capturing those ideas to discuss and decide on how to act on them quickly is a key to support process improvement efforts.

  A white board provides any easy and visual means with which to capture ideas. Also, you can create an Improvement tracking team and capture ideas that you track on an electronic Kanban board.
- **Institutionalize sharing**: Sharing best practices and communicating ideas helps all teams within an organization grow and improve. Developing a culture of learning is key to supporting this and other continuous improvement activities. Some ideas to consider:
  - In-house wikis
  - In-house distribution lists
  - Hackathon weeks or 10% hack time
  - Internal Agile support team to support teams who adopt Agile practices

    The Culture Game provides a good resource for Agile managers to help teams adopt Agile and share best practices.
- **Communities of practice**: Support internal common disciplines (for example, DBAs, SW Architects, UX design)

Working software

> *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."*
> *"Working software is the primary measure of progress."*
> - Agile manifesto

As the amount of software, features, and complexity increase, you'll need to adopt practices that help you produce consumable solutions.

- **Feature flags**: Use feature flags to enable or disable access to different features. Provide support for turning on features to early adopters to get working feedback.

- **Release trains**: Provide another type of cadence to deliver one or more features. Feature teams understand the pre-planned schedule of pushing out new features, and plan correctly. Release trains can correspond to the same sprint cadence establish for the organization, or occur at a different cadence. See Scaled Agile Framework for how to setup sprints and release trains.
- **Continuous integration**: Adopt processes that eliminate manual work and instead automate the flow of software through the test, build, and deploy cycles.
- **Internal Open Source**: Bring the value and ethos that's developed in the Open Source Software community to your internal development teams.

Related articles

Along with the above practices, you'll find more guidance around scaling your Agile tools in the following articles:

- Agile culture
- Add teams
- Portfolio management
- Visibility across teams
- Scaling Agile to large teams

Industry resources

- Agile manifesto
- Agile Alliance
- Scaled Agile Lean Development - The Principles

Practices that don't scale

- **Estimating large initiatives**: Part of waterfall project methods involved estimating resources and schedules. The larger the initiatives, the less likely these estimates were of any value. As projects grow, risks and unforeseen issues and impediments can arise, invalidating many estimates.
- **Velocity**: While team velocity can provide a useful metric for gaining insight into how much work each team can complete during a sprint cycle, you can't add team velocities to gain meaningful or useful metrics. Also, using velocity gained from many teams to reliably complete long range forecasts is problematic. Teams vary in the way they estimate their work, and those variations increase over time.
- **Top-down prescriptive solutions**: One size doesn't fit all, and one solution typically doesn't fit all teams. Supporting team autonomy means letting teams find their own solutions.

Manage priorities and gain visibility across teams

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Agile tools provide each team a wealth of ways to gain visibility into their work—to manage priorities and status and to monitor progress and trends. However, how do you gain visibility across several teams? What tools should you use?

You have three main ways to track progress across several teams.

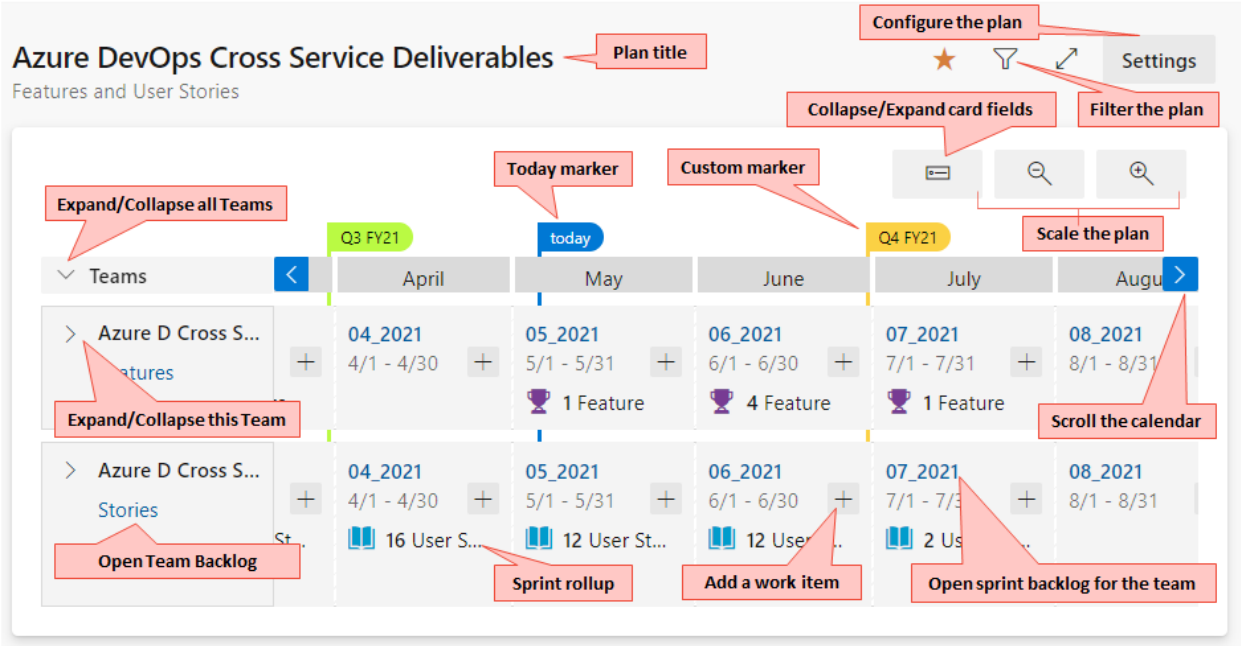
- Management teams can define Delivery Plans that provide visibility into the deliverables several teams have scheduled
- Each management team can use their Agile tools, and in particular portfolio backlogs, to gain visibility of the feature teams defined under their area path
- Management teams can create dashboards that monitor status, progress, and trends across several teams.

For an overview of all team tools, see Manage teams and configure team tools.

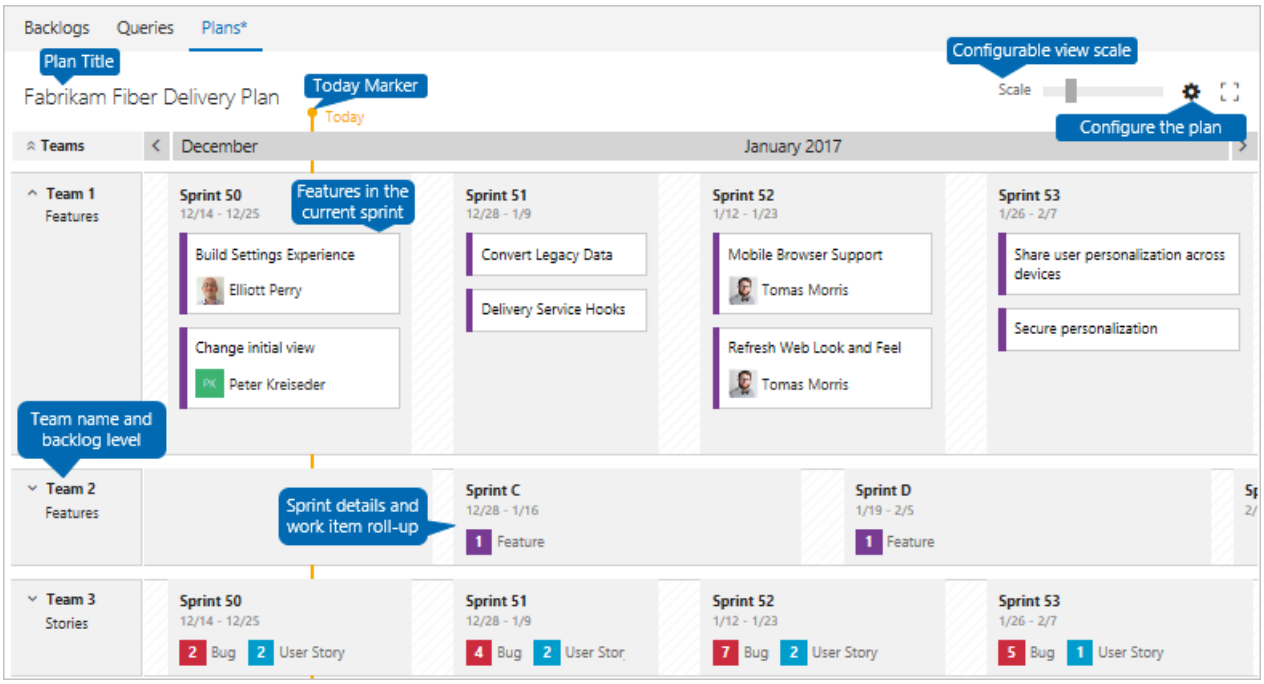
Delivery Plans support a view of team backlogs on a calendar timeline

With a Delivery Plan, you gain a tailor-made view across several teams and their development backlogs—stories, features, or epics. You can use these views to drive alignment across teams by overlaying several backlogs onto your delivery schedule.

When you configure a Delivery Plan, you select the teams and backlog levels of interest. You can then interact with the plan to update it and drill into more details. To learn more about Delivery Plans, see Review team plans.



When you configure a Delivery Plan, you select the teams and backlog levels of interest. You can then interact with the plan to update it and drill into more details. To learn more about Delivery Plans, see Delivery Plans.



Use portfolio backlogs to track features and epics

The first level of gaining visibility across several teams is to configure your teams and backlogs to support the views you want.

We recommend that you structure your teams as follows:

- Add a management team for a group of feature teams; these teams own epics and turn on only the Epic portfolio backlog level
- Add feature teams to manage features, stories and tasks, and turn on the stories and features backlog levels

The management team creates the epics. Then, they or their feature teams break down the epics into features and then map their features to the epics on the management backlog.

> **TIP**
> By breaking down large goals, epics, scenarios, or features into smaller ones, teams can make better estimates and identify risks and dependencies.

Limiting the backlog levels for each team—Epics for management teams and Features and Stories for feature teams—helps each team to stay focused on monitoring the progress of their work. For details on managing team backlog levels, see Select backlog navigation levels.

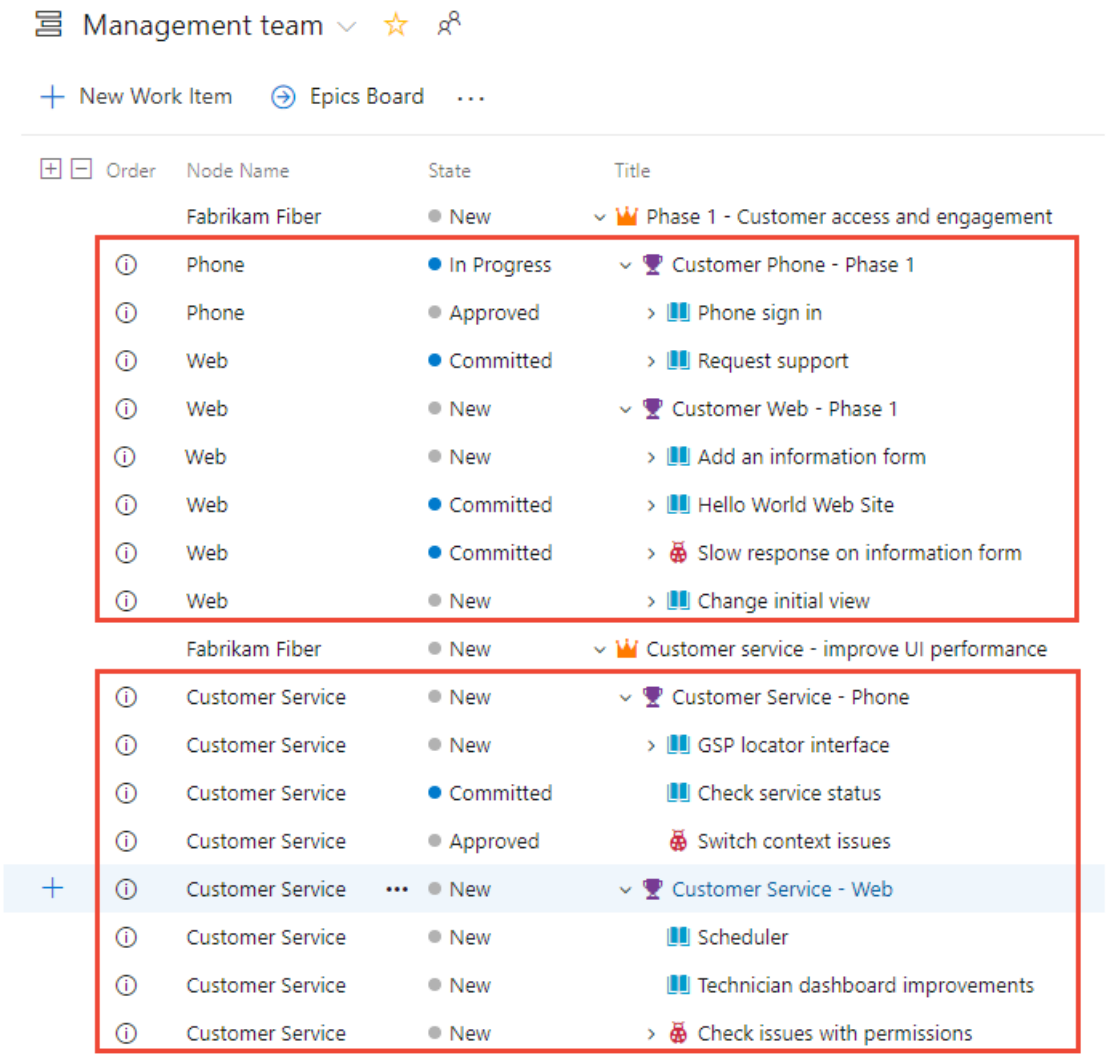
With the multi-team portfolio backlog view, you can:

- Review priorities with your team and reorder features to support current priorities
- You can drill down to see the status of each feature's child user stories or PBIs
- Filter the backlog based on keyword or tag to focus on specific teams or categorized items
- (Optional) You can use the mapping feature to map user stories or PBIs to features

View child items owned by other teams

Management teams can drill down from their portfolio backlog to see how **Epics** are progressing. Drilling down, you can see all the backlog items and features, even though they belong to one of three different teams: Customer Service, Phone, and Web.

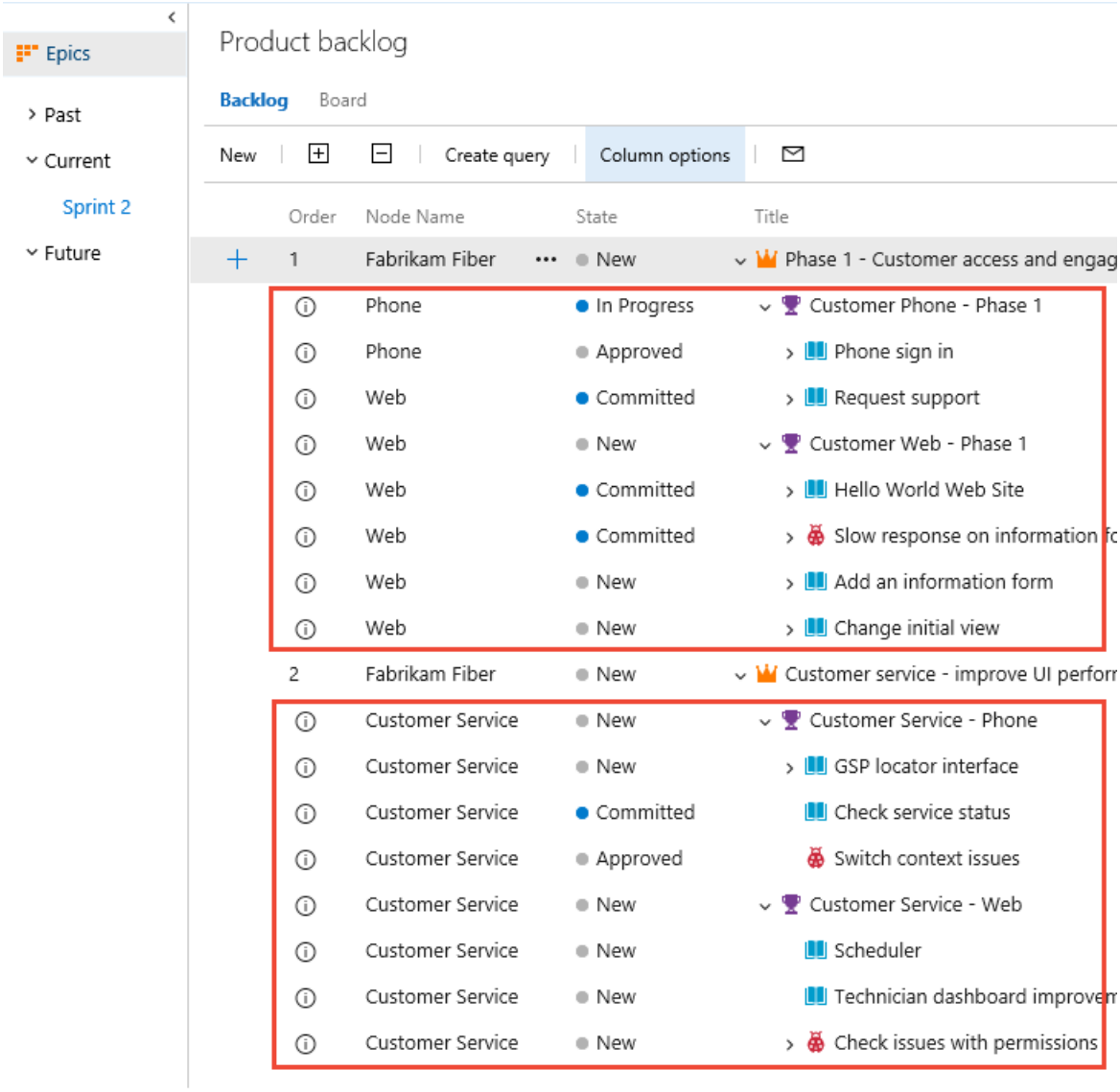
Items that are owned by other teams appear with an information icon, ⓘ .



TIP

Add the **Node Name** field as a column to identify the area path/team associated with the work items.

Items that are owned by other teams appear with an information icon, ⓘ .



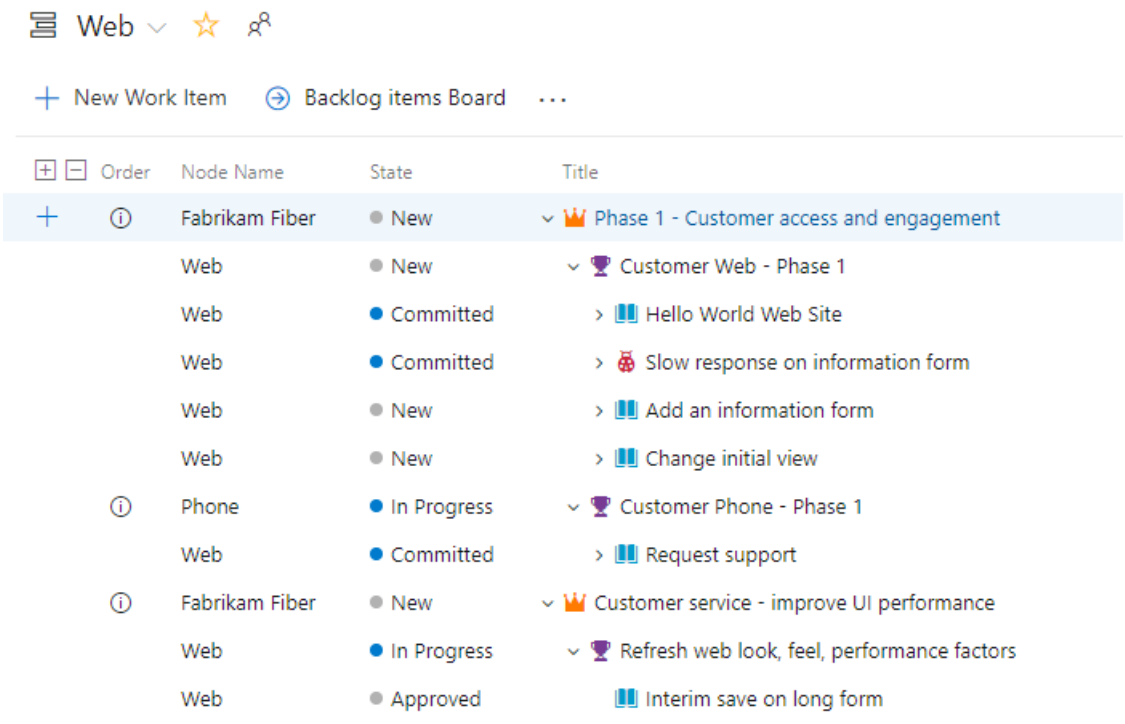
TIP

Add the **Node Name** field as a column to identify the area path/team associated with the work items.

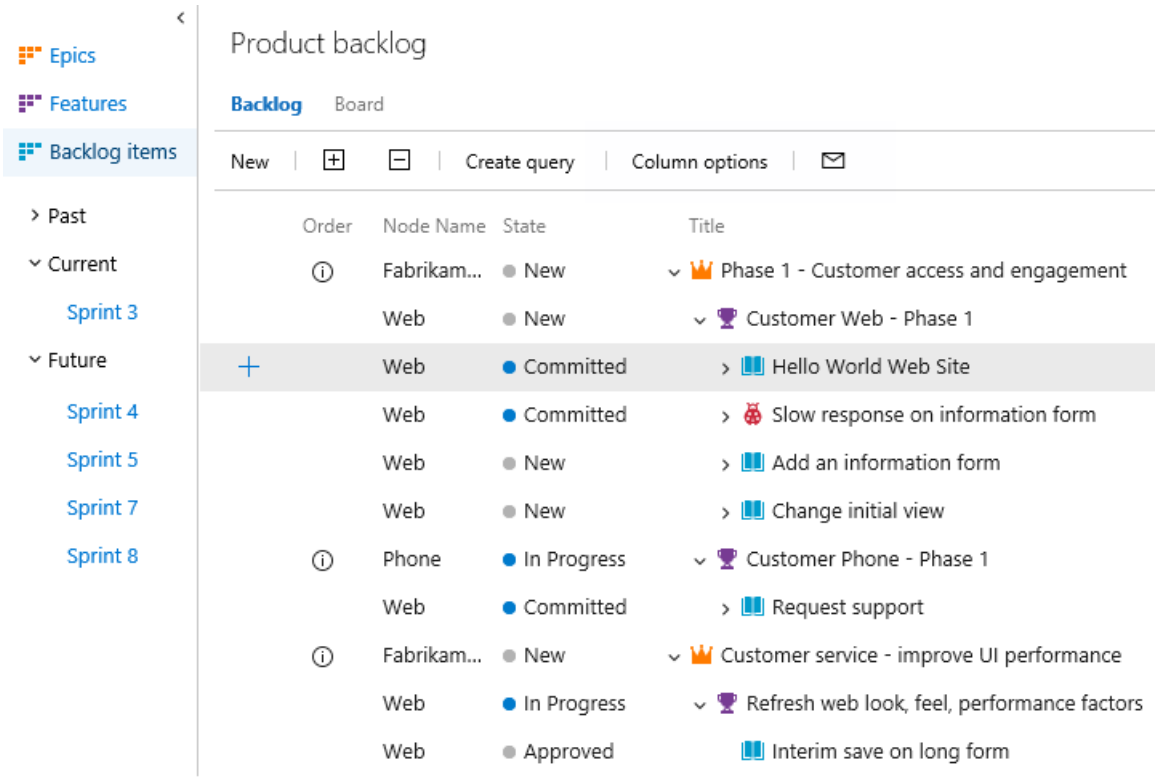
View backlog items and parent items owned by other teams

Feature teams can turn **Show parents** on their backlogs to see context and those items owned by other teams.

Items that are owned by other teams appear with an information icon, ⓘ .



Items that are owned by other teams appear with an information icon, ⓘ .



TIP

When estimating stories or product backlog items, start with one story point per person per day. Feature teams can later calibrate and adjust those estimates as needed. For example, the velocity of a seasoned team will be higher than a new team. The size of the work stays the same, but a seasoned team can just deliver faster.

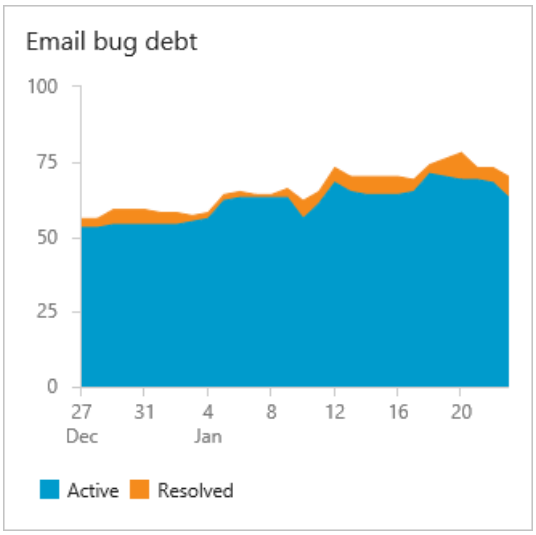
To learn more about this configuration, see Portfolio management, Add teams, and Organize your backlog.

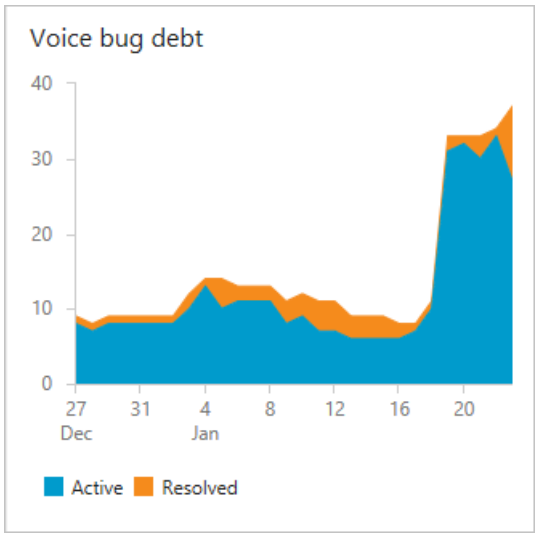
Add management dashboards with multi-team views

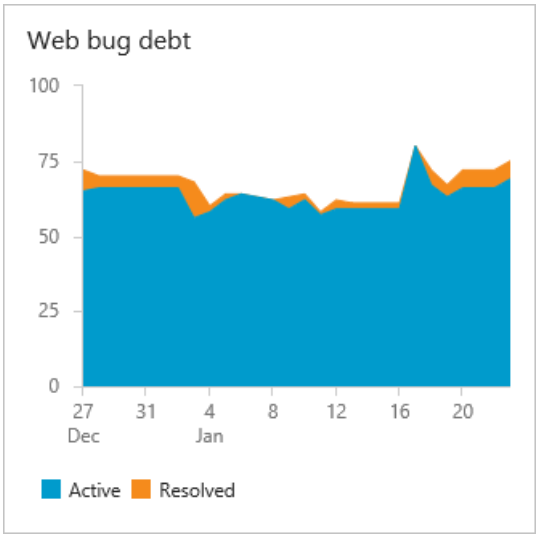
A second method for gaining visibility across teams is to define multi-team focused dashboards that let you view progress, status, and trends. You do define focused dashboards primarily by defining queries that either capture the progress of a single team or several teams. You can then create charts and view trends for each team or for several teams.

The two areas of most interest to management teams are project health and bug debt. The widget catalog provides 10+ widgets you can add to a dashboard to track the status, progress, and health of your project and teams. Also, you can find other widgets in the Visual Studio Marketplace, Azure DevOps tab.

For example, here we've added three query-based charts, one for each team, to a dashboard that shows the active and resolved bugs over the previous four weeks.







When defining multi-team dashboards, consider the following questions:

- What are you wanting to learn and how will it drive your organization's actions
- What time frame is of interest.

Review Agile culture and Practices that scale for guidance on team autonomy and organizational alignment.

Project health and progress against goals dashboard

Use the Query Results widget to provide a list of features by state:

- Completed features (Done or Closed)
- New features (New or Proposed)
- Features being actively worked (In Progress or Active)

Use the Chart for work items widget to add query-based charts. To learn more about creating query-based

charts, see Charts.

Technical debt, bug debt, and activity dashboard

Another measure of project health and the health of the teams is to monitor bug activity and bug debt. Consider the charts you can create that will help you answer these questions:

- Are bugs getting fixed? At a rate that's acceptable?
- How stale are bugs?
- Is the bug debt per team being maintained?
- Is the ratio of high priority bugs being kept within organizational goals?

For tips on creating queries based on counts or numeric fields, see Query by numeric field.

Use the Analytics Service to gain visibility across teams

You can add Widgets based on the Analytics Service to a dashboard that show progress for a team. From one dashboard, you can add widgets for any team within the project.

Track capacity when working on more than one team

You can track capacity for individuals that participate on more than one team. To learn how, see Set sprint capacity, Track capacity when working on more than one team.

Limitations of multi-team Kanban board views

While the management teams you configure can use the Kanban board to monitor feature progress by turning on the Features backlog, there are limitations inherent within these views. Even if the management team and the feature teams configure their Feature Kanban board columns with identical workflow mapping, updating the Features on one team's Kanban board won't be reflected on another team's Kanban board. Only when the work item state changes does the card column reflect the same on all boards.

> **IMPORTANT**
> Work items that appear on more than one team's Kanban board can yield query results that don't meet your expectations. Because each team can customize the Kanban board columns and swimlanes, the values assigned to work items which appear on different boards may not be the same. The primary work around for this issue is to maintain single ownership of work items by team area path. Another option is to add custom workflow states which all teams can use. For details, see Customize your work tracking experience.

Related articles

As you can see, there are many ways you can monitor progress and trends across several teams. The methods you choose will depend on your focus and organizational goals.

Here are some other articles that address working with multiple teams:

- Backlogs, boards, and plans
- Review team plans
- Add teams
- Portfolio management

End-to-end traceability

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services

One of the chief advantages of adopting a platform such as Azure DevOps is the support it provides to trace work from requirements to deployment. You gain insight at each step of decisions made and the software deployed. The scenarios illustrated in this article are supported when you use Azure Repos git repositories.

The following are some of the supported tasks to facilitate end-to-end traceability:

- Create a branch from a requirement
- Create a pull request of updated branch
- Validate the pull request using a build pipeline
- Create and run inline tests on requirements
- Merge the pull request into the main, default branch
- Deploy changes into production with deployment status to Azure Boards
- Monitor and report on requirements traceability

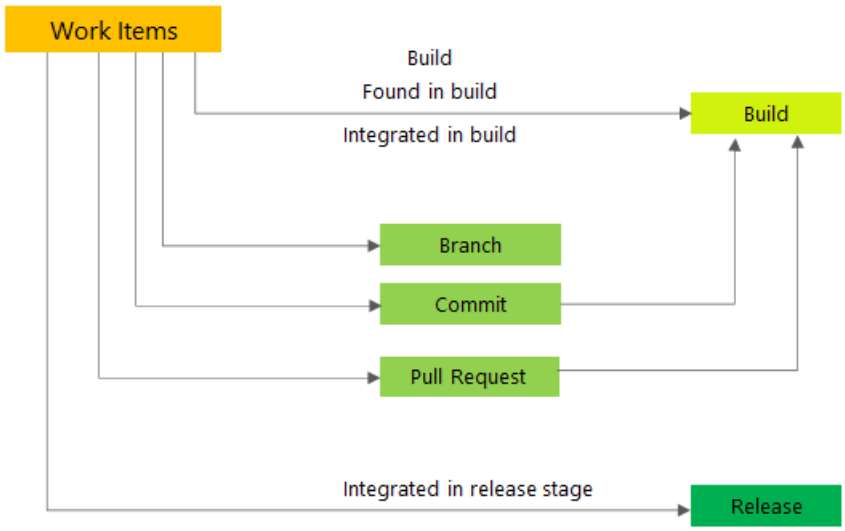
> **NOTE**
> *Tracking the origin of work through delivery, and the ability to trace work across the development lifecycle, is essential to achieving end-to-end traceability.*

End-to-end traceability is supported by linking various objects such as work items, branches, commits, pull requests, builds, and releases. Build in reports and Analytics support the ability to monitor traceability in real time. This article presents an overview of Azure DevOps traceability support without going into the details of how to enable and support traceability. More detailed information is provided under Related articles.

Traceability and linking

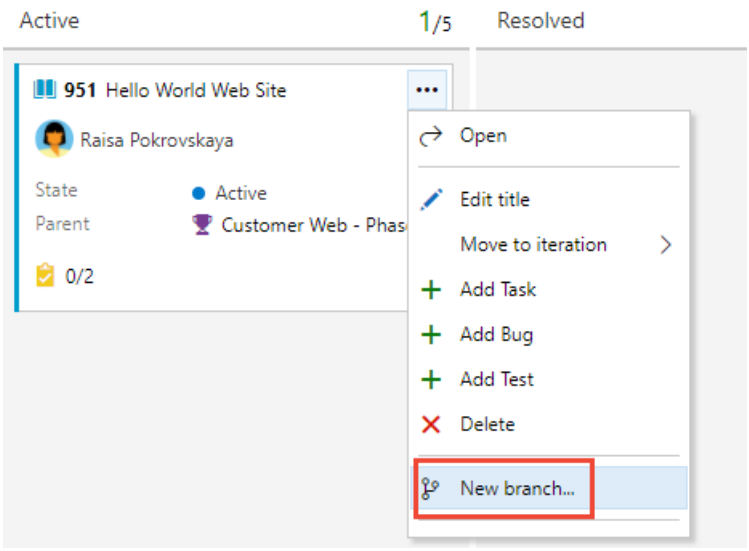
Throughout the development lifecycle, you can capture which code changes, builds, and releases support the completion of a work item. In this way, your team can understand what work was done or how a bug was fixed through the audit trail of changes to the code base.

The link types used to construct these links for Git repositories—as illustrated in the following image—are *Build*, *Found in build*, *Integrated in build*, *Branch*, *Commit*, *Pull Request*, and *Integrated in release stage*.

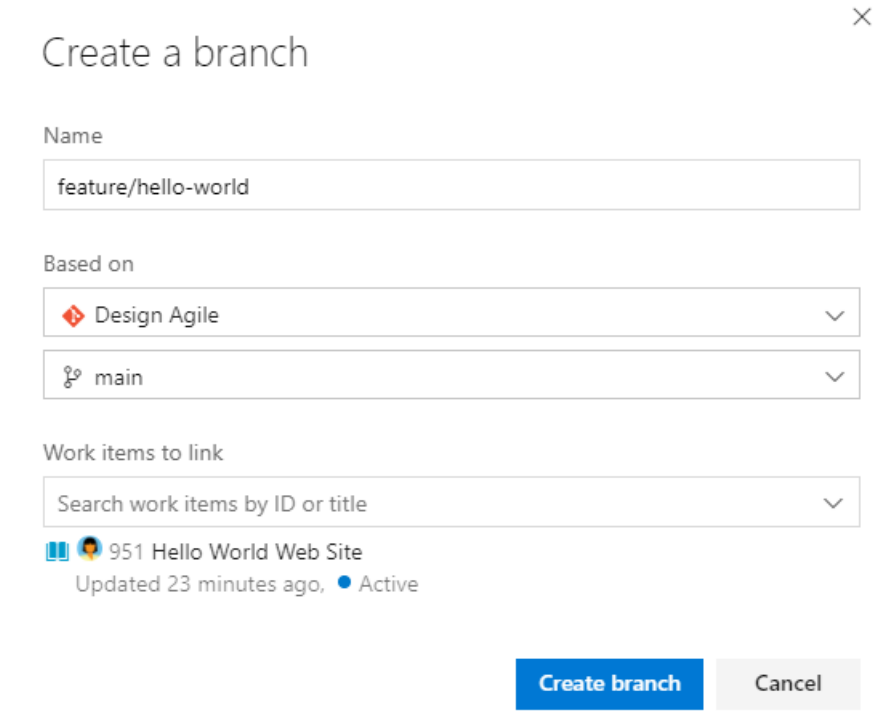


Branch from a requirement

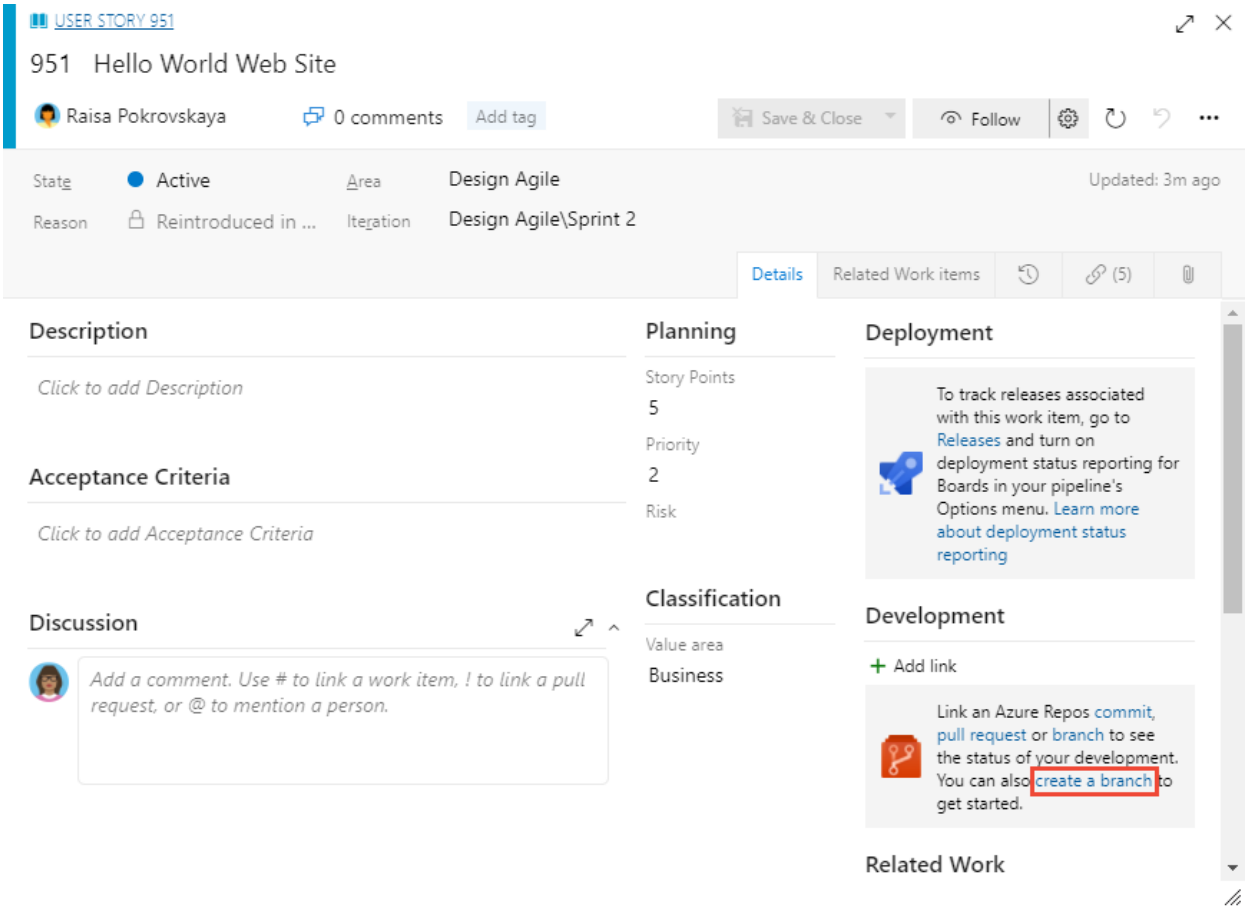
You can accomplish many tasks with a single click from the product Kanban board. Here we show creating a branch from a requirement by opening the work item card menu.



You can name and label a branch off the default main branch from the dialog that opens. The work item is automatically linked to the branch you created with the *Branch* link type.

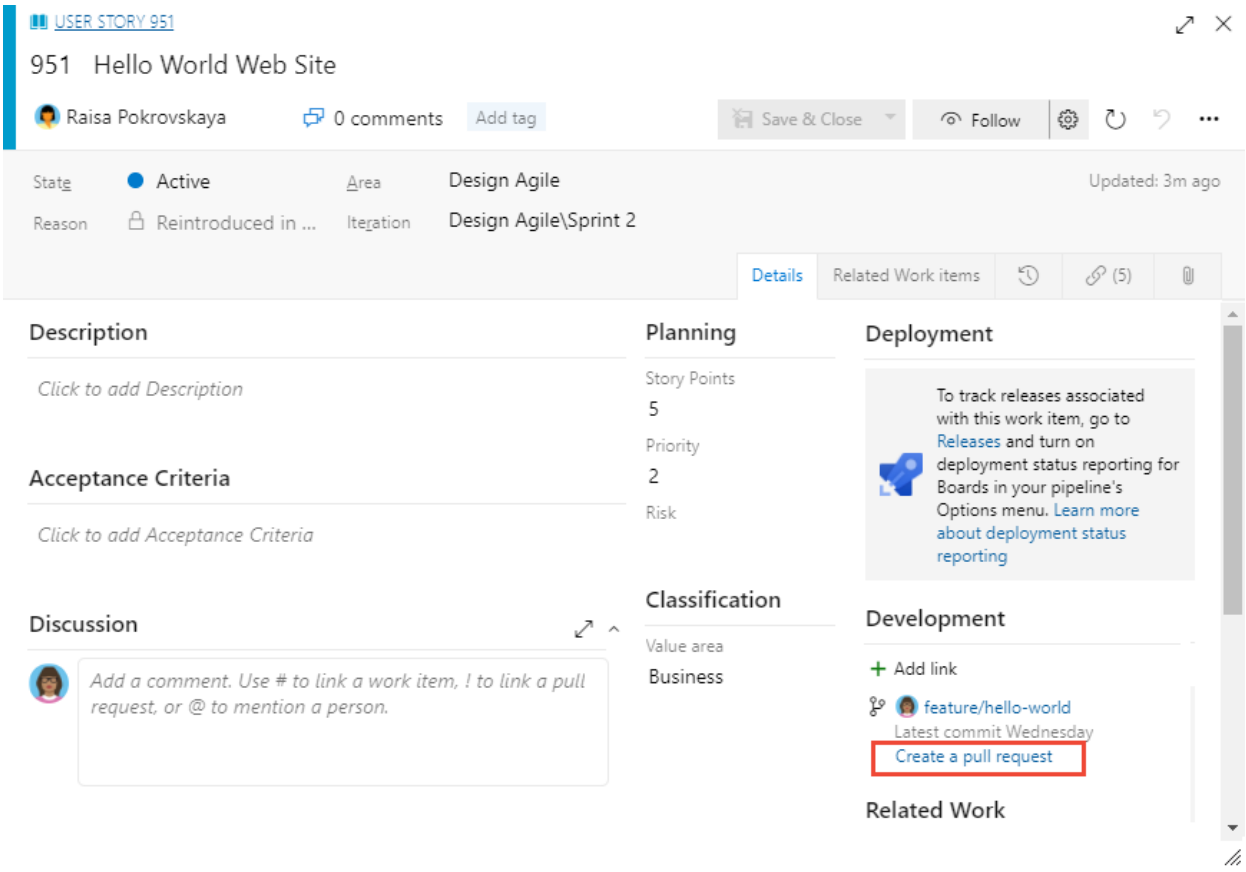


You can also accomplish this task through the work item form by choosing the **create a branch** link.



Create a pull request from a requirement

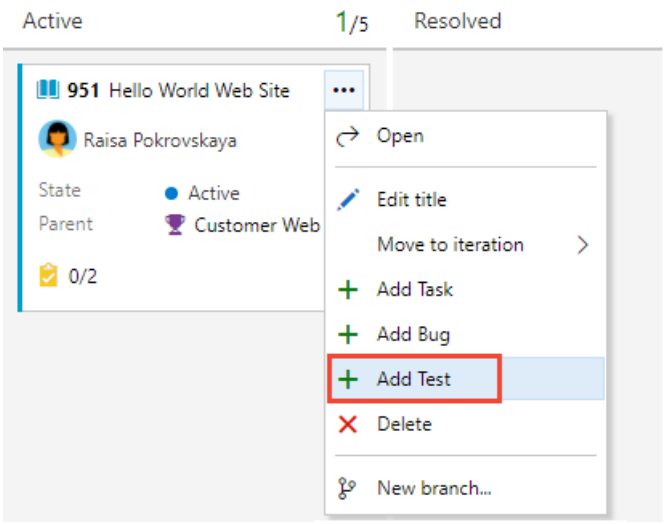
After code changes have been made to the new branch, developers can then create a pull request from the requirement by opening the work item form.



Another advantage of driving software development from the Kanban board and work item is to encourage developers to input comments as they work to further the audit trail of changes occurring.

Add and run tests from requirements

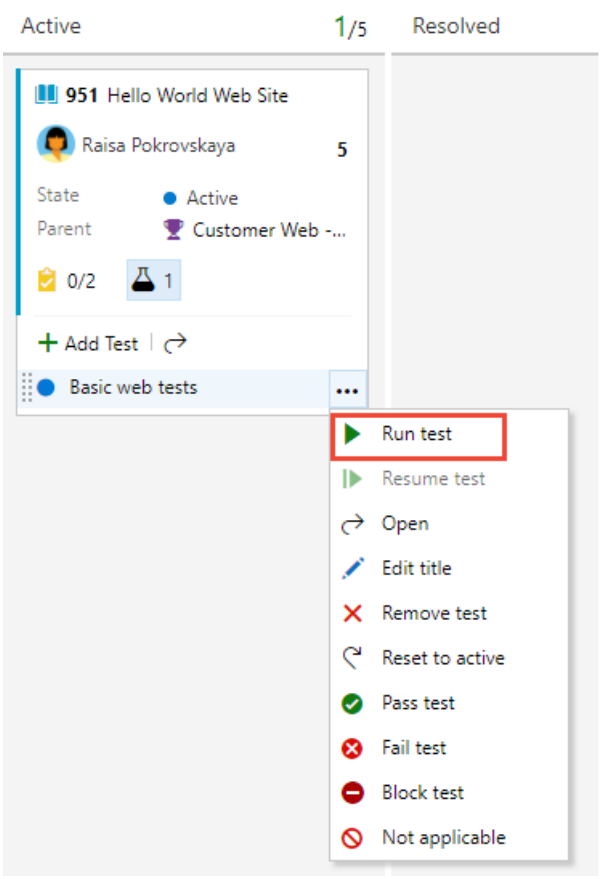
Just like you can create a new branch from the Kanban board, you can also add tests to the work item.



NOTE

Test traceability supports linking a test to a set of requirements and validating that the application works as expected.

After adding and defining the tests, you can run them from the Kanban board and set the test status.



Test integration with the Kanban board makes it easy for teams to get started with manual testing and then take advantage of the full testing capabilities provided by Azure Test Plans. The Kanban board shows the test added to support the requirement when test cases are created from the Kanban board or when requirement-based test suites are created under Test Plans.

Manual and automated testing

Teams that are moving from manual testing to continuous, automated testing, and have a subset of tests already automated, can execute them as part of a pipeline or on demand. Referred to as *planned testing*, automated tests can be associated to the test cases in a test plan and executed from Test Plans. Once associated, these tests contribute towards the quality metrics of the corresponding requirements.

Deploy changes into production

After you define a pipeline to build and release the code changes, you can track the deployment of the requirement to each release stage. From the work item form, you can quickly open the links to builds and releases from the **Deployment** and **Development** control sections.

Deployment and Development controls

When you open the work item form, you can see the stages the requirement has been deployed to, and drill down for greater details by choosing the links. Under the **Development** section, you can open the branch ,commit, or pull requests that have been linked to the requirement.



The **Deployment** control shows release information for those work items that have been associated to a Git commit which is part of a build being released.

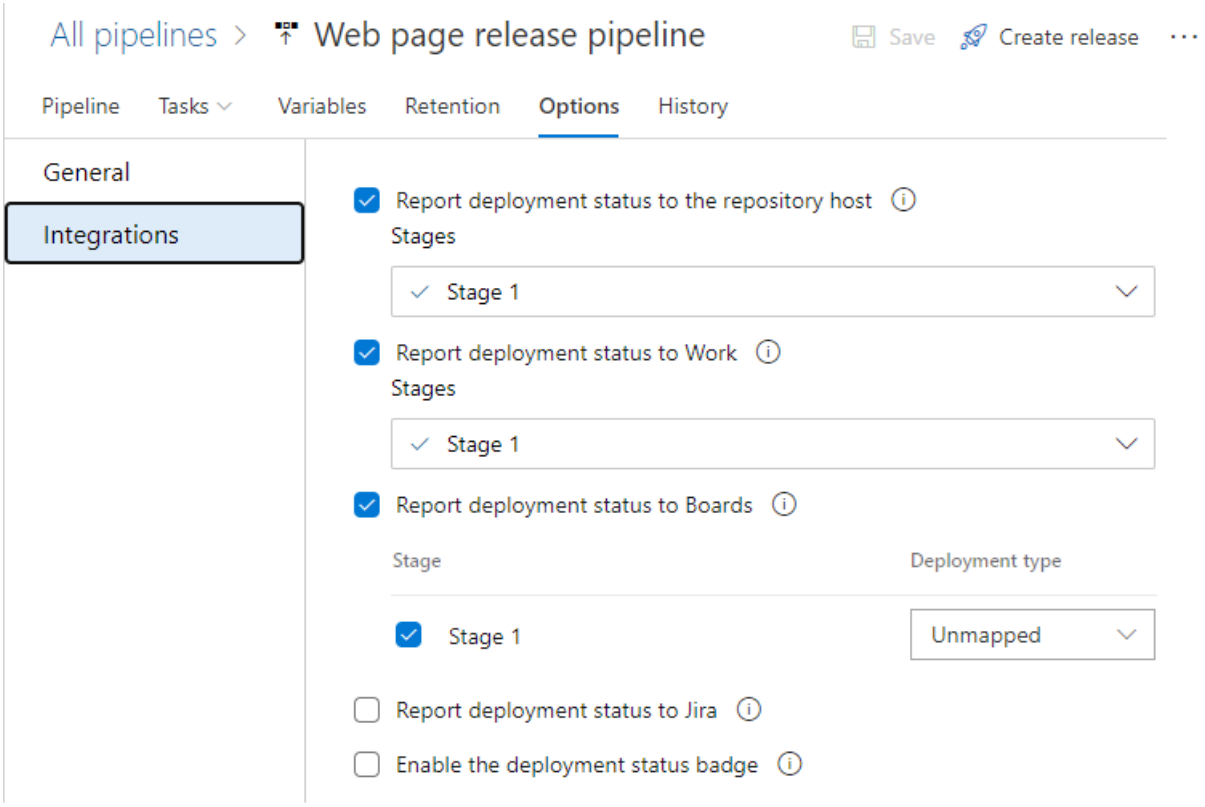
Release view

The following image illustrates the multiple environments that the release is targeting which the selected work item is associated with.



Release settings

You manage the deploy to production view options from the release settings.



The work item deployment control displays the status of releases within those work items that are associated with commits in the build and those release pipelines you've configured to report deployment information to Azure Boards.

Requirements Traceability Matrix

Requirements traceability relates and documents two or more phases of a development process, which can then be traced both forward or backward from its origin. Requirements traceability provides teams insights into indicators such as quality of requirements or readiness to ship requirements. A fundamental aspect of requirements traceability is association of the requirements to test cases, bugs and code changes.

> **NOTE**
> *The Requirements Traceability Matrix (RTM) is a document that links requirements throughout the validation process. The purpose of the Requirements Traceability Matrix is to ensure that all requirements defined for a system are tested in the test protocols.*

Requirements traceability reports

Requirements traceability reports

Bug traceability

View the bug with the test result, directly in context, within the Tests tab. The Work Items tab also lists any linked requirements for the test result.

Source traceability

Based on the build or release pipeline, you can choose the timeline or pipeline view to see what code changes were committed. You can analyze the code changes to identify the potential root cause of the test failure.

Test analytics

Test analytics for builds

To help teams find and fix tests that fail frequently or intermittently, use the top failing tests report. The build summary includes the Analytics page that hosts this report. The top-level view provides a summary of the test pass rate and results for the selected build pipeline, for the specified period. The default range is 14 days.



Test analytics for releases

Open a build or release summary to view the top failing tests report. This report provides a granular view of the top failing tests in the pipeline, along with the failure details.

For tests executing as part of release, access test analytics from the Analytics link at the top right corner. As with build, the summary provides an aggregated view of the test pass rate and results for the specified period.

Track the quality of requirements

To track the quality of work items that belong to the Requirements category. This includes work items such as User Stories (Agile), Product Backlog Items (Scrum) and Requirements (CMMI).



Test failures

Open a build or release summary to view the top failing tests report. This report provides a granular view of the top failing tests in the pipeline, along with the failure details.



Related articles

To learn more about any of the concepts introduced in this article, refer to the following articles.

Linking

- Configure repositories and branches to integrate with work tracking
- Configure pipelines to support work tracking
- Drive Git development from a work item
- Link and view work items to builds and deployments
- Link user stories, issues, bugs, and other work items
- Linking, traceability, and managing dependencies
- Link type reference

Testing

- Add, run, and update inline tests
- Associate automated tests with test cases

Reports and Analytics

- Requirements traceability
- Requirements tracking sample report
- Requirements tracking rollup sample report
- Review test results
- Test Analytics
- Review code coverage results
- Code coverage for pull requests

Link work items to other objects

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you link work items to other objects, you maintain an audit trail of related work for your team. All users can add work item links to the following internal (Azure DevOps) and external (Git) objects.

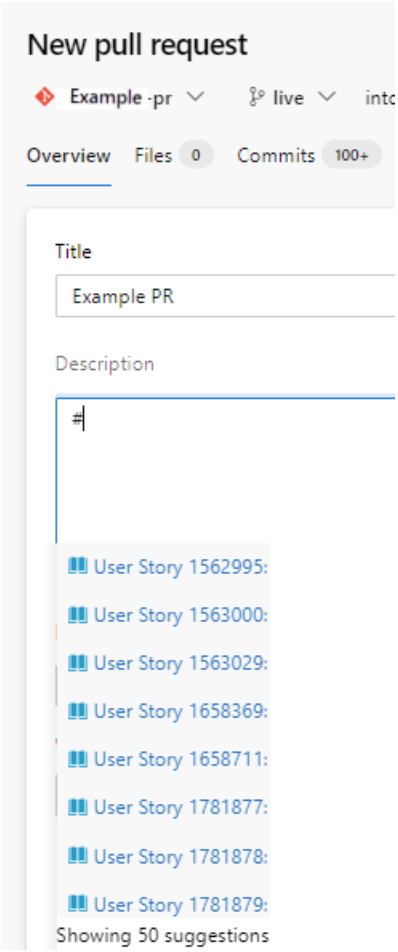
- **Build**: found in build, integrated in build
  - Link work items to builds and deployments
- **Release**: integrated in release, integrated in release environment
  - Automatically link work items to builds or releases
- **Repository**: pull request, description, branch, commit, comment, tag
  - Configure repositories and branches to integrate with work tracking
  - Link to work items from GitHub commits, pull requests, and issues
  - Azure Boards-GitHub integration
- **Wiki**:
  - Add & edit wiki pages
  - Wiki Markdown guidance

- **Build**: found in build, integrated in build
  - Link work items to builds and deployments
- **Release**: integrated in release, integrated in release environment
  - Automatically link work items to builds or releases
- **Repository**: pull request, description, branch, commit, comment, tag
  - Configure repositories and branches to integrate with work tracking
  - Link to work items from GitHub commits, pull requests, and issues
  - Azure Boards-GitHub integration

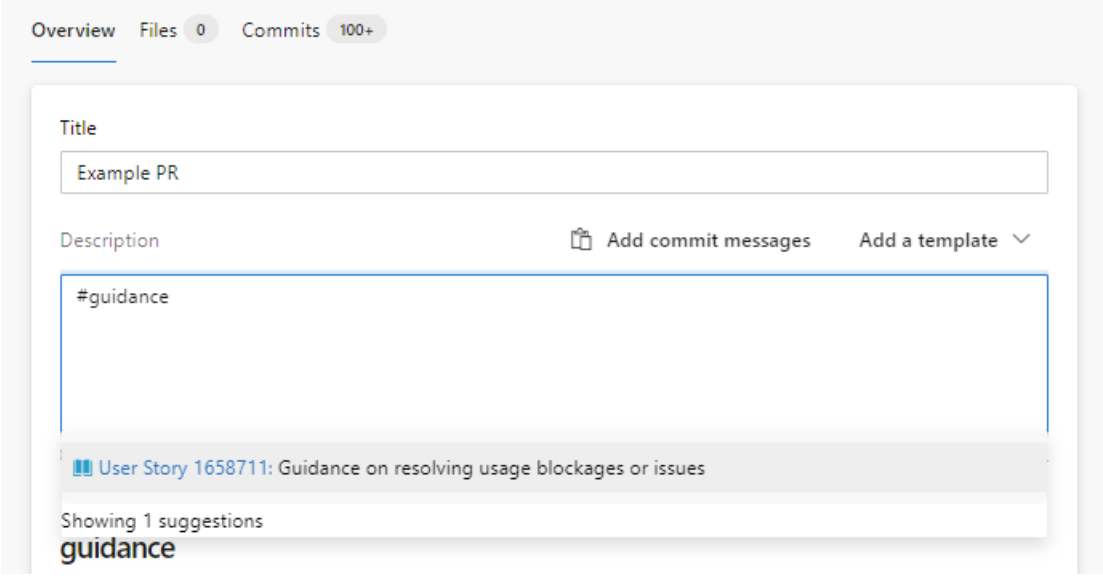
Link work items to pull requests

Link your work items to pull request commits, comments, description, and so on.

1. Enter `#` to trigger the `#ID` work item picker in your pull request. You see a list of 50 work items that you've recently modified or are assigned to you.



2. Enter up to five keywords that match the work item type, ID, or title to narrow the list of suggested work items.



Set work item state in pull request

When you create a pull request, you can set the *state* value of the linked work items in the description. Follow the syntax: `{state value}: #ID`. When you merge the pull request, the system reads the description and updates the work item state. In the following example, we set work items #300 and #301 to Resolved, and #323 and #324 to Closed.

> **NOTE**
> This feature requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

Azure DevOps considers the following criteria (in this order) when it attempts to set the state of #mentioned work items:

1. State
2. State Category
3. Keyword

Criteria logic for work item state

The following table describes the criteria logic for work item state.

CRITERIA	ACTION
If the value matches a state,	**Then** set it to that state.
Else If the value matches a state category,	**Then** set the work item to first state in that category. See the following note.
Else If the value matches a keyword,	**Then** set the work item to matching keyword state. See the following section.
Else	Ignore it and do nothing.

Criteria logic for work item keyword

The following table describes keyword logic, which helps with intent matching. For example, you might enter "Resolves", but you really meant "Resolved".

KEYWORD	ACTION
Proposed, Proposes, Propose	Set to the first state in the Proposed category.
InProgress	Set to the first state in the In Progress category.
Completed, Completes, Complete	Set to the first state in the Completed category.
Resolved, Resolves, Resolve	Set to the first state in the Resolved category.
Fixes, Fixed, Fix	Close work item. Except Bug, which gets set to Resolved.

> **NOTE**
> We don't support category matching on projects using a Hosted XML process. Category matching is only available for projects using an inherited process.

Link work items to a build in same or other projects

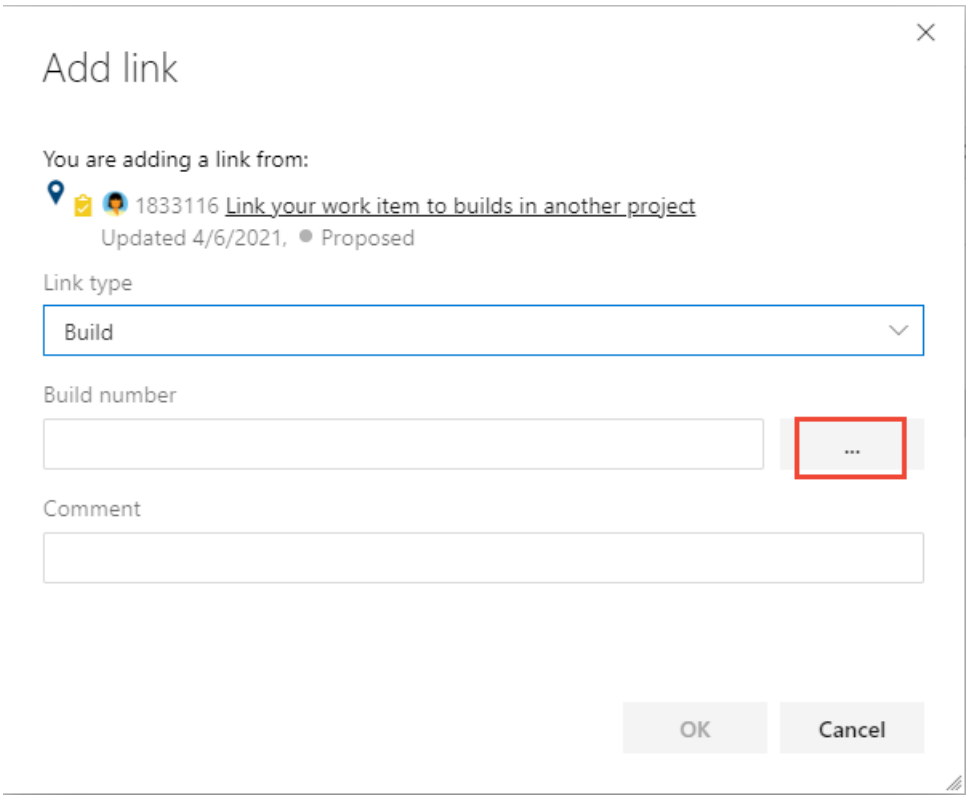
You can link work items to existing builds from the **Add link** dialog. These builds can be within your project or to other projects in your organization or collection.

NOTE

This feature requires installation of Azure DevOps Server 2020.1 update. To learn more, see Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards.

1. From the **Links** tab of a work item, choose **Add link > Existing item**.
2. From the **Add link** dialog, choose one of the build link types—**Build**, **Found in build**, **Integrated in build**— and specify the build number.

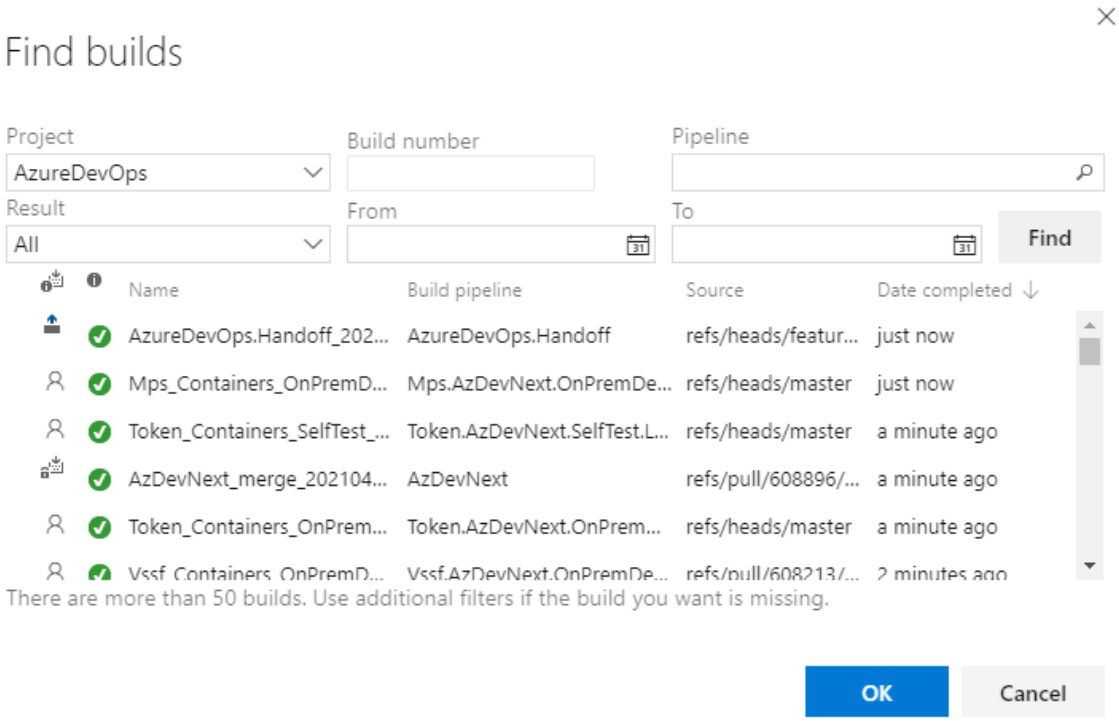
   If you don't know the build number—a combination of the pipeline and build name—you can search for it by choosing the ··· icon.



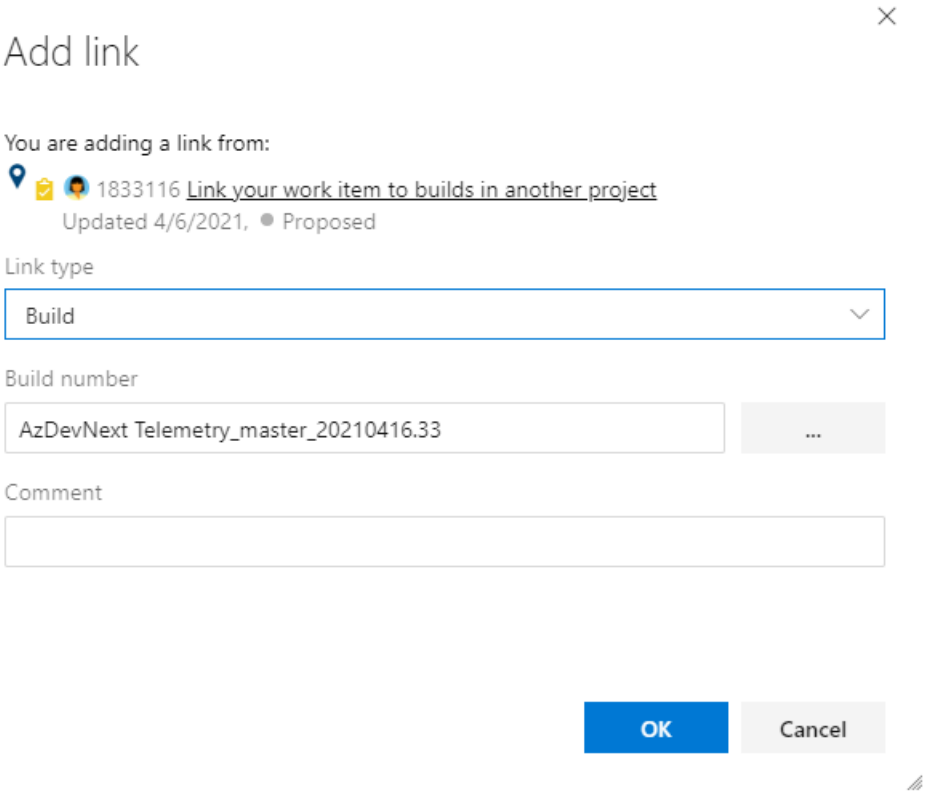
3. From the **Link builds** dialog, choose the parameters to filter your search of builds.

   If linking to a build in a different project, then first choose the **Project** whose build you want to link to.

   For example, you can specify a build number, select a build pipeline, a build result—such as, **All**, **succeeded**, **partially succeeded**, **failed**, or **canceled**. Or, with **All** selected for **Result**, choose **Find** to list the available builds to link to.



4. Choose the build from the list you want to link to and then choose **OK**.
5. From the **Add link** dialog, choose **OK** to complete the operation.

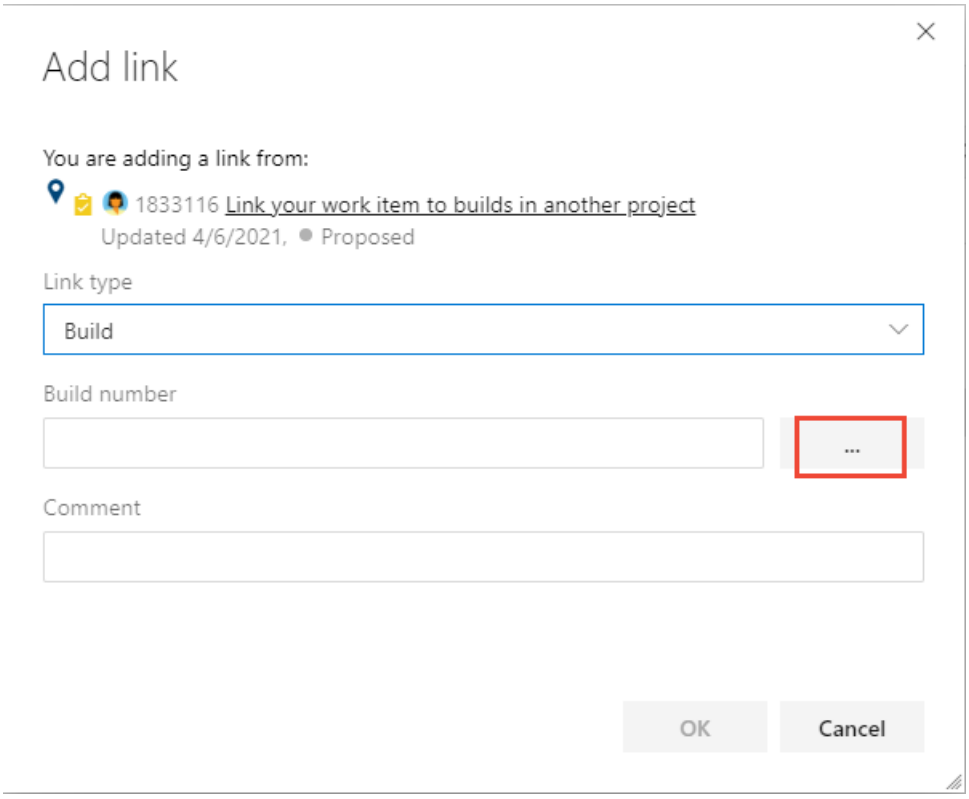


Link work items to a build

You can link work items to existing builds from the **Add link** dialog.

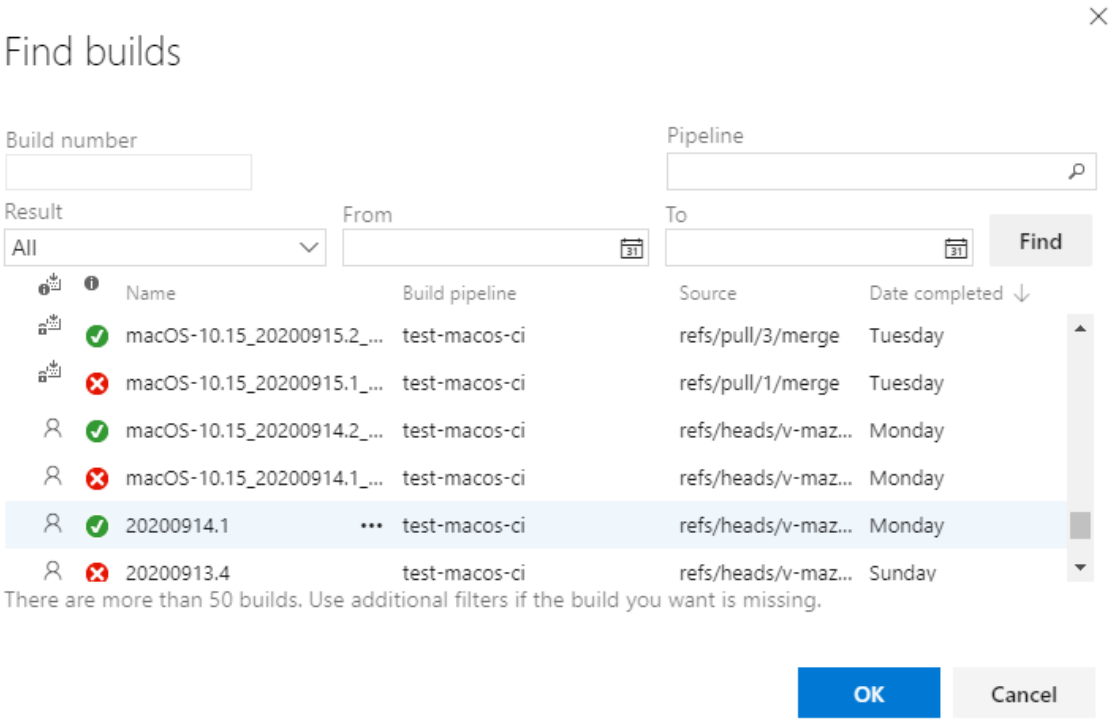
1. From the **Links** tab of a work item, choose **Add link** > **Existing item**.
2. From the **Add link** dialog, choose one of the build link types—**Build**, **Found in build**, **Integrated in build**— and specify the build number.

If you don't know the build number—a combination of the pipeline and build name—you can search for it by choosing the ··· icon.



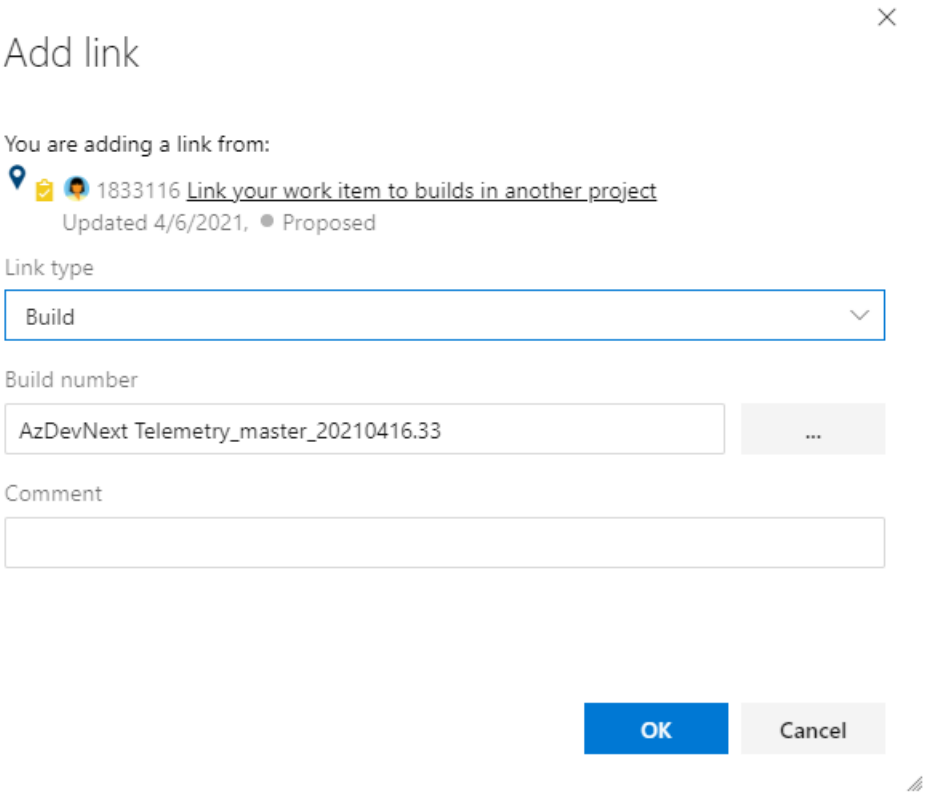
3. From the **Link builds** dialog, choose the parameters to filter your search of builds.

   For example, you can specify a build number, select a build pipeline, a build result—such as, **All**, **succeeded**, **partially succeeded**, **failed**, or **canceled**. Or, with **All** selected for **Result**, choose **Find** to list the available builds to link to.



4. Choose the build from the list you want to link to and then choose **OK**.

5. From the **Add link** dialog, choose **OK** to complete the operation.



For more information, see Link work items to user stories, issues, bugs, and other work items

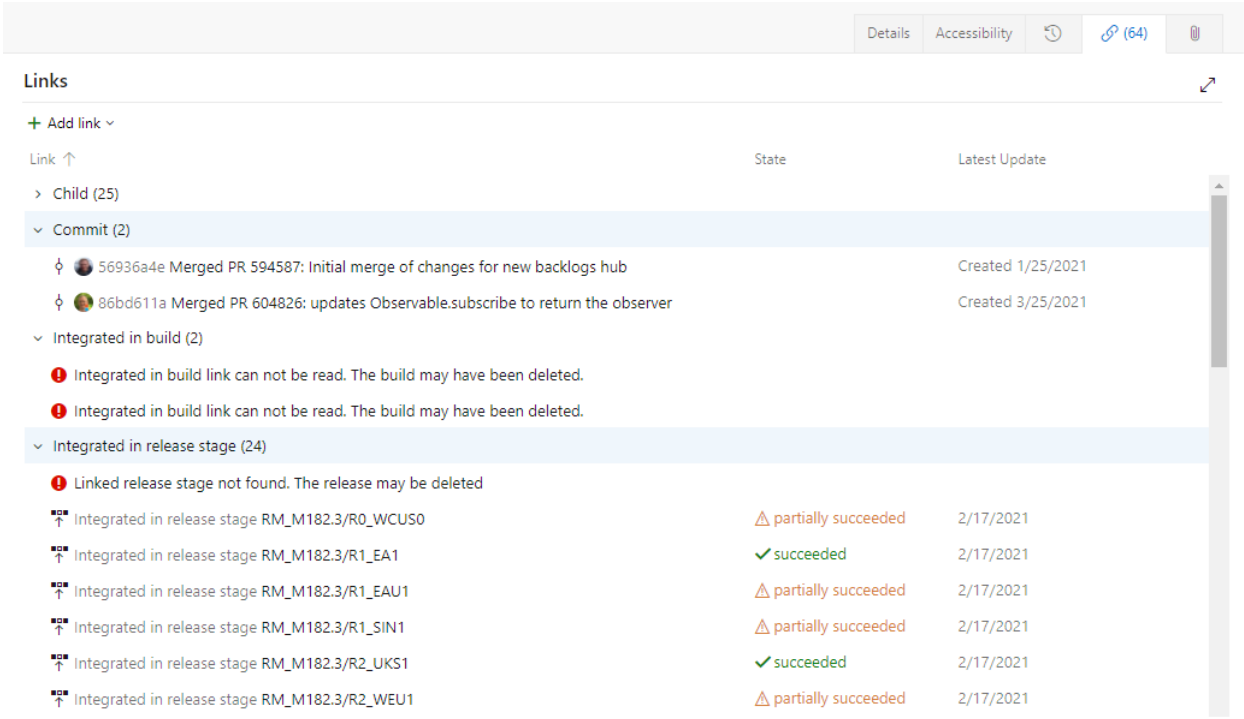
View list of linked objects

To view the list of all objects linked to a work item, open the work item and choose the **Links** tab. The links tab indicates the count of all linked objects.



Linked objects are grouped under their link type, with a count within each group. You can expand or collapse each group, and sort within each group by **State**, **Latest Update**, or **Comment** by choosing the corresponding column title.

For example, the following **Links** tab shows a portion of the 64 linked objects for a work item.



Links prefaced with the red exclamation mark indicate that the build, release, or other object has been deleted. This is usually due to retention policies which automatically delete these objects after a certain time period has passed.

> **NOTE**
> You can't create a work item query to list linked objects. Work item queries only return work items that are linked to other work items. However, you can create a query that lists work items that contain external links. For more information, see Query by link or attachment count.

Related articles

- End-to-end traceability
- Drive Git development from a work item
- Link type reference
- Save work with commits
- View and manage pull requests

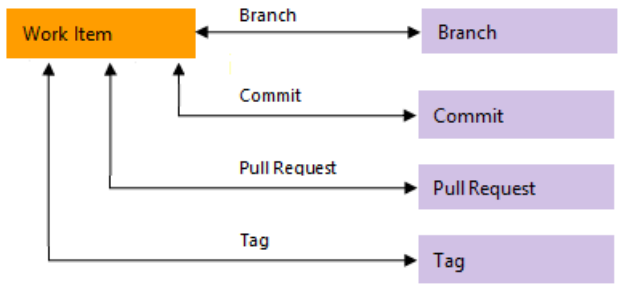
Configure repositories and branches to integrate with work tracking

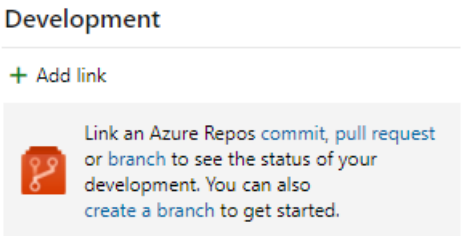
10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To support traceability of your Git code with work tracking, you can exercise one or more features and configure several options.

The following table summarizes the integration points between Azure Boards and Azure Repos, Git. Through various link types, you can track code changes—commits and pull requests for Git—that support development of user stories and features. The link types used to construct these links include *Branch , Commit, Pull Request*, and *Tag*.



Feature	Description
Manually link work items to Git branches, commits, pull requests, and tags	You can link from a work item or from a Git object. For details, see Link to work items from other objects, View list of linked objects.
Drive Git development from work item(s)	You can initiate a Git branch or link to Git commits or pull requests and drive your Git development cycle for a work item from within the work item form.  For details, see Drive Git development from a work item.
Automatically link and transition work items with Git commits	Enable or disable the following options for a single Git repository. To learn how, see Configure Git repositories to support work tracking. • Automatically create links for work items mentioned in a commit comment

- Allow mentions in commit comments to close work items
- Remember user preferences for completing work items with pull requests.

Check for linked work items in a Git branch

Encourage traceability by checking for linked work items on pull requests. For details, see Configure branch policies to support work tracking.

Auto complete work items with pull requests

When you link a work item to a pull request (PR), you have the option to automatically complete those work items when you successfully complete the PR. The system defaults to your selection for future PRs. For details, see Auto complete of work items with pull requests.

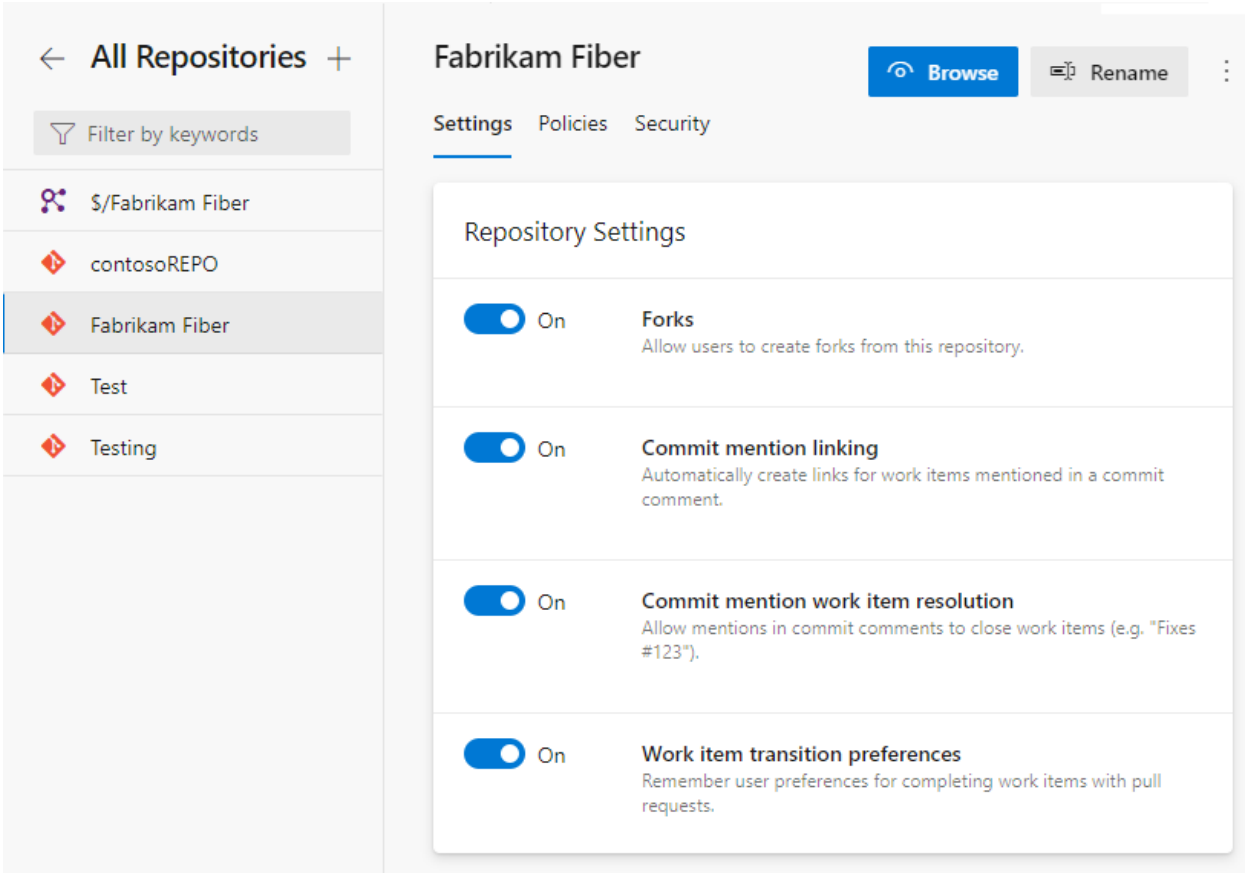
Query for external links

You can query for work items that contain links to branches, commits, pull requests or tags. To learn more, see Query by link or attachment count.

Configure Git repositories to support work tracking

You can enable several options to automate linking of work items from commits or pull requests.

To configure the repository, open **Project Settings > Repositories**, choose a repository, and then view the **Settings** tab.



Enable the following options:

- **Commit mention linking** When enabled, commit messages containing `#` followed by a valid work item ID will automatically link the commit to the mentioned work item. Disable this option when pushing a repository previously contained by a different account or service. If you import a repository, we

automatically turn this option off.

- **Commit mention work item resolution** Enable this option to support closing work items mentioned in commits with the `Fixes` prefix, for example `Fixes #*WorkItemID*`.
- **Work item transition preferences** When enabled, the default option, the system remembers each user's last choice to complete linked work items during pull request completion. Depending on your team's preference, you may choose to disable this option to discourage users from completing work items with their pull requests. When disabled, users must opt-in to completing work items for each pull request.

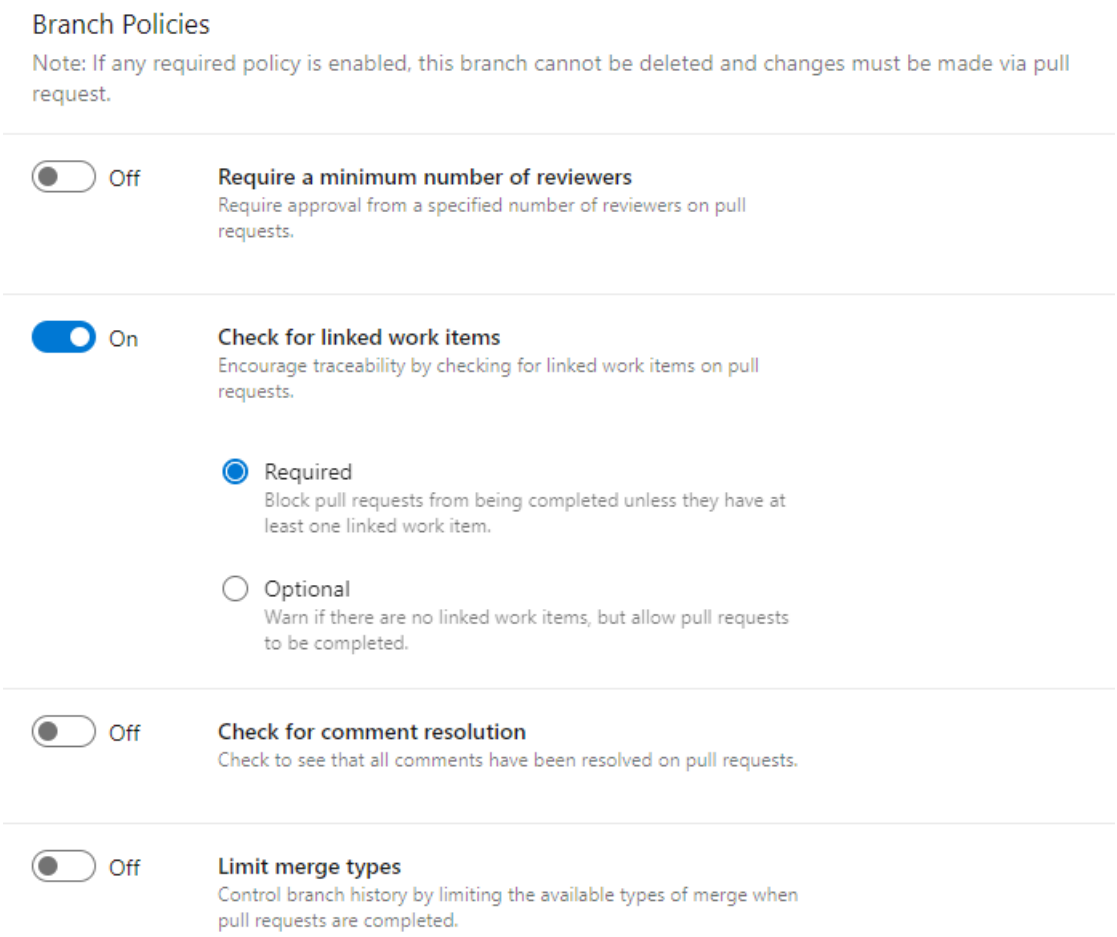
Configure branch policies to support work tracking

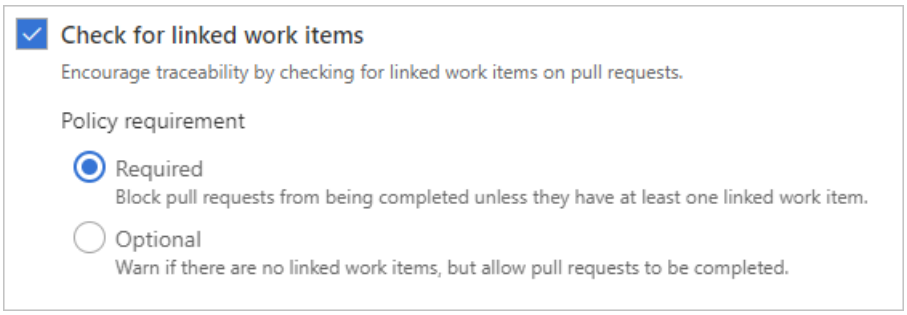
To ensure that changes to a branch have links to work items, you configure the branch policy for a Git repository.

To configure the branch, open **Project Settings** > **Repositories**, choose a Git repository, and then view the **Policies** tab. Scroll down and choose a branch under the **Branch policies** section. Linking work items provides additional context for your changes and ensures that updates go through your work item tracking process.

Turn on the **Check for linked work items** option.

- Choose **Required** to mandate all pull requests have at least one linked work item in order to be completed.
- Choose **Optional** to allow pull requests to be completed without linked work items, but issue a warning that there are no linked work items.





Related articles

- Link to work items from other objects
- Link type reference
- Drive Git development from a work item
- End-to-end traceability

Configure pipelines to support work tracking

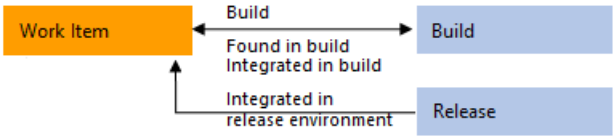
10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To support integration and traceability across Azure DevOps Services with pipelines, you can configure several options. You can report pipeline status, copy the syntax for status badges, and set up automatic linking of work items to builds and releases.

Supported pipeline and work tracking integration features

Several features provide support for end-to-end traceability as user stories and features move through the development cycle. As with Azure Repos, you can link work items to pipeline objects with the following link types: *Build, Integrated in build*, and *Integrated in release*. Note, that the *Integrated in release environment* link can only be created by enabling the **Report release status to Boards** option in Classic release pipelines.



The following table summarizes the integration points between Azure Boards and Azure Pipelines. Options and configuration steps differ depending on whether you are configuring a YAML or Classic pipeline, and your Azure DevOps version. Most options are supported for pipelines run against an Azure Repos Git repository unless otherwise noted.

Feature

Description

Supported versions

Manually link work items to builds

You can link from a work item to builds within the same project or other projects within the organization. For details, see Link to work items from other objects.

All versions

View builds linked to from a work item

You can view all builds linked to from a work item, whether manual or automatically linked, from the Links tab. For details, see Link to work items from other objects, View list of linked objects.

All versions

Automatically link work items to builds

Required to populate the **Development** control with *Integrated in build* links. The work items or commits that are part of a release are computed from the versions of artifacts. For example, each build in Azure Pipelines is associated with a set of work items and commits. For details, see Automatically link work items later in this article.

YAML, Azure DevOps Server 2020 and later

Classic, TFS 2017.2 and later

Automatically link work items to releases and report deployment status to a work item (Classic only)

Required to populate **Deployment** control in work item form with *Integrated in release stage* links. For details, see Report deployment status to Boards later in this article.

Azure DevOps Server 2020 and later

View list of work items linked to a build or release

Review and open the work items included in a build or release.

YAML, Azure DevOps Server 2020 and later
Classic, TFS 2017.2 and later

Create work item on failure (Classic)

Automatically create a work item when a build fails, and optionally set values for work item fields. For details, see Create work item on failure later in this article.

TFS 2018 and later versions

Query Work Items task, ensure the number of matching work items returned from a query is within a threshold.

Use this task to ensure the number of matching items returned by a work item query is within the configured thresholds. For details, see Query Work Items task, Control deployments with gates and approvals.

Azure DevOps Server 2020 and later versions

Prerequisites

- To configure the integration options for a Classic release pipeline, you must have permissions to edit the release.
- To link work items to commits and pull requests, you must have your **Edit work items in this node** permissions set to **Allow** for the Area Path assigned to the work item. By default, the Contributors group has this permission set.
- To view work items, you must have your **View work items in this node** permissions set to **Allow** for the Area Path assigned to the work item.

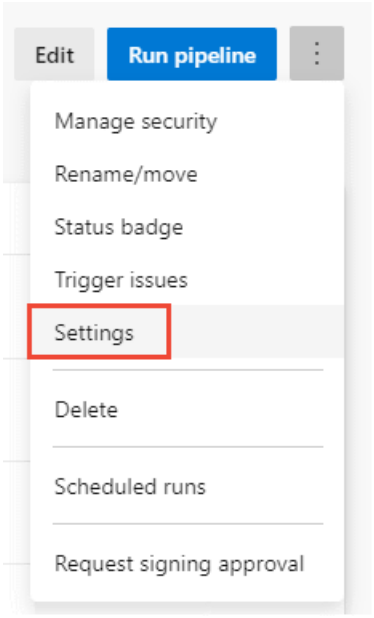
Open pipeline settings, build options, or integration options

- YAML
- Classic Build
- Classic Release

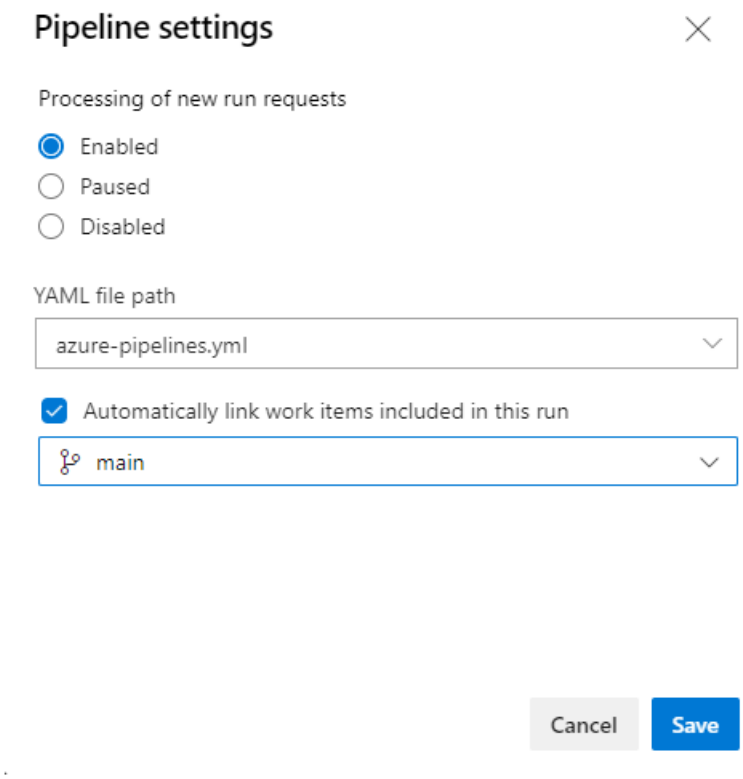
Open Pipeline settings

For YAML-defined release pipelines, you configure the integration through the **Pipeline settings** dialog.

1. Open the pipeline, choose ⋮ **More actions**, and then choose **Settings**.



The Pipeline Settings dialog appears. For details on automatic linking, see Automatically link work items later in this article.



This setting isn't available for Azure DevOps Server 2019 or earlier versions.

Automatically link work items to builds or releases

By enabling automatic linking, you can track the builds or releases that have incorporated work without having to manually search through a large set of builds or releases. Each successful build associated with the work item automatically appears in the **Development** control of the work item form. Each release stage associated with the work item automatically appears in the **Deployment** control of the work item form.

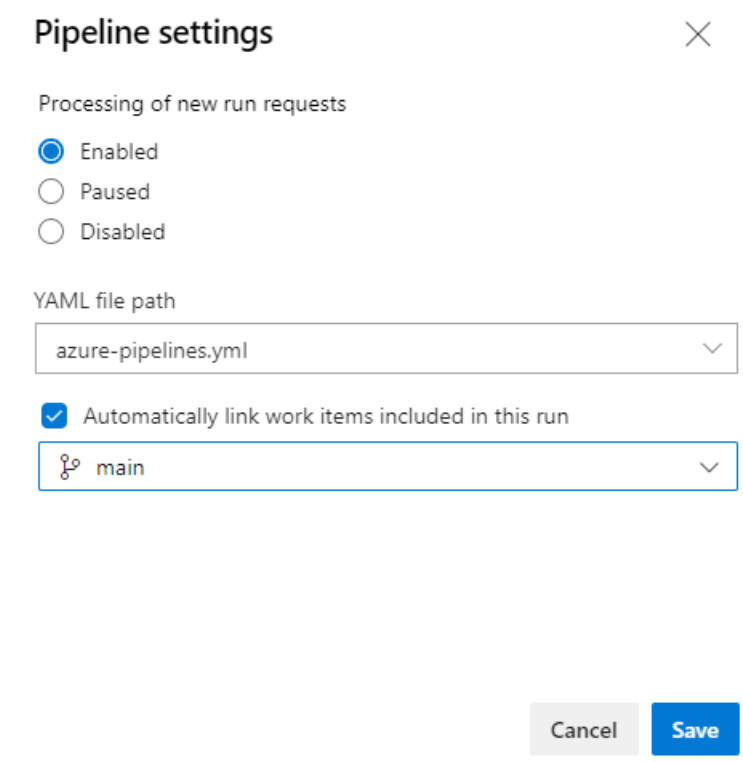
Automatically link work items to builds

By enabling automatic linking, you can track the builds that have incorporated work without having to manually

search through a large set of builds. Each successful build associated with the work item automatically appears in the **Development** control of the work item form.

- YAML
- Classic Build
- Classic Release

1. Open **Pipeline settings** as describe in Open Pipeline settings.
2. Enable **Automatically link new work in this build**.

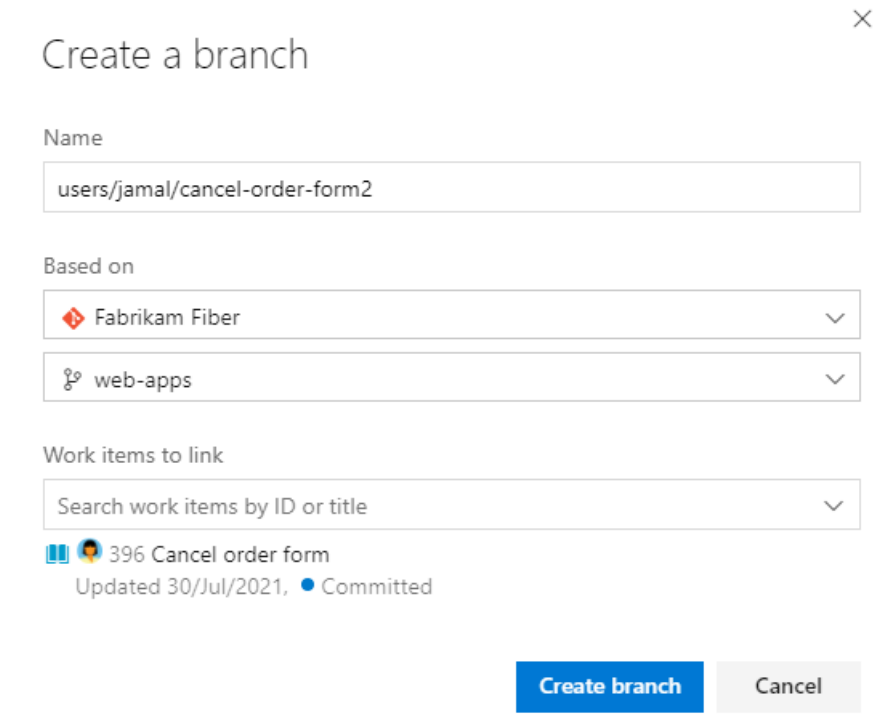


Once enabled, **Integrated in build** links are generated for all work items linked to the selected pull request with each release run.

This feature isn't supported for YAML pipelines in Azure DevOps Server 2019.

What work items are included in automatic linking?

When developing your software, you can link work items when you create a branch, commit, or pull request. Or, you can initiate a branch, commit, or pull request from a work item, automatically linking these objects as described in Drive Git development from a work item. For example, here we create a new branch from the Cancel order form user story.



When automatically linking work items to builds, the following computations are made:

- For a first-time build:
  - Identify all work items linked to the branch, commits, and pull requests associated with the build.
- For subsequent builds:
  - Identify all work items associated with the current commit (C1) being built.
  - Identify all work items associated with the commit (C2) of the last successful build of the same source branch.
  - Identify all work items associated with the commits between C1 and C2 in the commit tree.

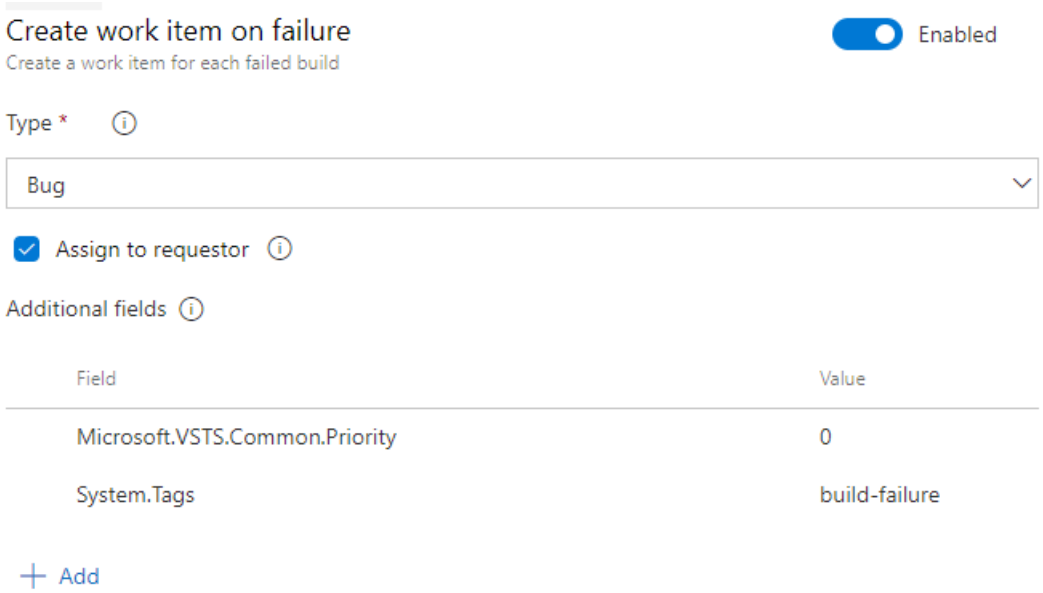
Create work item on build failure (Classic)

If a build pipeline fails, you can automatically create a work item to track getting the problem fixed. You can specify the work item type and set options to automatically assign it to the requestor or other fields. The requestor corresponds to the person that triggered the build.

> **TIP**
> The option to **Create work item on failure** is only supported for Classic pipelines. To accomplish this with a YAML pipeline, you can use a marketplace extension like Create Bug on Release failure or you can implement it yourself using Azure CLI or REST API calls.

1. Open pipeline build options as describe in Build properties.
2. Enable **Create work item on failure** and choose the type of work item to create. Optionally check the **Assign to requestor** checkbox to set the **Assign To** field and add fields to set within the work item to create.

   For example, here we choose the Bug work item type and specify the Priority and Tags fields and their values.



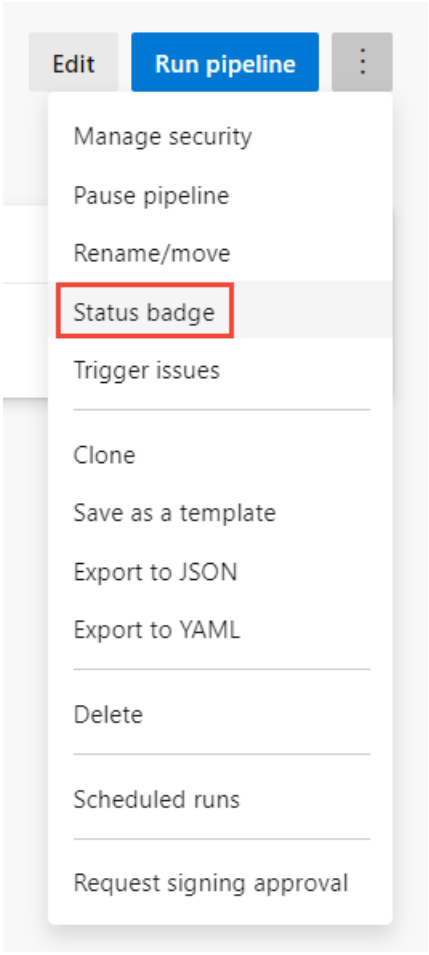
3. Save your pipeline.

To learn the reference name for a field, look it up from the Work item field index. For custom fields you add through an inherited process, Azure DevOps assigns a reference name based on friendly field name prefixed with *Custom.*. For example, you add a field named **DevOps Triage**, the reference name is Custom.DevOpsTriage. No spaces are allowed within the reference name.

Get or enable a status badge

- YAML
- Classic Build
- Classic Release

1. Open pipeline ⋮ **More Actions** and choose **Status badge**.



2. Choose the branch and scope of interest, and then choose **Copy to clipboard** to copy the image or Markdown syntax.



Related articles

- Define your multi-stage continuous deployment (CD) pipeline
- Link and view work items to builds and deployments
- Release pipelines (Classic) overview

- Configure repositories to support work tracking.
- How to retrieve all work items associated with a release pipeline using Azure DevOps API
- Drive Git development from a work item
- Link to work items from other objects
- End-to-end traceability
- Linking, traceability, and managing dependencies
- Link type reference

Drive Git development from a work item in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

One of the ways your team can drive their development and stay in sync is to link your work items to the objects created during development, such as branches, commits, pull requests, and builds. You can begin that linking by creating a branch from one or more work items. Later, you can create pull requests, quickly open commits, and maintain a record of development operations performed to complete specific work.

Review this article to learn:

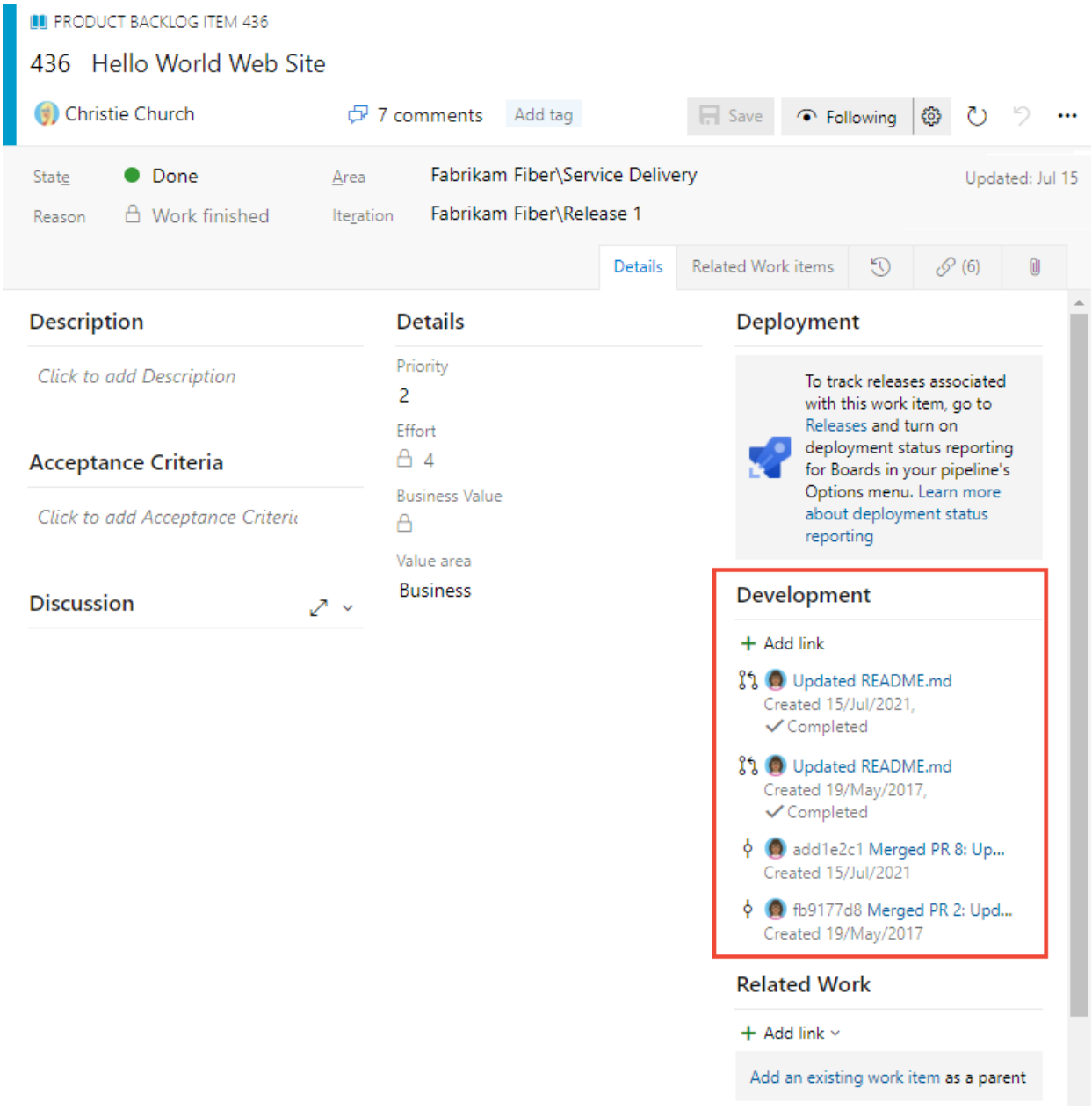
- How to create a new branch or pull request from a work item
- Complete the pull request
- Perform a squash merge
- Create a branch for several work items
- Link a work item to existing development and build objects

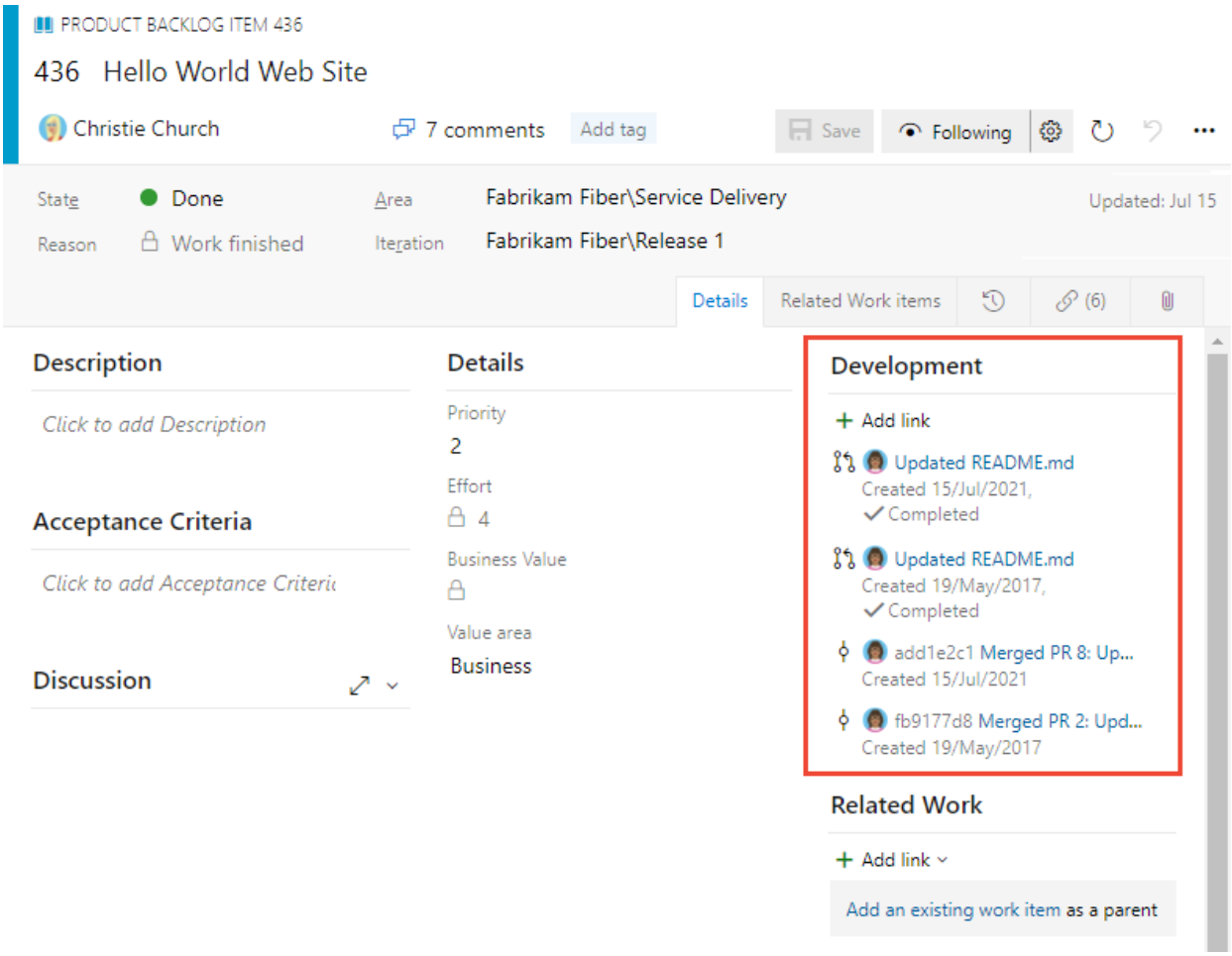
This article addresses creating new branches, adding links to commits, and adding pull requests to a Git repository hosted on Azure DevOps. To link to GitHub commits and pull requests, see Link GitHub commits and pull requests to work items.

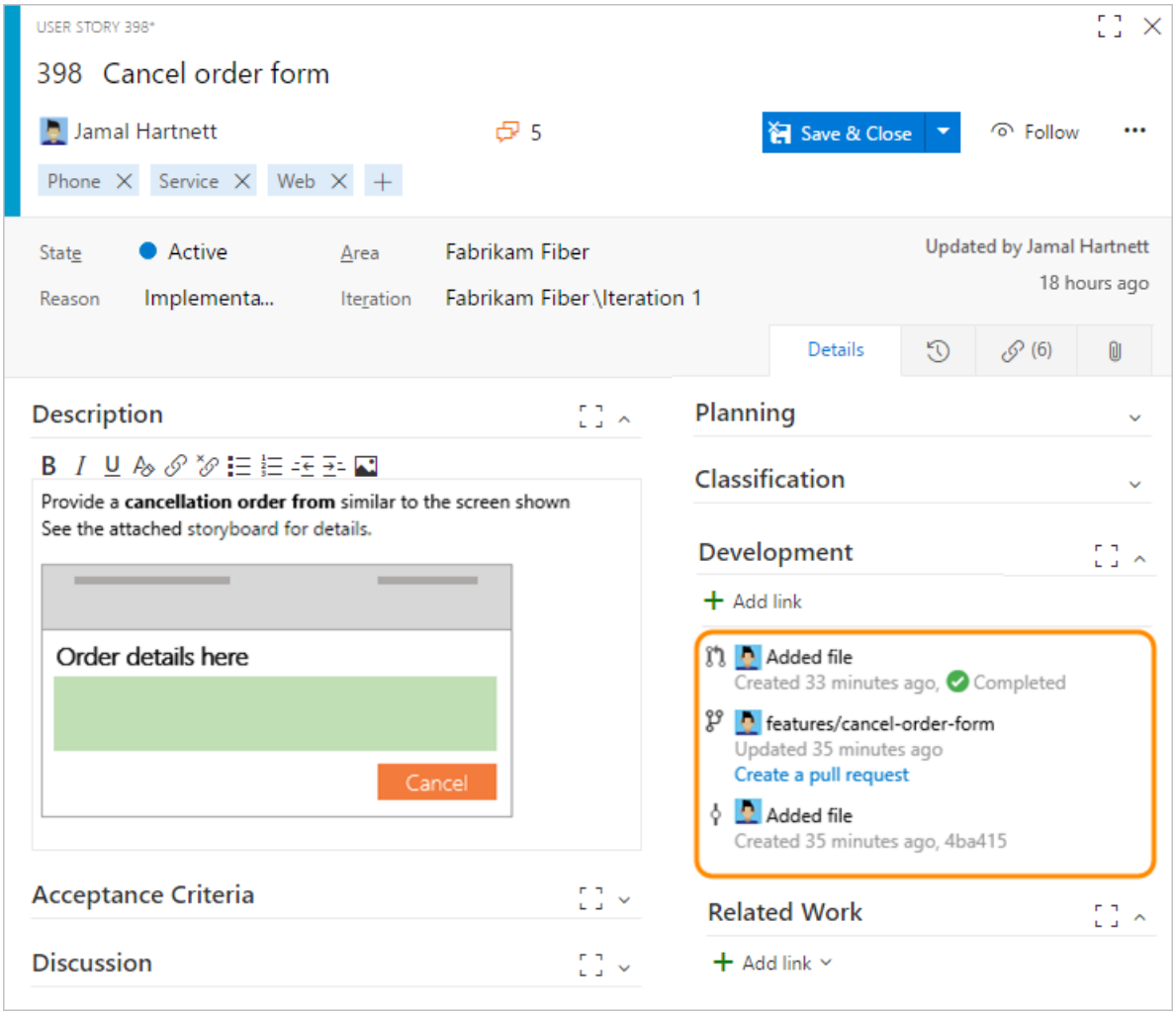
> **TIP**
> You can set up automatic linking and other settings that link work items to Git commits, pull requests, builds, and more. To learn how, see the following resources:
>
> - Configure repositories and branches to integrate with work tracking
> - Configure pipelines to support work tracking
> - Link and view work items to builds and deployments.

Development control

The **Development** control records all Git development processes that support completion of the work item. This control can show your team information needed to take the next development step and minimize navigational steps to accomplish common development tasks. It also supports traceability, providing visibility into all the branches, commits, pull requests, and builds related to the work item.


PRODUCT BACKLOG ITEM 436
436 Hello World Web Site
Christie Church
7 comments
Add tag
Save
Following
State
Done
Area
Fabrikam Fiber\Service Delivery
Updated: Jul 15
Reason
Work finished
Iteration
Fabrikam Fiber\Release 1
Details
Related Work items
(6)
Description
Click to add Description
Acceptance Criteria
Click to add Acceptance Criteri
Discussion
Details
Priority
2
Effort
4
Business Value
Value area
Business
Deployment
To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting
Development
Add link
Updated README.md
Created 15/Jul/2021,
Completed
Updated README.md
Created 19/May/2017,
Completed
add1e2c1 Merged PR 8: Up...
Created 15/Jul/2021
fb9177d8 Merged PR 2: Upd...
Created 19/May/2017
Related Work
Add link
Add an existing work item as a parent





From it, you can quickly access branches, pull requests, and commits which are linked to the work item. Also,

you can start a pull request for a branch you've created or linked to from the work item.

Keep in mind that the **Development** control only appears within the web portal work item form. The work item tracking experience and forms that appear in Visual Studio or other supported clients don't display several of the features that are available from the web portal.

Prerequisites

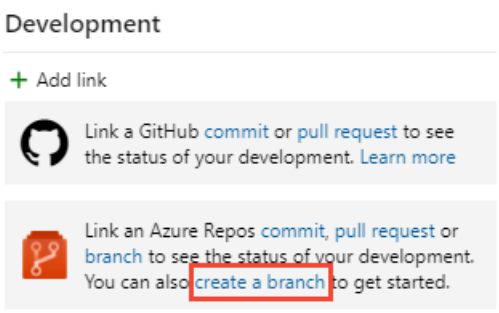
- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.

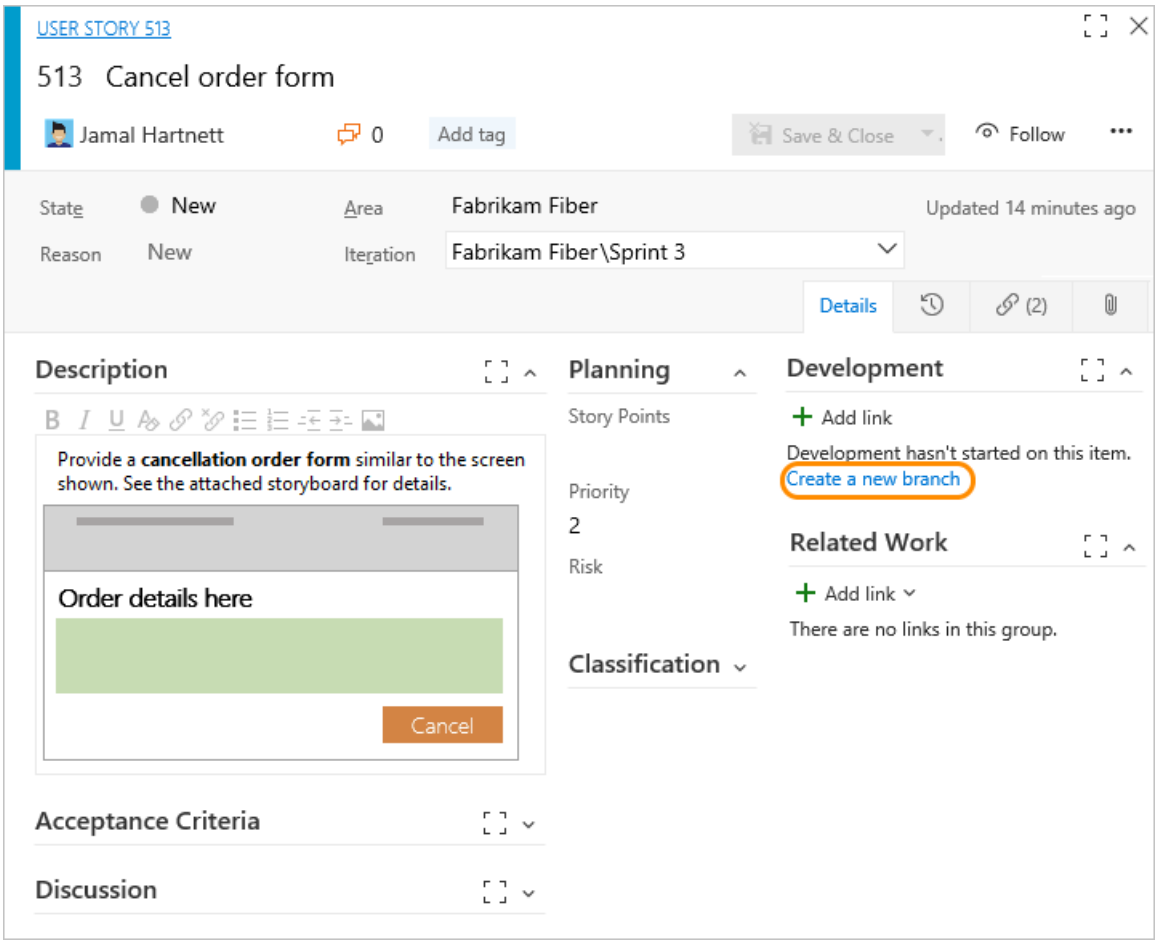
- Connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To view or modify work items, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.

Workflow process

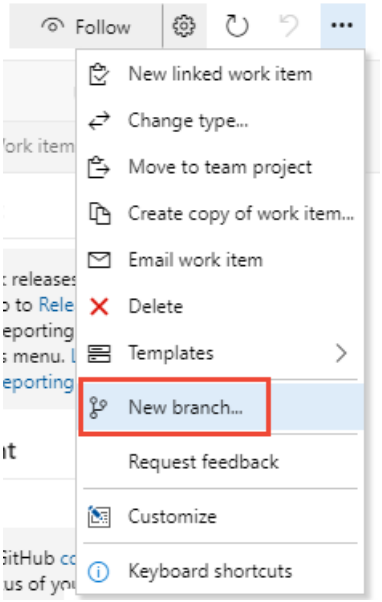
Consider creating a new branch when there are no linked code artifacts. If a branch exists, but no pull requests, consider creating a pull request. Here's a typical workflow sequence when working with a Git repository.

1. Start work on the work item by creating a branch. You can add a new Git branch from within the Development section...

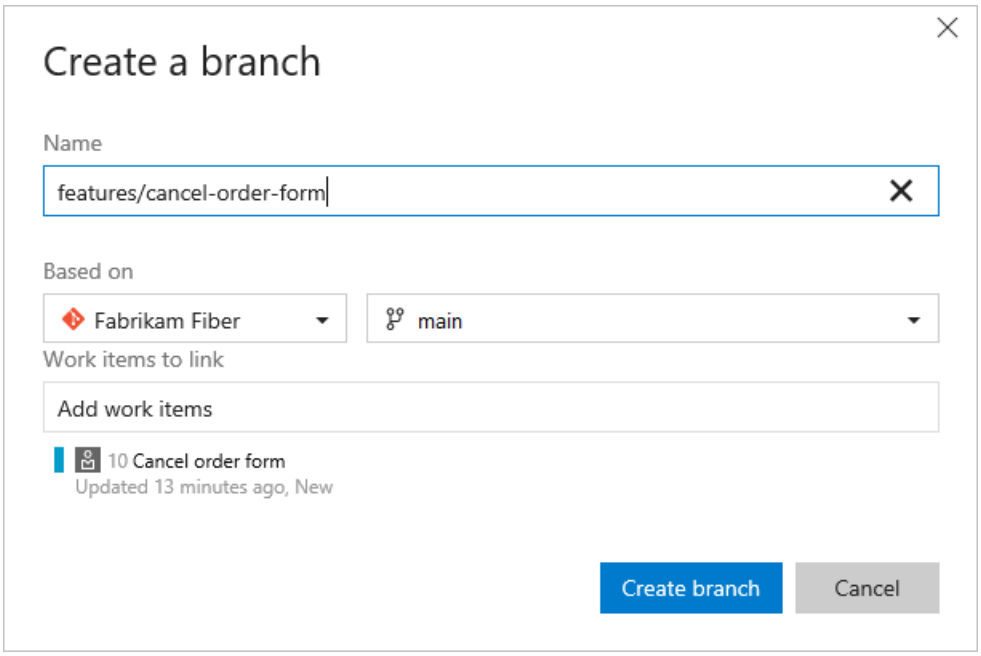




... or, from the form's ··· Actions menu.



Name the branch and select the repository on which it's based.



Branches you create are automatically linked to the work item.

> **NOTE**
> You can only create a branch once you've added files to the main branch, which we recommend you label `main` or other distinctive label. The system automatically adds a README file to the initial repo created with each new project.

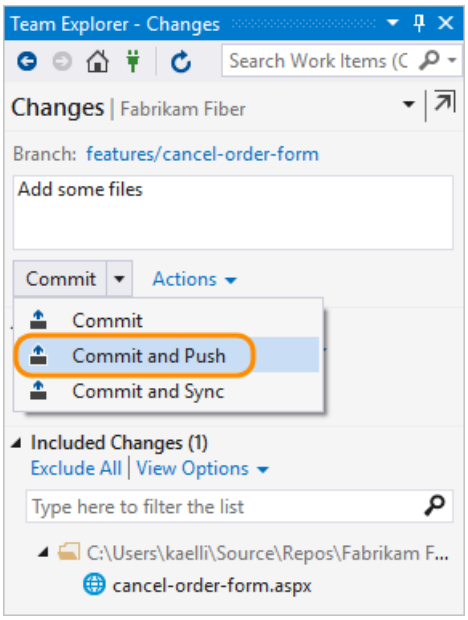
2. The system will open to the repository and branch that you created.

   You can edit a file within the web portal.

   Or, if you have extensive file edits or need to add files, then you'll need to work from Visual Studio or other supported IDE. You'll want to add a new local branch from the branch you created. For details, see Update code with fetch and pull, Download changes with fetch. (While any code editing and committing process will work, we work best with an edition of Visual Studio.)

3. Add or modify files in the branch that you created.

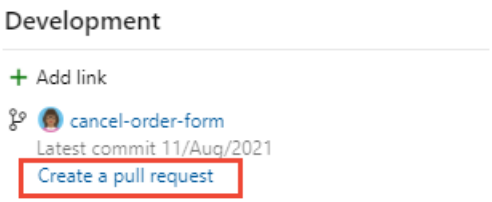
   From Visual Studio or other supported IDE, commit and push changes from your local branch to the repository.



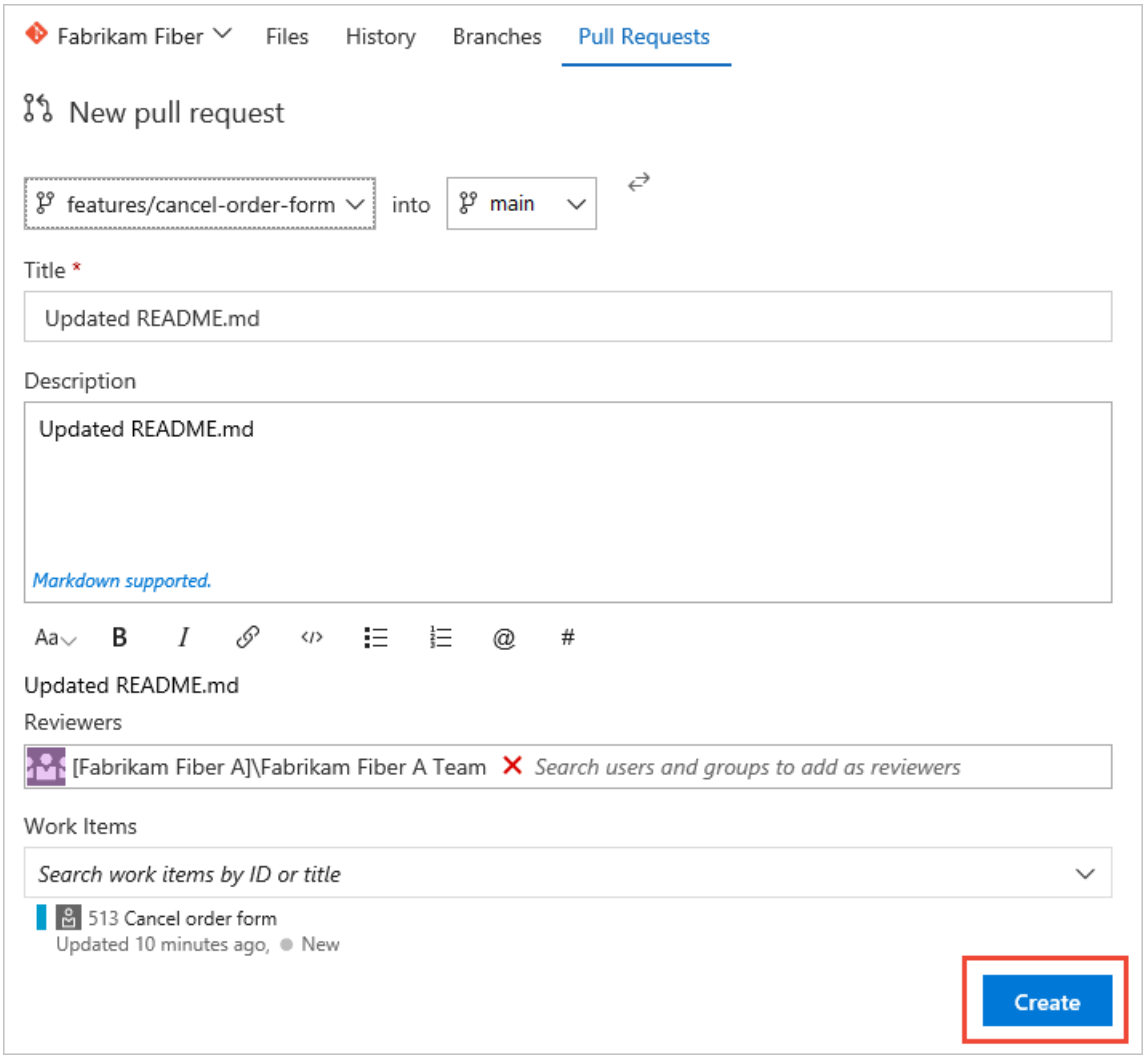
If this is the first time pushing changes from a new branch, you'll need to publish the branch before pushing your changes. For more information, see Share code with push.

4. Create a pull request from the work item form.

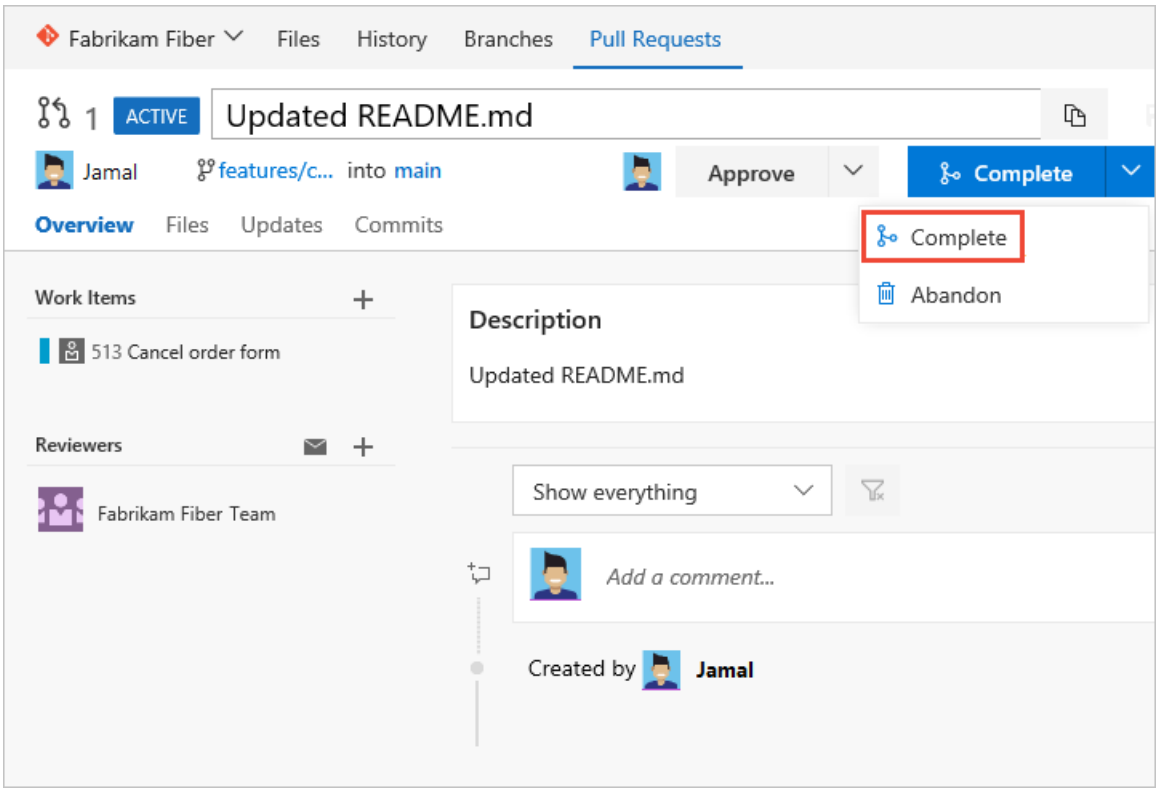
   You create a pull request to merge the changes you made to a main branch and get your changes reviewed by other members of your team.



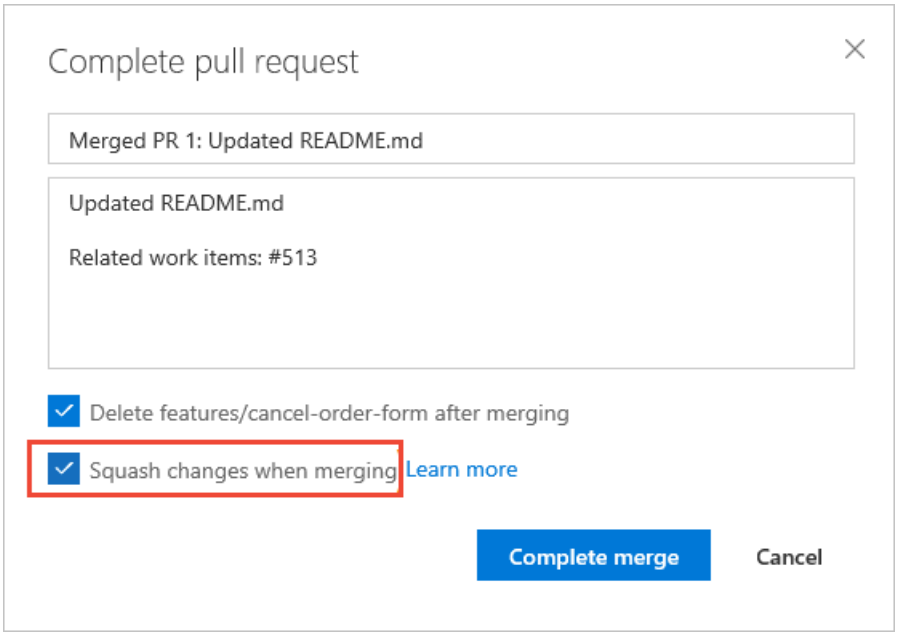
5. Your view will switch to **Code**, Pull Requests page. Complete creating the pull request as shown.



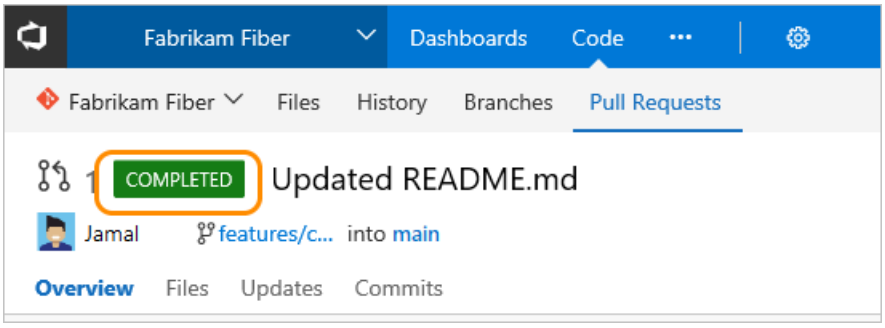
> **NOTE**
>
> Once you've created a pull request, you can't create a new pull request for the same branch until you complete the previous pull request.



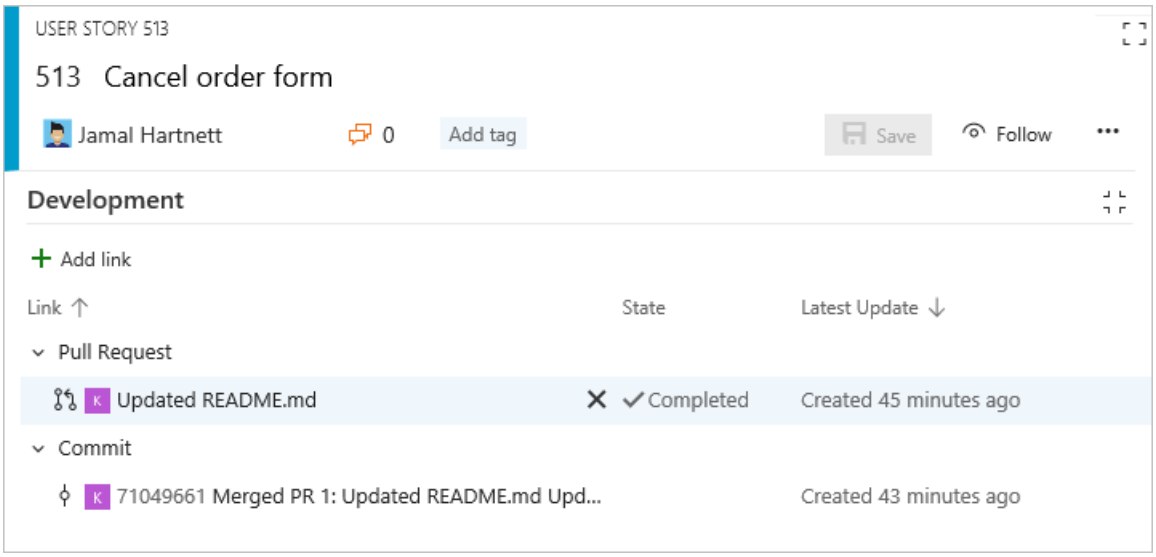
Check the box for **Squash changes when merging** and then complete the merge.



6. Upon completion, you should see a similar screen as follows.



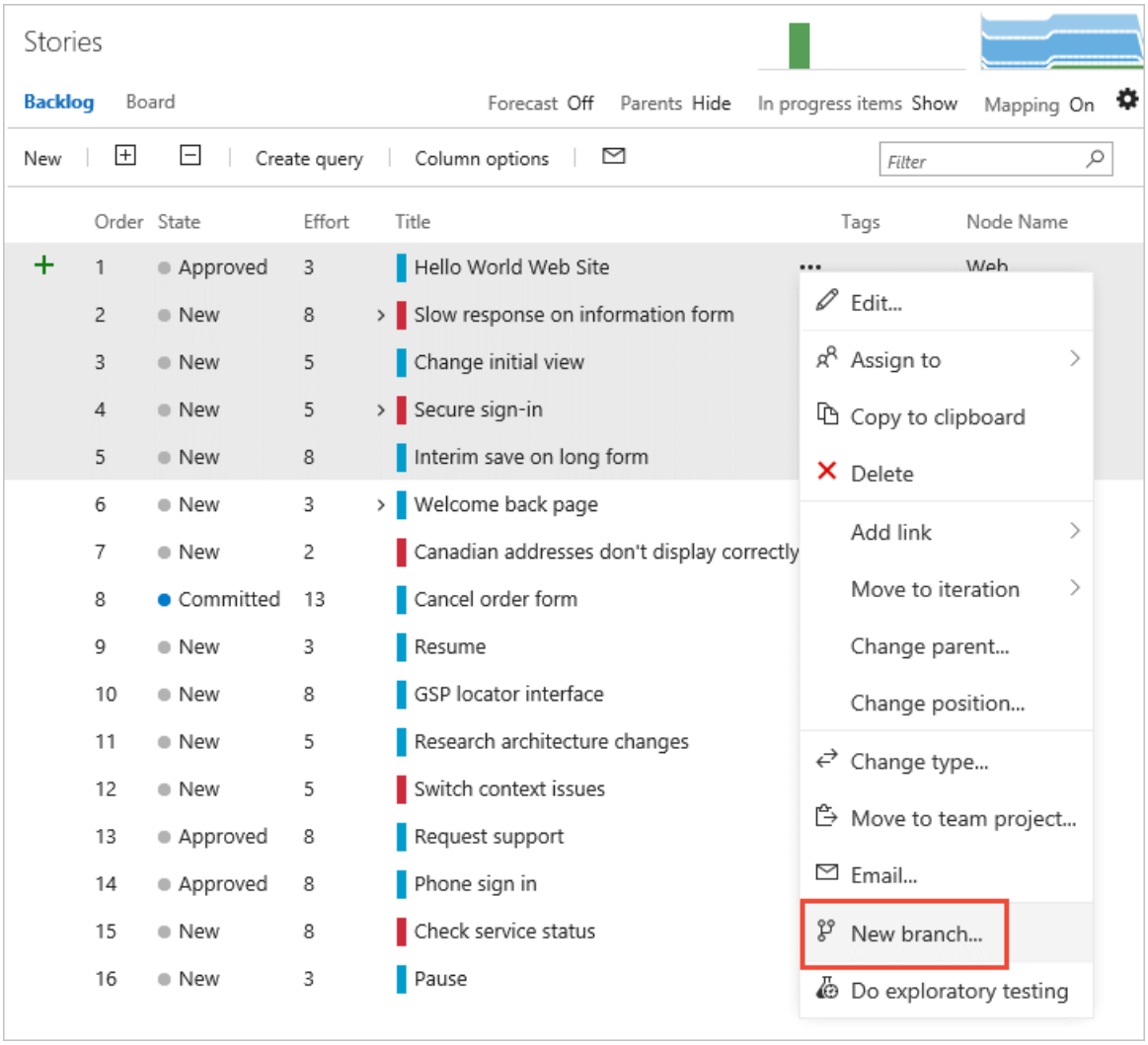
7. Open the work item form or refresh the form, expand the **Development** section (choose **Maximize Development**), and you'll see the links that have been added to support the operations you completed.



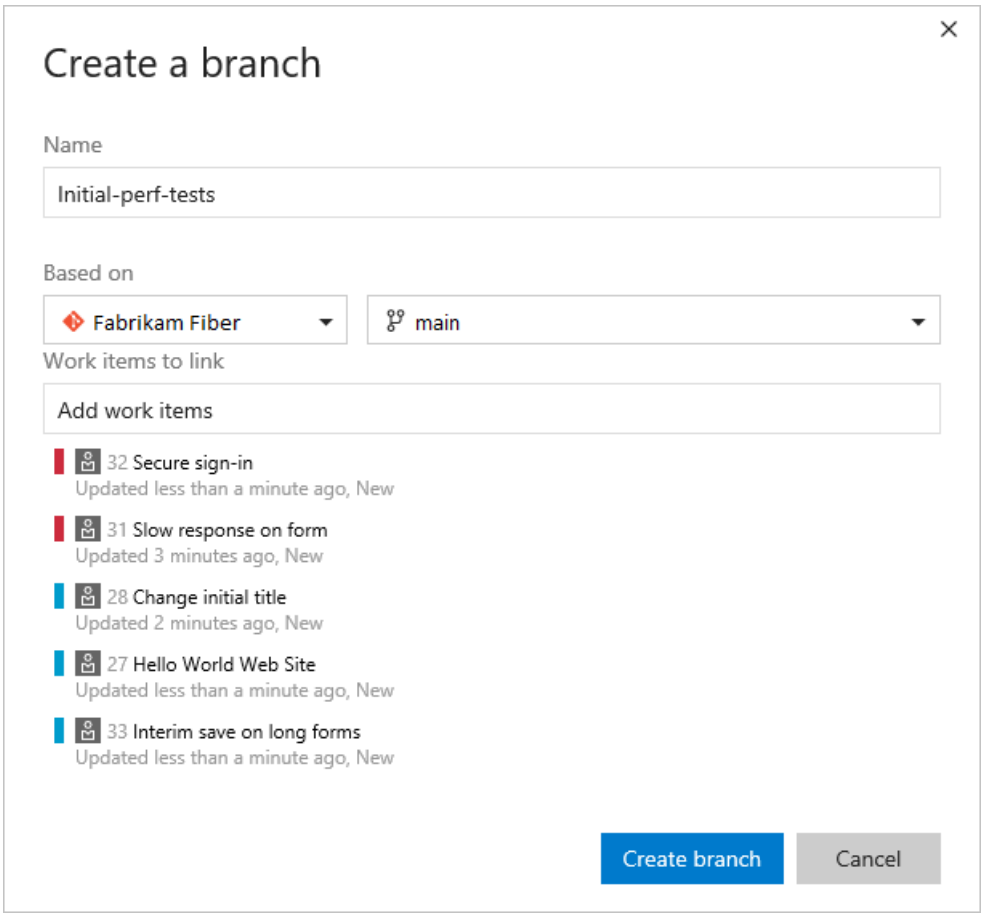
Create a branch for several work items

You can also add a new branch from the work item listed on the backlog or Kanban board without having to open the work item. Using multi-select, you can select several work items and create a new branch where they're all linked to the branch.

For example, here we select the first five items to link to a new branch.

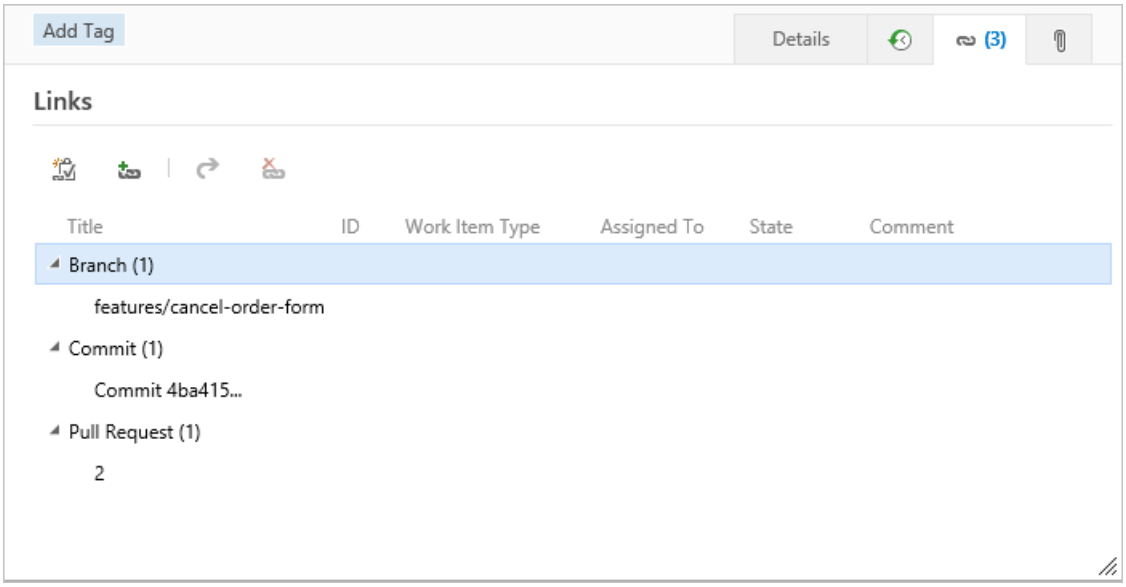


And, we specify the name of the branch.

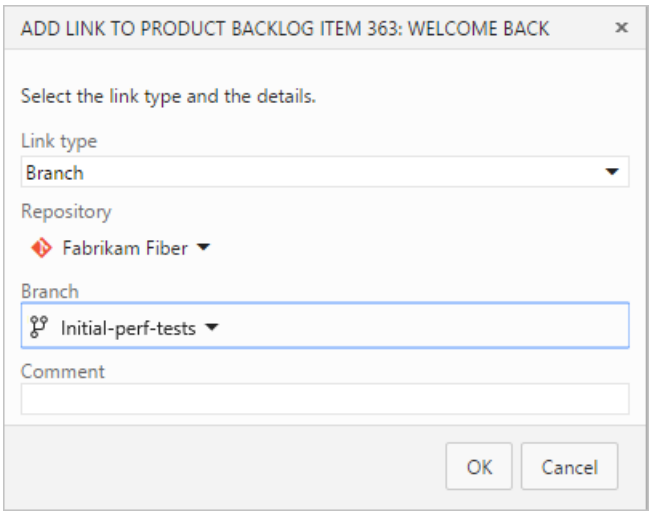


Link to existing development and build objects

All items listed under the Development section also appear under the **Links** tab. All development actions started from the Development section are also recorded under the **History** tab.



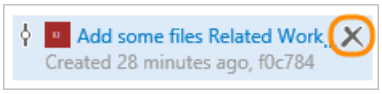
To link a work item to an existing object, choose the Add links icon and then choose the link type.



Linking, traceability, and managing dependencies.

Remove a link

If you want to remove a link, you can do so from the **Development** section by highlighting it first and then choose **Remove link** ✕ .



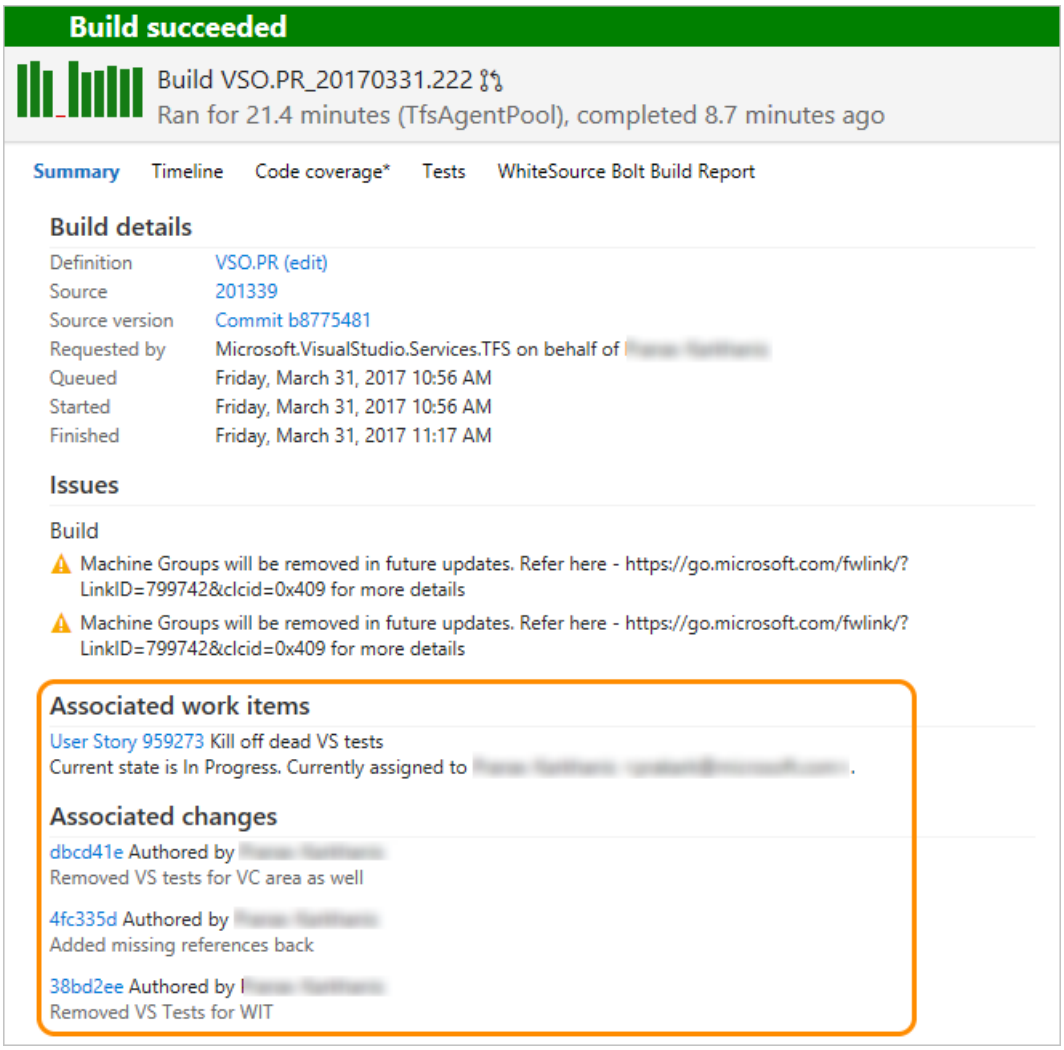
Or, you can select it from the **Links** tab and choose **Actions** ··· for the link and then choose the **Remove link** ✕ option.

Related articles

- Configure repositories and branches to integrate with work tracking
- Configure pipelines to support work tracking
- Add work items
- Git overview
- Link GitHub commits and pull requests to work items
- Link to work items from other objects

Associated work items in build

With Git commits, any work items that have been linked to a commit are listed under the Associated work items in the build summary page.



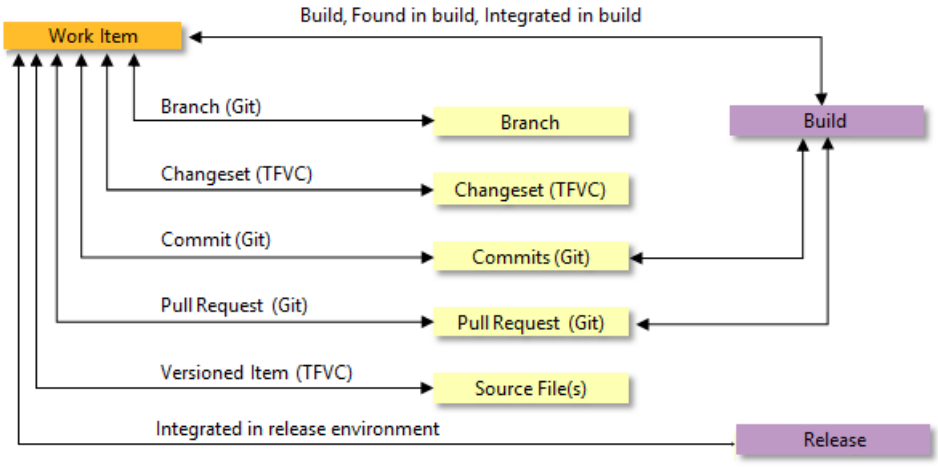
Link types showing in the Development section

Links shown in this section appear because of these actions:

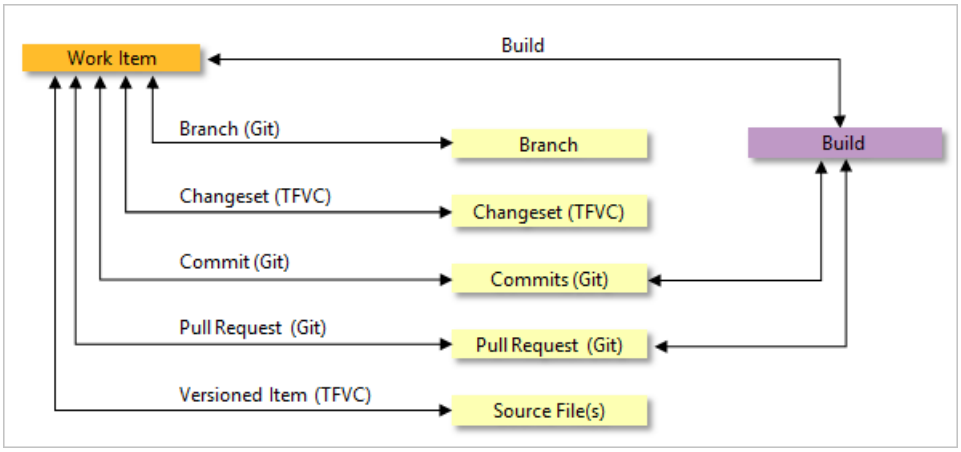
- Creating a branch, commit, or pull request from the work item
- Specifying the work item ID during a commit, pull request, or other supported Git or TFVC operation
- Specifically linking the work item from the **Development** section or **Links** tab to a source code branch, build, or other supported Git or TFVC operation.

Hovering over any entry listed under the Development section activates the hyperlink to the associated object.

The link types you can add within the development section are Branch, Build, Changeset, Commit, Found in build, Integrated in build, Pull Request, and Versioned Item.



The link types you can add within the development section are Branch, Build, Changeset, Commit, Pull Request, and Versioned Item.



To learn more about the links control or to customize the Development links control, see LinksControlOptions elements, Development links control.

Link work items to builds and deployments in Azure Boards

10/4/2022 • 4 minutes to read • Edit Online

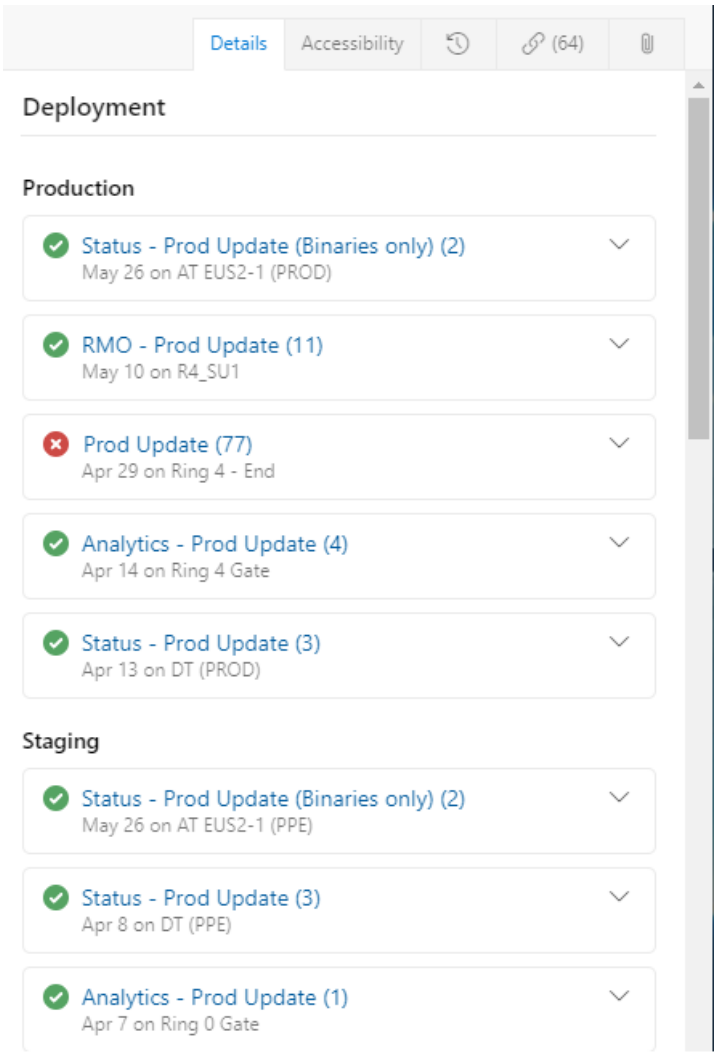
Azure DevOps Services | Azure DevOps Server 2022 | Azure DevOps Server 2020

One of the main ways Azure DevOps supports traceability is by linking objects. Work items link to Git branches, commits, pull requests, builds, and more. Work item forms provide two controls to show and quickly navigate to development objects. The **Deployment** control is described in this article, and the **Development** control is described in Drive Git development from a work item.

With the **Deployment** control, you can determine at a glance whether a feature or user story has been deployed and to what stage. You gain visual insight into the status of a work item as it is deployed to different release environments and quick navigation to each release stage and run.

> **NOTE**
> The **Deployment** control requires configuration of a Classic release pipeline. It doesn't support linking to release stages defined for a YAML pipeline.

As shown in the following image, the **Deployment** control shows release information for two release stages those work items that have been linked to a Git commit or pull request for a release pipeline configured to integrate with Azure Boards.

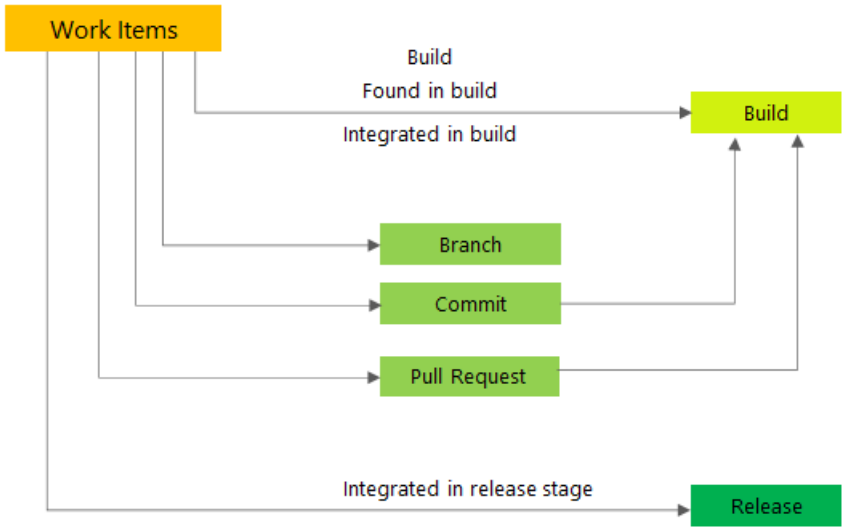


How linking is supported

Work items linked to a Git repository branch, commit, or pull request participate in populating the **Deployment** control.

You can view all links through the work item form **Links** tab.

- Work items associated with commits in the build will show the status of the release
- Only work items co-located with the same project where the release pipeline is defined are linked to.



To learn how to associate work items to commits, see Drive Git development from a work item or Link to work

items from other objects. To view objects linked to a work item, see View list of links for a work item.

Prerequisites

- To configure the integration options for a Classic release pipeline, you must have permissions to edit the release.
- To link work items to commits and pull requests, you must have your **Edit work items in this node** permissions set to **Allow** for the Area Path assigned to the work item. By default, the Contributors group has this permission set.
- To view work items, you must have your **View work items in this node** permissions set to **Allow** for the Area Path assigned to the work item.

To populate the **Deployment** control, complete the following steps:

1. Define a Classic release pipeline and set up the release stages as described in Define your multi-stage continuous deployment (CD) pipeline.
2. Configure the pipeline as described in Configure pipelines to support work tracking, Report deployment status to Boards.
3. Link work items to a commit or pull request in Azure Repos Git repository. For details, see:
   - Drive Git development from a work item
   - Link to work items from other objects
4. Run the pipeline.

Deployment control and work item types

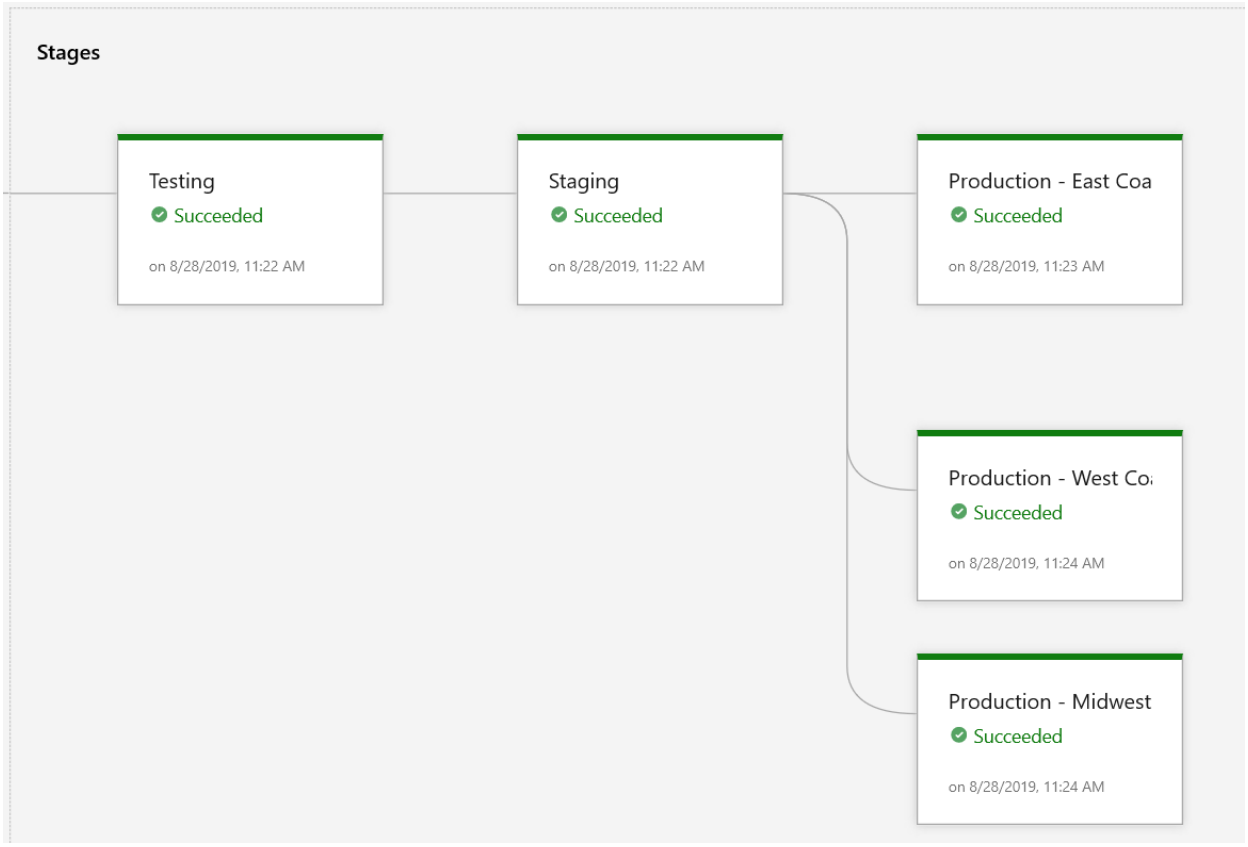
By default, the Deployment control appears on the work item forms for User Story (Agile), Product Backlog Item (Scrum), Issue (Basic), Requirement (CMMI), Feature, Epic, Bug, Task, and Test Case work item types. It's automatically enabled for custom work item types you define using the Inherited process. If you don't use the control, you can choose to Hide from layout.

If your project is customized using a Hosted XML process or you need to add it to a custom work item type for an On-premises XML process, you'll need to update your work item type definitions to display the control. For details, see Hosted XML process model, Add release deployment support to a work item type.

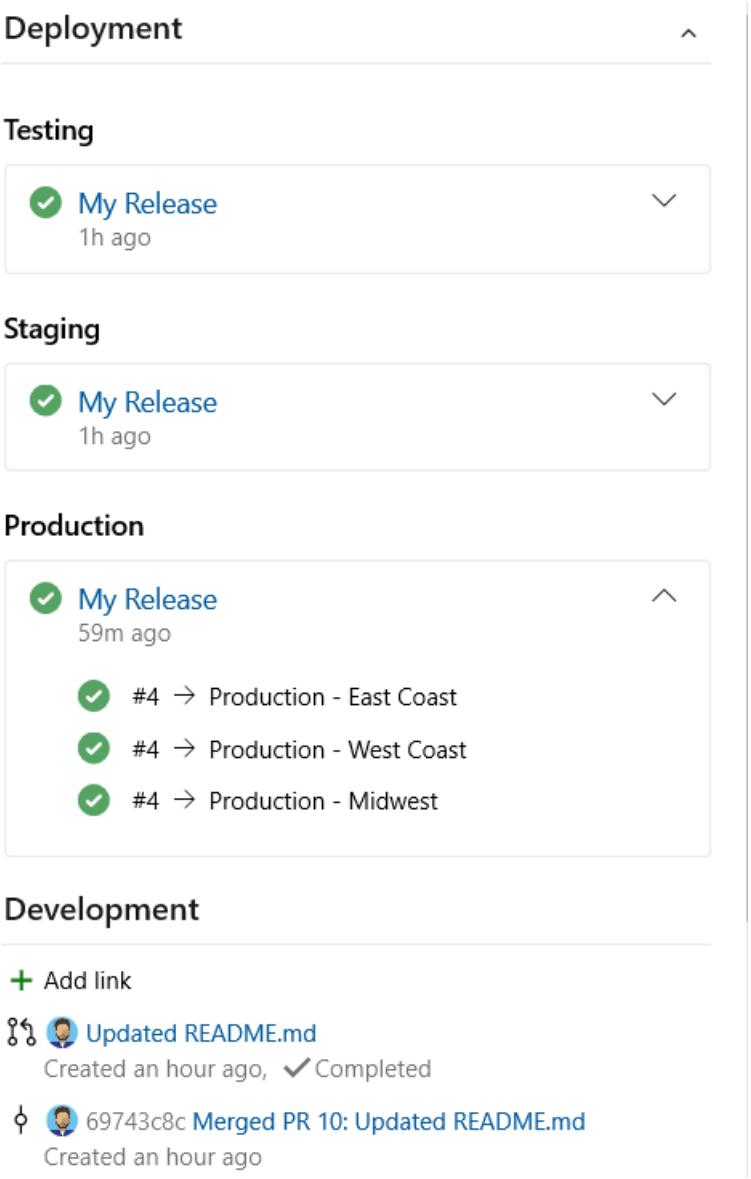
Deployment control

The work item deployment control displays the status of releases within those work items that are associated with commits in the build and those release pipelines you've configured to report deployment information to Azure Boards.

The following example shows multiple environments that the release is targeting which the selected work item is associated with.

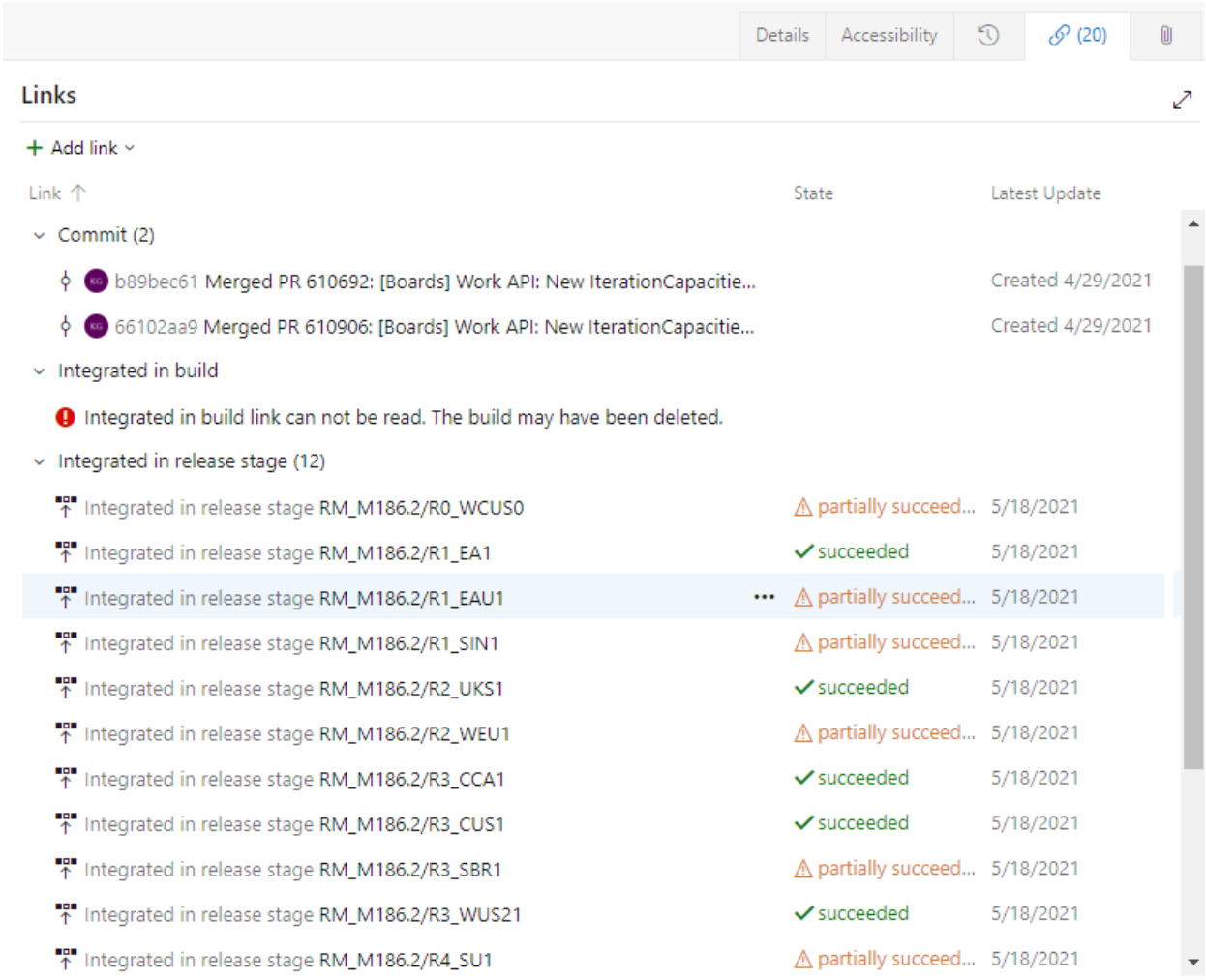


When you open the work item, you can see the stages the release is being deployed, in real time.



View a list of links for a work item

To view and navigate to the builds and releases linked to a work item, choose the **Links** tab. Links are grouped under their link type and listed in the order they were created. Choose the **State** or **Latest Update** column headings to sort by the column. Links prefaced with the red exclamation mark indicate that the build, release, or other object has been deleted. This indicator occurs because of retention policies, which automatically delete these objects after a certain time period has passed.



Unlink work items

Once a work item is linked to a commit or pull request, it will continue to show up as part of the release stages. For example, if you have a work item that didn't pass testing criteria, you may want to remove it from the builds and releases.

To remove the work item from participating in future builds and releases, delete the link to the most recent commit and pull request. You can do that by opening the **Links** tab for the work item as shown in the previous section.

Query for linked work items

You can't query for work items that are included in releases. However, you can create a query for work items with an `External Link Count > 0`. Include other query filters to optimize your search criteria.

Related articles

Azure Repos, Git

- Configure repositories and branches to integrate with work tracking
- Drive Git development from a work item

Azure Pipelines

- Define your multi-stage continuous deployment (CD) pipeline
- Release pipelines (Classic) overview
- Configure pipelines to support work tracking.

- How to retrieve all work items associated with a release pipeline using Azure DevOps API

Link work items

- Link to work items from other objects
- End-to-end traceability
- Linking, traceability, and managing dependencies
- Link type reference

Process customization

- Hosted XML process model, Add release deployment support to a work item type

Autocomplete work items with pull requests

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

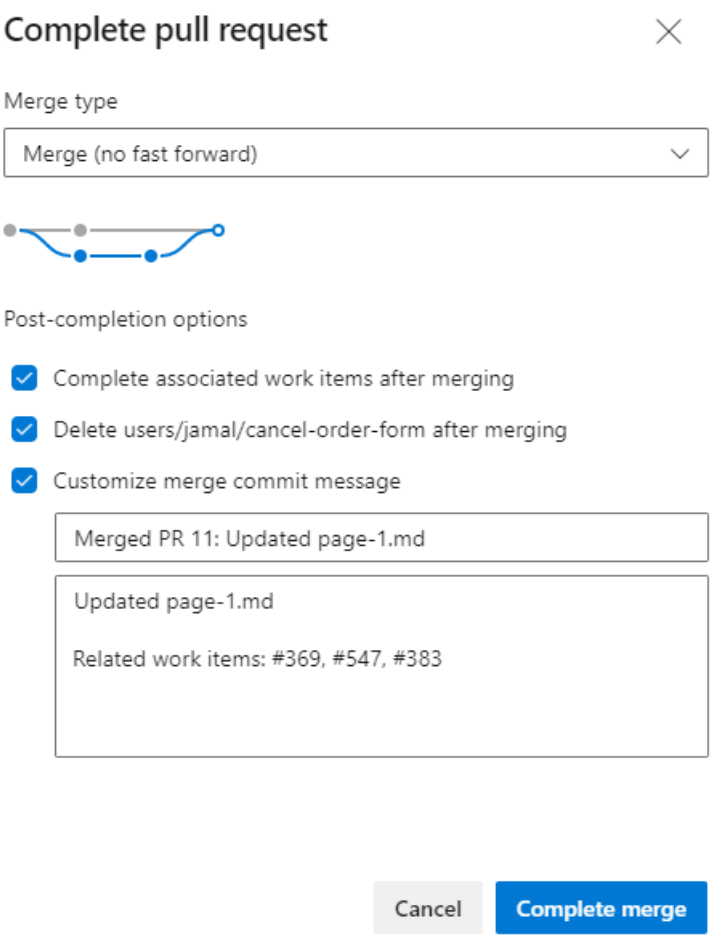
When you link a work item to a pull request (PR), you can automatically complete those work items when you complete the PR. Or, you can specify the workflow state to transition the work item to upon merging the PR.

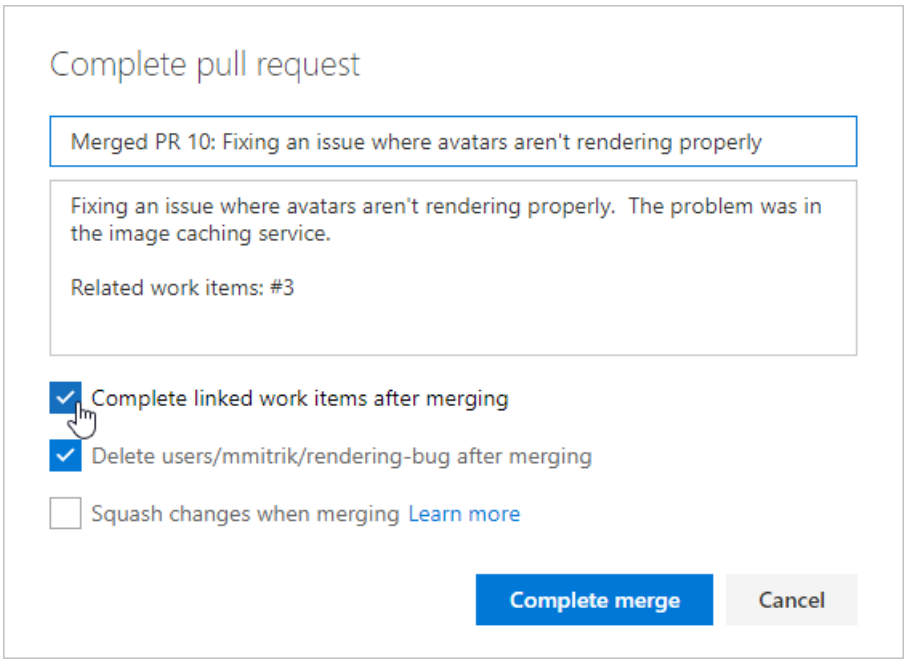
When you link a work item to a pull request (PR), you can automatically complete those work items when you complete the PR.

To learn more about pull requests, see Create, view, and manage pull requests.

Autocomplete work items

As shown in the following image, check the box to **Complete linked work items after merging**. The system defaults to your selection for future PRs.





In the following circumstances, the system won't automatically update the work item state to Done, Closed, or the state that belongs to the Closed category for the work item type:

- The work item, whose work item type is managed with the Inheritance process model, is already in a State that belongs to the Resolved category. In this instance, the system won't update the State. For example, if a bug derived from the Agile process is in a Resolved state, the system won't transition it to Closed.
- The work item is already in a State that belongs to the Completed category. No further transition is required.
- The WIT associated with the work item contains one or more workflow field rules that prevent the work item being saved to a next state. For example, a rule requires that another field must be defined as part of closing the work item.
- For on-premises deployments and Azure Boards Hosted process model, you must modify the workflow to specify actions (**ACTION** element) to take place when transitioning the workflow. See Change the workflow for a work item type, Specify Actions.

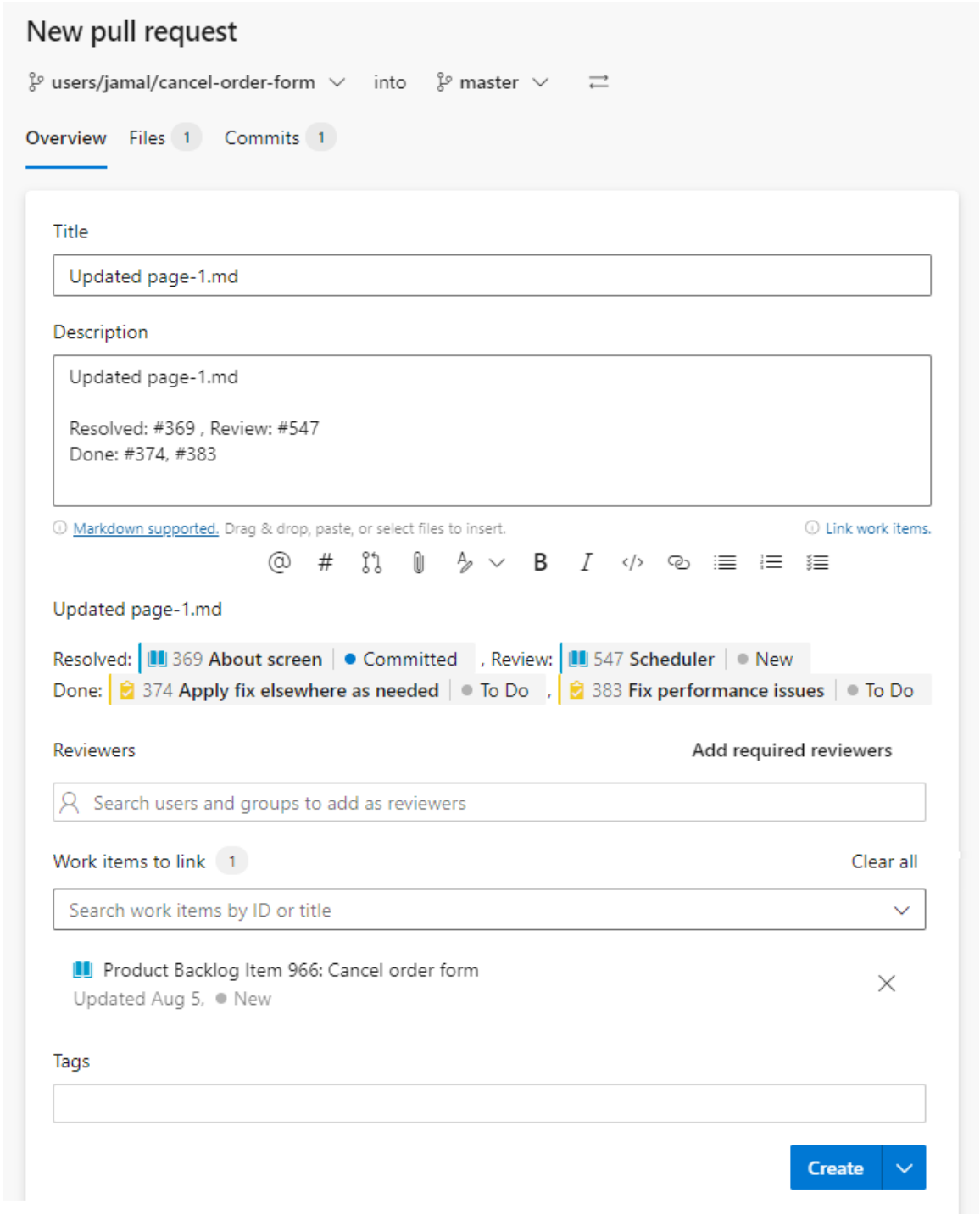
To learn more about process models, see Customize your work tracking experience.

Specify the workflow state of linked work items

To transition a work item to a specific workflow state, you can enter the information in the pull request Description. Prefix the **#ID** with a valid workflow state for the work item you mention.

> **NOTE**
> This feature requires Azure DevOps Server 2020.1 update or later version.

As shown in the following image, two user stories are transitioned, one to **Resolved** and the other to **Review**. Also, two tasks are set to **Done**.



Related articles

- Create, view, and manage pull requests
- Customize the workflow (Inheritance process)
- Customize your work tracking experience
- How workflow states and state categories are used in Backlogs and Boards

About requesting and providing feedback

10/4/2022 • 2 minutes to read • Edit Online

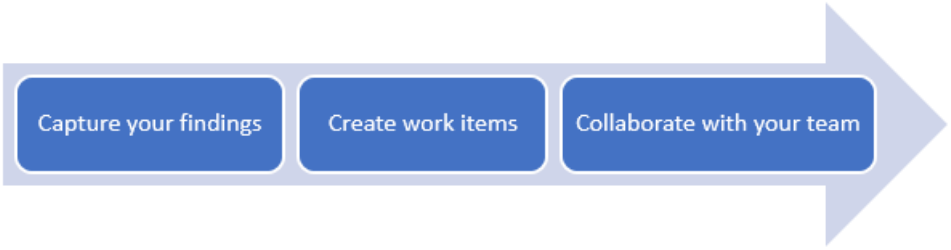
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can request feedback using one of two tools, through the Test & Feedback extension or through the Request feedback link you access from a dashboard.

> **NOTE**
> This article is about using Azure DevOps feedback tools. To provide feedback about Azure DevOps, see Provide product and content feedback.

About the Test & Feedback extension

Using the Test & Feedback extension is a simple, three step process:



- **Capture your findings** quickly and easily using the tools in the extension. Capture notes, screenshots with annotations, and screen recordings to describe your findings and highlight issues. Additionally, in the background the extension automatically captures rich data such as user actions as an image action log, page load data, and system information about the browser, operating system, memory, and more that can serve as a starting point for debugging.
- **Create work items** such as bugs, tasks, and test cases directly from the extension. The captured findings automatically become a part of the work item. Users can file a bug to report an issue with the product, or create a task that indicates a new work requirement. The extension can also be used to create test cases for scenarios discovered during exploration.
- **Collaborate with your team** by sharing your findings. Export your session report in Standalone mode, or connect to Azure DevOps or Team Foundation Server (2015 or later) for a fully integrated experience including exploring user stories and backlog items, simplified tracking and triaging of bugs and tasks, and managing feedback requests in one place.

As users perform exploratory testing, you can get insights from the sessions. View completed exploratory sessions and derive meaningful insights across all sessions. Get end-to-end traceability such as a breakdown of the work items created, the work items explored and not explored, session owners, and more.

To learn more, see the following articles:

- Install the Test & Feedback extension
- Request stakeholder feedback
- Provide stakeholder feedback

Microsoft Feedback client

The Visual Studio 2015 Microsoft Feedback client is a downloadable tool that you install on your desktop. It supports similar features for capturing findings to those provided by the Test & Feedback extension. It doesn't support creating work items. You can download the tool from Feedback Client for Microsoft Visual Studio Team Foundation Server 2015.

To learn more, see the following articles:

- Get feedback (Work tracking)
- Provide feedback with the Feedback client
- Set feedback permissions

Related articles

- Give reviewers permissions to provide feedback
- Default permissions and access set for collaboration tools
- Give us feedback, get support

Request stakeholder feedback using the Test & Feedback extension

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Stakeholders can respond to feedback requests for user stories and features generated in Azure DevOps using a lightweight end-to-end flow based on the Test & Feedback extension. Only users with Basic access can request feedback. Basic users can provide feedback using the flow described in this topic.

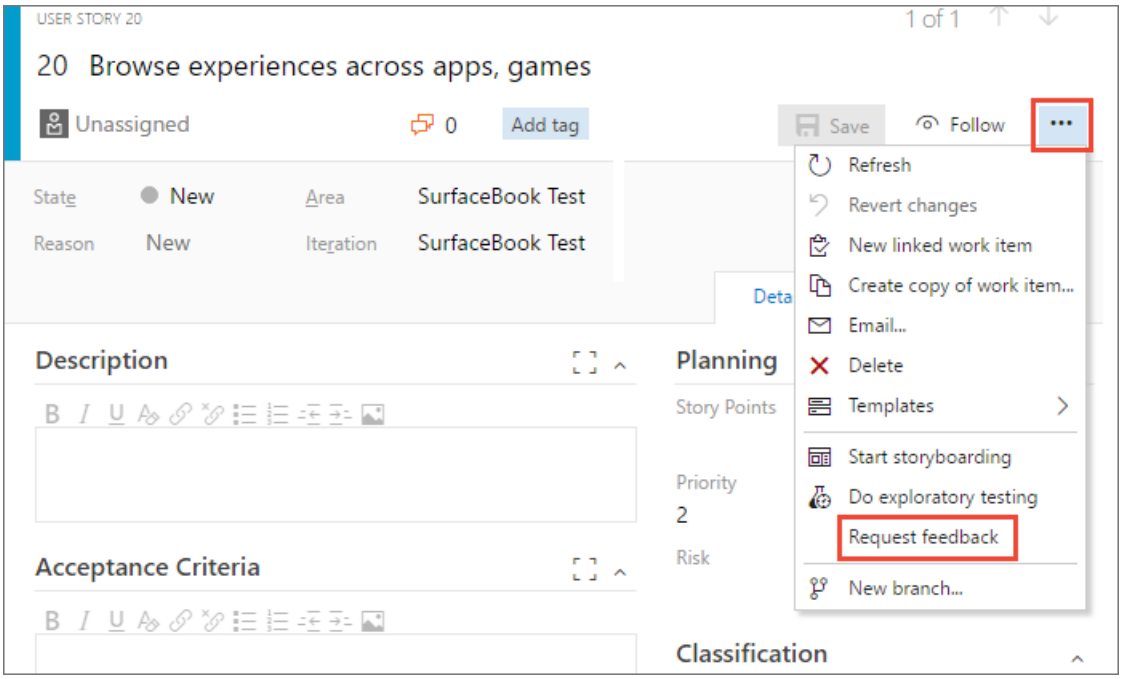
- Request feedback
- Provide feedback
- Voluntary feedback
- Track requests

> **NOTE**
> This lightweight end-to-end flow is applicable only for web apps and by using Azure DevOps. To get feedback for desktop apps, or for earlier versions of TFS, use the feedback flow described in Get feedback about the Microsoft Feedback Client.

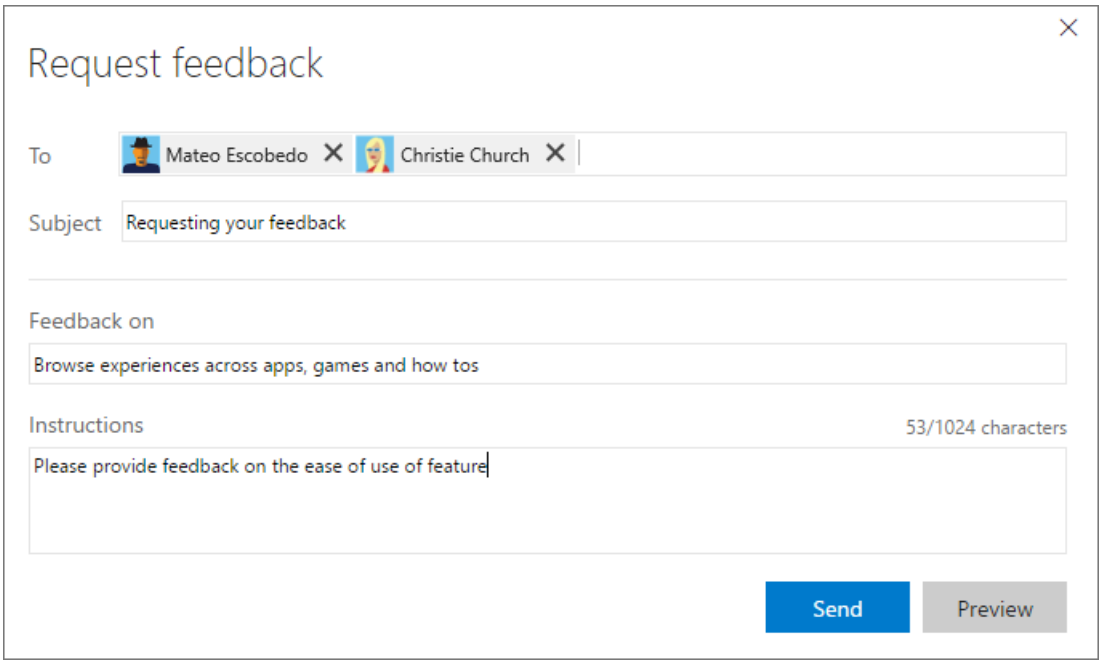
Request feedback from stakeholders

Request feedback from stakeholders directly from an Azure DevOps work item.

1. Open the work item form for the user story or feature for which you want to request feedback.
2. Open the shortcut menu from the ellipses (...) and choose **Request feedback**.



3. Type or select the names of the stakeholder(s) you want to send the request to, and optionally add any instructions or notes that will help them provide meaningful feedback.



4. Choose **Send** to generate emails to all the selected stakeholders.

> **NOTE**
>
> Teams can request feedback from other team members, such as users having Basic access. Just add their names in the feedback request form so that a **Request feedback** email is sent to them. Also see Can users with Basic access respond to feedback requests?.

Related articles

- Provide stakeholder feedback using the Test & Feedback extension
- Voluntary stakeholder feedback using the Test & Feedback extension
- Track stakeholder feedback using the Test & Feedback extension
- Exploratory test and submit feedback directly from your browser
- Overview of manual and exploratory testing

Provide feedback using the Test & Feedback extension

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Stakeholders and other users can respond to feedback requests using the Test & Feedback extension in two ways:

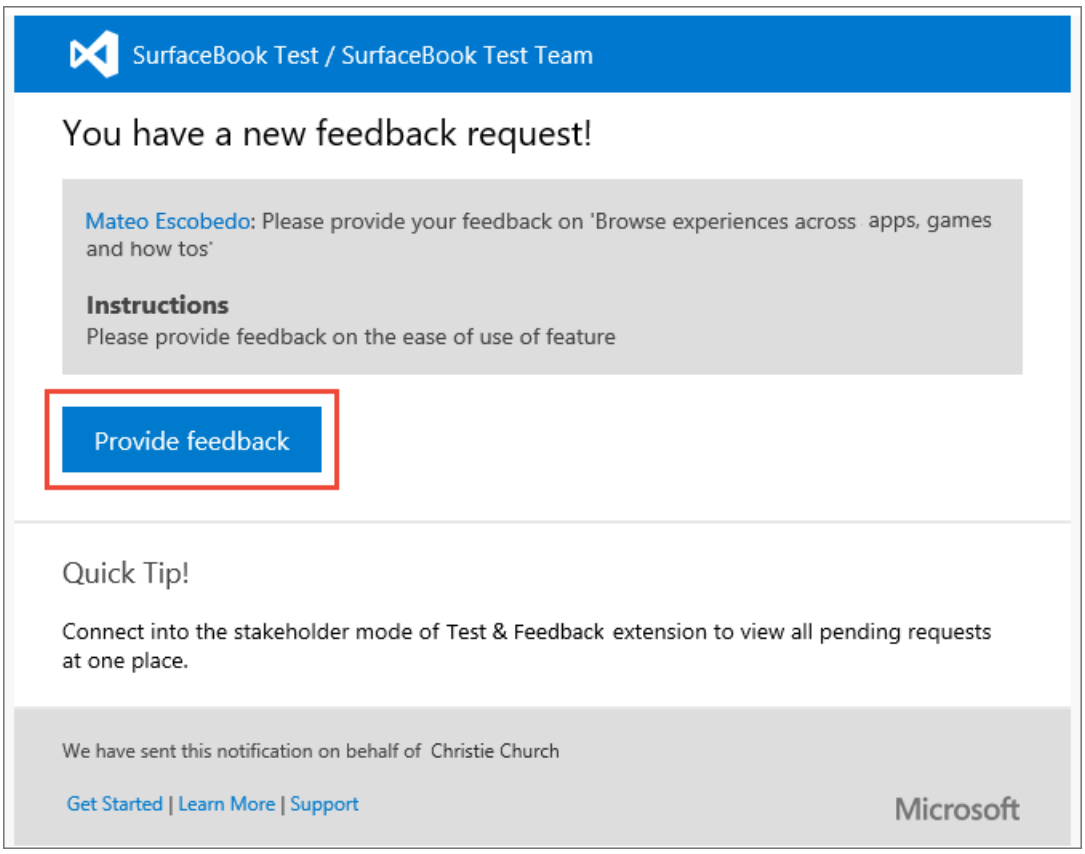
- From the link in a feedback request email
- Directly from the Test & Feedback extension

Before you start, ensure you have installed the Test & Feedback extension. This is required in order to respond to feedback requests.

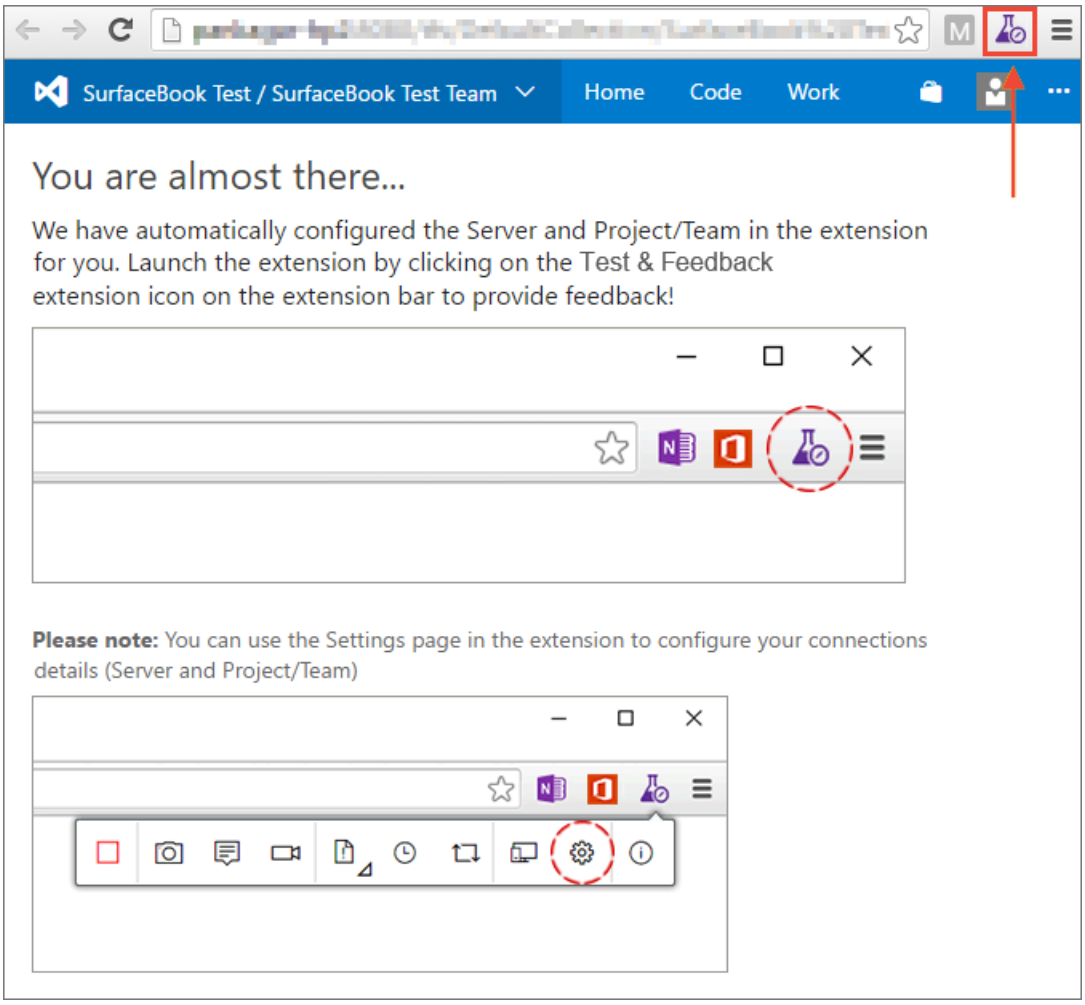
> **NOTE**
> Any user with a Stakeholder access can use the Test & Feedback extension in **Stakeholder** mode. This mode is designed to allow the widest possible range of users to assist test teams by providing feedback.

Provide feedback directly from a feedback request email

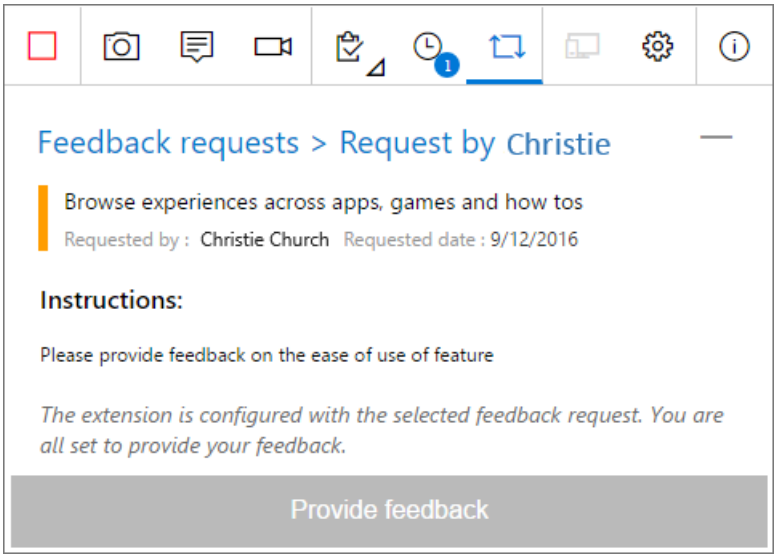
1. Open the feedback request email and choose the **Provide feedback** link.



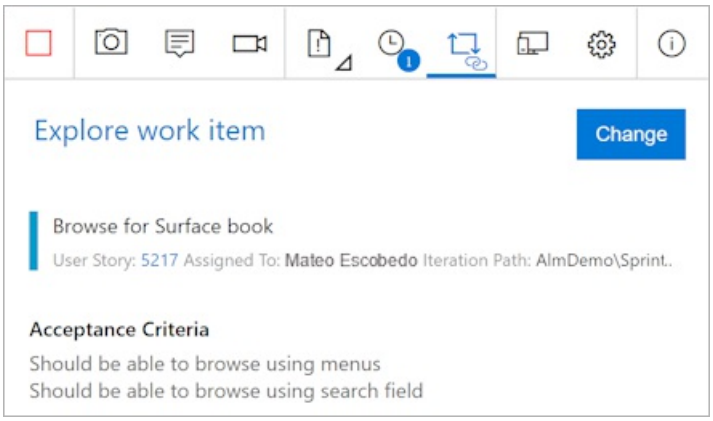
2. The Azure DevOps landing page opens to confirm that the extension has been automatically configured with the feedback request. Choose the icon in the toolbar to launch the extension.



If you are a **Stakeholder**, you see the **Feedback requests** page. Read the instructions (if any) in the feedback form to understand how to give the feedback and what the requestor requires.

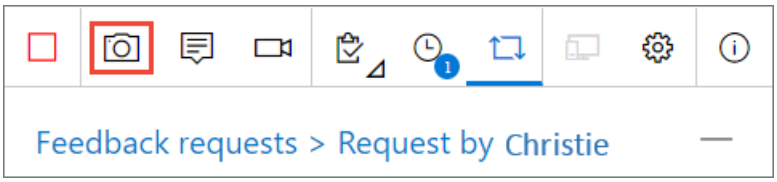


If you are a **Basic** user, you see the **Explore work item** traceability page showing details of the user story on which feedback was requested, and the user acceptance criteria (if any).



3. Read any instructions in the email and this page to understand how to give the feedback, and on what feature.

4. Open the application you need to provide feedback on and begin your feedback. For example, choose **Capture screenshot** to take a screenshot.

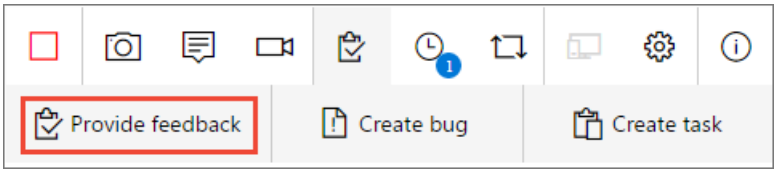


You can use all the capabilities of the extension such as capturing screenshots, notes, and screen recordings. For more details, see this topic.

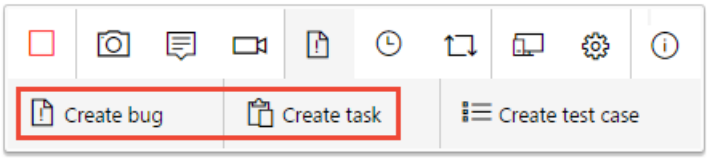
> Some browsers may not provide all the capture capabilities. See Supported web browsers for the extension.

5. When you are done capturing feedback:

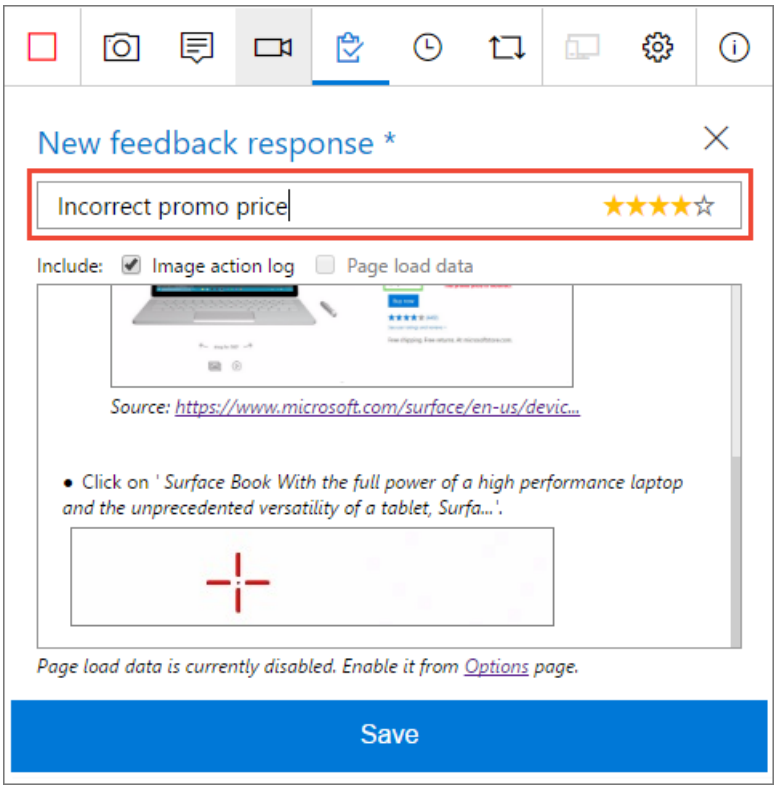
   - If you are a **Stakeholder**, choose **Provide feedback**. You can optionally choose to create bugs and tasks when you submit your feedback. The process is the same as described in this topic.



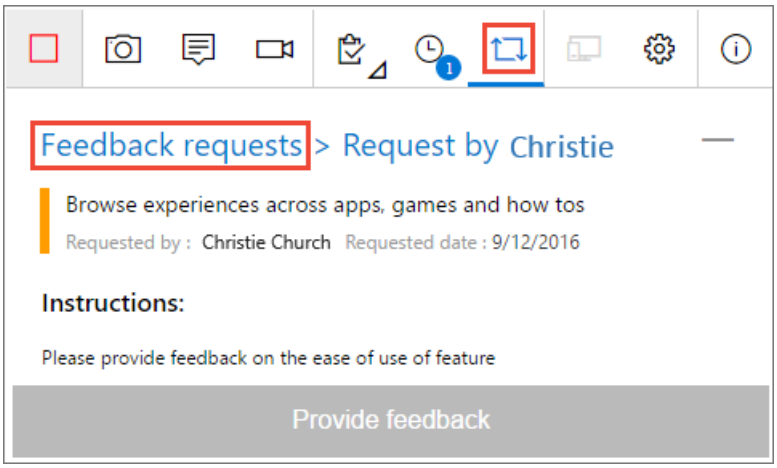
   - If you are a **Basic** user, create a bug or a task.



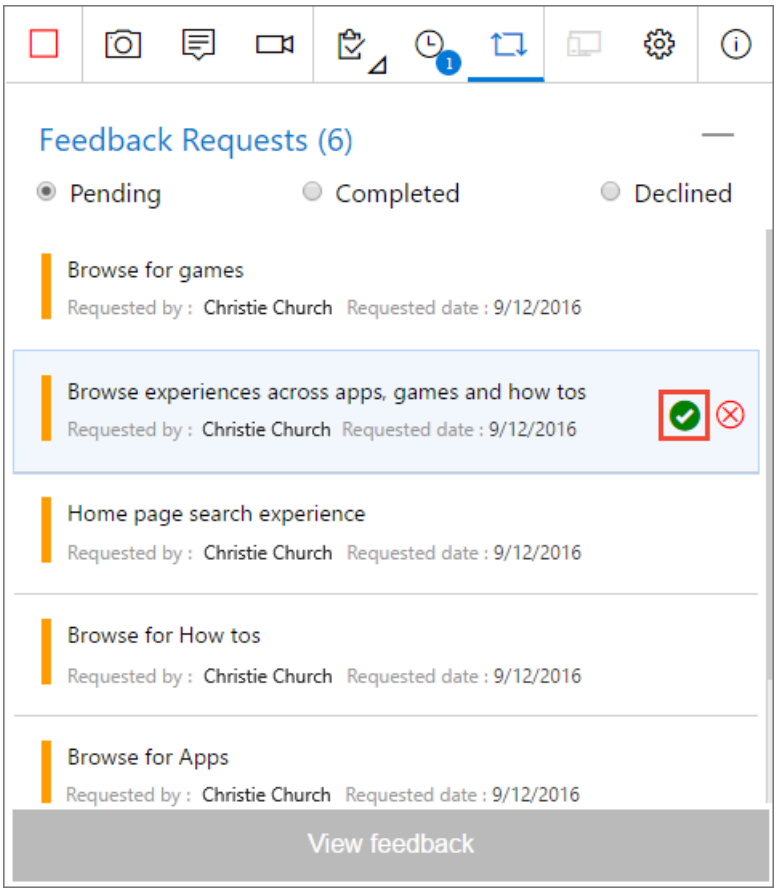
6. All your feedback captured is shown in the response form, bug, or task. Type a suitable title and, optionally, select a star rating for the feature you've been testing.



7. Save your feedback. This create a work item in Azure DevOps containing all your feedback.
8. Continue to capture more feedback if required. You can submit multiple feedback responses, bugs, and tasks for the same feedback request.
9. If you are a **Stakeholder**:
   - When you are done providing feedback, go to the **Feedback requests** page and choose **Feedback requests**.



   - In the pending feedback requests page, mark the feedback request as **Completed**.

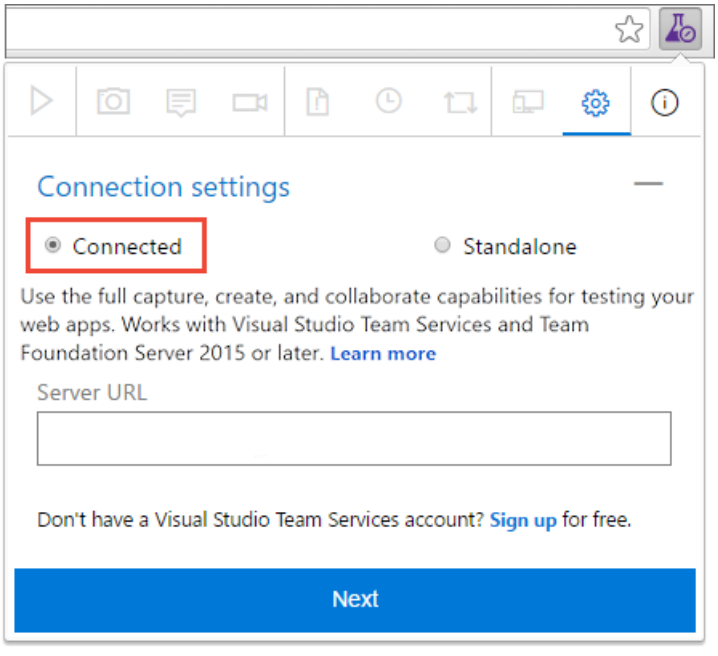


10. Choose the **Stop** icon to end your feedback session.

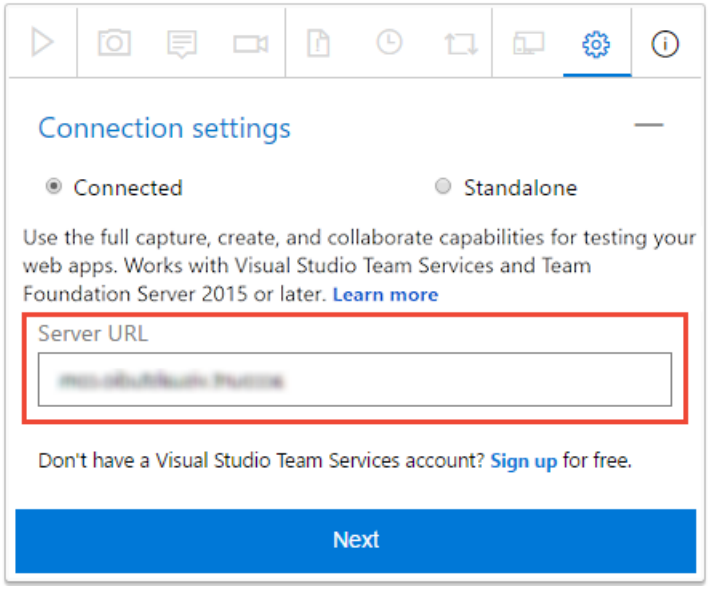


Provide feedback directly from the Test & Feedback extension

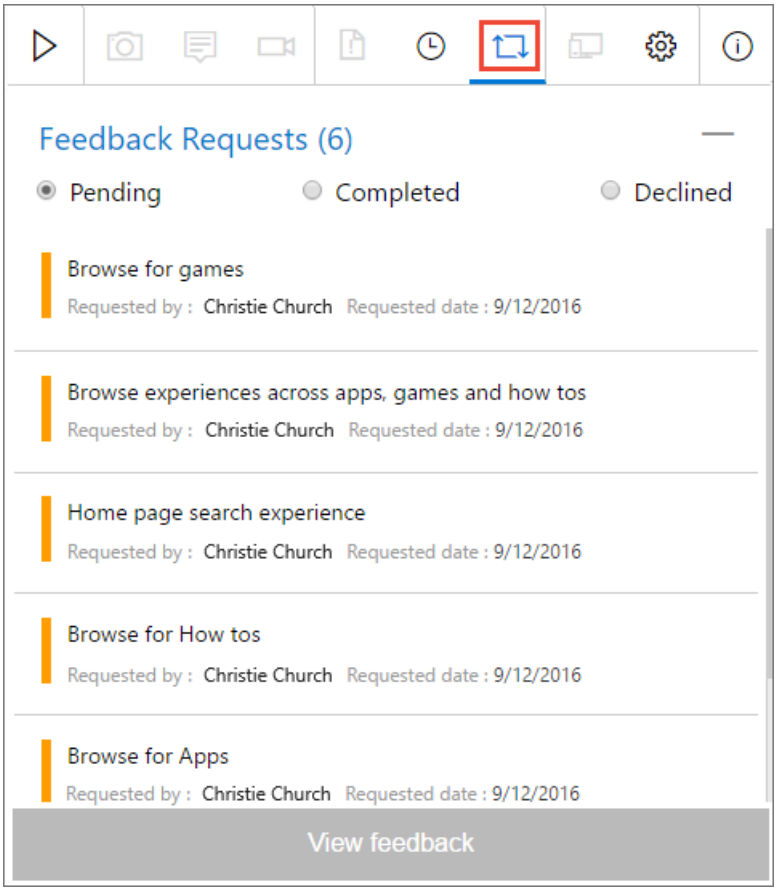
1. Open the Test & Feedback extension in your browser using the icon in the toolbar.
2. In the **Connection settings** page, choose **Connected** mode.



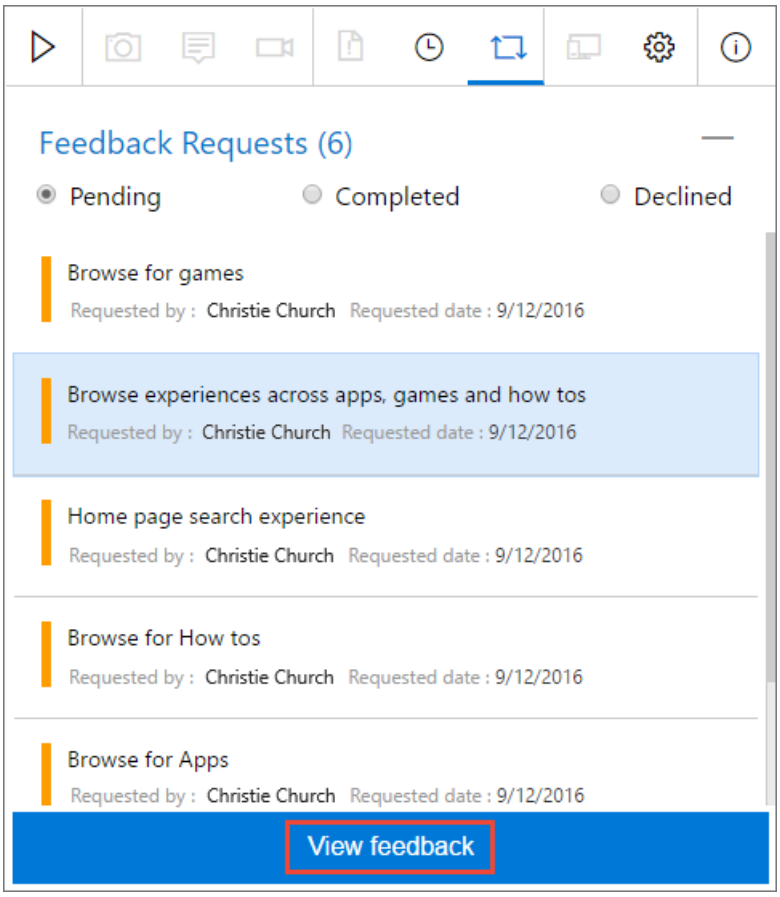
3. Connect to the server and the project or team that is requesting feedback.



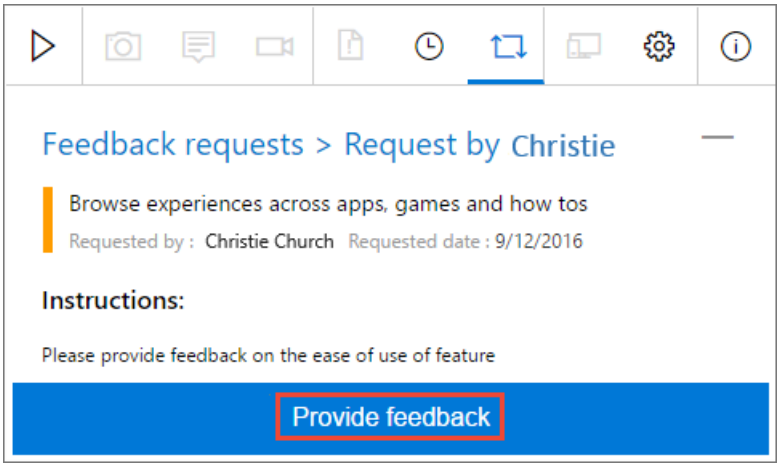
4. Open the **Feedback requests** page to see all your feedback requests from the project or team you connected to.



5. Select the feedback request you want to respond to and choose **View feedback**.



6. Read the instructions in the feedback request details page, then choose **Provide feedback**.



7. Capture and submit your feedback as shown above.

Related articles

- Request stakeholder feedback using the Test & Feedback extension
- Voluntary stakeholder feedback using the Test & Feedback extension
- Track stakeholder feedback using the Test & Feedback extension
- Exploratory test and submit feedback directly from your browser
- Overview of manual and exploratory testing

Get insights across your exploratory testing sessions

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

View completed exploratory testing sessions and derive meaningful insights at team or individual level, and for a specific period.

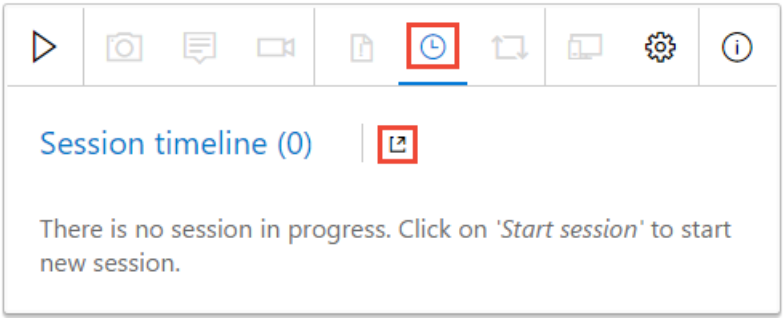
Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or run manual or automated tests, you must have **Basic** access or higher.

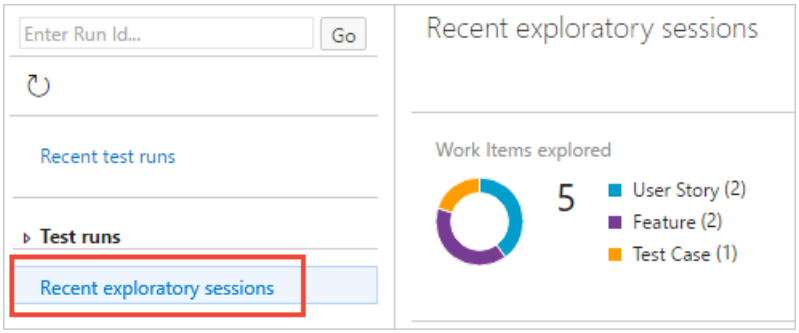
To learn more, see Manual test access and permissions.

1. Open the **Recent exploratory sessions** page. You can do this:

   - From the Test & Feedback extension by choosing the "view" icon on the **Session timeline** page.

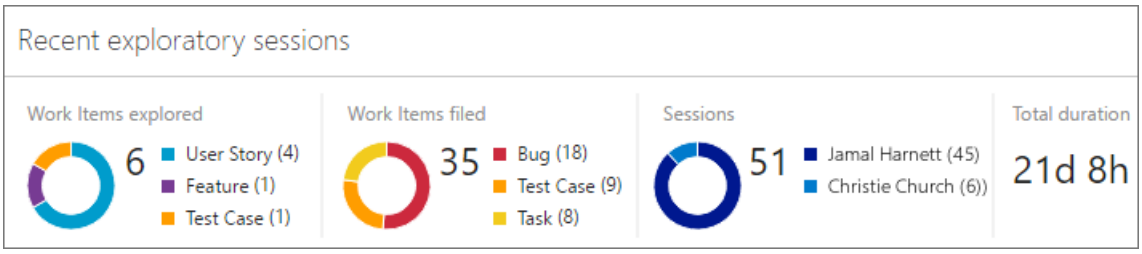


   - From the **Test Plans** web portal. by opening the **Runs** page and choosing **Recent exploratory sessions**.

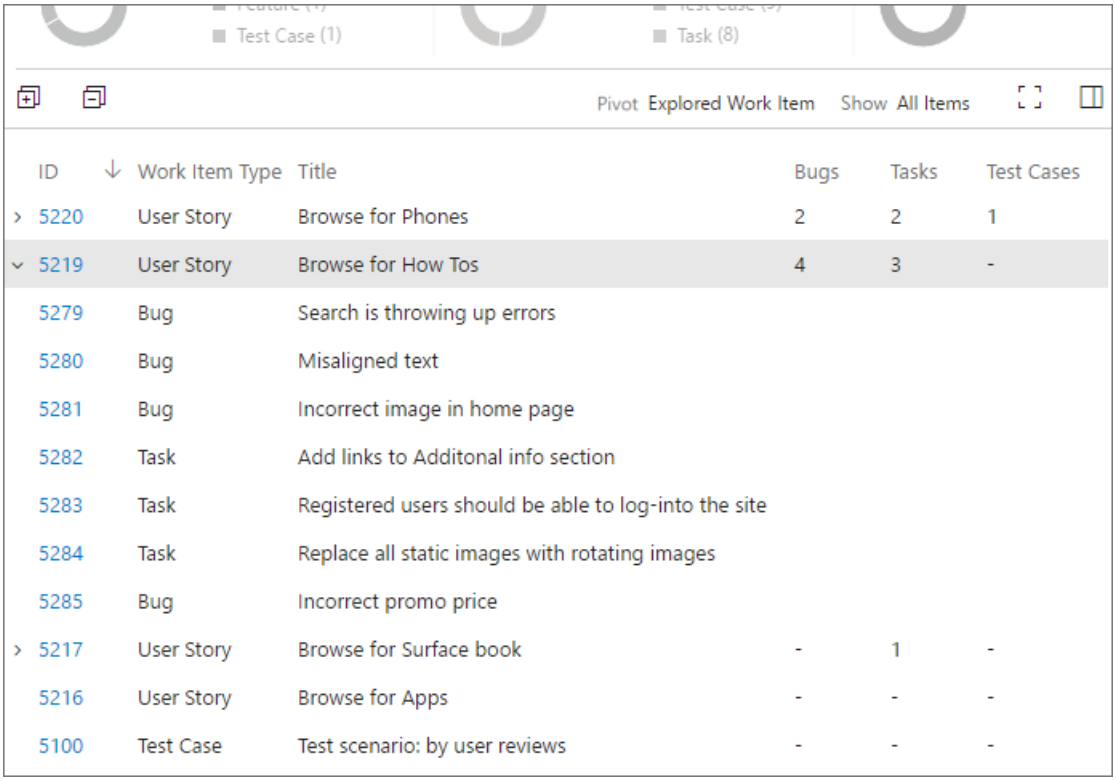


2. Explore the **Recent exploratory sessions** page. It contains three main sections:

   Summary view - shows a graphical breakdown of the work items explored, work items created, session owners, and the total time for these sessions.

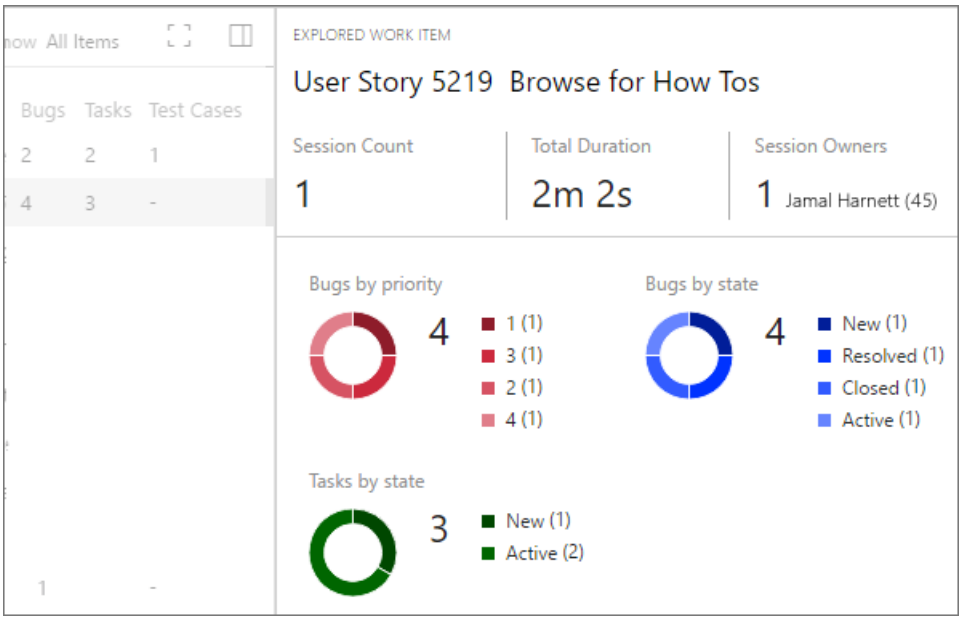


Pivot view - shows a collapsible nested and sortable list of items grouped in different ways.



ID	Work Item Type	Title	Bugs	Tasks	Test Cases
5220	User Story	Browse for Phones	2	2	1
5219	User Story	Browse for How Tos	4	3	-
5279	Bug	Search is throwing up errors			
5280	Bug	Misaligned text			
5281	Bug	Incorrect image in home page			
5282	Task	Add links to Additonal info section			
5283	Task	Registered users should be able to log-into the site			
5284	Task	Replace all static images with rotating images			
5285	Bug	Incorrect promo price			
5217	User Story	Browse for Surface book	-	1	-
5216	User Story	Browse for Apps	-	-	-
5100	Test Case	Test scenario: by user reviews	-	-	-

Details view - shows the work item selected in the Pivot view or a summary of information about a selection of items.

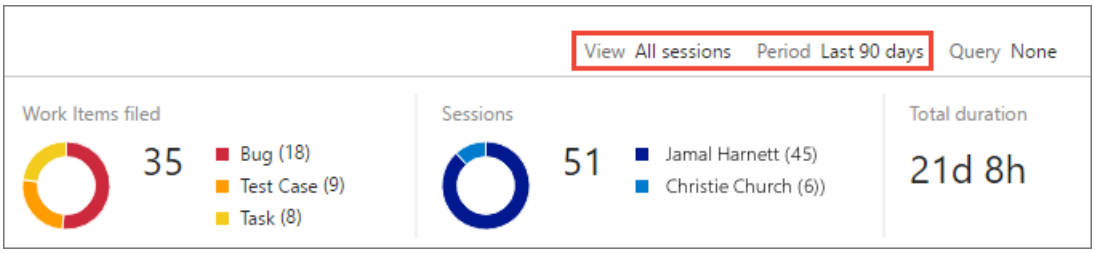


Get insights from your exploratory testing sessions:

Use the **Recent exploratory sessions** page to get insights about your app from the information collected during your exploratory testing sessions.

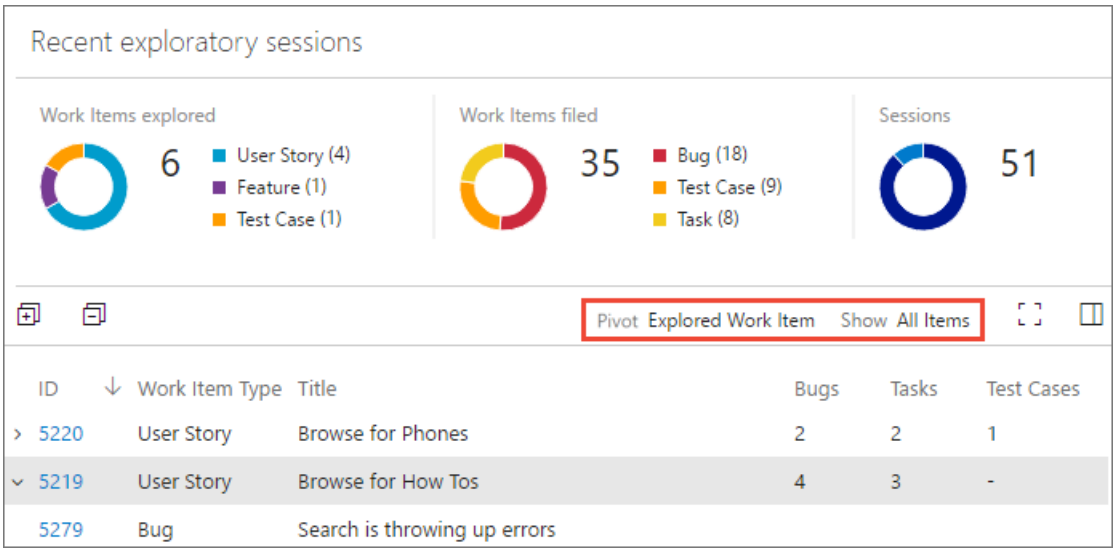
1. **Set the scope for the data**. Summary view provides the highest level view of your test results. Use it to get insights into the overall effort and results of the exploratory testing sessions.

   - Use the **View** filter to scope the summary view to all sessions or just your own sessions.
   - Use the **Period** filter to scope the summary view to sessions in the period from the last 7 to the last 90 days.



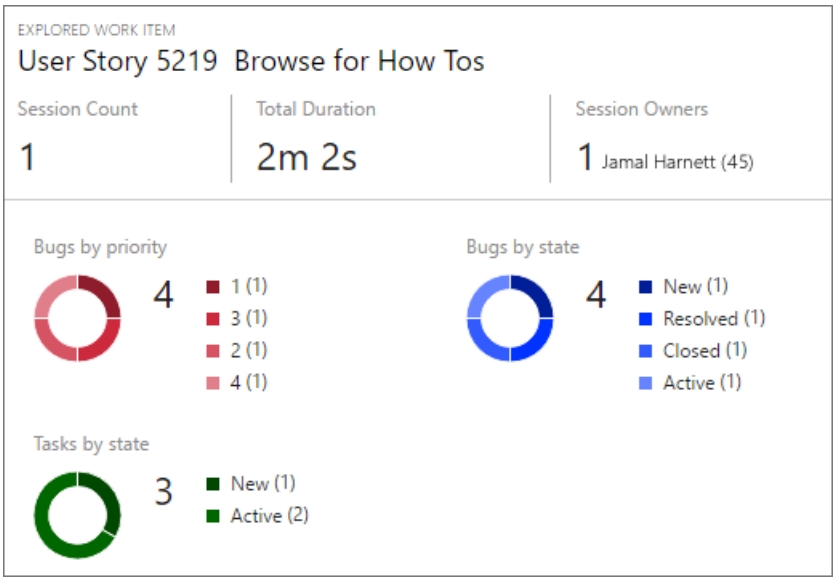
2. **Pivot the data on the type of work item**. Pivot view lets you to focus on all the work items you created in your exploratory testing sessions, or just on bugs, tasks, or test cases; and group the results in different ways.

   - Use the **Pivot** filter to group the work items as a nested list based on those that have been explored, those that have not been explored (requires a query), by the session in which they were created, or by session owner.
   - Use the **Show** filter to show all items; or just bugs, tasks, or test cases.



3. **Get deep insights from Details view**. Details view gives insights into the items selected in Pivot view. Depending on the type of item you select, you see the work item as an editable form, or a series of charts.

   - Select a row in Pivot view to see a summary of all the related information in Details view. For example, if you have pivoted the list based on sessions, select a session to see a summary of all the information from the work items in just that session.
   - Select a child row in Pivot view to display the work item form for that individual item. For example, if you have pivoted the list based on explored work items, expand a work item and select a child bug, task, or test case to see the work item form for just that item.



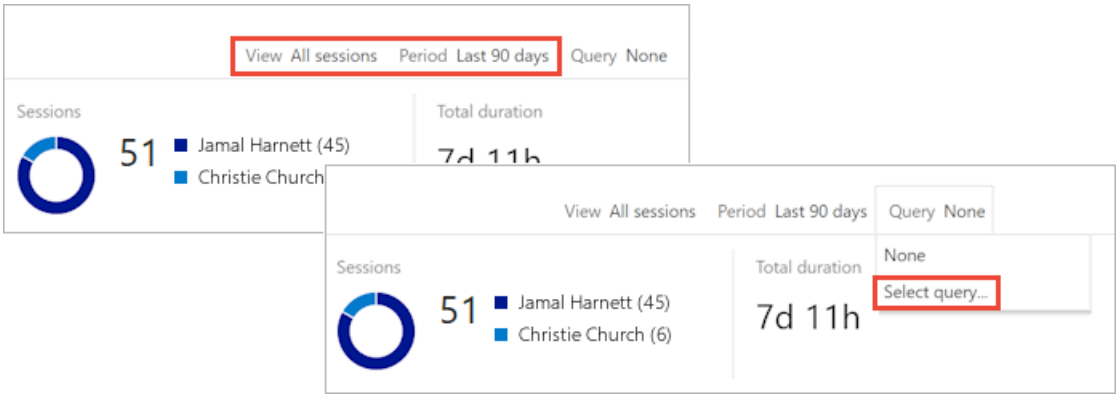
Discover work items not yet explored

Use a query to explore the work items that users have not yet explored.

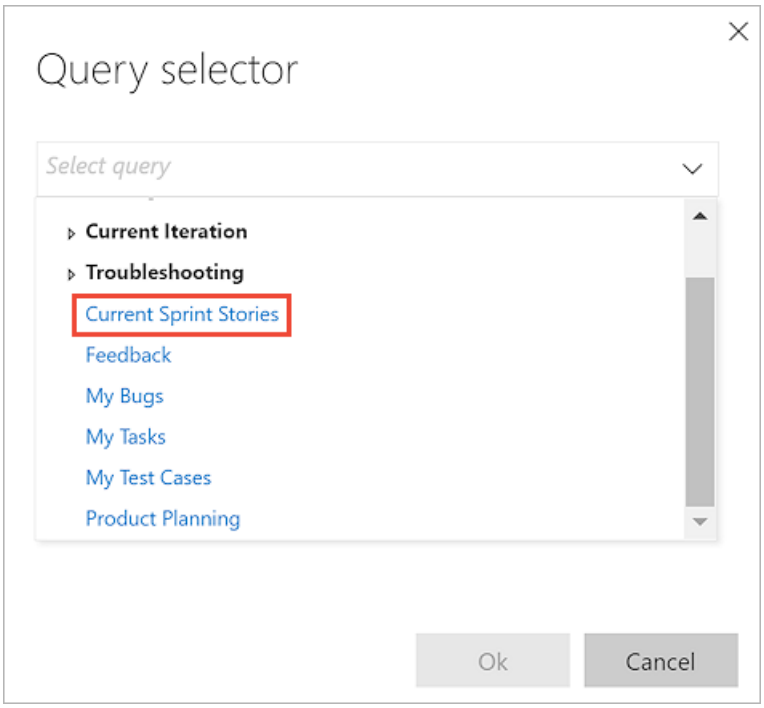
1. Create a shared query in Azure DevOps that selects work items that can be explored using the Test & Feedback extension, such as work items in the epic category, feature category, requirement category, requirement-based suites, or test cases.

   > You must use a **shared** query. If this query returns a mix of supported and unsupported work items, only those in supported categories will be displayed.

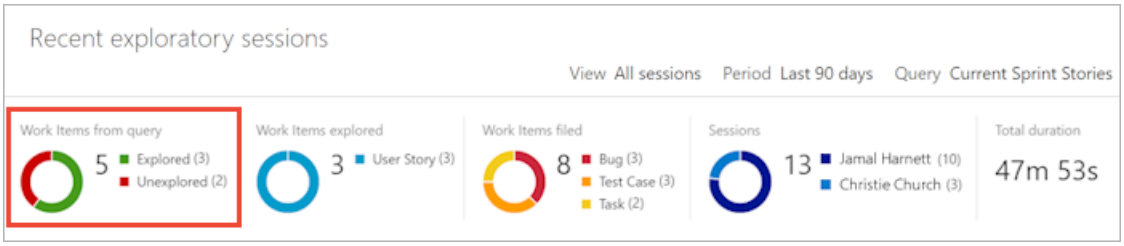
2. Use the **View** and **Period** filters to scope the view to the type of session (all sessions or just your own sessions) and the time span (from the last 7 to the last 90 days). Then open the **Query** list and choose **Select query**.



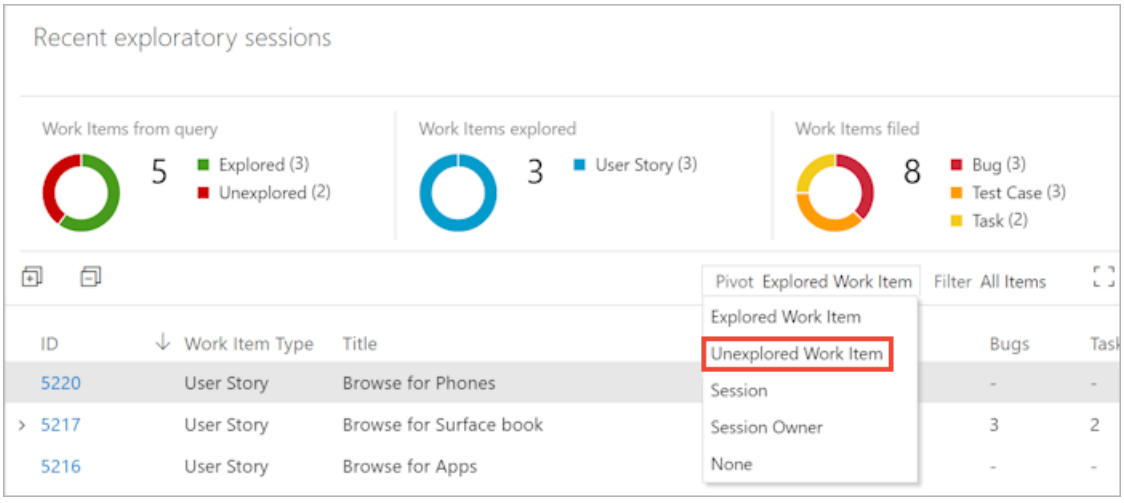
3. In the **Query selector** dialog, choose the shared query you created earlier.



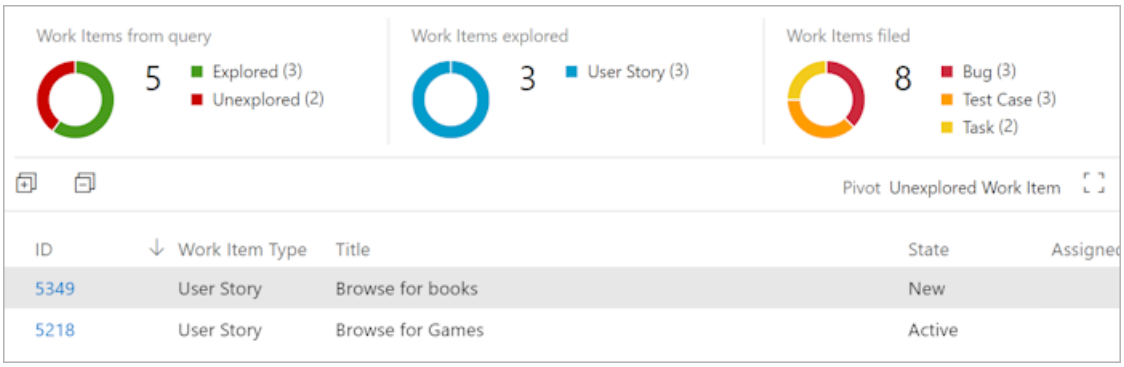
4. View all the work items returned by the query in Summary view. You see a breakdown of explored and unexplored work items, work items filed, sessions, and total session duration.



5. Open the **Pivot** list and choose **Unexplored Work Item**.



The view now shows only the unexplored work items.



See Also

- Use the Test & Feedback extension in Connected mode
- Add findings to existing bugs with exploratory testing
- Explore work items with exploratory testing
- Use the Test & Feedback extension in Standalone mode
- Exploratory testing with Microsoft Test Manager
- Overview of manual and exploratory testing

Install the Test & Feedback extension

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The **Test & Feedback extension** helps teams perform exploratory testing and provide feedback. Everyone in the team, such as developers, product owners, managers, UX or UI engineers, marketing teams, early adopters, and other stakeholders can use the extension to submit bugs or provide feedback and contribute to the quality of your product.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To view or run manual or automated tests, you must have **Basic** access or higher.

To learn more, see Manual test access and permissions.

Supported web browsers for the extension

The Test & Feedback extension is currently available for Google Chrome, Microsoft Edge (Chromium Only), and Mozilla Firefox version 50.0 and higher.

Some browser versions do not currently support all the features of the Test & Feedback extension.

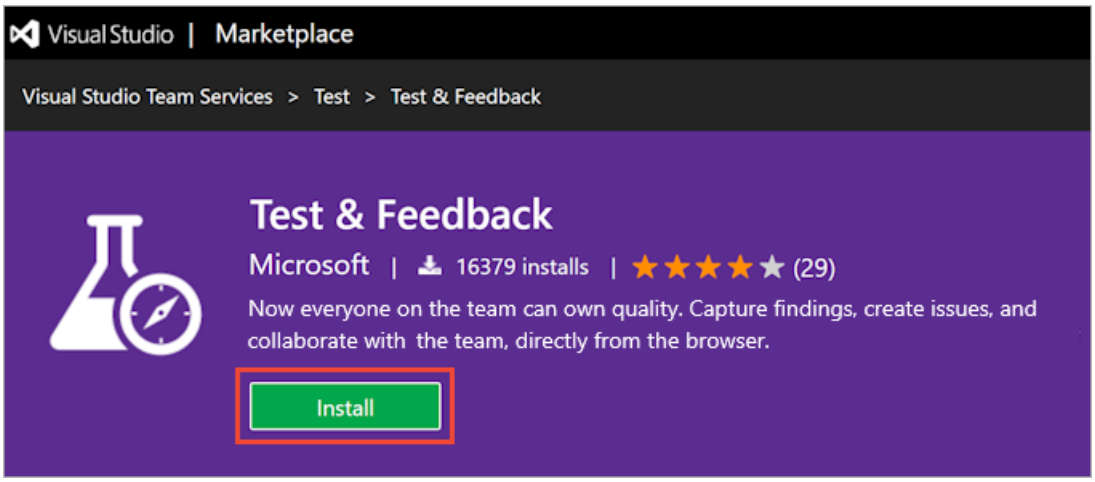
FEATURE	CHROME	MICROSOFT EDGE	FIREFOX
Capture screenshots with inline annotations	Yes	Yes	Yes
Capture notes	Yes	Yes	Yes
Capture screen recordings	Yes	Yes	No
Capture page load data	Yes	Yes	No
Capture user actions log	Yes	Yes	Yes
Capture system information	Yes	Yes	No
Create bugs	Yes	Yes	Yes
Create tasks and test cases	Yes	Yes	Yes
Create feedback requests	Yes	Yes	Yes
Export session report for sharing	Yes	Yes	Yes

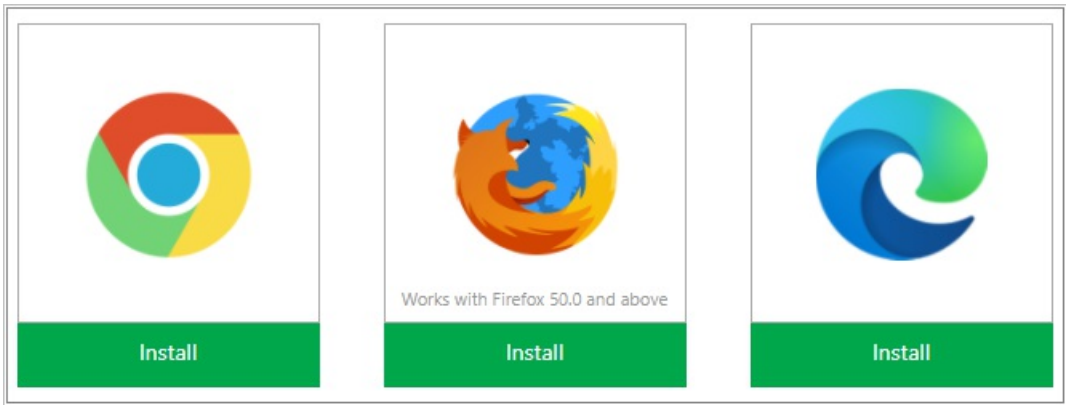
FEATURE	CHROME	MICROSOFT EDGE	FIREFOX
End-to-end traceability for work items	Yes	Yes	Yes
Simplified bug and task tracking and triaging	Yes	Yes	Yes
View and get insights from sessions	Yes	Yes	Yes
View similar existing bugs	Yes	Yes	Yes
Test app on devices using cloud providers such as Perfecto	Yes	Yes	No
Manage feedback requests	Yes	Yes	Yes

For more information, see Visual Studio Marketplace, Azure DevOps tab.

Install the extension

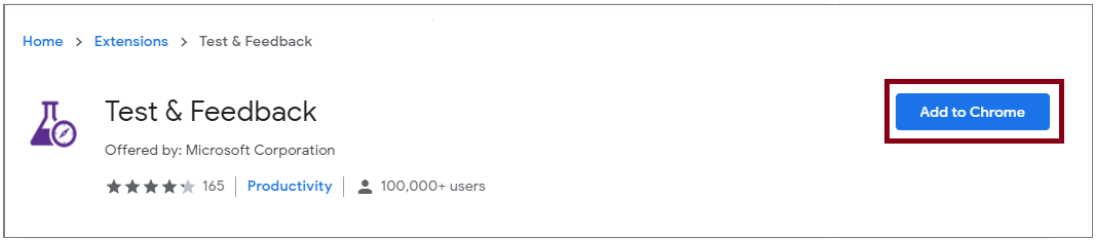
1. Check the list of supported browsers and decide which you want to use.
2. Download and install your chosen browser, if you haven't already, then open it.
3. Go to Visual Studio Marketplace > Test & Feedback and choose **Install**.

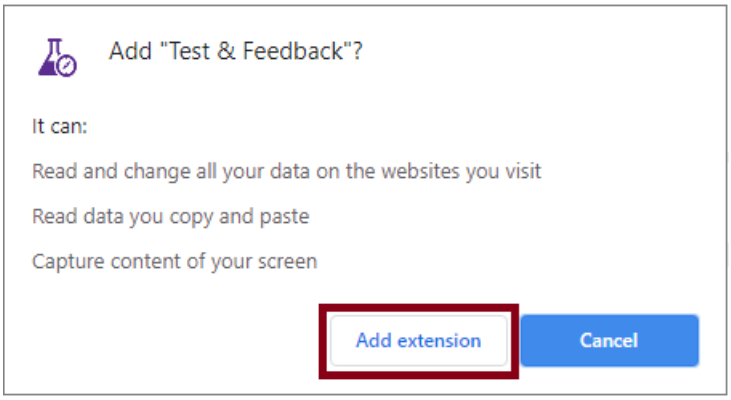




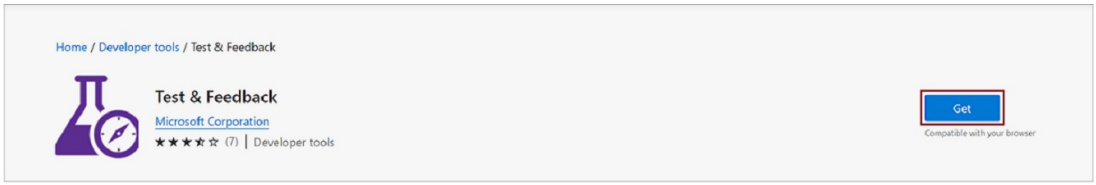
4. Follow the instructions shown to install the Test & Feedback extension in your browser:

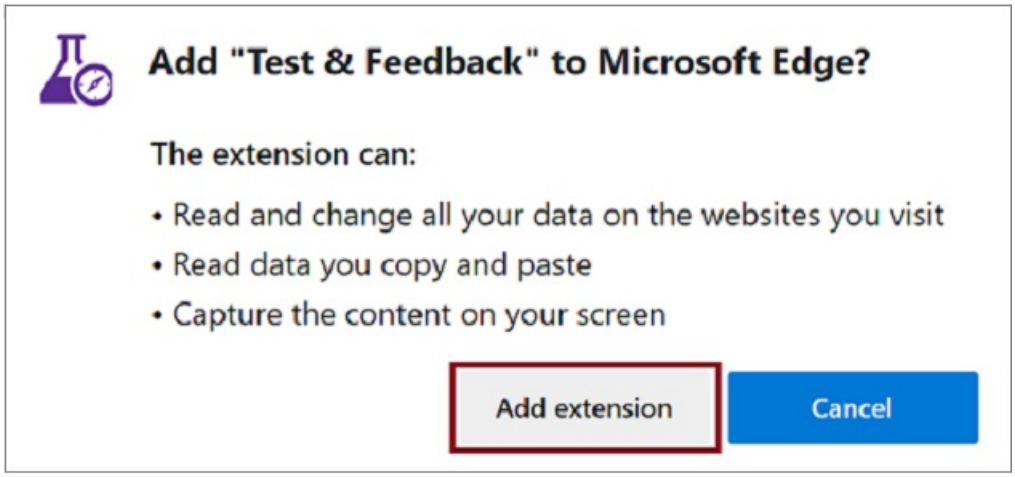
- If you are using Google Chrome, choose the **Install** link for Chrome from the above image to open the Google Chrome web store and follow the instructions to install the extension.



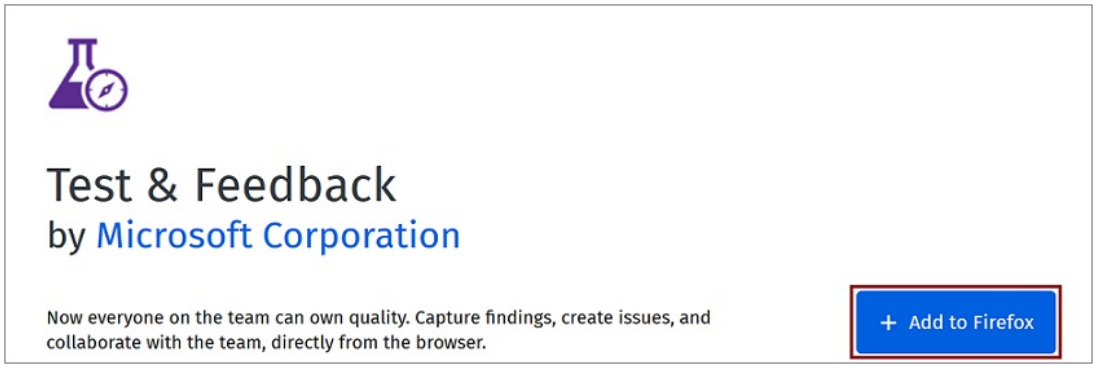


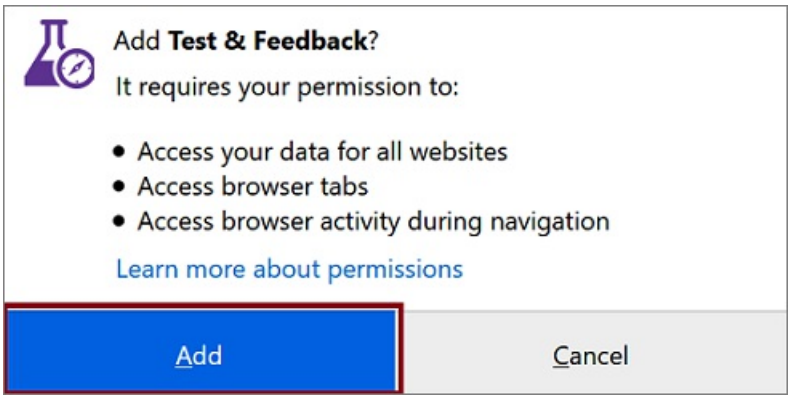
- If you are using Microsoft Edge (Chromium), choose the **Install** link for Edge from the above image to open the Microsoft Edge Add-ons page and follow the instructions to install the extension.





- If you are using Mozilla Firefox 50.0 and higher, choose the **Install** link for Firefox from the above image to open the Firefox Browser Add-ons page and follow the instructions to install the extension.

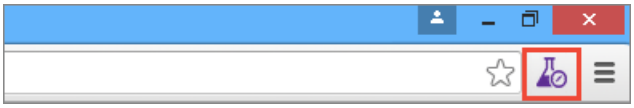




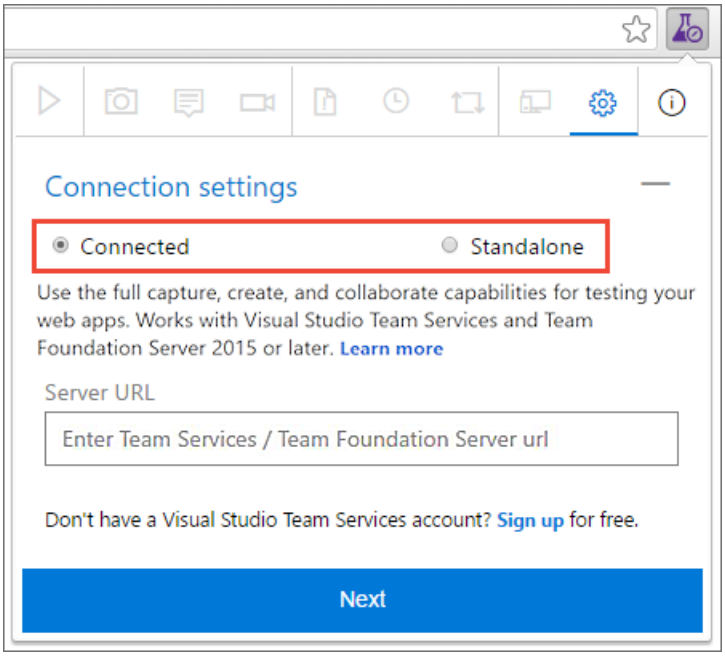
You need to install the extension or add-on only once. Afterwards your browser will update it automatically.

Select an exploratory testing mode

1. Open the extension you installed in your browser by choosing the icon.



2. Decide if you want to use the extension in Connected or Standalone mode.



Connected mode

Available to all users of Azure DevOps and TFS 2015 or later:

- Users with **Basic** access or higher: Full capture and create capabilities to submit bugs, tasks, and test cases. Includes collaboration capabilities such as end-to-end traceability, rich insights across completed exploratory sessions, simplified tracking and triaging for bugs and tasks, and more.
- Users with **Stakeholder** access: Full capture and create capabilities, except for test cases, to submit feedback and respond to feedback requests from the team.
- Feedback experience is available only in Azure DevOps and TFS 2017 or later.

Standalone mode

Available to everyone. No connection to Azure DevOps is required. Take notes and screenshots with inline annotations to capture issues. Create bugs and export a session report to share findings.

If you have problems connecting to Azure DevOps, you may find the topic TF31002: Unable to connect useful.

Related articles

- FAQs for manual testing

Next step

Use the Test & Feedback extension in Connected mode

Exploratory testing with the Test & Feedback extension in Connected mode

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To use the Test & Feedback extension in **Connected** mode you must connect to an Azure DevOps project. This automatically configures the extension based on your access level:

- Users with **Basic** access can use the extension to perform exploratory testing, as described below.
- Users with **Stakeholder** access can use the extension to respond to feedback requests or to provide feedback voluntarily. More details.
- Users with **Basic** or **Stakeholder** access can use extension to respond to feedback requests sent by the team by choosing the **Provide feedback** link in the email. More details.

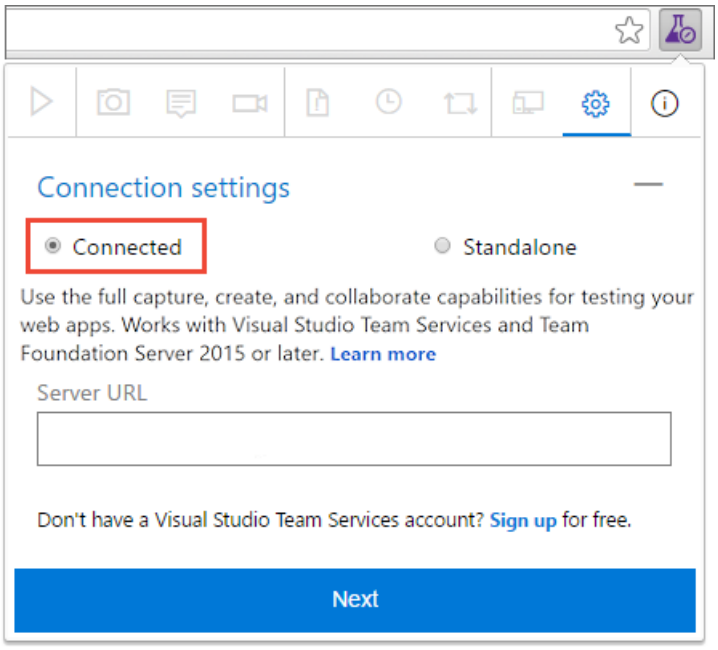
Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To request or provide feedback, you must have **Stakeholder** access or higher.
- To add or modify bugs or other work item types, you must have the **Edit work items in this node** permission set to **Allow** under the corresponding **Area Path**.
- To add new tags, you must have the **Create tag definition** permission set to **Allow**.

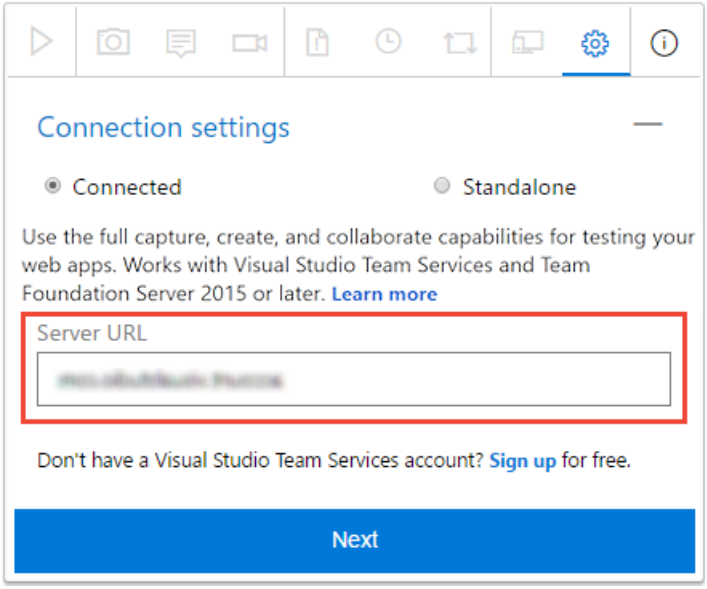
To learn more, see Set permissions and access for testing.

Connect to Azure DevOps

1. If you want to use Azure DevOps, and you haven't already done so, sign up for a subscription now. Make sure you create a project when you create your subscription.
2. If you haven't already, install the Test & Feedback extension.
3. Open the extension in your web browser and select **Connected** mode.

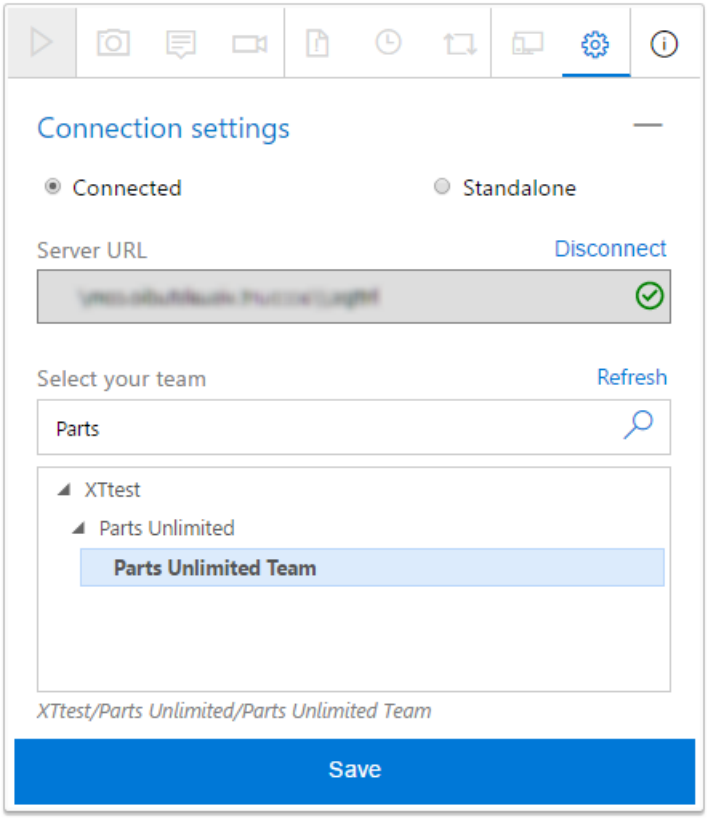


4. Enter the Azure DevOps URL you want to connect to and choose **Next**.



   If you are connecting for the first time, you may be prompted to sign in.

5. After connecting to the server, the extension shows all the collections, projects and teams in that server. Select the project or team you want to connect to and choose **Save**.



If there are many projects or teams, use the search textbox to find the one you need.

The extension is now ready to be used in **Connected** mode. Depending on your access level (Basic or Stakeholder) you will see the appropriate UI for either exploratory testing or providing feedback. The extension remembers your selection and remains connected until the session cookies expire or you explicitly disconnect from the server.

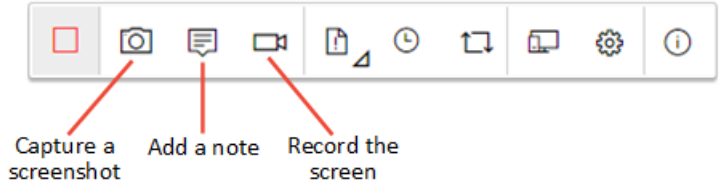
Create bugs or tasks

After you have connected, you are ready to begin testing your app.

1. Start your exploratory testing session.

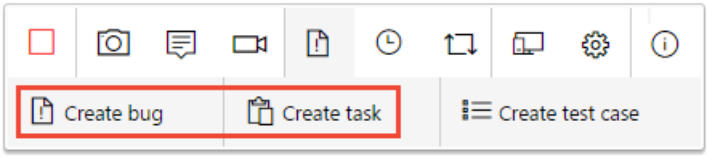


2. Open the web application you want to test, and start exploring it.
3. When you find an area that has a bug, take a screenshot of any part of the screen, make notes, or record your actions as a video.

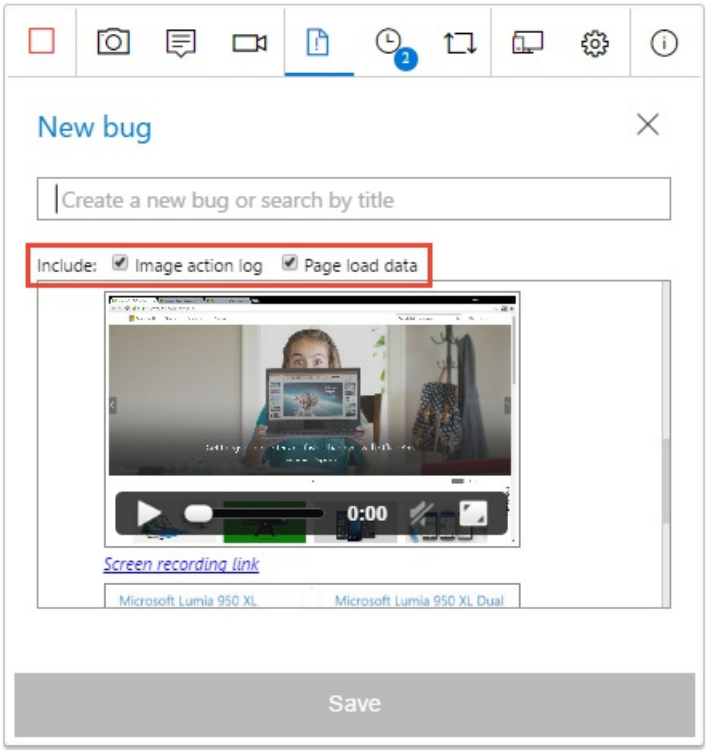


> Some browsers may not provide all of the capture capabilities. See Supported web browsers for the extension.

4. When you are done exploring and capturing information, create a bug or a task.

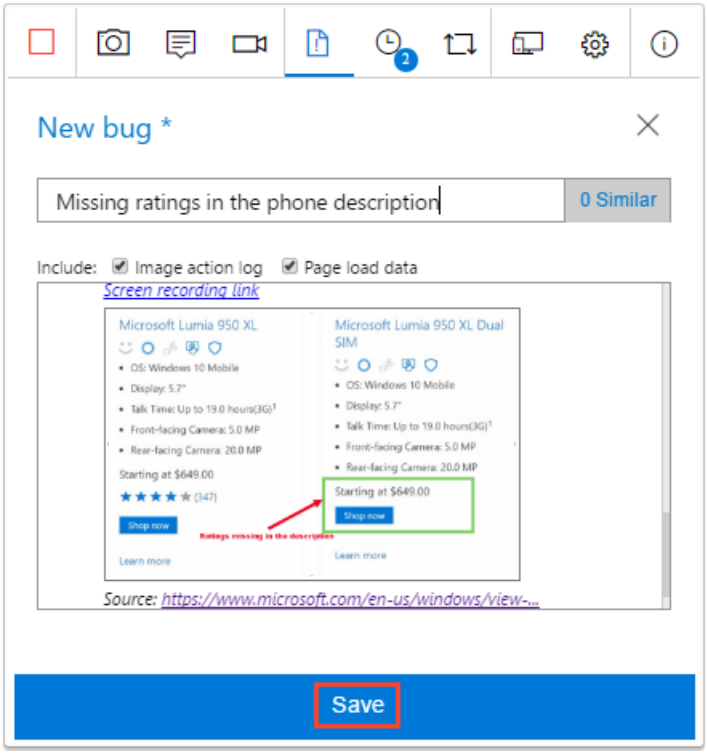


5. The bug or task form contains all your captured information. It also contains an image action log describing your interactions with the page (such as mouse clicks, keyboard typing events, touch gestures, and more) and page load data. Uncheck these options if you do not want to include this data in the bug or task.



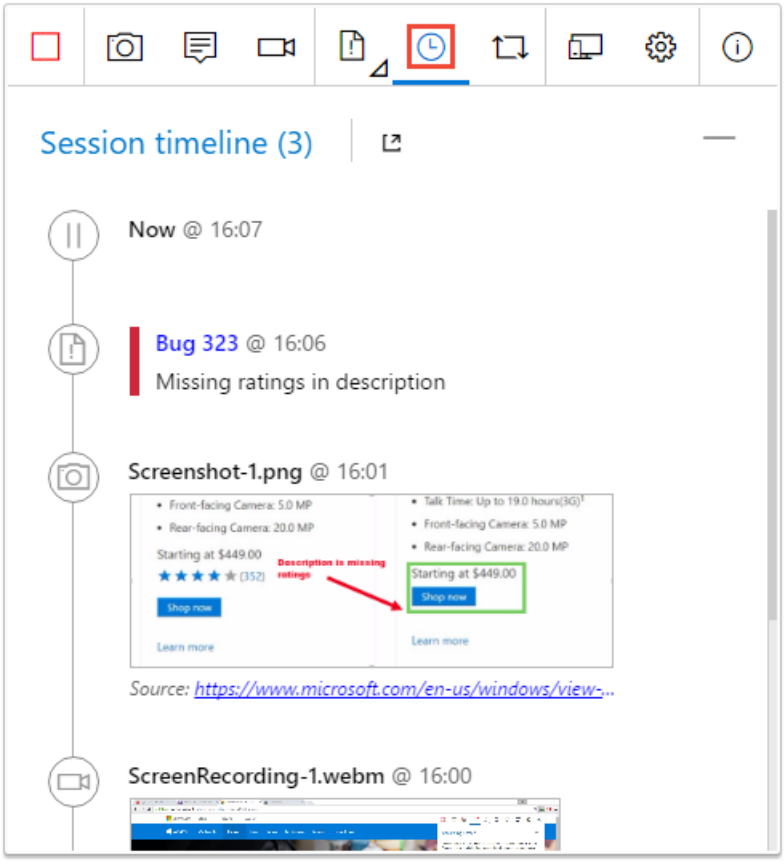
> The image action log is the sequence of steps you took that led to the issue. It can be used to reproduce the issue and understand the context. Page load data provides preliminary information about the time it takes to load the pages, such as the resource timings and navigation timelines.

6. Enter a title for the bug or task and add any additional notes you require to the description. Then save the bug or task.



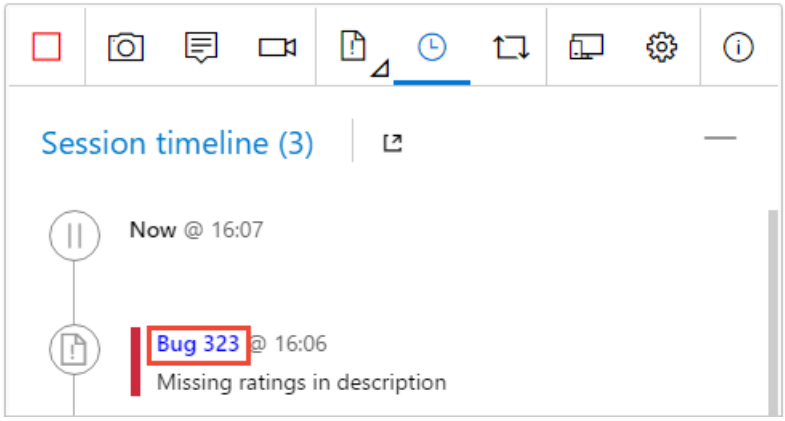
You can also add your findings to an existing similar bug.

7. View a list of all your activities in reverse chronological order in the **Session timeline** page. It shows all the screenshots, videos, and notes you've captured, the work items such as bugs, tasks, and test cases you've already filed, and the work items you've explored.

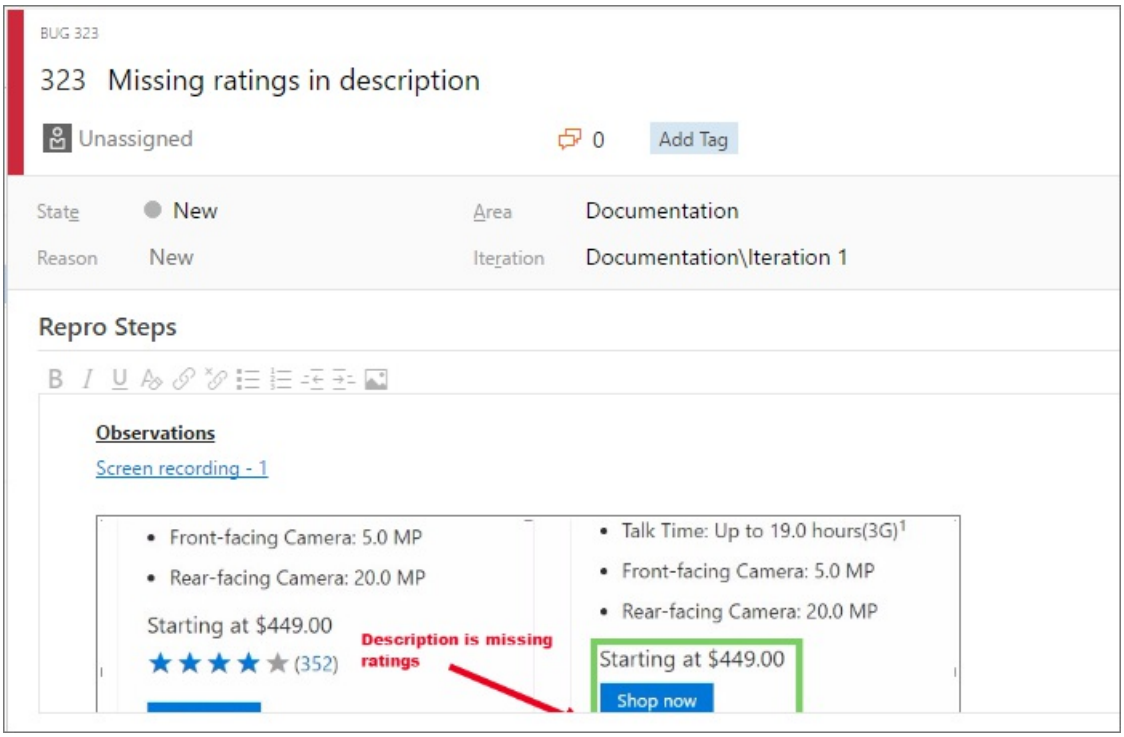


You can use the extension to explore work items in Azure DevOps.

8. To view a bug or task in Azure DevOps, choose the link in the session timeline.



This opens the work item form in Azure DevOps.

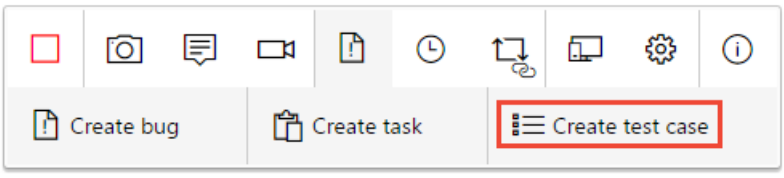


How do I play the video recordings I created with the extension?

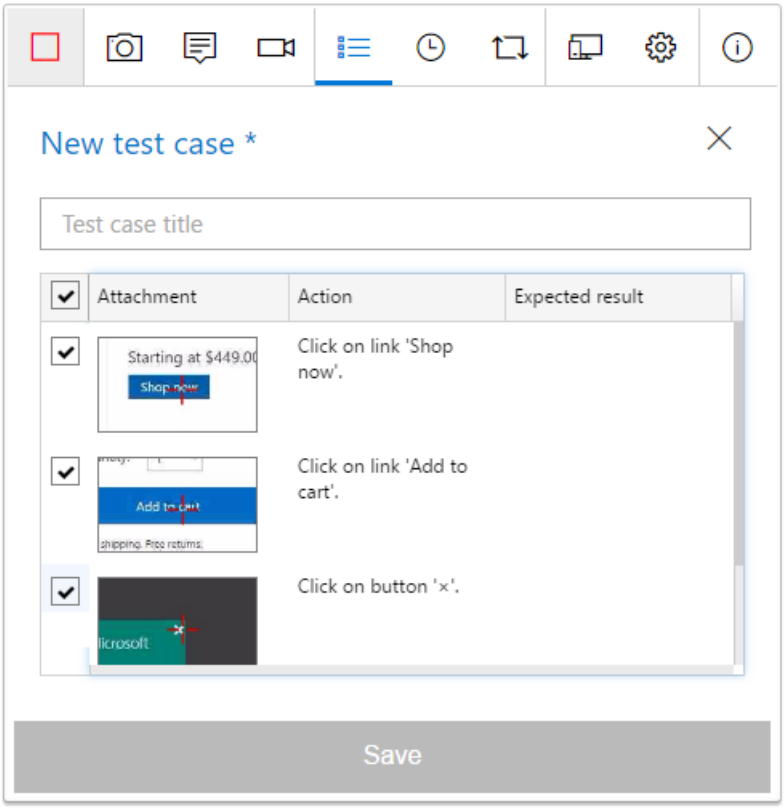
Create test cases

The extension lets you create test cases as you explore your application.

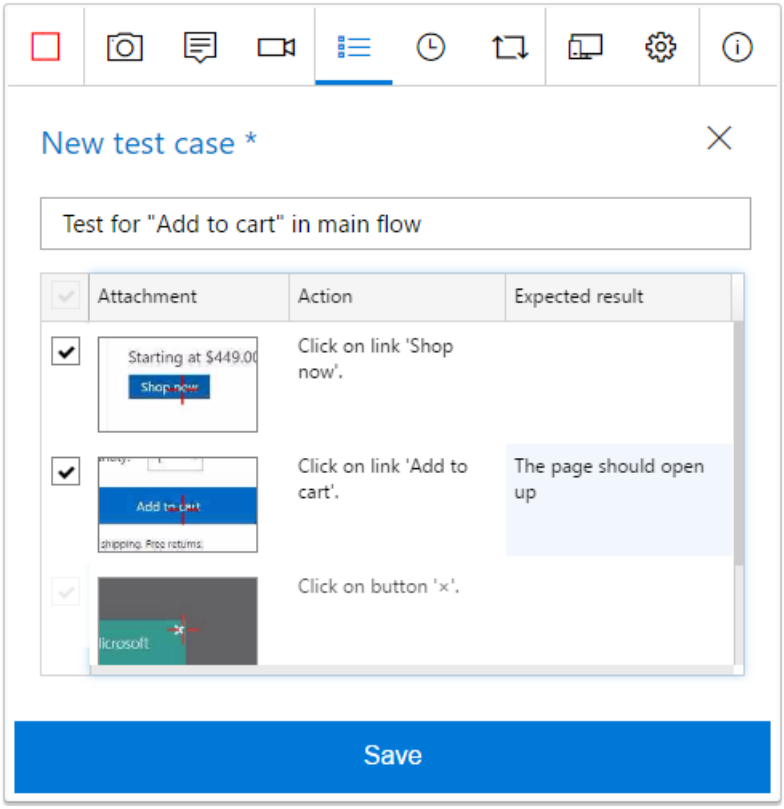
1. When you find a scenario where you want to create a test case, choose **Create test case**.



2. The test case form contains a list of all your actions up to this point while exploring the app (it reads them from the image action log).



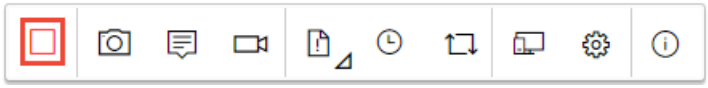
3. Enter a title for the test case and then edit it as required. For example, uncheck the action steps you don't want to include in the test case, edit the captured text, and add the expected result. Then save the test case.



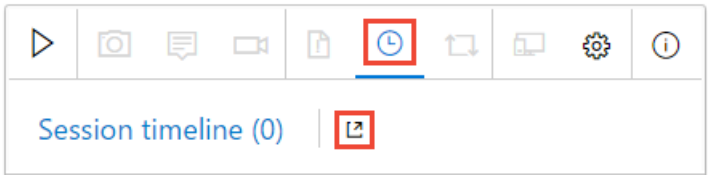
4. Continue exploring the application. Create more bugs, tasks, or test cases as required.

End your testing session

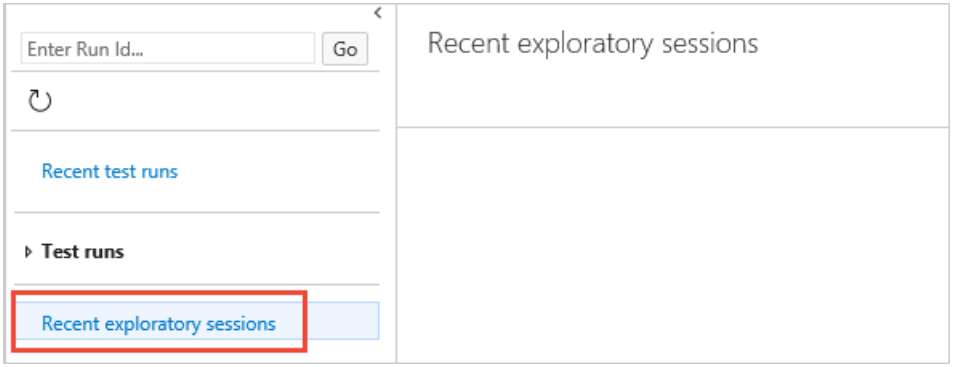
1. When you're done, stop your session.



2. If you are using Azure DevOps, TFS 2017, or later version, open the **Session timeline** page and choose the "view" icon to see your completed exploratory sessions in Azure DevOps.



Alternatively, open the **Recent exploratory sessions** list directly in the **Runs** page of the **Test Plans** web portal.



See your exploratory session results

After you file bugs, create tasks, or create test cases, all these show up in the "Recent exploratory sessions" page in Azure Test Plans.

- See how you can view your sessions and get insights.

How do I play the video recordings I created with the extension?

Exploratory testing with the Test & Feedback extension in Standalone mode

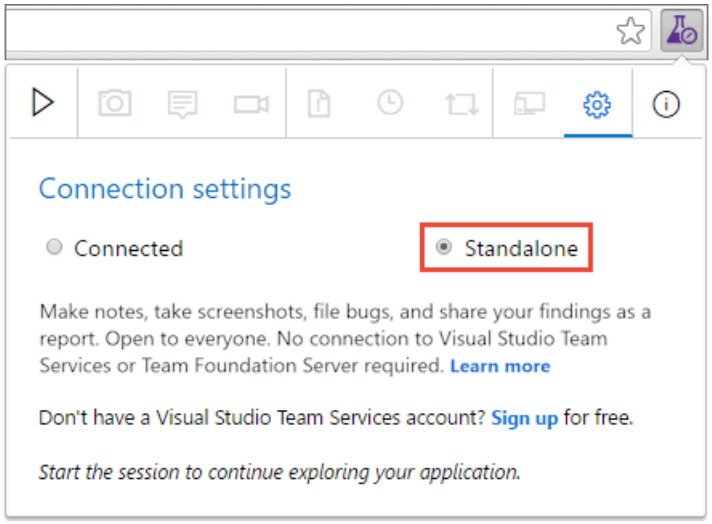
10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

All teams can use the Test & Feedback extension in **Standalone** mode. Users don't need an Azure DevOps subscription or Team Foundation Server connection to use this mode.

Start testing in Standalone mode

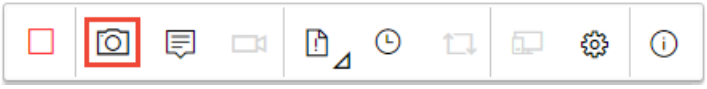
1. If you haven't already, install the Test & Feedback extension.
2. Open the extension in your web browser and select **Standalone** mode.



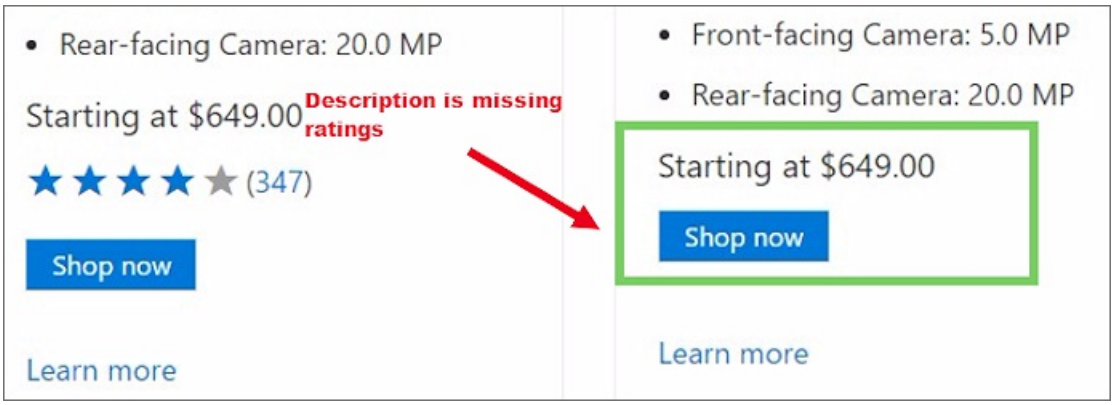
3. Open the web application you want to explore and start the testing session.



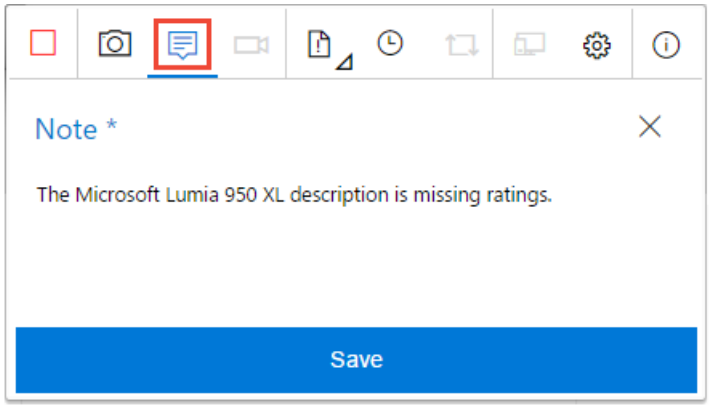
4. When you find an area that has a bug, take a screenshot of the entire screen or any part of it.



5. You can annotate the screenshot using the tools available in the inline annotation toolbar.

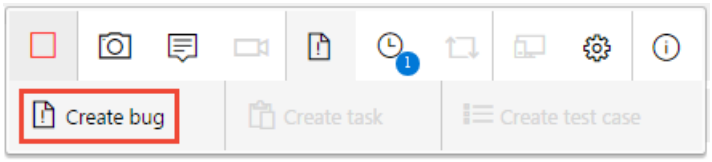


6. Make notes about the issue to share with your team, and then save the note.

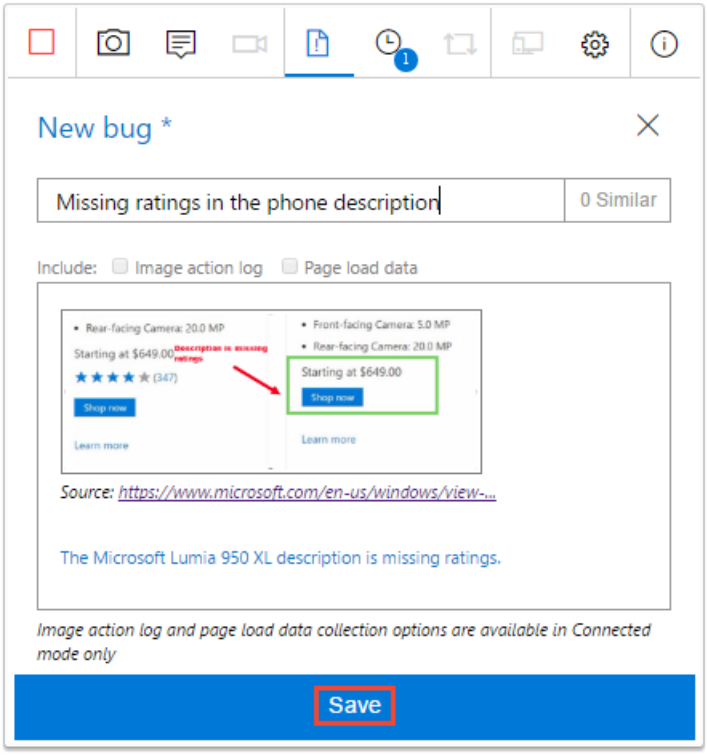


Create a bug

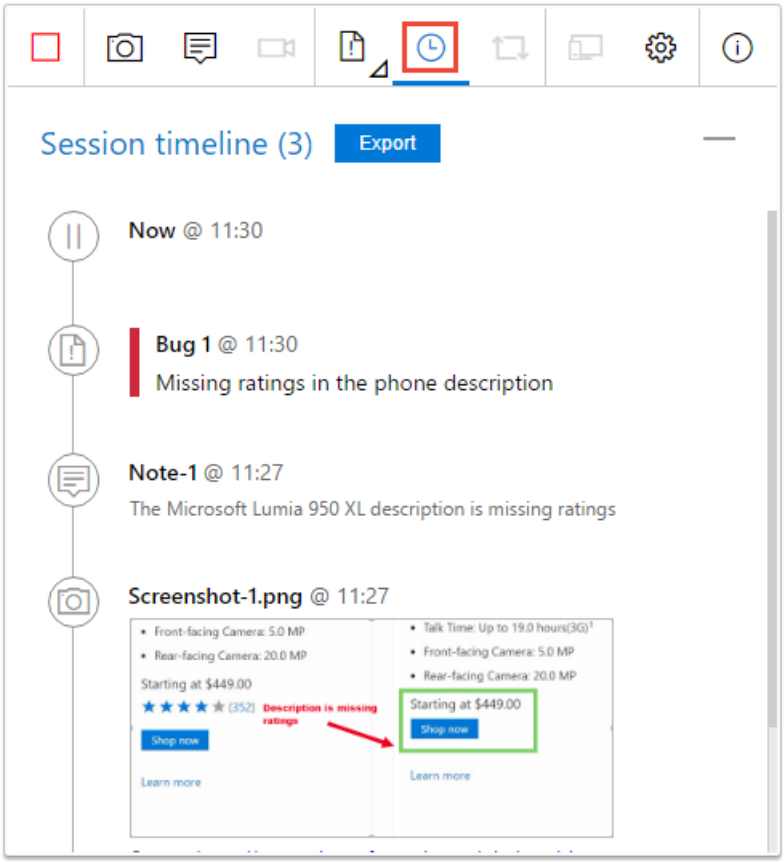
1. When you have finished capturing information for an issue, choose **Create bug**.



2. The bug form contains all your captured information. Enter a title for the bug and add any additional notes you require to the description. Then save the bug.



3. View a list of all your activities in reverse chronological order in the **Session timeline** page. It shows all the screenshots and notes you've captured and the bugs you've already created.

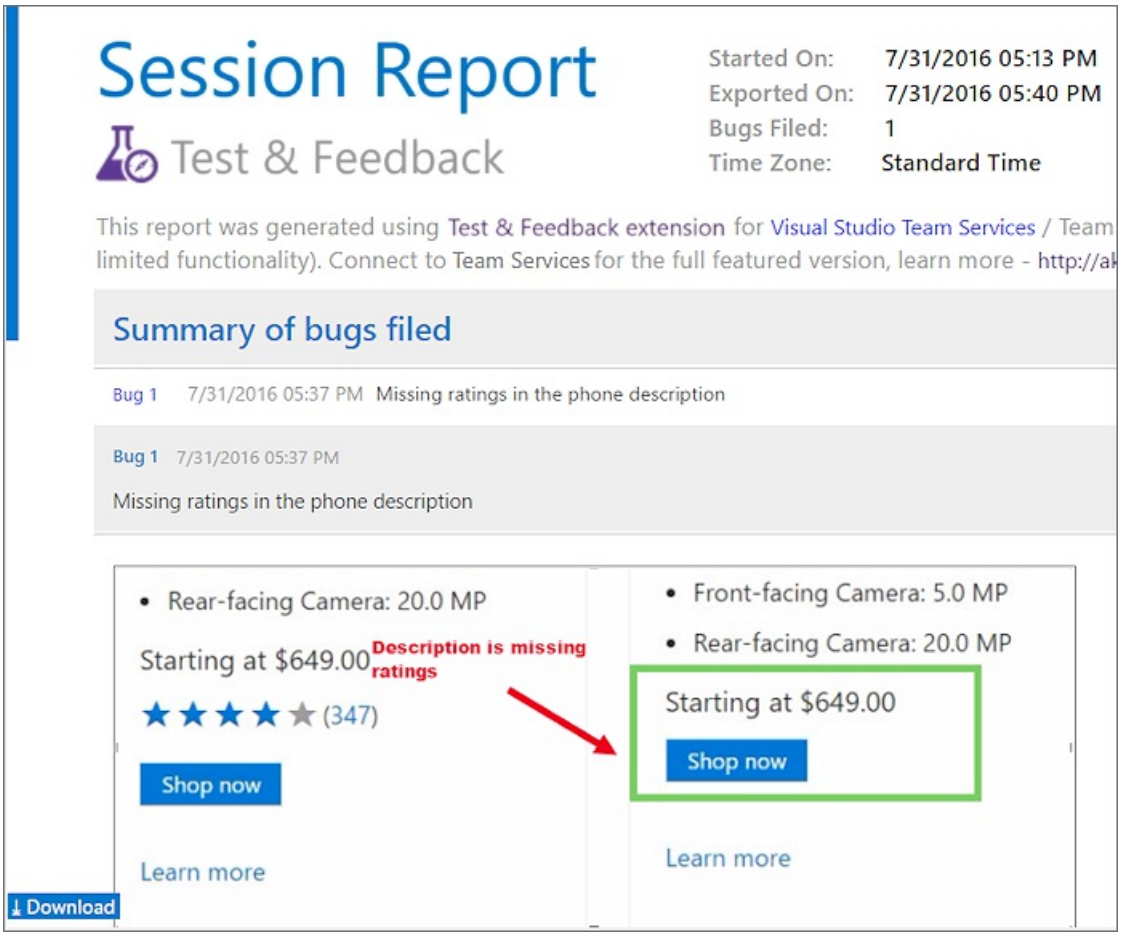


End your testing session

1. Continue exploring the application. Create more bugs as you encounter issues with the app.
2. When you're done, stop your session.



The extension automatically creates a session report that contains details of all the bugs created during the session, and any attachments.



3. The report is saved in the default Downloads folder of your web browser. Share it with the rest of your team as an email attachment, or copy it to OneNote, Word, or in any other format you prefer.

How do I play the video recordings I created with the extension?

Next step

Use the extension in Connected mode

Explore work items with the Test & Feedback extension

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use the Test & Feedback extension to explore existing work items and associate them with a new or an in-progress exploratory session. After a work item is associated with a session, all new bugs, tasks and test cases created in the current session are automatically linked to that work item. This enables end-to-end traceability, and simplifies tracking and management of issues.

You can explore:

- Work items belonging to a requirement category, a feature category, or an epic category
- Requirements-based test suites and test cases.

You can explore a work item from the Kanban board or from the extension. You can also explore multiple work items in the same session.

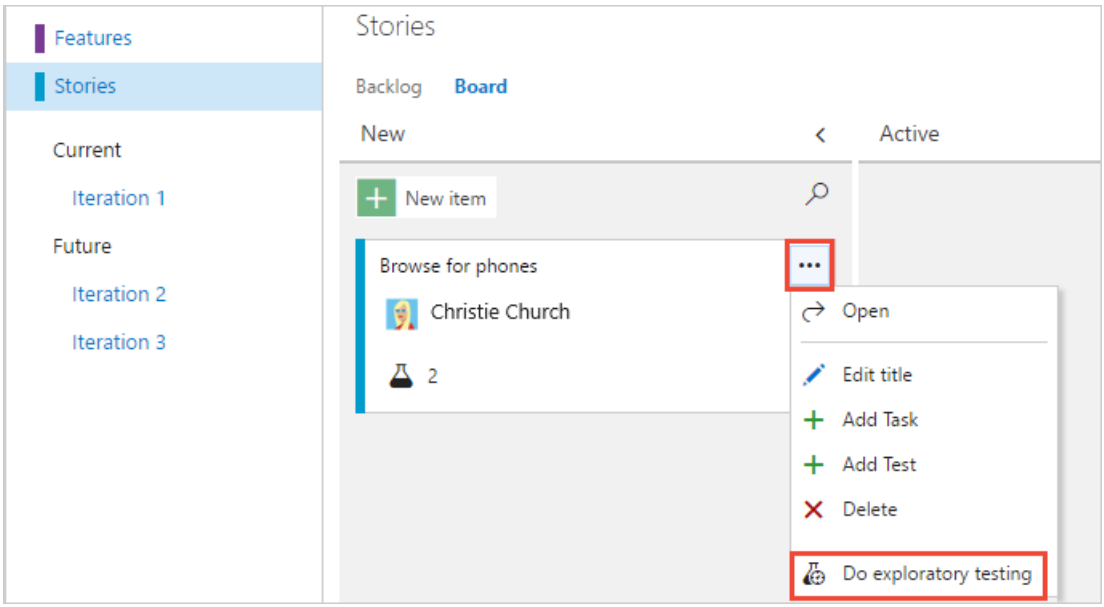
Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To request or provide feedback, you must have **Stakeholder** access or higher.
- To add or modify bugs or other work item types, you must have the **Edit work items in this node** permission set to **Allow** under the corresponding **Area Path**.
- To add new tags, you must have the **Create tag definition** permission set to **Allow**.

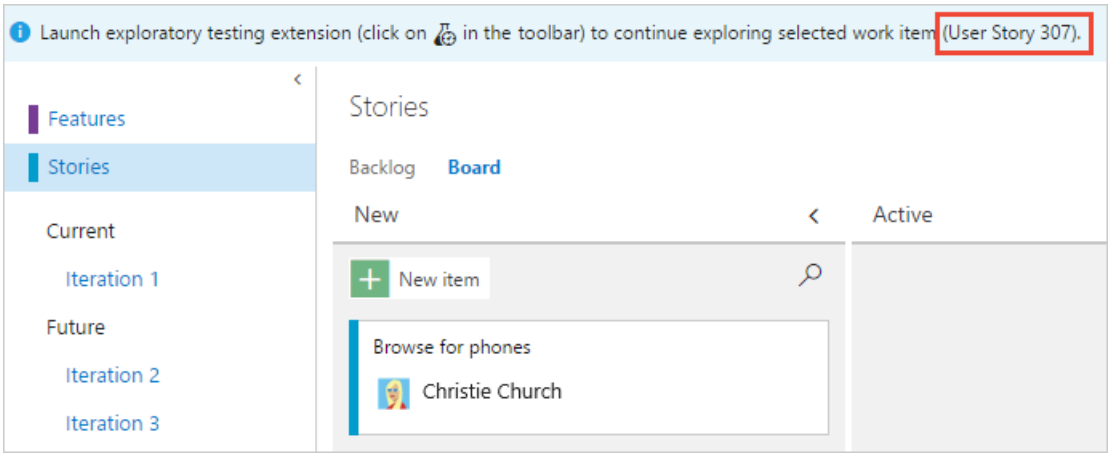
To learn more, see Set permissions and access for testing.

Explore work items from the Kanban board

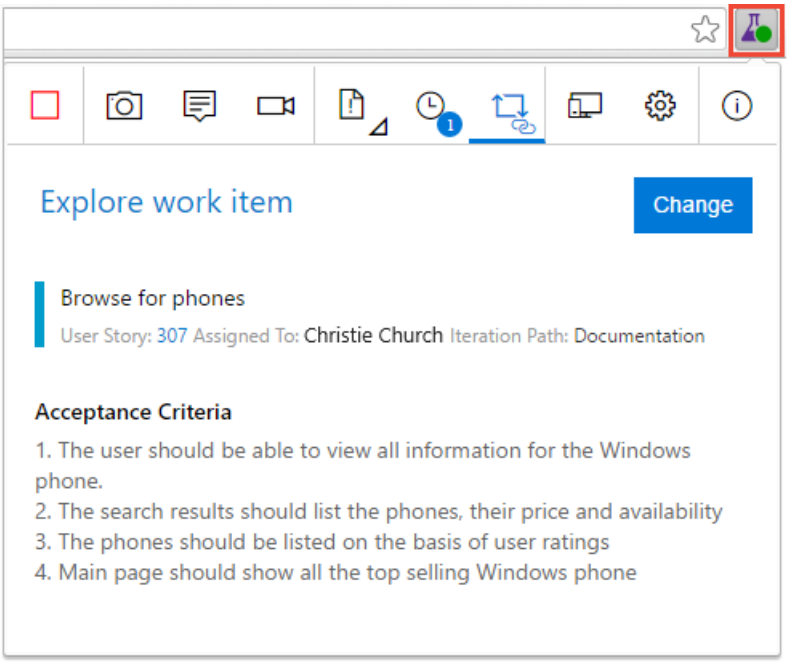
1. In the Kanban board, open the shortcut menu of the work item you want to explore, and choose **Do exploratory testing**.



2. A banner in the Work hub shows which work item is associated with your session.

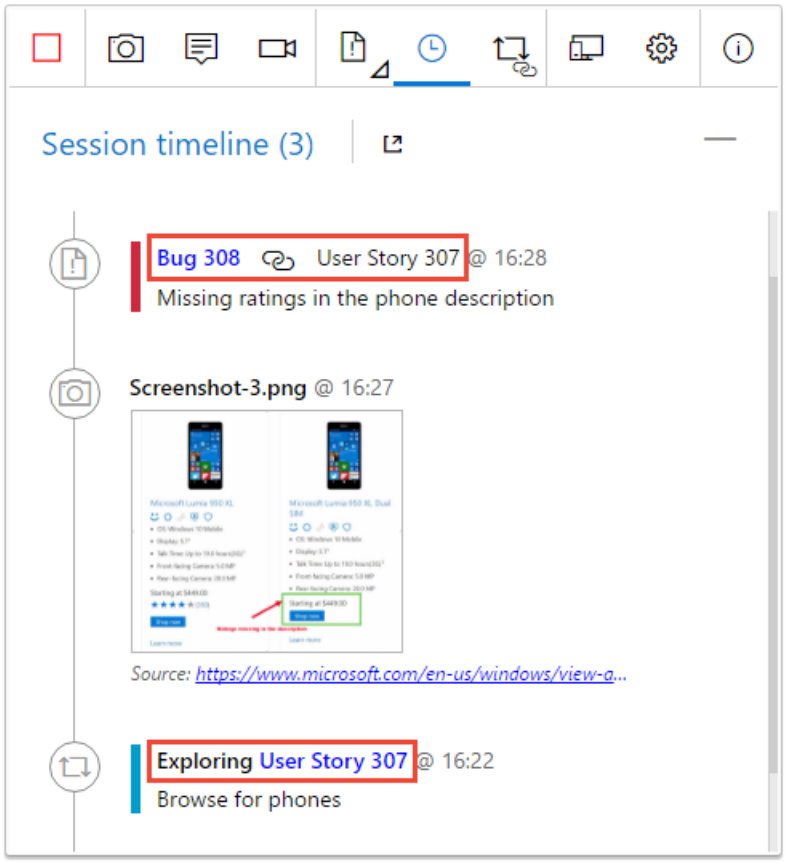


3. Launch the Test & Feedback extension. If there are acceptance criteria for the work item, these are shown.



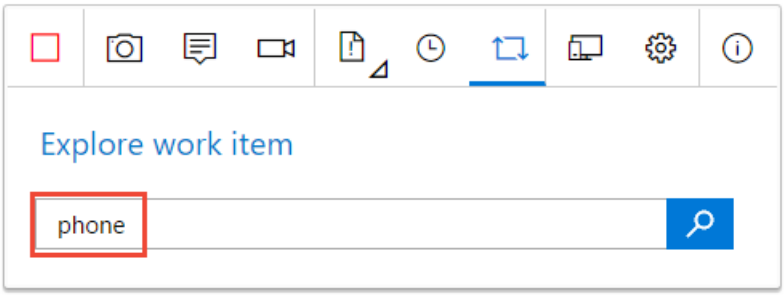
If you have not already started a session, start one now. The work item is automatically associated with the current or new session.

4. All bugs, tasks, and test cases you create will automatically be linked to the current work item.



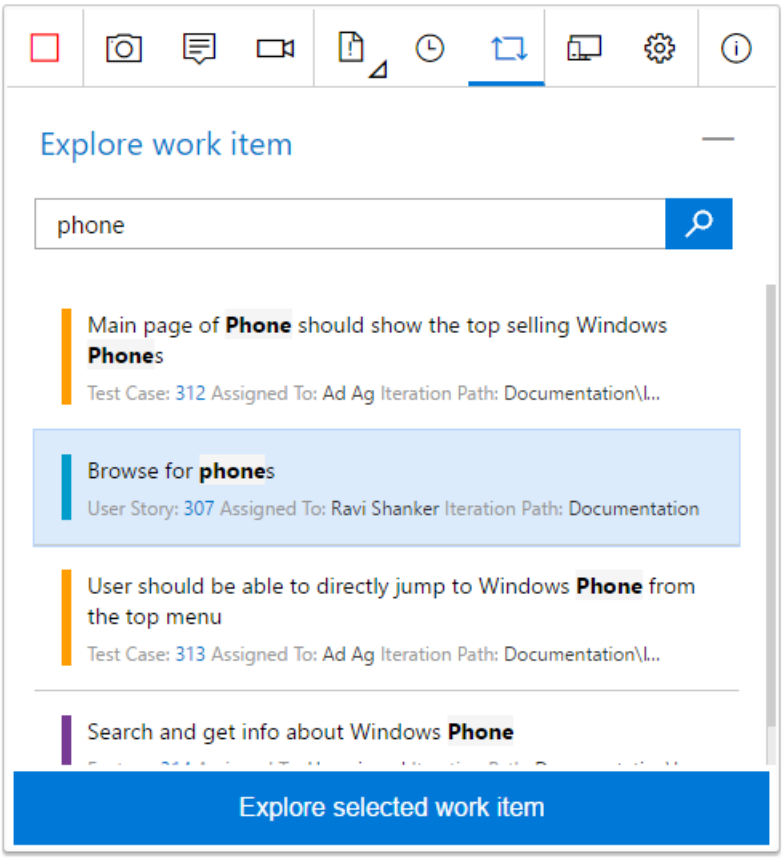
Explore work items from the Test & Feedback extension

1. Open the **Explore work item** page in the extension and search for the work item you want to explore.

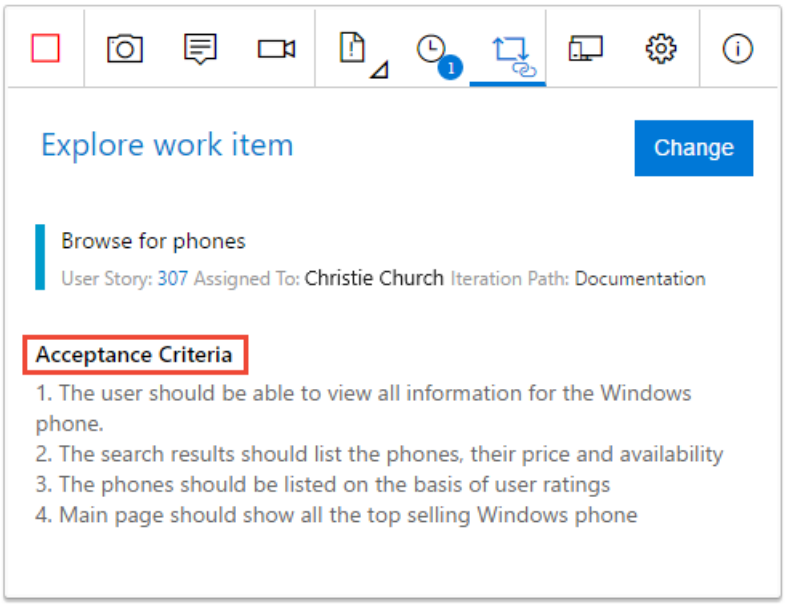


You can search using the work item identifier or keywords in the work item title.

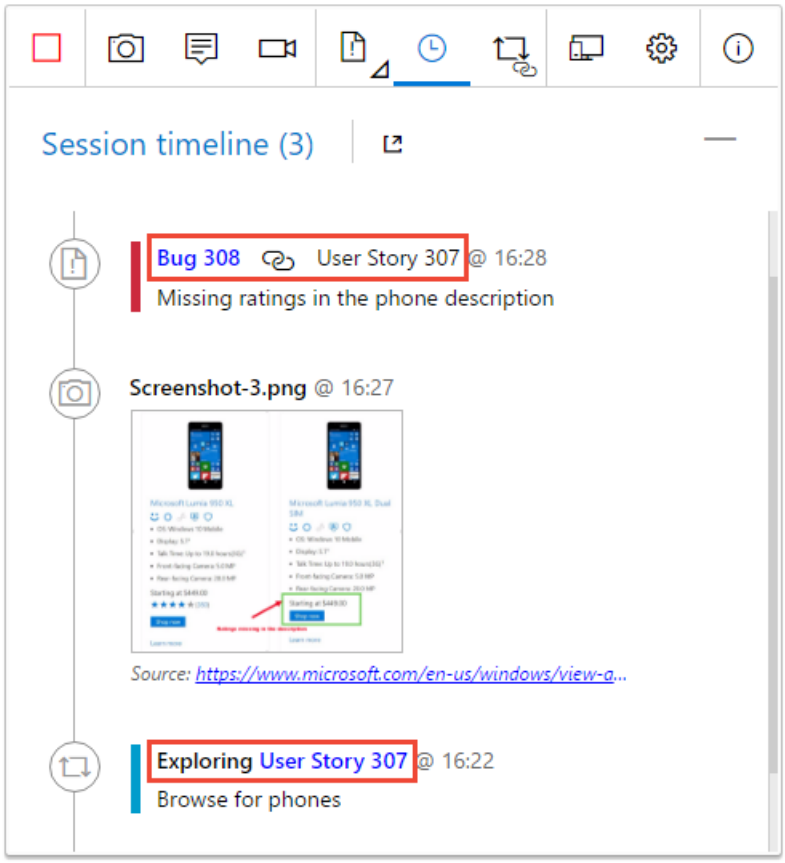
2. Select the work item in the search results and choose **Explore selected work item**.



3. The work item is now associated with the in-progress session. If there are acceptance criteria, these are shown.



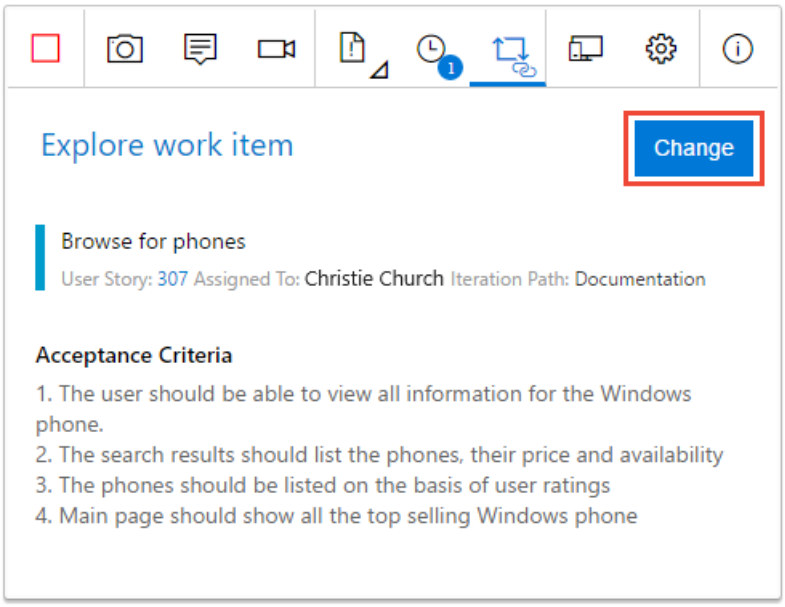
4. All bugs, tasks, and test cases you create will automatically be linked to the current work item.



Explore multiple work items in the same session

To explore another work item, you must first dissociate the current work item from the in-progress session.

1. Open the **Explore work item** page and choose **Change**.



2. Associate the new work item with the in-progress session as described above.

See your exploratory session results

After you file bugs, create tasks, or create test cases, all these show up in the "Recent exploratory sessions" page in Azure Test Plans.

- See how you can view your sessions and get insights.

See Also

- FAQs for manual testing
- Use the Test & Feedback extension in Connected mode
- Add findings to existing bugs with exploratory testing
- Get insights across your exploratory testing sessions
- Use the Test & Feedback extension in Standalone mode
- Exploratory testing with Microsoft Test Manager
- Overview of manual and exploratory testing

Add findings to existing bugs with exploratory testing

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

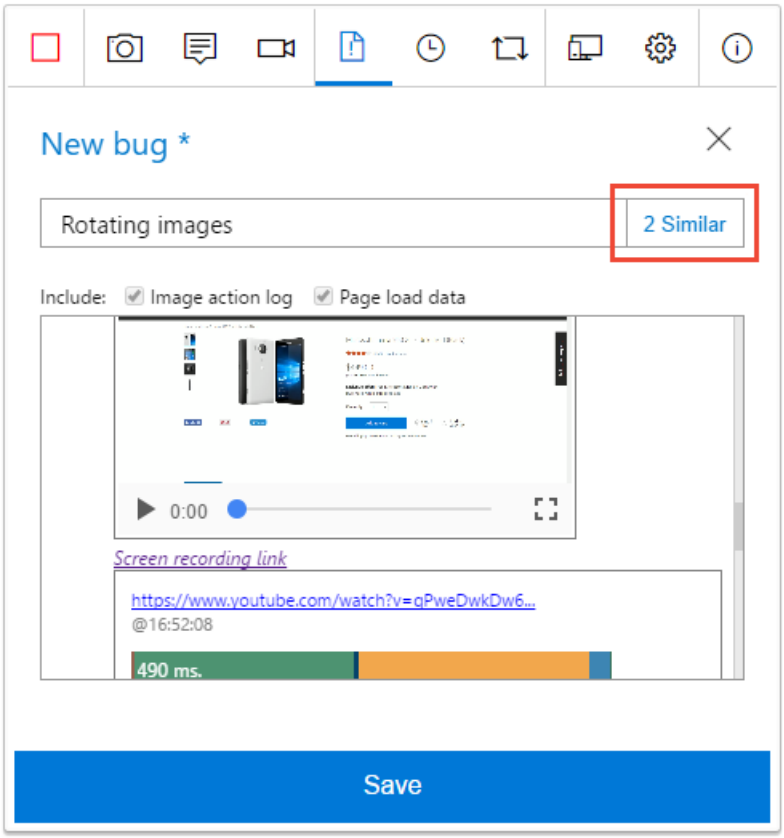
To help avoid duplication, the Test & Feedback extension automatically searches for and displays existing bugs, based on the keywords in the title, as you file a new bug. You can choose to continue creating a new bug or add your findings to an existing bug.

Prerequisites

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project. To get added, Add users to a project or team.
- To request or provide feedback, you must have **Stakeholder** access or higher.
- To add or modify bugs or other work item types, you must have the **Edit work items in this node** permission set to **Allow** under the corresponding **Area Path**.
- To add new tags, you must have the **Create tag definition** permission set to **Allow**.

To learn more, see Set permissions and access for testing.

1. As you type the title for a new bug, in the background the extension searches for similar bugs that might be related to the issue you've found and displays a link to the results. Choose this link to see the results that have similar title keywords.

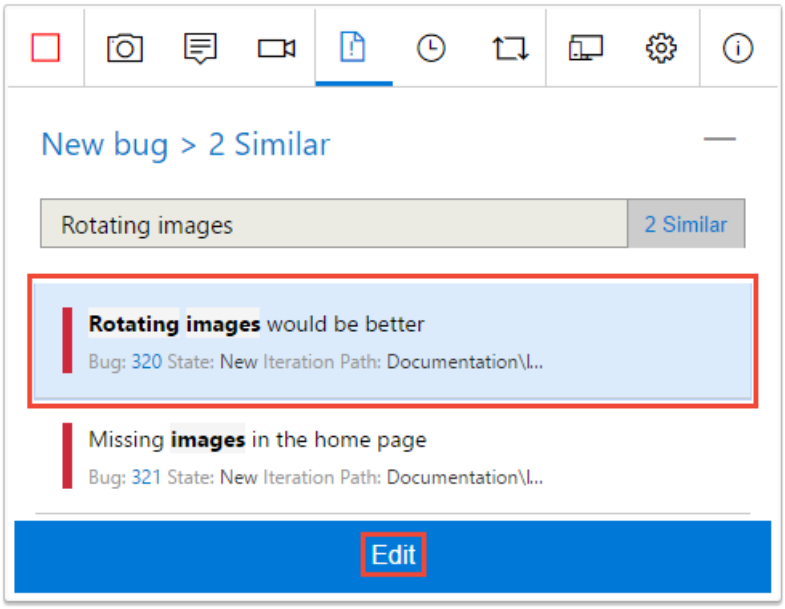


The form displays **0 Similar** if it does not find any matching bugs. In this case, or if you don't see a "similar" link, you can create a new bug containing your screenshots, notes, and videos as described in

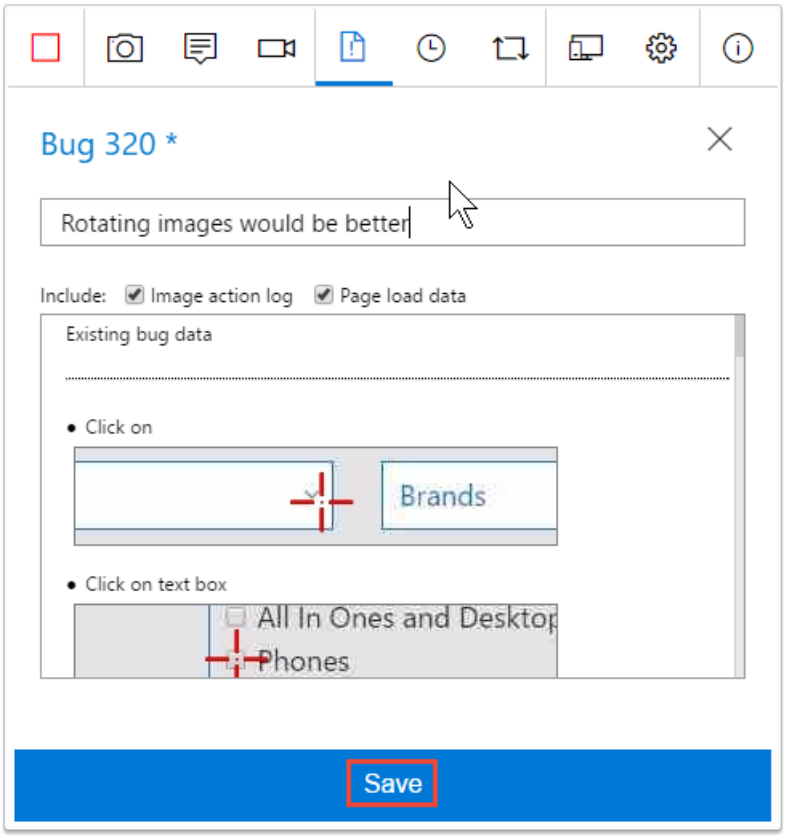
this topic.

2. If you see a bug you want to update, instead of creating a new one:

   - Select it in the list and choose **Edit**.

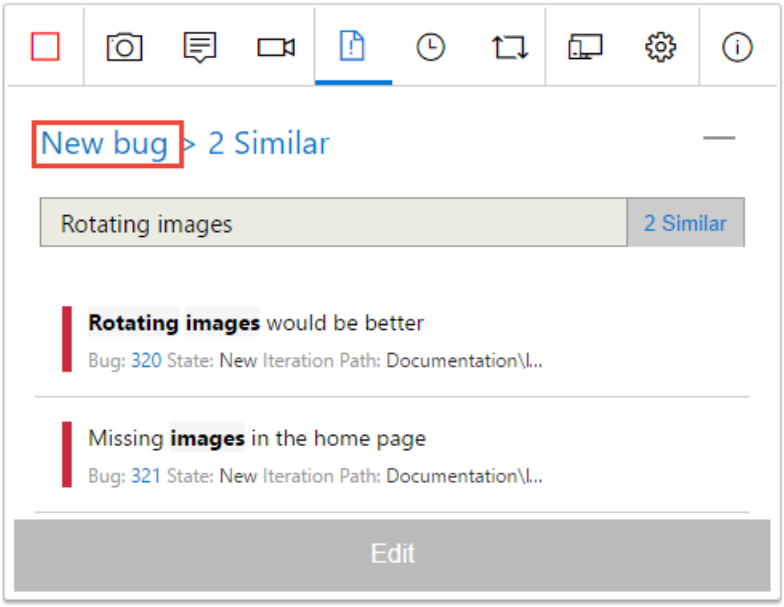


   - The extension appends all your screenshots, notes, and videos to the existing bug.
   - Save the updated bug.



3. If, instead, you decide not to update an existing bug, ignore the "similar" link and:

   - Choose the **New bug** link to return to the bug details form.



- Enter the details for the new bug and save it as described in this topic.

4. Continue exploring your app, filing bugs and tasks, and creating test cases.

See your exploratory session results

After you file bugs, create tasks, or create test cases, all these show up in the "Recent exploratory sessions" page in Azure Test Plans.

- See how you can view your sessions and get insights.

See Also

- Use the Test & Feedback extension in Connected mode
- Explore work items with exploratory testing
- Get insights across your exploratory testing sessions
- Use the Test & Feedback extension in Standalone mode
- Exploratory testing with Microsoft Test Manager
- Overview of manual and exploratory testing

Voluntarily provide stakeholder feedback using the Test & Feedback extension

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Stakeholders can respond to feedback requests for user stories and features generated in Azure DevOps using a lightweight end-to-end flow based on the Test & Feedback extension. Only users with Basic access can request feedback. Basic users can provide feedback using the flow described in this topic.

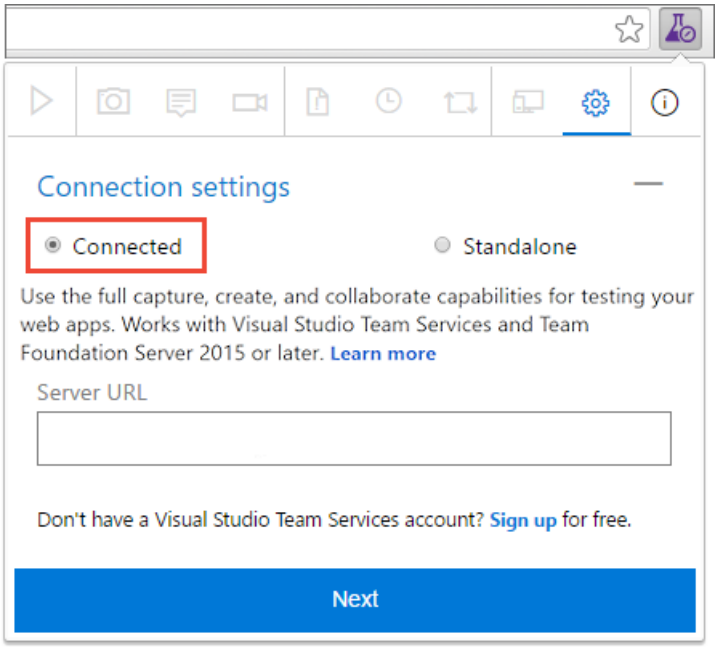
- Request feedback
- Provide feedback
- Voluntary feedback
- Track requests

> **NOTE**
> This lightweight end-to-end flow is applicable only for web apps and by using Azure DevOps. To get feedback for desktop apps, or for earlier versions of TFS, use the feedback flow described in Get feedback about the Microsoft Feedback Client.

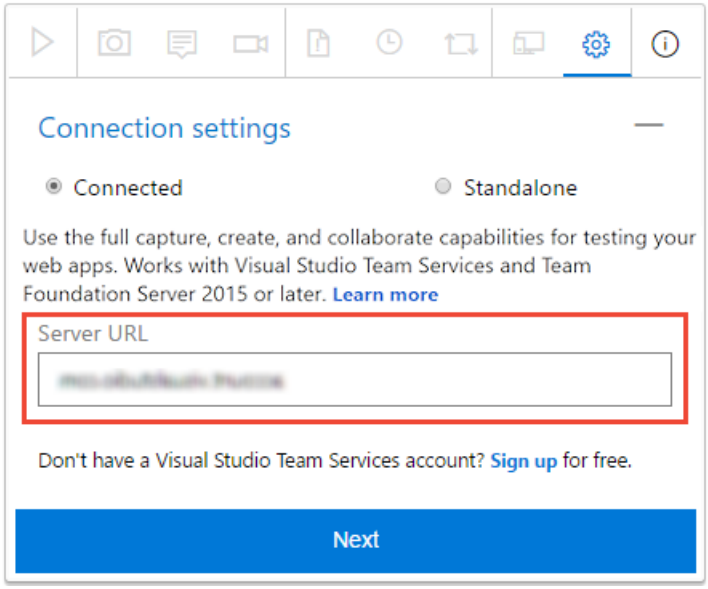
Provide voluntary feedback

You can use the Test & Feedback extension to provide feedback voluntarily, even if you haven't received a specific feedback request.

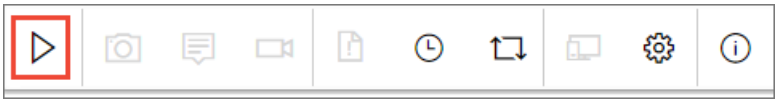
1. Open the Test & Feedback extension in your browser using the icon in the toolbar.
2. In the **Connection settings** page, choose **Connected** mode.



3. Connect to the server and the project or team that is requesting feedback.



4. Start the exploratory testing session.



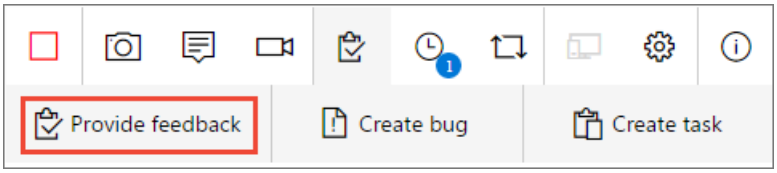
5. Open the application you want to provide feedback on and begin your feedback. For example, choose **Capture screenshot** to take a screenshot.



You can use all the capabilities of the extension such as capturing screenshots, notes, and screen recordings.

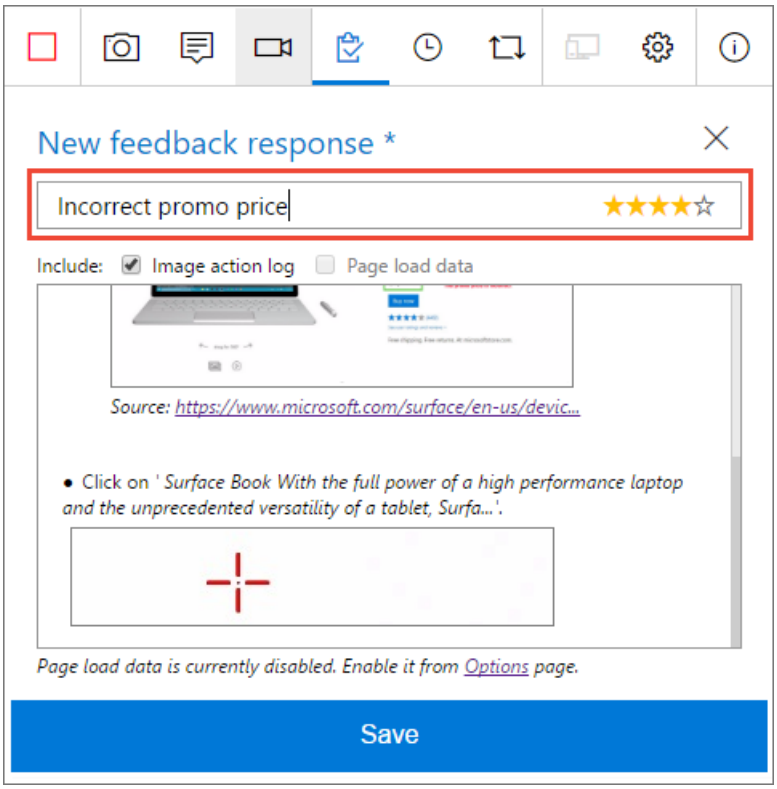
> Some browsers may not provide all of the capture capabilities. See Supported web browsers for the extension.

6. When you are done capturing feedback, Choose **Provide feedback**.



You can optionally choose to create bugs and tasks when you submit your feedback. The process is the same as described here.

7. All your feedback captured is shown in the response form. Type a suitable title and, optionally, select a star rating for the feature you've been testing.



8. Save your feedback. This create a work item in Azure DevOps containing all your feedback.
9. Continue to capture more feedback if required. You can submit multiple feedback responses, bugs, and tasks for the same feedback request.
10. Choose the **Stop** icon to end your feedback session.



Related articles

- Request stakeholder feedback using the Test & Feedback extension
- Provide stakeholder feedback using the Test & Feedback extension
- Track stakeholder feedback using the Test & Feedback extension
- Exploratory test and submit feedback directly from your browser
- Overview of manual and exploratory testing

Track stakeholder feedback

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

All feedback is captured in a *Feedback Response* work item. You can track feedback, whether captured by the Test & Feedback extension or the Microsoft Feedback client, through a work item query.

Prerequisites

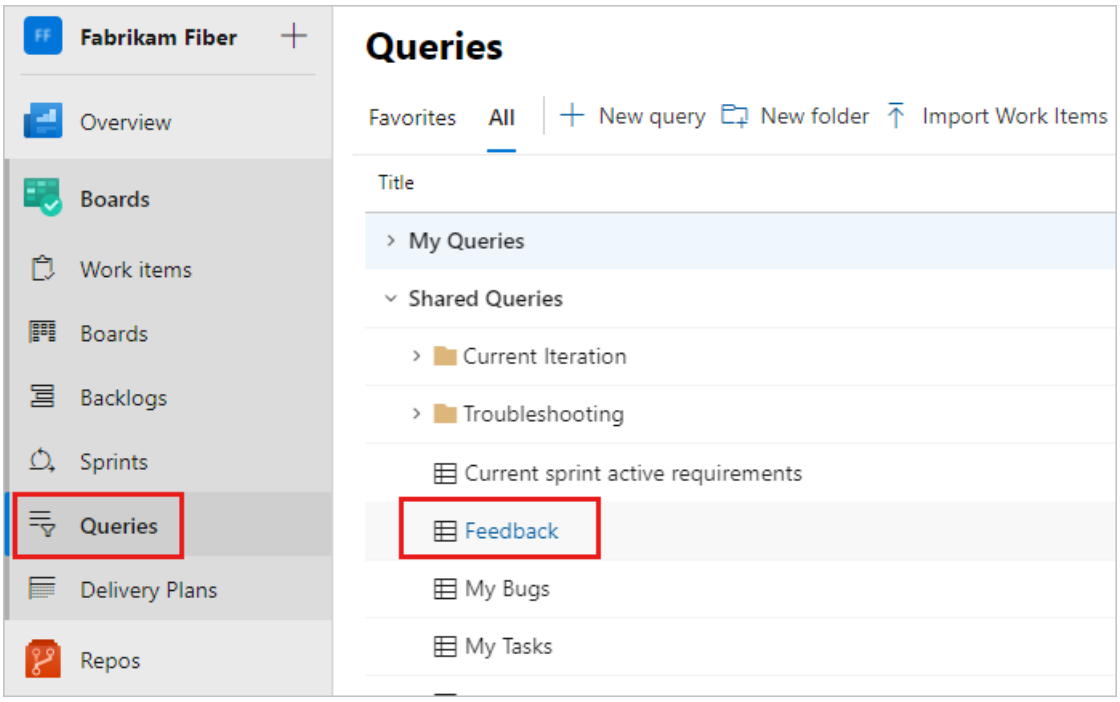
- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

Track feedback requests

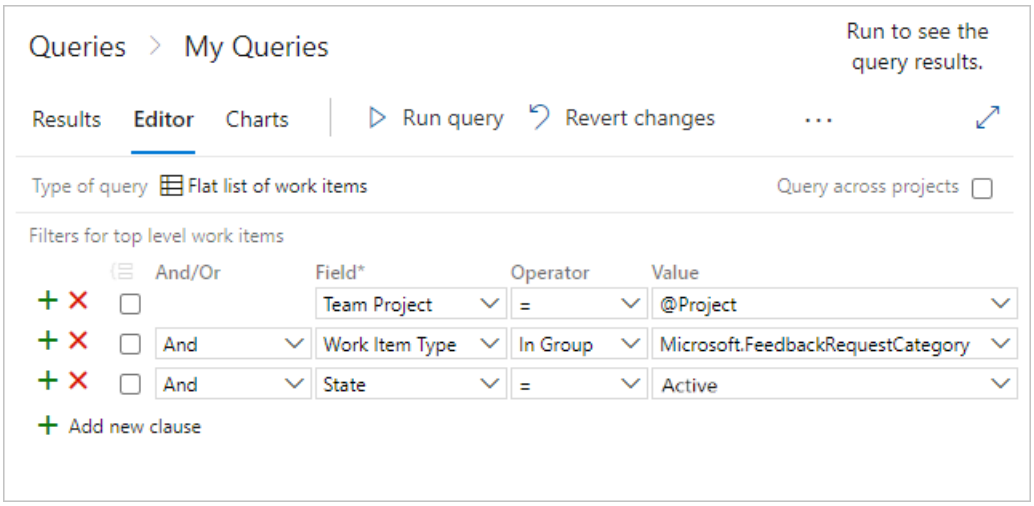
To view feedback, use the **Feedback** shared query. Select your project and open **Boards** > **Queries**. Under **Queries**, select **All**. In the Shared Queries, select **Feedback**.



This query displays a list of all the feedback responses received. To learn more, see Web portal navigation.

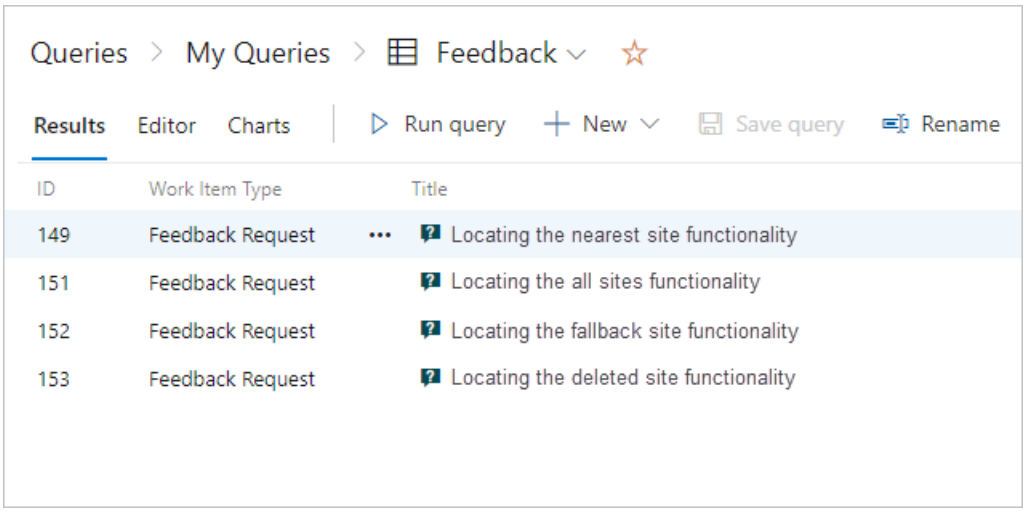
To create a feedback query, follow these steps:

1. Select **Boards** > **Queries** and then select **New query**.
2. In the **Editor** for your new query, enter the following values:



- Team Project = @Project
- Work Item Type In Group Microsoft.FeedbackRequestCategory
- State = Active

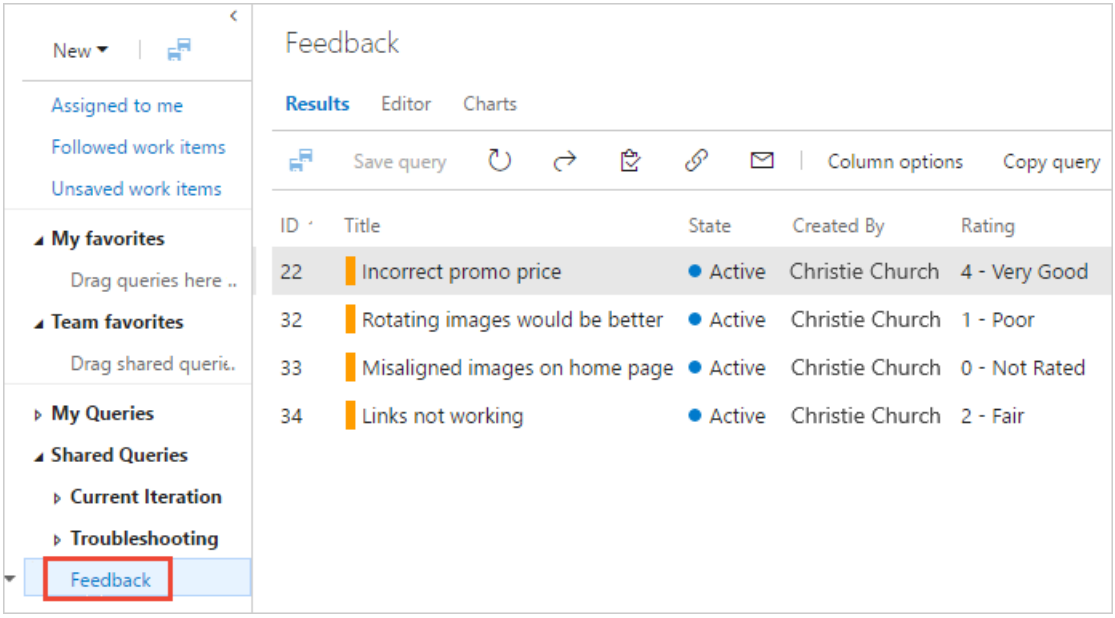
3. Select **Save query** and enter a name.
4. Select **Run query** to see a list of active feedback responses for your team project.



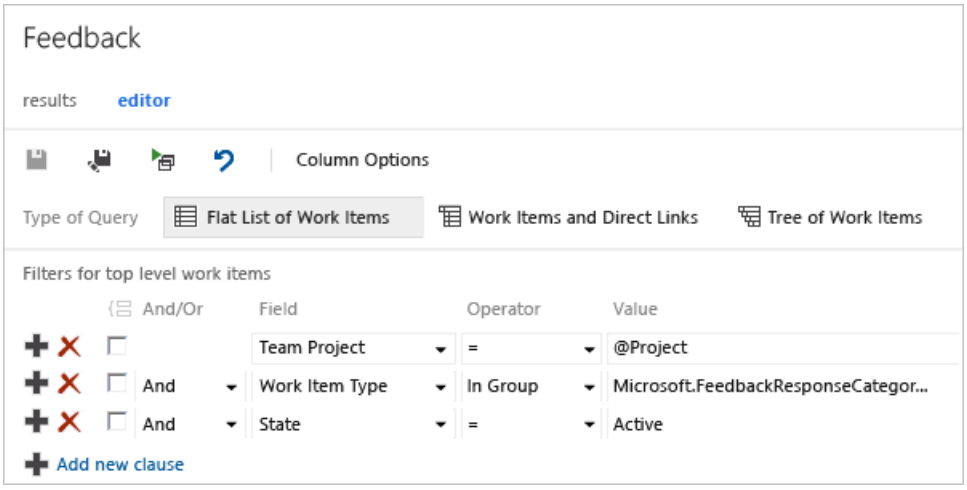
5. Select a response work item to see the details of the feedback.

1. Select your project and open **Boards>Queries** or **Work>Queries**. To learn how, see Web portal navigation.

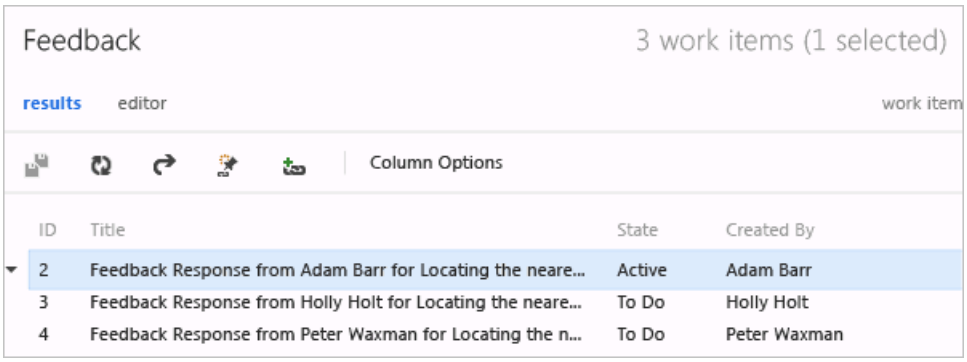
2. In the list of shared queries, select **Feedback**. This query displays a list of all the feedback responses received.



Or, create a feedback query with the parameters, as shown.



3. You should see a list of all active feedback responses for your team project.



4. Open the response work item to see the details of the feedback.

Related articles

- What is Azure Test Plans?
- Request feedback using the Test & Feedback extension
- Get feedback
- Provide feedback using the Test & Feedback extension
- Define a work item query

Get feedback

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Once you have working software, you're ready to get feedback from your stakeholders. You can ask reviewers to provide videos, screenshots, type-written comments, and ratings. Their feedback is captured into work items that you can review and use to create a bug or suggest a new backlog item.

Before requesting feedback, make sure that you provide stakeholders who'll you request feedback from the necessary permissions.

> **NOTE**
> You can also request feedback from stakeholders for web apps using the Test & Feedback extension. For desktop apps, you must use the feedback request form documented in this topic and stakeholders must reply using the Microsoft Feedback Client.

Prerequisites

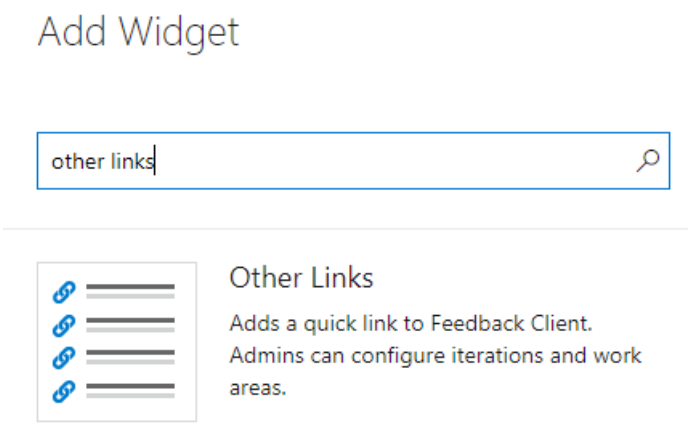
- You must connect to a team project. If you don't have a project yet, create one in Azure DevOps Services.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To request feedback, you must be granted **Basic** access or higher. For details, see About access levels.
- To provide or review feedback, you must be granted **Stakeholder** access or higher.
- To view or modify feedback responses, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to the Request feedback feature just like users with **Basic** access. For details, see About access levels.

- You must connect to a project. If you don't have a project yet, create one.
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, Add users to a project or team.
- To request feedback, you must be granted **Basic** access or higher. For details, see About access levels.
- To provide or review feedback, you must be granted **Stakeholder** access or higher.
- To view or modify feedback responses, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To send feedback requests, the server administrator must configure an SMTP server.

Add the Other links widget to your dashboard

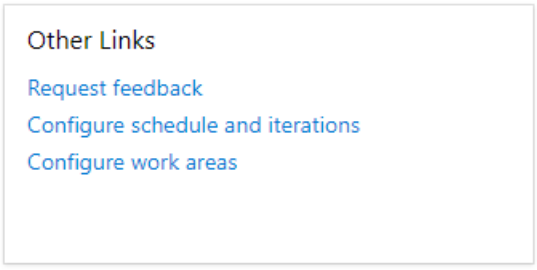
Add the **Other links** widget to a web portal team dashboard. For details, see Add widgets to a dashboard



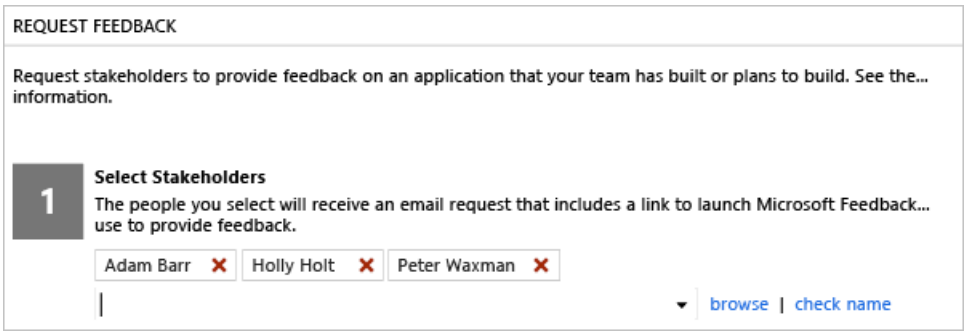
Request feedback

To request feedback, you fill out a form that generates an email request to your stakeholders.

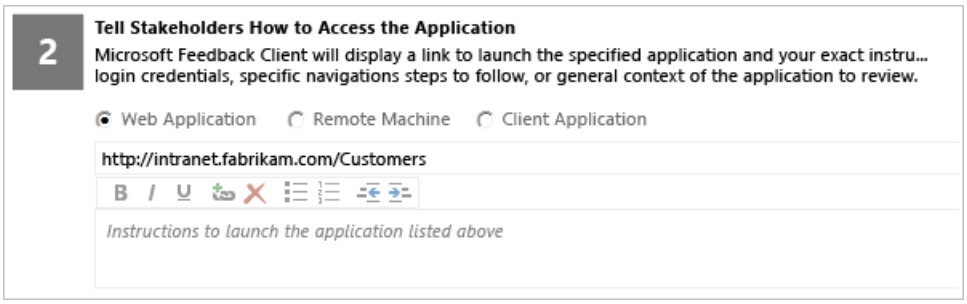
1. From the dashboard, choose the **Request feedback** link from the Other links widget.



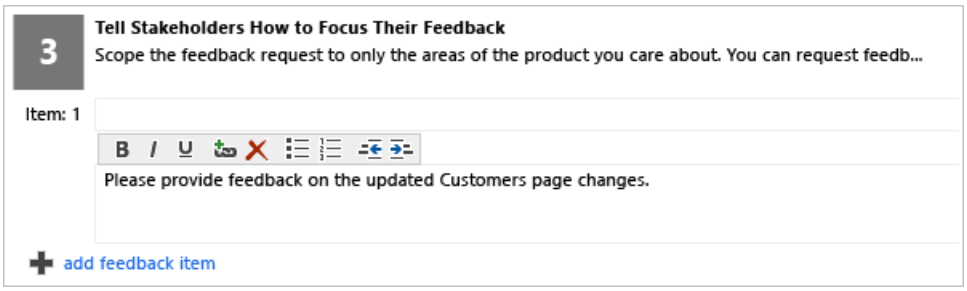
2. Add the feedback reviewers. If you don't see the names you want in the browse list, grant them permissions to provide feedback.



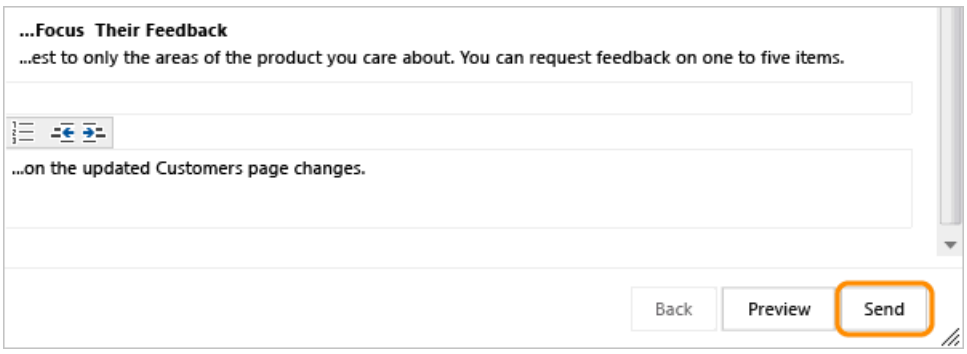
3. Tell your reviewers how to run the app they'll be reviewing.



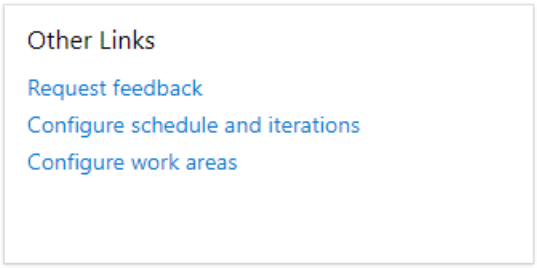
4. For each area of interest, decide what type of feedback you want. Set the context for the reviewers by providing enough background information. Add up to four more areas of interest with the **add feedback item** link.



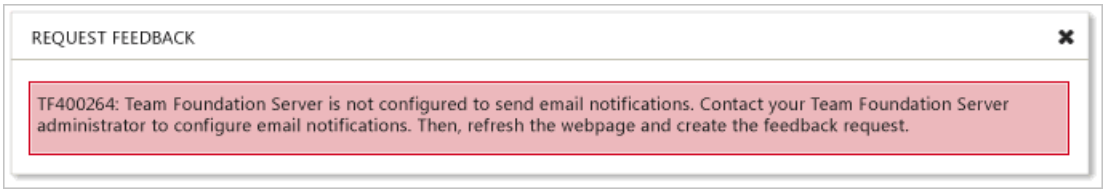
5. Send the request.



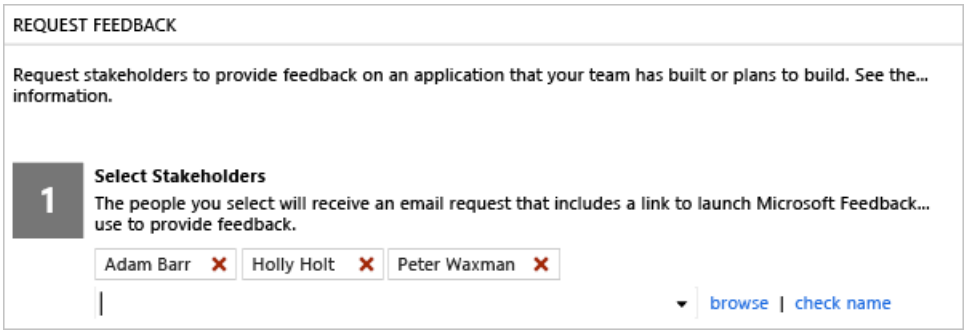
1. From the dashboard, choose the **Request feedback** link from the Other links widget.



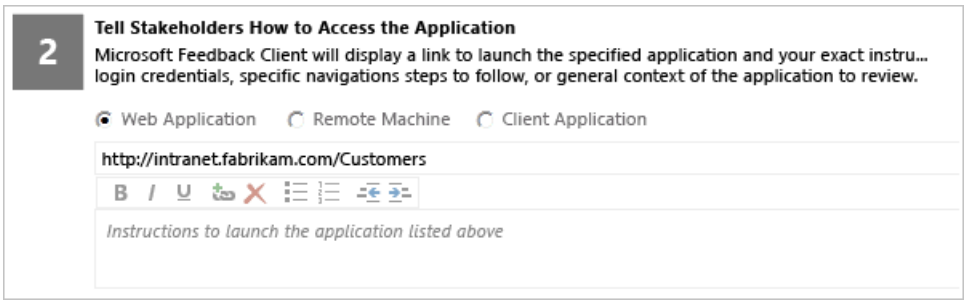
If the following message appears, you need to configure an SMTP server.



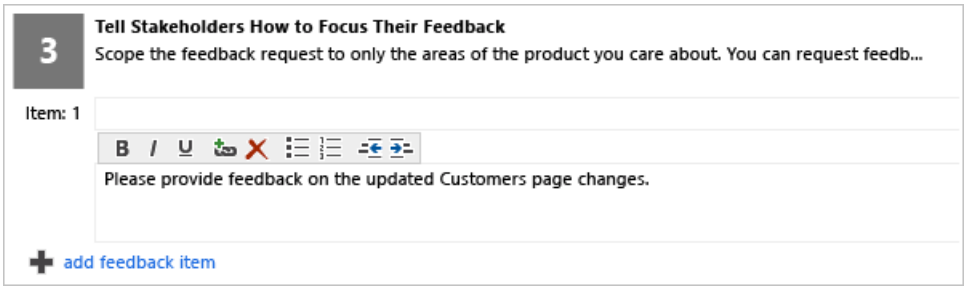
2. Add the feedback reviewers. If you don't see the names you want in the browse list, grant them permissions to provide feedback.



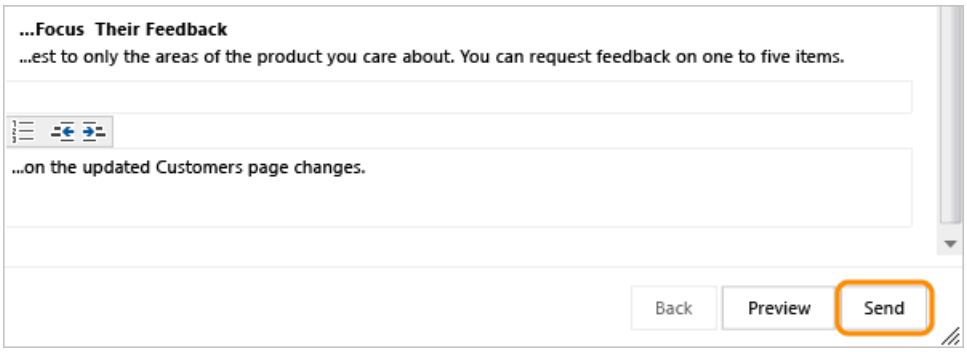
3. Tell your reviewers how to run the app they'll be reviewing.



4. For each area of interest, decide what type of feedback you want. Set the context for the reviewers by providing enough background information. Add up to four more areas of interest with the **add feedback item** link.



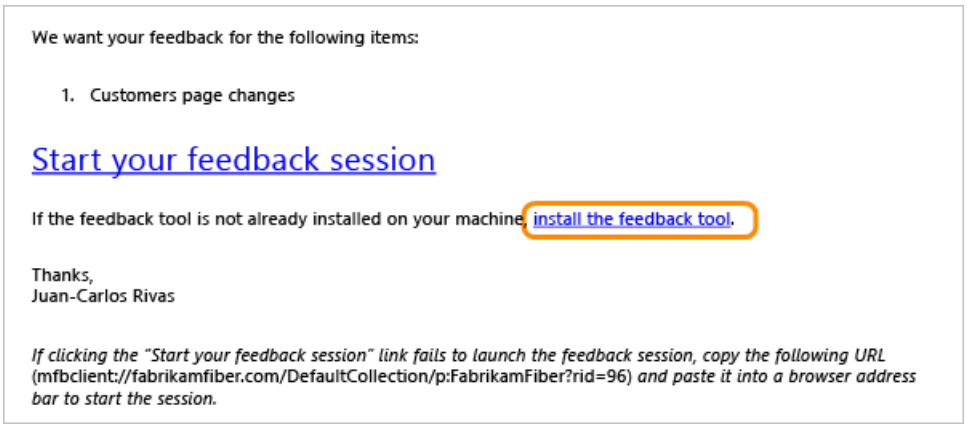
5. Send the request.



Provide feedback

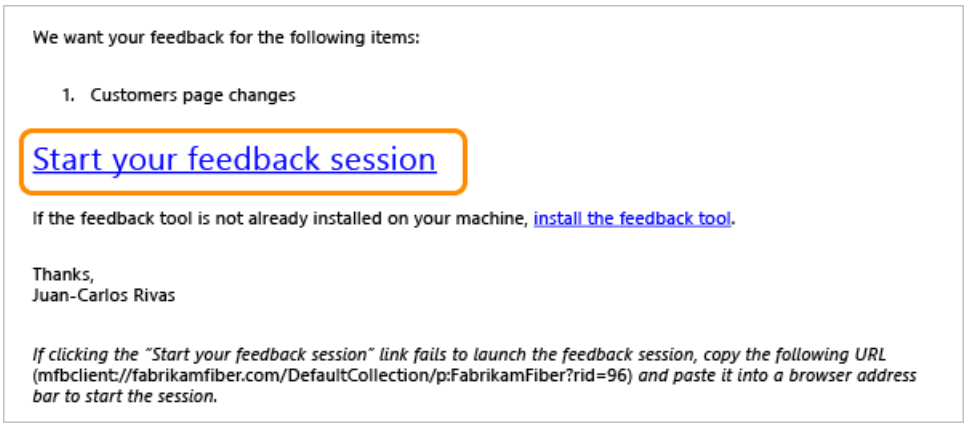
Reviewers launch your application and provide feedback through the free Microsoft Feedback Client.

1. Reviewers who don't have a version of Visual Studio installed can download the feedback client directly from the feedback request they receive.

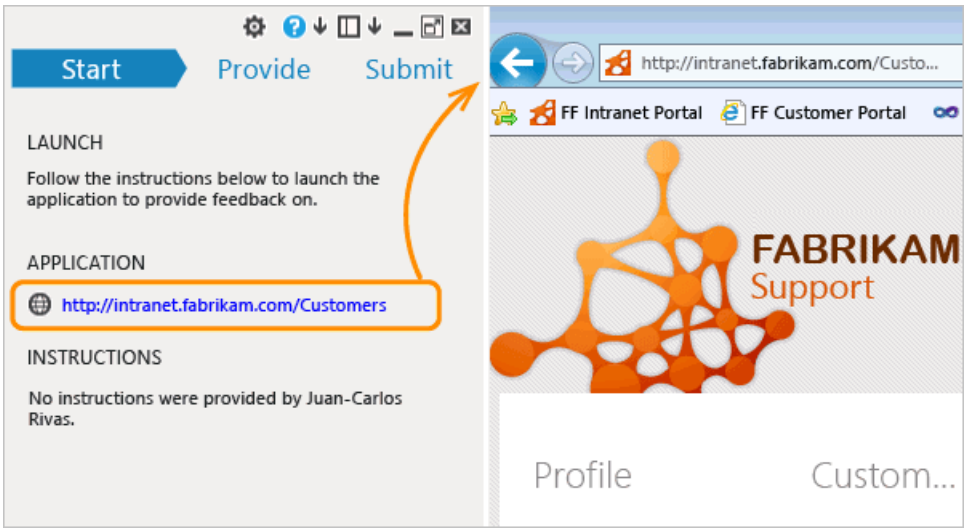


   Or, they can go to the Visual Studio download site.

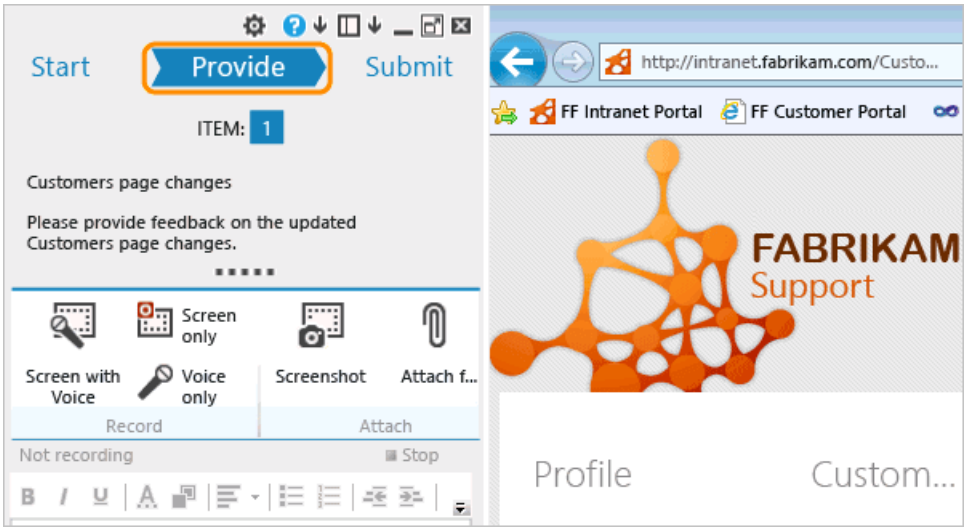
2. Reviewers start the feedback session.



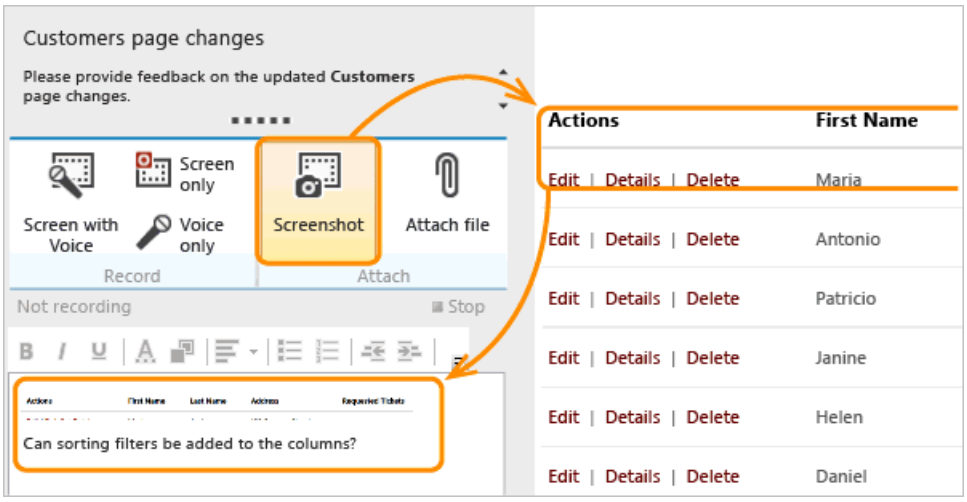
3. They launch the app to review from the feedback tool.



4. They begin providing feedback.



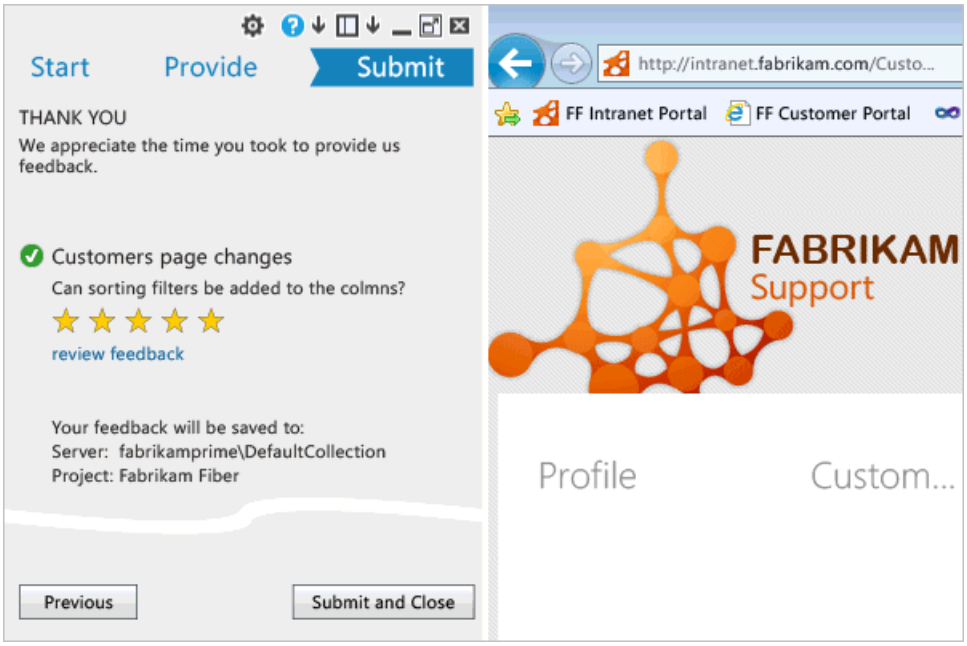
5. Reviewers can add screenshots, comments, and file attachments, and even record the feedback session. Results show up on the lower part of the screen. In this case, you can see the comment that the stakeholder wrote after attaching the screenshot.



NOTE

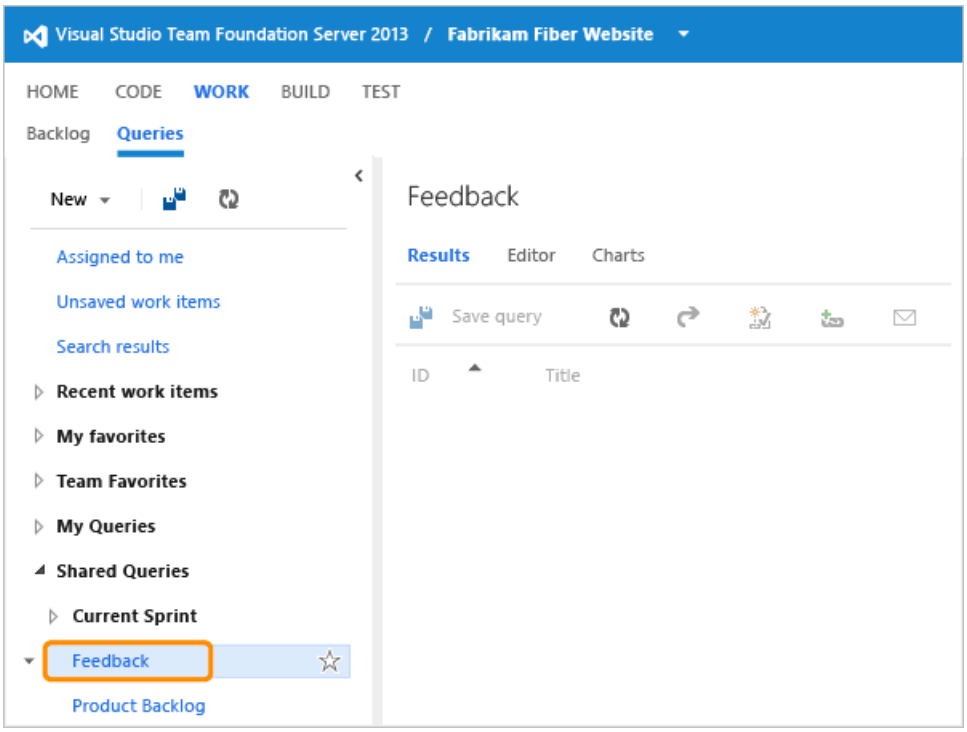
Security Note: Unless you stop recording, everything is recorded—all steps that you take as well as anything you say. If you provide sensitive data such as user names and passwords, you will capture this information in the recording. However, you can always delete a recording by deleting the image for the recording session that appears in the feedback tool's text box.

6. Reviewers can modify or even delete parts of their feedback, such as a recording, before they submit their feedback.

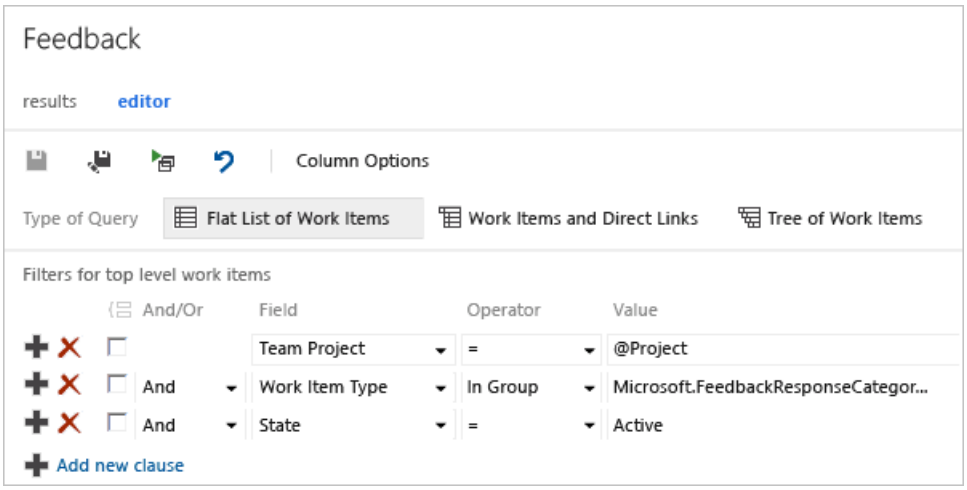


Review feedback

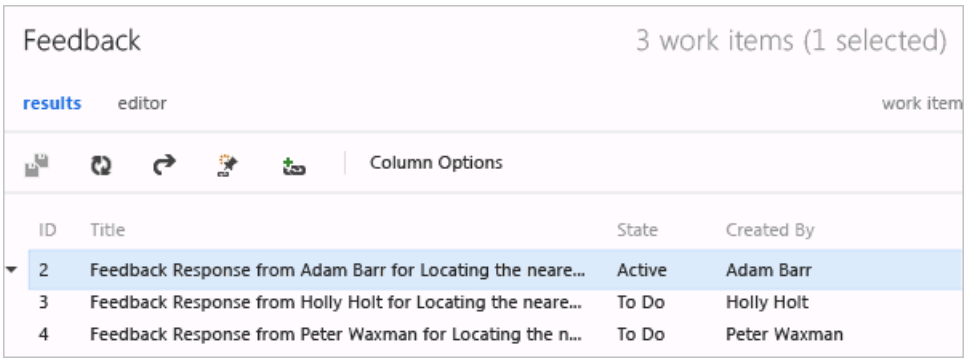
1. Open the Feedback query.



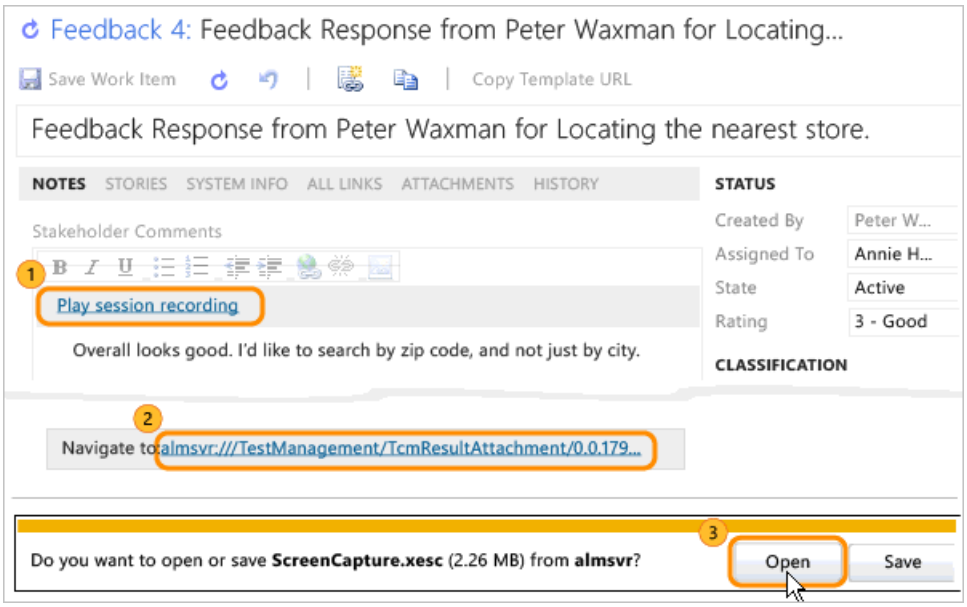
Or, create a feedback query with the parameters, as shown.



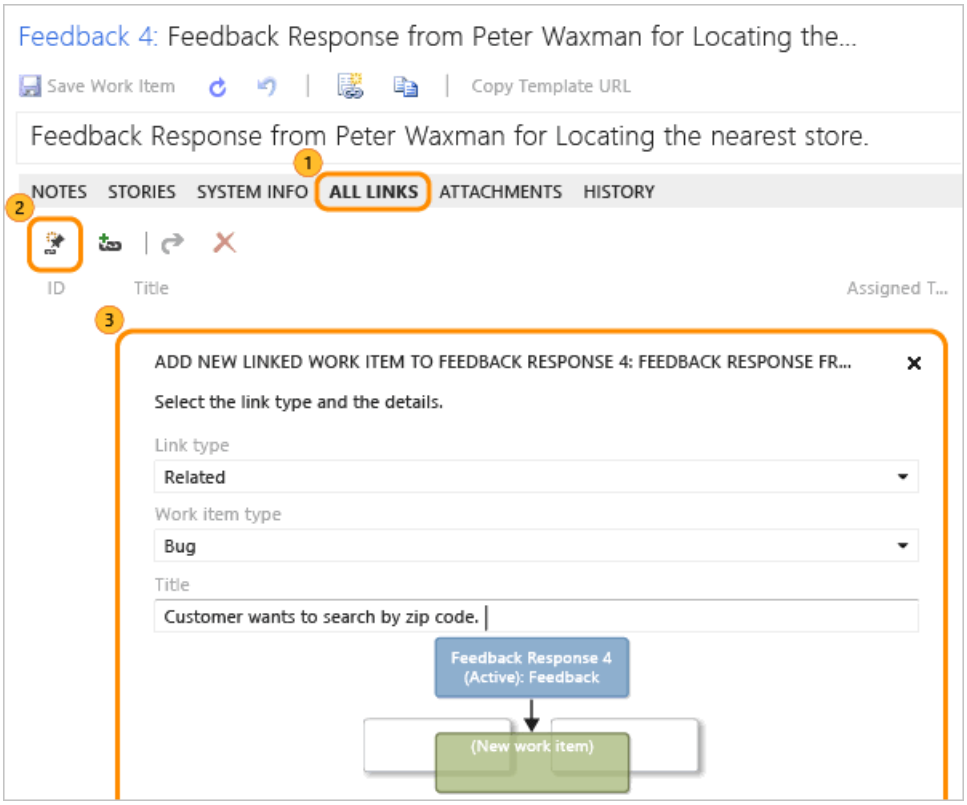
You should see a list of all active feedback responses for your team project.



2. Open a response item and play or save a recording.



3. Or, you can create a bug or backlog item linked to the feedback.



With the feedback experience, you can engage stakeholders frequently to provide continuous feedback. Interacting with your working apps, your stakeholders can record rich and actionable data that the system automatically stores in the form of video or audio recordings, comments, and annotated screenshots. You can then take action on each feedback response by assigning it to a team member or creating bugs or backlog items to the linked feedback.

Related notes

- You can change the audio device or annotation tool using the Settings icon change settings icon on the Microsoft Feedback Client.
- If you access the Microsoft Feedback Client from a remote machine, you can enable remote audio.
- You can install the Test & Feedback extension from the Marketplace, Test & Feedback.

Give feedback using Microsoft Feedback Client

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can respond to a request for feedback using the Microsoft Feedback Client. This tool allows you to launch an application, capture your interaction with it as video and capture your verbal or type-written comments as well. To support traceability, your feedback is stored in the data store for Azure DevOps Services or an on-premises Team Foundation Server (TFS).

You can install the install the Test & Feedback extension from the Marketplace, Test & Feedback.

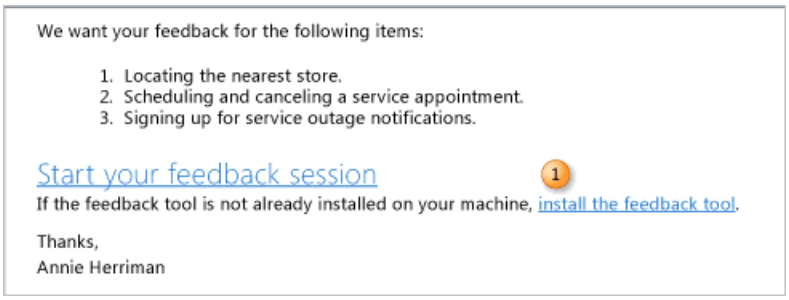
Initiate a feedback session

When you receive an email request for feedback, it contains a link to launch Microsoft Feedback Client, and a link to install the feedback tool if it is not already installed.

To install Microsoft Feedback Client

You can skip this procedure if you already have Microsoft Feedback Client installed or if you will access the feedback tool on a remote machine.

1. Open your email request, and then choose the Install the feedback tool link to install Microsoft Feedback Client.



   If you are viewing the email from an internet-based email client, you might need to copy the address behind the link and paste it into the address bar of your browser.

2. Restart your computer.

To initiate a feedback session from an email request

1. Open your email request, and then choose the **Start your local feedback session** link

   > **NOTE**
   > If you are viewing the email from an internet-based email client, you might need to copy the address behind the link and paste it into the address bar of your browser.

   If you're accessing the feedback tool on a remote computer, open the shortcut menu for the **Start your local feedback session** link, and then choose **Copy Hyperlink**. On the remote computer, choose **Start**, choose **Run**, paste the hyperlink into the box, and then choose the **OK** button.

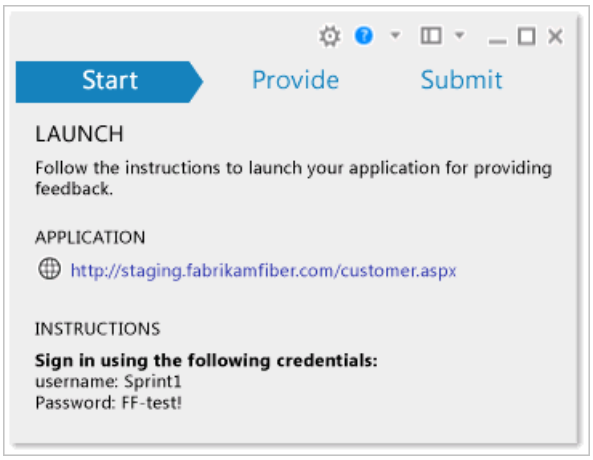
   The Feedback Client launches and the Windows Security dialog box appears.

2. Enter the credentials provided to you for connecting with Team Foundation Server, and then choose the

OK button.

Launch the application

The feedback tool opens to the Start page. Launch the application by following the instructions provided.



1. From the Start page, choose the **Application** link to open, start, or install the application for which you have been requested to provide feedback.
2. Follow any additional instructions provided to sign in or access the application.

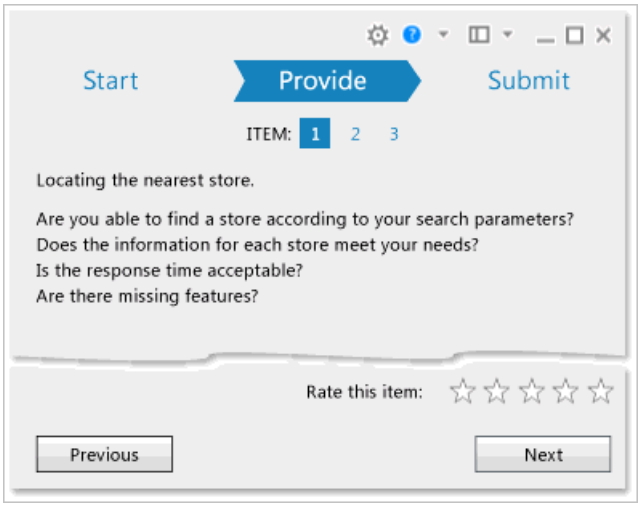
   For example, you would choose the `http://staging.fabrikamfiber.com/customer.aspx` link and then enter the user name and password provided.

Provide feedback on requested items

You can record your interactions with the application, record verbal comments, enter text, clip and annotate a screen, or attach a file. Recordings appear as images within the feedback tool's text box.

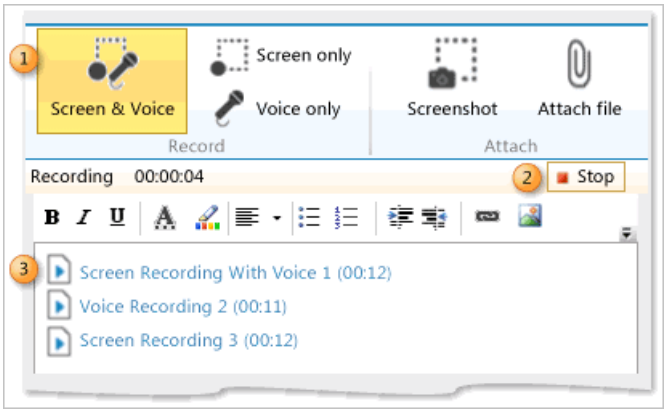
To review descriptions of items submitted for feedback

On the **Provide** page, one or more items appear for you to provide feedback. For each item, you can get context on what's being asked and then you can give feedback free form through video or audio recordings, text, screenshot, or file attachments. When finished with one item, choose the **Next** button to move to the next item.



1. On the toolbar, choose the number of the ITEM that you want to review.
2. Review the information provided.
3. Interact with the application, and determine what feedback you want to provide, based on the instructions for each item's request.

To start, stop, or delete a screen or voice recording



You can change settings defined for the audio device and annotation tool at any time. For more information, see Change the audio device or annotation tool.

1. **To start recording:** Choose one of the icons: **Screen & Voice**, **Screen only**, or **Voice only**.

   > **IMPORTANT**
   >
   > **Security Note:** Unless you stop recording, all steps that you take and remarks that you make while recording screen and voice will be recorded. If you provide sensitive data such as user names and passwords, you will capture this information in the recording. However, you can always delete a recording by deleting the image for the recording session that appears in the feedback tool's text box.

2. **To stop recording**: Choose the **Stop** button.

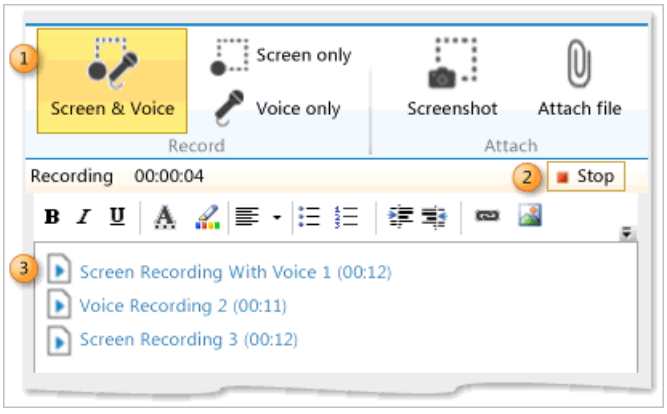
   The recording stops, and an image appears in the text box that indicates the type of recording and the length of the recording session.

3. **To delete a recording**: Choose the image for the recording that you want to delete, and then press the **Delete** key.

To add text, capture a screenshot, or attach a file

You can add text, capture a screenshot, annotate a screenshot, or attach a file as part of your feedback. You can perform these operations while you continue to record your feedback.

By annotating screenshots, you can indicate corrections or improvements by adding text or images to the screenshot that you captured. By default, Microsoft Paint opens automatically when you open a screenshot image that you captured within the feedback tool. You can also configure another annotation tool to open automatically whenever you capture a screenshot. For more information, see Change the audio device or annotation tool.



1. **To add text**: In the tool's rich-text box, enter the information that you want to include when you submit your feedback.

You can use the toolbar to format your comment, add a hyperlink, and insert an image.

2. **To capture a screenshot**: Choose the camera icon, and then select the rectangular area that you want to capture.

   The image appears in the rich-text box.

3. **To annotate the screenshot**: Double-click the image.

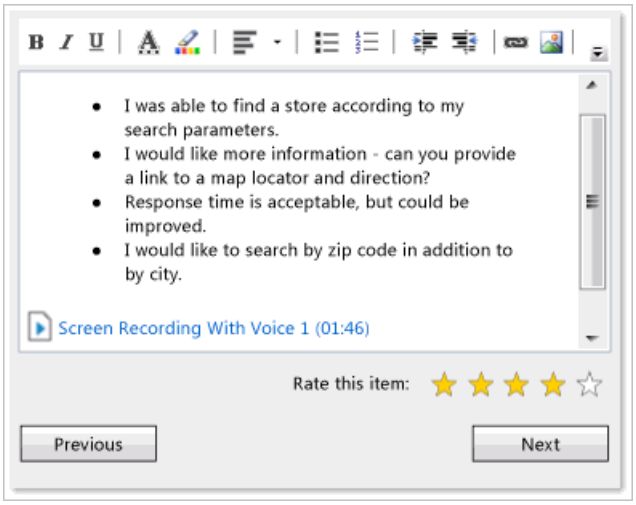
   Microsoft Paint, or whichever annotation tool you have configured, opens with the image displayed. To specify corrections or improvements, perform the following steps:

   a. Draw or enter text onto the image.
   b. Choose the **Save** icon, and close the annotation tool.

   The image appears within the feedback tool with the changes that you made.

4. **To attach a file**: Choose the attachment icon, and in the **Open** dialog box, browse to the file that you want to attach, and then choose the **Open** button.

To rate an item and move to the next item

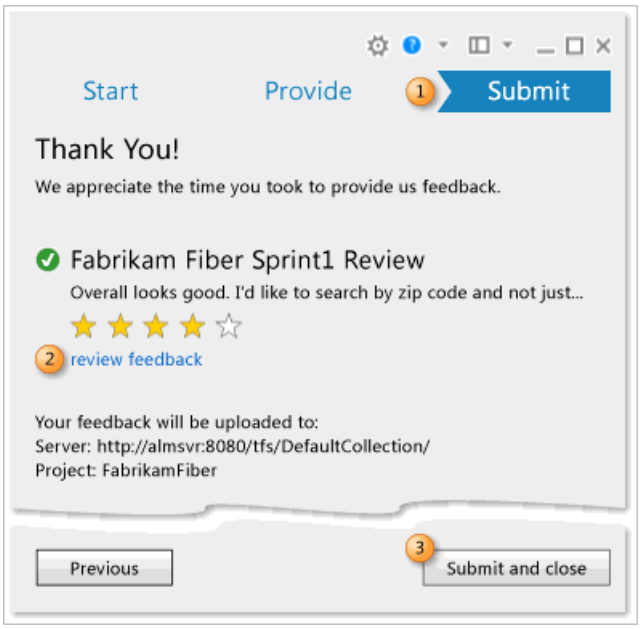


1. (Optional) Choose one or more stars to indicate your overall rating of the item that you have just reviewed.

2. Choose the **Next** button to move to the next item to review.

Review and submit your feedback

After you enter your feedback for each item, you can review, make corrections or additions, and then submit your feedback.



1. Choose the **Submit** button, which also stops any recordings that you may have started.
2. (Optional) On the **Submit** page, choose the **review feedback** link to return to the **Provide** page for the first feedback item. Review your feedback for each item and choose the **Next** button to advance to the next item.

   You can delete or modify any content within the text box for any item.
3. On the **Submit** page, choose the **Submit and close** button.

   Microsoft Feedback Client closes, and the system creates a feedback response work item which contains the feedback that you submit with the title, "Feedback Response from *YourUserName* for *RequestFeedbackItemTitle*."

Related notes

- You Initiate a feedback request from the home page of your web portal or using the Other links widget from a team dashboard.
- You can change the audio device or annotation tool using the Settings icon change settings icon on the Microsoft Feedback Client.
- If you access the Microsoft Feedback Client from a remote machine, you can enable remote audio.
- To record audio, you must have an audio recording device configured on your computer. If you access the feedback tool from a remote device, you might have to enable remote audio capture.

Requirements to provide feedback

- To provide feedback using the Microsoft Feedback Client you must have security credentials that will allow you to connect to Azure DevOps Services or an on-premises TFS and have the permissions described in Give reviewers permissions to provide feedback.
- Your computer must meet the system requirements for installing Microsoft Feedback Client. For more information, see the following page on the Microsoft website: Microsoft Feedback Client Download.

Track stakeholder feedback

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

All feedback is captured in a *Feedback Response* work item. You can track feedback, whether captured by the Test & Feedback extension or the Microsoft Feedback client, through a work item query.

Prerequisites

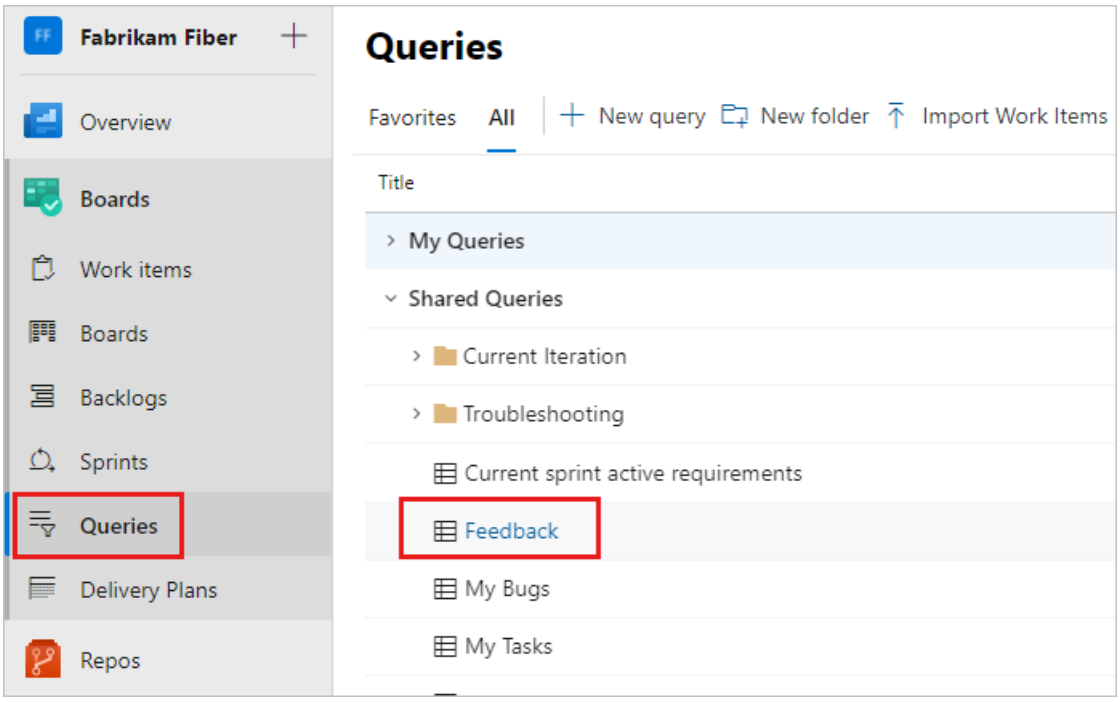
- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

> **NOTE**
> Users with **Stakeholder** access for a public project have full access to query features just like users with **Basic** access. For details, see Stakeholder access quick reference.

- By default, all project members and users with **Stakeholder** access can view and run all shared queries. You can change the permissions set for a shared query folder or shared query. For details, see Set query permissions.
- To add and save a query under **Shared queries**, you must be granted **Basic** access or higher. Also, you must have your **Contribute** permission set to **Allow** for the folder you want to add the query to. By default, the **Contributors** group doesn't have this permission.

Track feedback requests

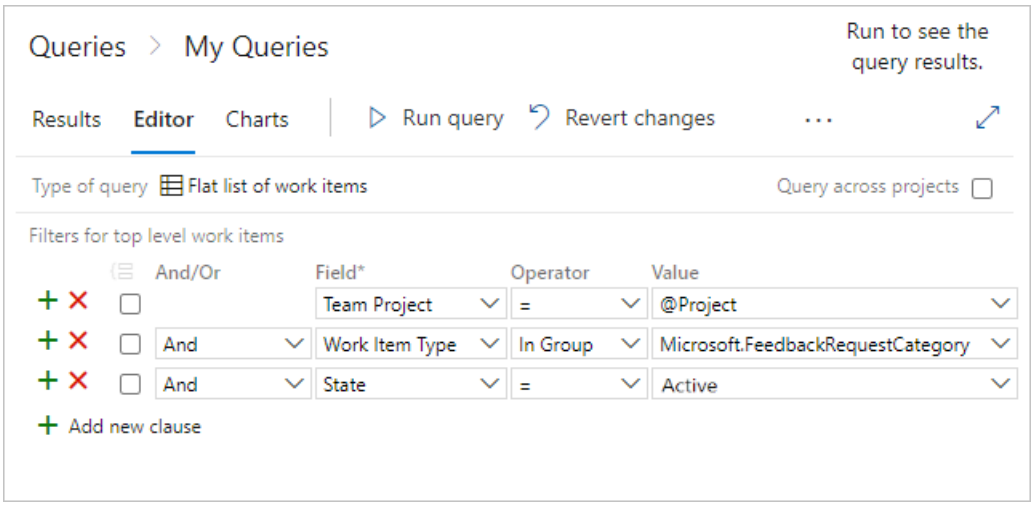
To view feedback, use the **Feedback** shared query. Select your project and open **Boards** > **Queries**. Under **Queries**, select **All**. In the Shared Queries, select **Feedback**.



This query displays a list of all the feedback responses received. To learn more, see Web portal navigation.

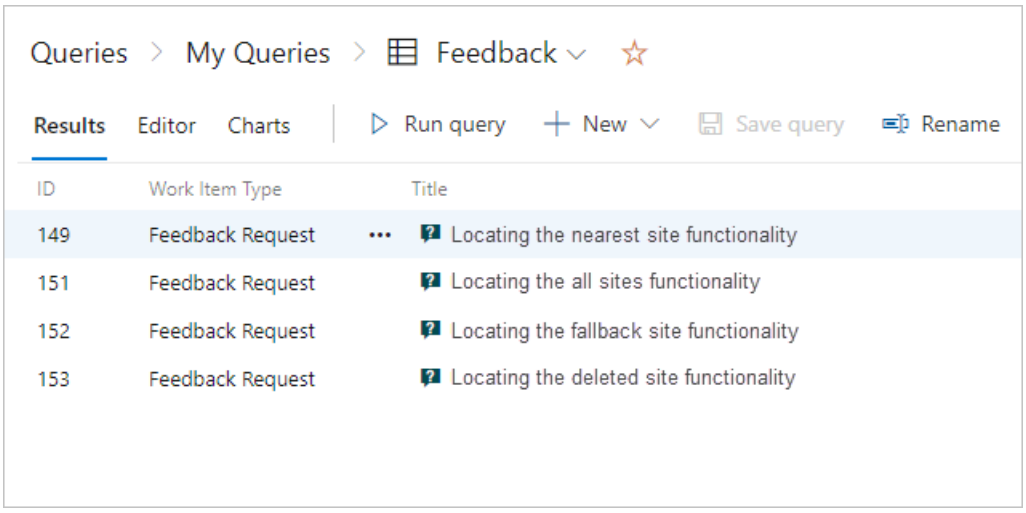
To create a feedback query, follow these steps:

1. Select **Boards** > **Queries** and then select **New query**.
2. In the **Editor** for your new query, enter the following values:



- Team Project = @Project
- Work Item Type In Group Microsoft.FeedbackRequestCategory
- State = Active

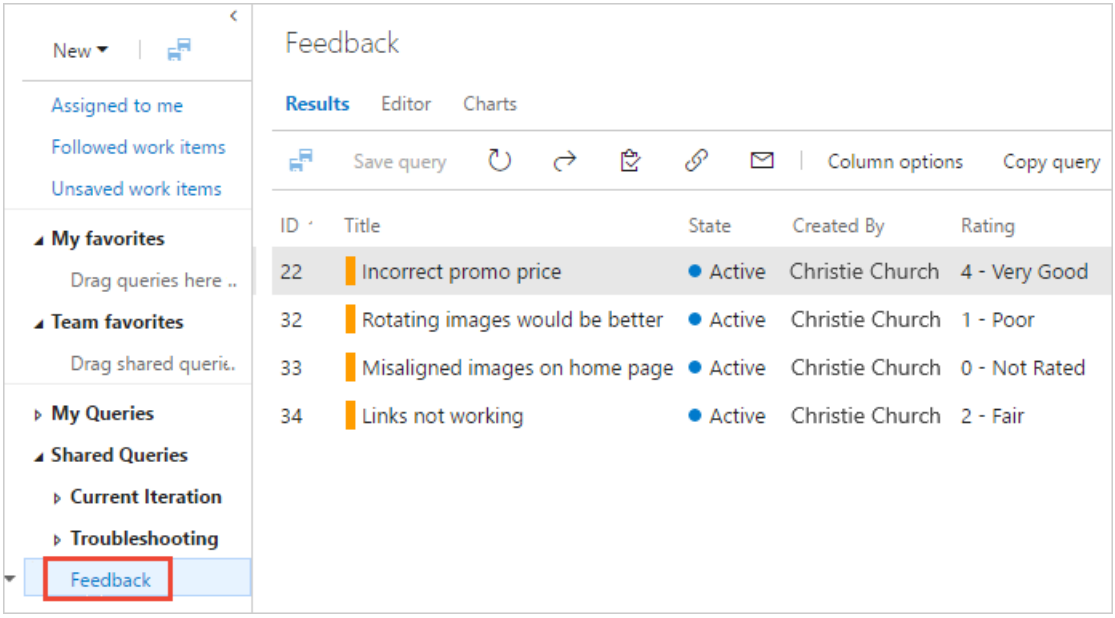
3. Select **Save query** and enter a name.
4. Select **Run query** to see a list of active feedback responses for your team project.



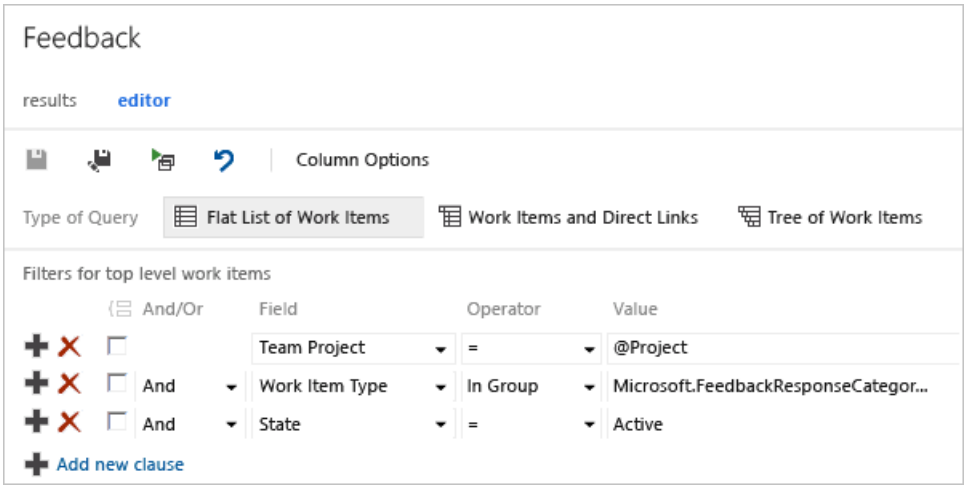
5. Select a response work item to see the details of the feedback.

1. Select your project and open **Boards>Queries** or **Work>Queries**. To learn how, see Web portal navigation.

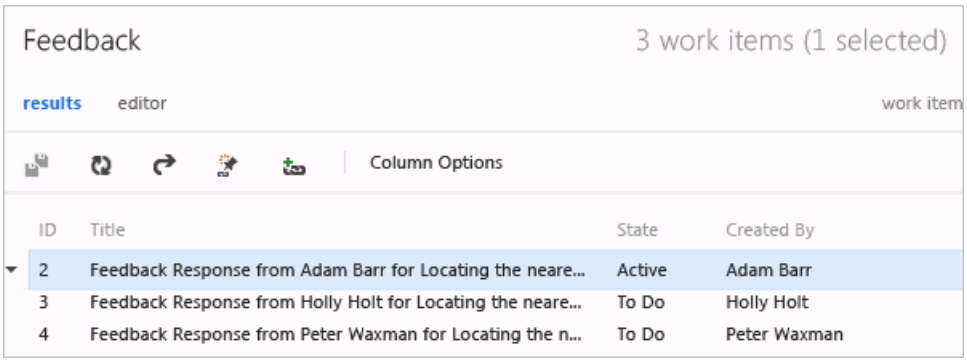
2. In the list of shared queries, select **Feedback**. This query displays a list of all the feedback responses received.



Or, create a feedback query with the parameters, as shown.



3. You should see a list of all active feedback responses for your team project.



4. Open the response work item to see the details of the feedback.

Related articles

- What is Azure Test Plans?
- Request feedback using the Test & Feedback extension
- Get feedback
- Provide feedback using the Test & Feedback extension
- Define a work item query

Give reviewers permissions to provide feedback

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You must grant permissions to provide feedback to users that you plan to request feedback from. Reviewers who aren't members of your team require special permissions to provide feedback using the Microsoft Feedback Client.

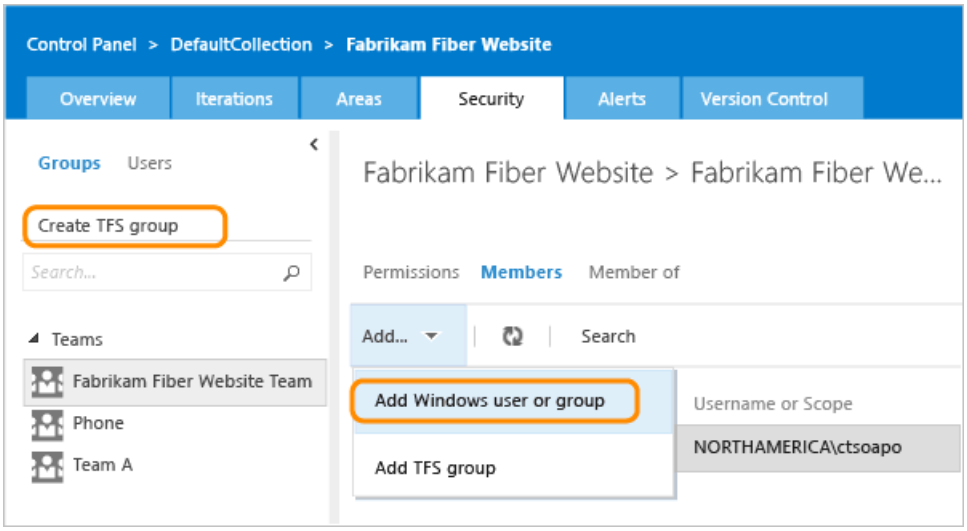
Add reviewers to your team project

1. From the web portal of your team project home page, open the administration context.



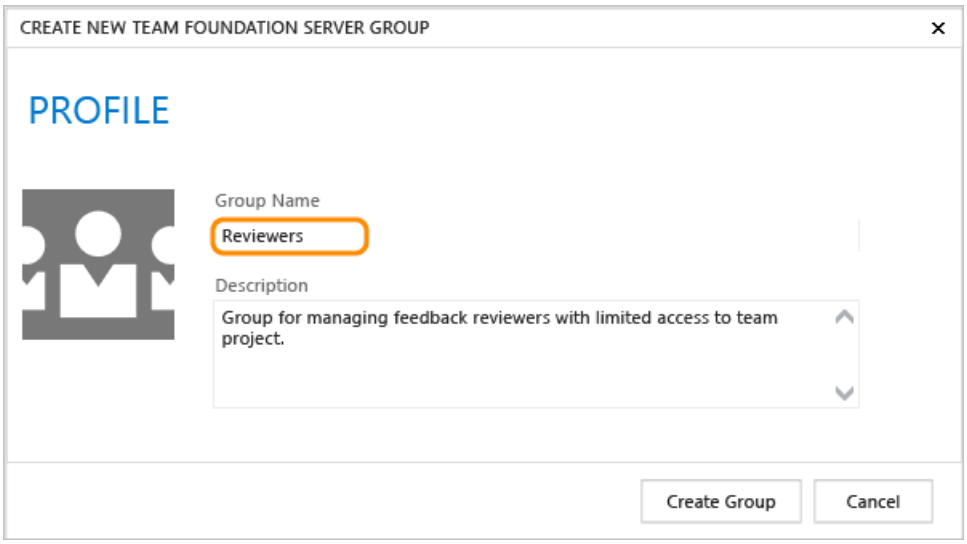
   If you aren't a member of the **Project Administrators** group, get added. See Change project-level permissions. You need to be a member in order to add users and groups to a team project, change permissions, and grant them access to the web portal.

2. Create a group for your reviewers.

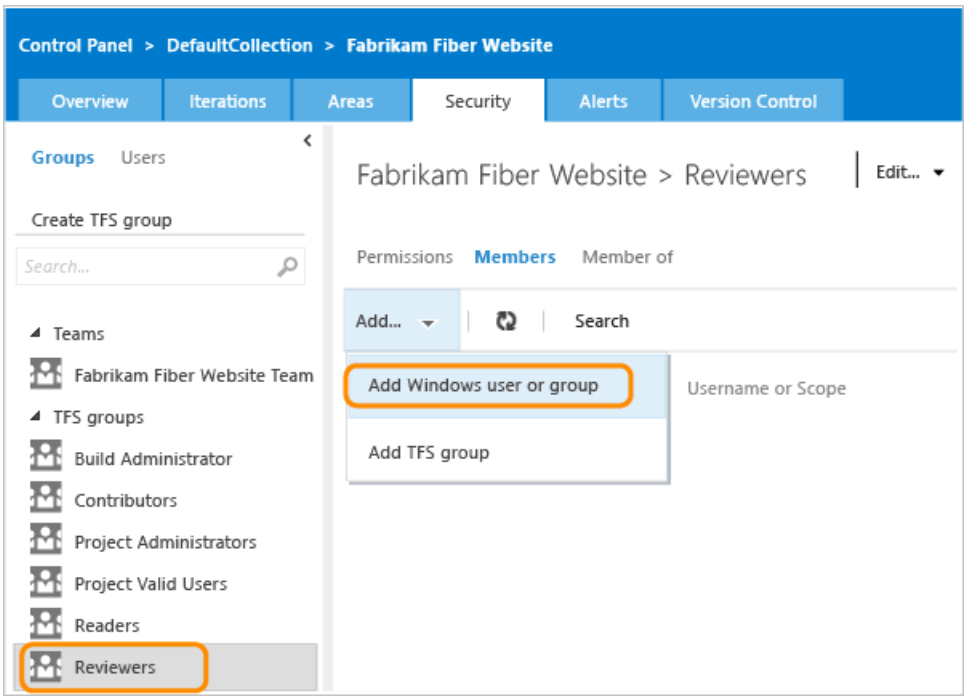


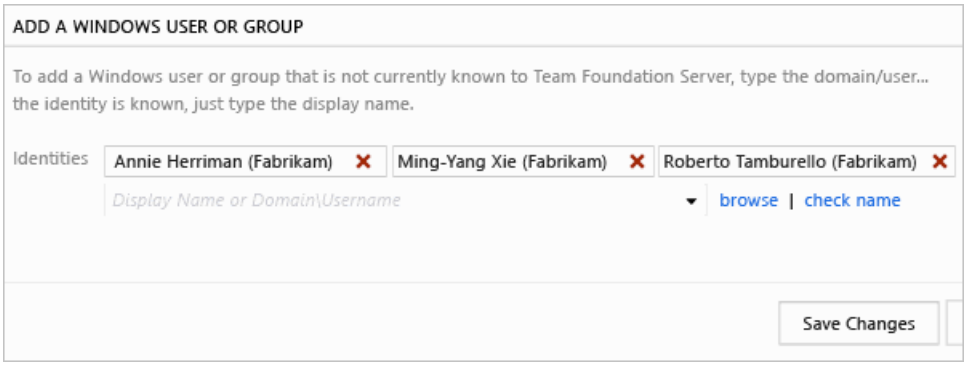
> **TIP**
> If you have a lot of reviewers, create an Azure Devops security group to help you manage permissions more efficiently. To learn how, see Add or remove users or groups, manage security groups.

3. Name your group.



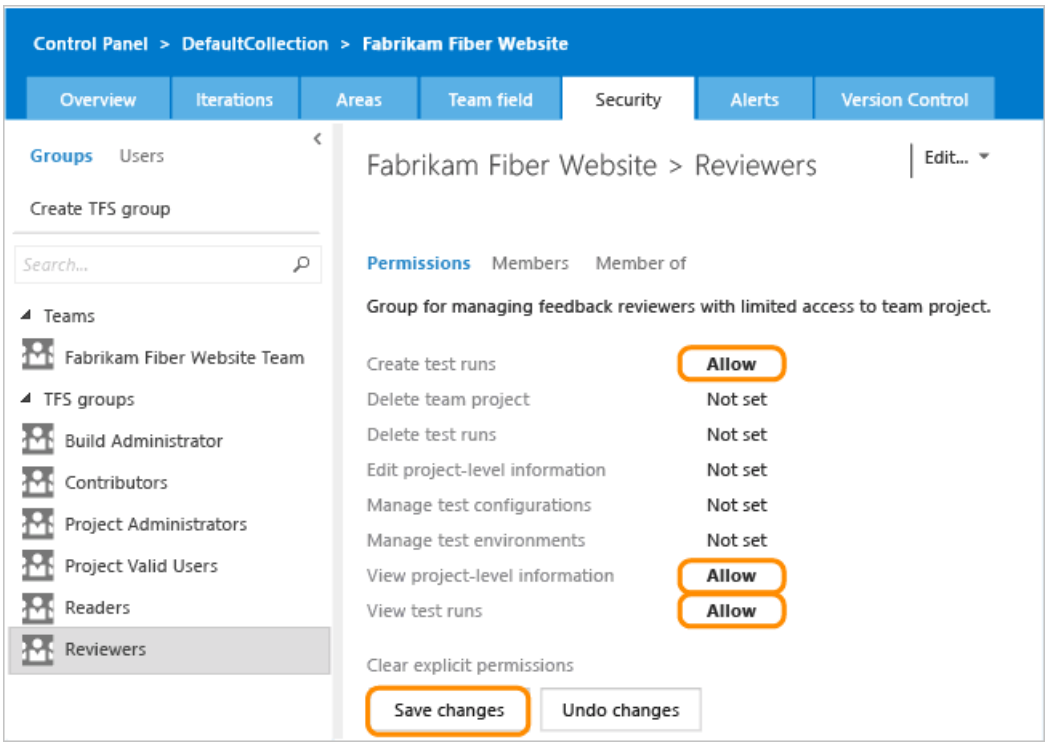
4. Add accounts to your group.





Set permissions so reviewers can provide feedback

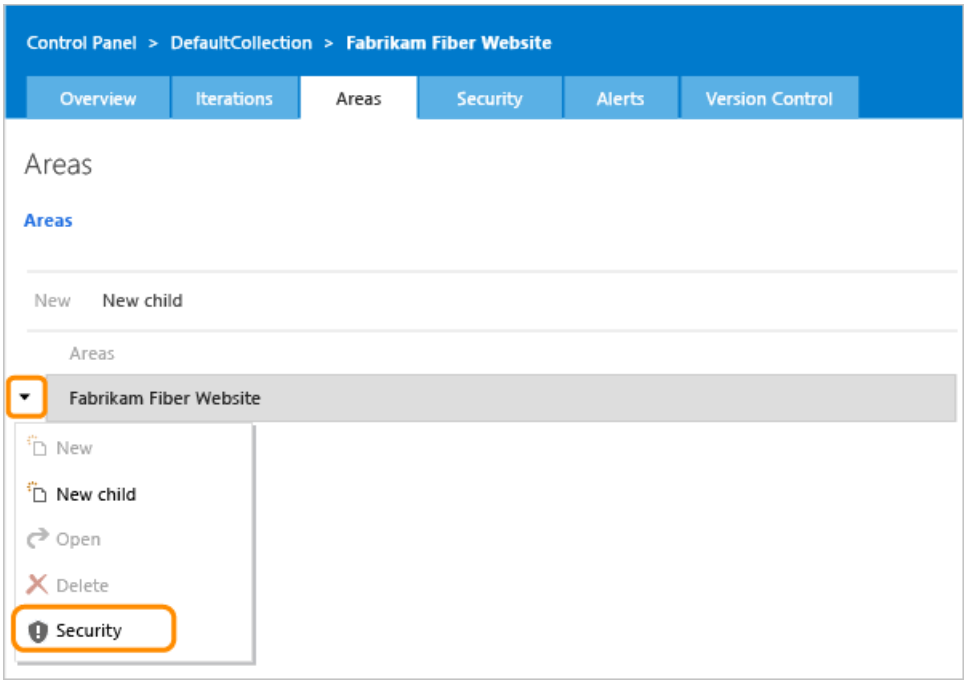
Allow reviewers to **Create test runs**, **View project-level information**, and **View test runs**.



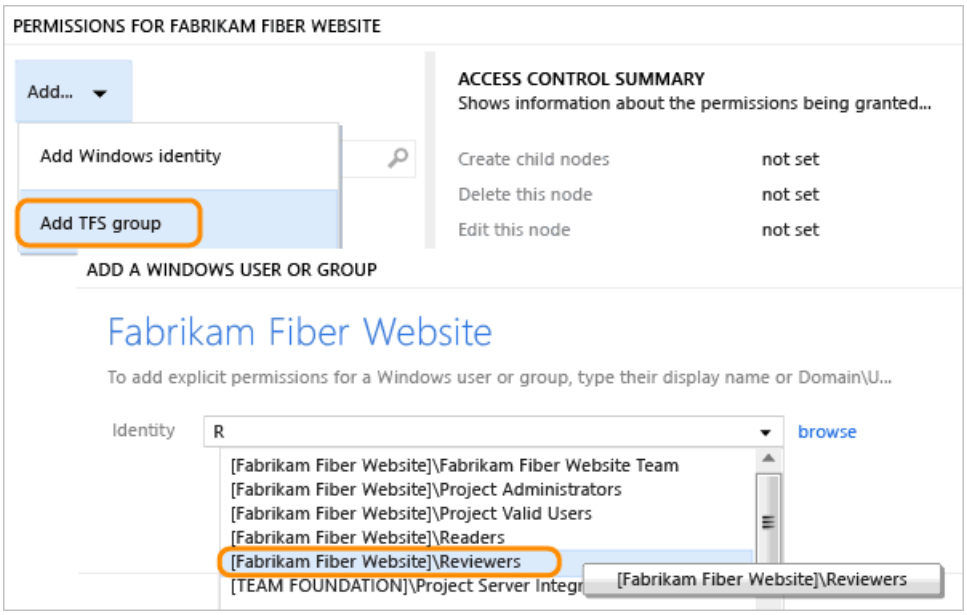
Set permissions so reviewers can modify work items

Since feedback is captured in a feedback response work item, reviewers need to be able to modify work items in the product areas they will review.

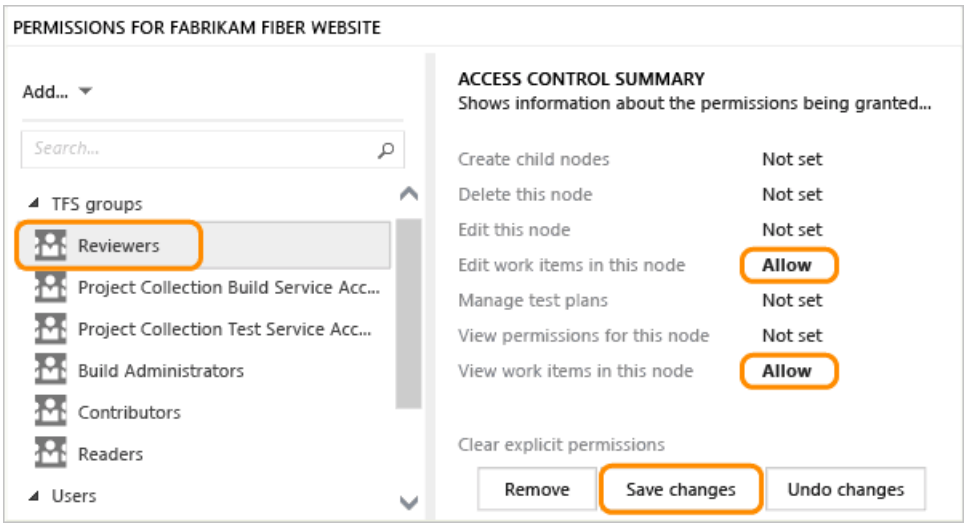
1. Open security for the team project.



2. Add the reviews group to the **VSO Groups** or **TFS Groups**.



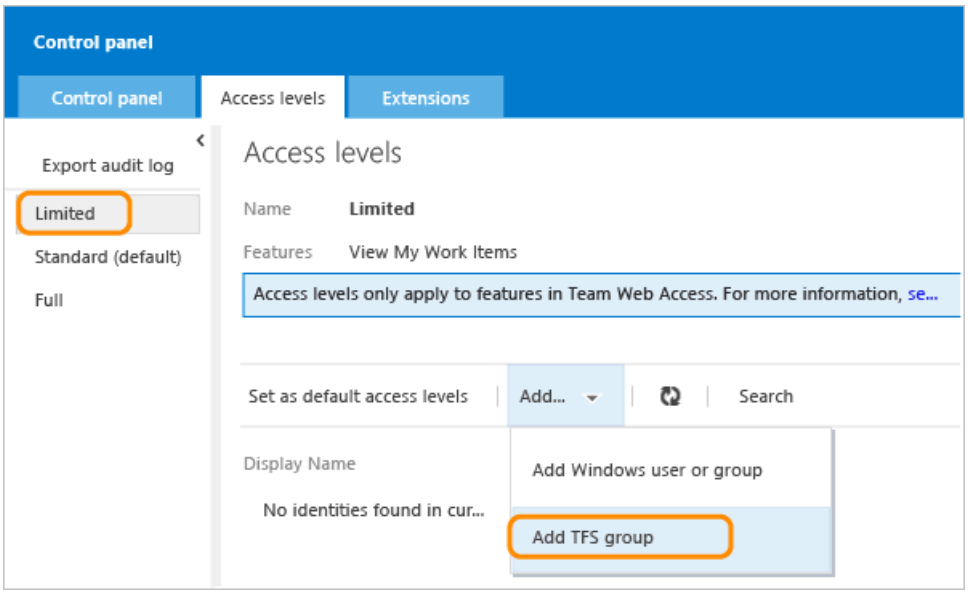
3. Allow reviewers to Edit work items in this node and View work items in this node.



If you want, allow reviewers the ability to modify their feedback submissions

Sometimes additional ideas occur after reviewers submit their feedback. By providing access to the web portal, reviewers can revisit and further annotate their feedback submissions.

- **Azure DevOps Services:** Assign the **Stakeholder** license to accounts that you add to your Reviewer group
- **On-premises Azure DevOps Server:** Add your Reviewer group to the **Stakeholder** group on the **access levels** page. If you don't see this tab, get administrative permissions.



Your reviewers will be able to view and modify only those work items that they create, which includes feedback responses. The Stakeholder group provides limited access to features and data for those members of your organization who do not have an Azure DevOps client access license (CAL).

Related articles

- Initiate a feedback request
- Respond to a feedback request
- Work as a stakeholder

Enable remote audio capture

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To record audio, you must have an audio recording device configured on your computer, or on a remote machine if you access Microsoft Feedback Client, Test Runner, or Exploratory Testing from a remote device.

To record audio as part of a feedback or testing session on a remote machine that is running Microsoft Feedback Client, Test Runner, or Exploratory Testing window, you must configure audio redirection settings on the remote machine. You must also enable record from this computer when you connect to the remote machine.

> **IMPORTANT**
> We recommend that you open Microsoft Feedback Client on your local computer. Redirection will decrease audio quality. Microsoft Feedback Client does not require you to record audio as part of the feedback session.

To set a default microphone on your local computer

To record audio from your local computer, you must set a microphone device as the default device.

1. Choose **Start, Control Panel**.
2. In the Control Panel window, choose the **Hardware and Sound** link, and then choose **Sound**.
3. In the Sound window, choose the **Recording** tab.
4. Select a recording device from the list provided and then choose **Set Default**.

To enable remote audio capture

> **IMPORTANT**
> If you edit the registry incorrectly, you might severely damage your system. Before you make any changes to the registry, you should back up any valued data on the computer.

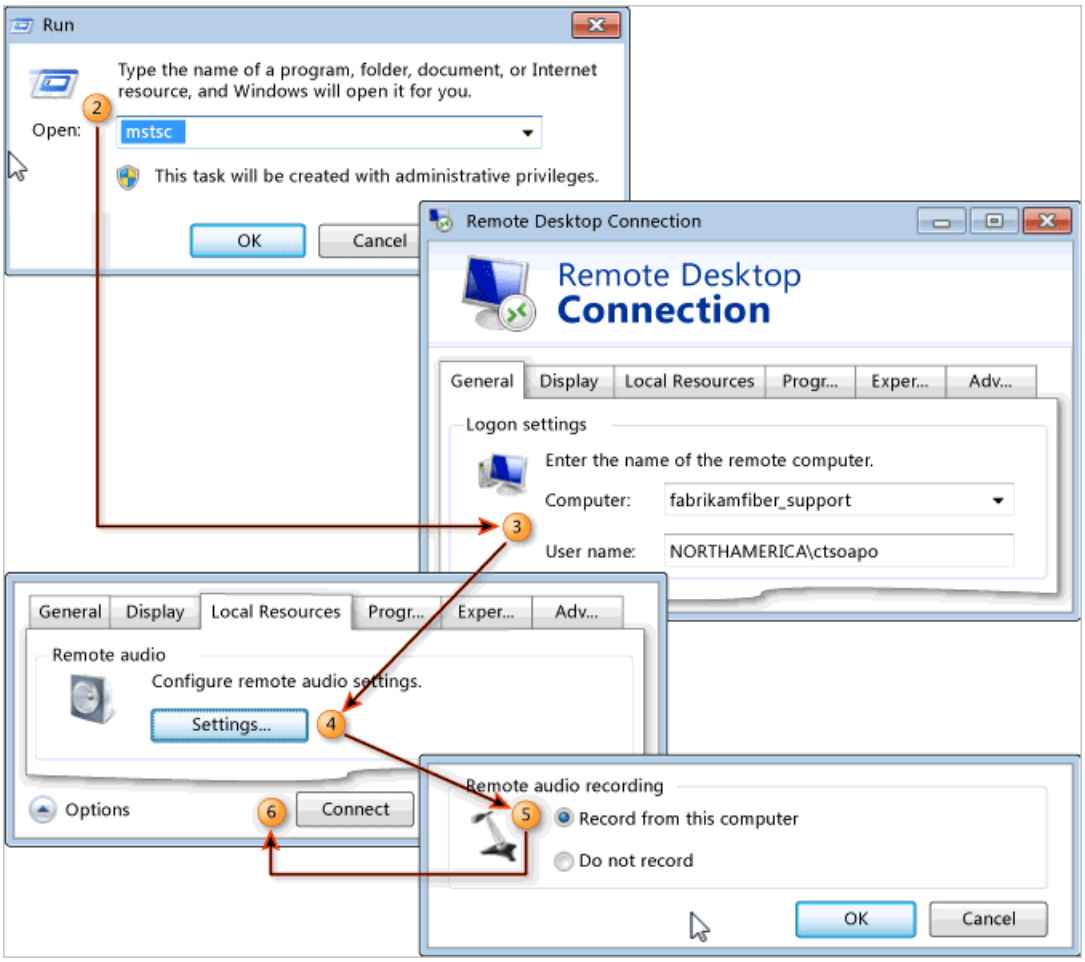
1. Choose **Start**, choose **Run**, enter **regedit**, choose the **OK** button, and then set the value of the following registry key to **0**.

   ```
   HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\fDisableAudioCapture
   ```

   For more information about how to set values of registry keys, see the following page on the Microsoft website: How to add, modify, or delete registry subkeys and values by using a registration entries (.reg) file.

2. To connect to the remote machine, choose **Start**, choose **Run**, enter `mstsc`, and then choose the **OK** button.

   The **Remote Desktop Connection** dialog box opens.



3. In the **Computer** list, choose or enter the name of the computer to which you want to connect, and, in the **User name** box, enter your user name.
4. On the **Local Resources** tab, choose the **Settings** button.
5. Under **Remote audio recording**, choose the **Record from this computer** option button, and then choose the **OK** button.
6. Choose the **Connect** button.
7. Exit and restart whichever client tool that you are using, for example Microsoft Feedback Client, Test Runner, or Exploratory Testing.

   You must restart you client in order for it to register the audio settings.

Related articles

- Exploratory testing
- Run your tests
- Provide feedback using the Microsoft Feedback Manager

Change the audio device or annotation tool

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can change the default settings used by Microsoft Feedback Client for the annotation tool or audio device. The annotation tool that you set automatically opens when you double-click an image within the feedback tool, and the audio device is used when you start a **Screen & Voice** or a **Voice only** recording. You might want to make a change if you do not have the default tools available on the computer that you use to provide feedback.

To change a setting, first launch Microsoft Feedback Client. For more information, see Provide feedback.

To change the annotation tool or annotation settings

1. On the toolbar at the top of Microsoft Feedback Client, choose the **Change settings** button.
2. In the **Change settings** dialog box, choose the **Browse** button.
3. In the **Open** dialog box, navigate to the annotation tool and executable file that you want, and then choose the **Open** button.
4. (Optional) Enter any arguments that can be passed to the annotation tool.

   For example, **%f** represents the file name to pass to the tool.
5. (Optional) Select the **After taking screenshot, immediately open in the specified annotation tool** check box to enable the annotation tool to open automatic.
6. Choose the **Save** button.

To change the audio device

1. On the toolbar at the bottom of Microsoft Feedback Client, choose the **Change settings** icon.
2. In the **Change settings** dialog box, choose the **Change** button to change the default audio device.
3. In the **Open** dialog box, navigate to the audio device that you want, select the **Default audio device** check box, and then choose the **Open** button.
4. Choose the **Save** button.

Related articles

- Provide feedback

Connect Azure Boards to an Office client

10/4/2022 • 13 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To support your work tracking efforts, you can use Microsoft Excel. You can either work in online mode, where you're connected to either Azure Boards or Azure DevOps Server. Or, work in offline mode, where you access the local computer and document.

To support your work tracking efforts, use Microsoft Excel and Microsoft Project. You can either work in online mode, where you're connected to Azure Boards or Team Foundation Server (TFS). Or, work in offline mode, where you access the local computer and document.

> **IMPORTANT**
> All Office integration tasks require that you have installed a version of Visual Studio or the free Azure DevOps Office Integration 2019.

In this article you'll learn about the following items:

- Which Office clients support connection to Azure DevOps
- Prerequisites for connecting from an Office client to Azure DevOps
- Importance of publishing and refreshing work from an Office client
- How to connect Excel to an Azure Boards project
- How to work offline and reconnect to Azure DevOps
- Marketplace extensions that support integration with Office clients

Supported Office clients and Azure DevOps versions

The following table indicates the Office clients supported for each Azure DevOps version starting with Team Foundation Server 2013 (TFS 2013).

> **NOTE**
> macOS isn't supported. Even if you've installed Visual Studio for Mac, connection to Azure DevOps from Excel or any other Office client isn't supported.

Azure DevOps/Visual Studio version	Excel	Project[1]	PowerPoint Storyboarding[2]
Azure DevOps Services			
Azure DevOps Server 2020			
Azure DevOps Server 2019			
Visual Studio 2019			
Azure DevOps Office Integration 2019			

✔

TFS 2018
Visual Studio 2017

✔

✔

✔

Notes:

1. Support for Project integration and the **TFSFieldMapping** command is deprecated for Azure DevOps Server 2019 and later versions. You might find support using one of the Marketplace extensions.
2. Support for linking PowerPoint files to work items from within PowerPoint has been deprecated starting with Visual Studio 2019 and Azure DevOps Office Integration 2019. You can still link to PowerPoint using the Storyboard link from within a work item. Also, the Visual Studio Gallery for PowerPoint Storyboarding has been deprecated.

Prerequisites

Connection from an Office client to an Azure Boards project requires that you've installed the necessary software and have the necessary permissions.

- To connect Excel to Azure Boards, you must have installed Office Excel 2010 or later version, including Office Excel 365.
- Installed Azure DevOps Office Integration 2019 (free).

  > **NOTE**
  > The only way to get the Azure DevOps Office Integration plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Office Integration installer. The plug-in supports connection to Azure Boards and Azure DevOps Server from Excel.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To connect Excel to Azure Boards, you must have installed Office Excel 2010 or later version, including Office Excel 365.
- Installed Azure DevOps Office Integration 2019 (free).

  > **NOTE**
  > The only way to get the Team Foundation plug-in is by installing one of the latest editions of Visual Studio or the TFS Standalone Office Integration installer. The TFS Office Integration 2017 plug-in supports connection to Azure Boards and TFS from Excel, Project, and the PowerPoint-based storyboarding tool.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To connect Excel to Azure Boards, you must have installed Office Excel 2010 or later version, including Office Excel 365.
- To connect Project to Azure Boards, you must have installed Office Project 2010 or later version, including Office Project 365.

- To connect PowerPoint to Azure Boards, you must have installed Office PowerPoint 2010 or later version installed. If you're interfacing with TFS 2013 or TFS 2015, you must have Office PowerPoint 2007 or later version installed.
- Installed Visual Studio 2013 or later version or Team Foundation Server Standalone Office Integration (free)

> **NOTE**
> The only way to get the Team Foundation plug-in is by installing one of the latest editions of Visual Studio or the TFS Standalone Office Integration installer. The TFS Office Integration 2017 plug-in supports connection to Azure Boards and TFS from Excel, Project, and the PowerPoint-based storyboarding tool.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.

To learn more about compatibility requirements, see Compatibility with Azure DevOps.

- Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Visual Studio 2013 or later version or Team Foundation Server Standalone Office Integration (free)
- Permissions to connect to the project in Azure Boards.

- Office Excel 2010 or later version, including Microsoft Office Excel 365
- Office Project 2010 or later version, including Office Project 365
- Visual Studio 2013 or later version or Team Foundation Server Standalone Office Integration (free)
- Permissions to connect to the project in Azure Boards or TFS.

To learn more about compatibility requirements, see Azure DevOps client compatibility.

> **IMPORTANT**
> You may receive the following error if you install Microsoft Office 2010 on the same computer as a previous version of Office.
>
> **Team Foundation Error**
>
> **Interface not registered (Exception from HRESULT: 0x80040155)**
>
> You may be able to resolve this error by repairing Office. You can access the Repair option by opening the **Control Panel**, choose **Uninstall or change a program**, open the context menu for Office 2010, and then choose **Change**. See also, Azure DevOps-Office integration issues.

Publish and refresh work items

When you add or update work items from Excel, local copies of your work items are created. To keep data in sync, it's important to refresh your local file when you open it and publish and refresh frequently during a long online session.

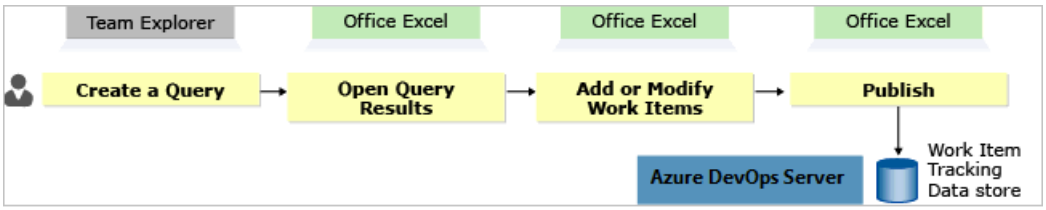
When you add or update work items from Excel or Project, local copies of your work items are created. To keep data in sync, it's important to refresh your local file when you open it and publish and refresh frequently during a long online session.

At first, the data in the local document matches the data in the database. But you or other team members can change the data about work items and cause the two to differ. To view the most recent changes from the database, refresh the document. The refresh downloads the latest values in the data store to your local document. To write changes from the document to the database, publish the changes. Publishing uploads the

changes you made to the work item tracking data store.

To keep work items in sync from your local data store and Azure Boards, publish and refresh often.

Azure DevOps and Excel



To work in Excel, see Bulk add work items with Excel.

Connect an Azure DevOps project to Excel

> **NOTE**
> While this section illustrates how to connect Excel to an Azure Boards project, the steps for connecting from Project or PowerPoint are similar.

To add or modify work items by using Excel, you connect your worksheet to a project. Establishing this connection binds the document to the Azure DevOps project to exchange information.

> **NOTE**
> When you connect to Azure Boards in the cloud, the **Team Project Collection** is automatically selected as there is only one collection associated with your Azure DevOps Services organization. When you connect to Azure Boards in an on-premises server, you choose the **Team Project Collection** prior to choosing the project.

You can start work from the web portal, Excel, or Visual Studio/Team Explorer. Your worksheet is associated with either a list of work items or a work item query.

- Open query in Excel (web portal)
- Connect from client to Azure Boards
- Connect client to Azure DevOps on-premises
- Open Excel from Visual Studio

This connection method requires that you have installed Azure DevOps Open in Excel. It also requires Visual Studio 2017 or later version.

1. From your web browser, (1) check that you've selected the right project, (2) choose **Boards>Queries**, and then (3) choose **All**.

   ![Open Boards

   Queries, vertical nav](/azure/devops/boards/queries/media/view-run-queries/open-queries-vert.png)

2. Choose the query you want to open in Excel.
3. From the **Results** tab, choose the ··· actions icon

   ![QUery Results, context menu, Open in Excel Open Boards

   Queries, vertical nav](media/connect/open-in-excel-from-portal.png)

To learn more, see Bulk add work items with Excel.

> **TIP**
> You can use multiple worksheets within an Excel workbook to work with different input or query lists. However, you can only connect to one project per workbook.

If you move your Azure DevOps project to a different project collection in the same instance of Azure DevOps, your documents automatically reconnect. However, if the project moves to a different instance of Azure DevOps, you must manually reconnect your documents to the new server.

Work offline and reconnect to Azure Boards

One advantage of working in Excel or Project is that you can work offline and add or modify work items. The following procedures show you how to disconnect an Excel work item list or a Project plan from Azure Boards. You can then reconnect later to synchronize the document with the work item database.

> **NOTE**
> If the project that contains work items for your Excel or Project document is moved to a different organization or Azure DevOps Server instance, you must reconfigure the server to which the document connects. For more information, see Connect Azure DevOps project to Excel provided earlier in this article.

Disconnect a document file from the network

To disconnect an Excel or Project document file from the network:

1. Open the document that you want to change while you're offline.
2. Refresh the work item list to retrieve the latest information from the work item database.
   - If you're using Excel, on the **Team** ribbon, in the **Work Items** group, choose **Refresh**.
   - If you're using Project, on the **Team** menu, choose **Refresh**.
3. If you're using Excel, add to the work item list the columns for all fields that you want to modify.

   You can't add columns when the work item list is disconnected from the server.
4. Disconnect your computer from the network, or save the work item list file and copy it to another computer.

   An error message might appear that tells you the Office program couldn't establish a connection with an Azure DevOps Server.
5. Modify or update the work item list as needed.

   > **NOTE**
   > You can't create most types of links between work items when the work item document is disconnected from the system. The exceptions are parent-child links in an Excel tree list, and both parent-child and predecessor-successor links in a Project plan.

Reconnect a file to Azure Boards

To reconnect an Excel or Project document file:

1. Reconnect your computer to the network, or copy the file to a computer that is connected to Azure Boards.
2. If you changed the document while you were offline, follow one of these steps:

- If you're using Excel, on the **Team** ribbon, in the **Work Items** group, choose **Publish**.
- If you're using Project, on the **Team** menu, choose **Publish Changes**.

3. If you didn't change the document while you were offline, follow one of these steps:

   - If you're using Excel, on the **Team** ribbon, in the **Work Items** group, choose **Refresh**.
   - If you're using Project, on the **Team** menu, choose **Refresh**.

4. Resolve any data validation errors or conflicts that occur.

Marketplace extensions

The following Marketplace extensions support integration between Azure DevOps and Office products.

- Azure DevOps Open in Excel: Opens a selected query in Excel.
- Office 365 Integration: Pushes notification of configurable Azure DevOps events to an Office 365 Group.

For more extensions that integrate with Microsoft Project, see Azure Boards migration and integration, Project and portfolio management.

Related articles

- FAQs: Work in Excel connected to Azure Boards
- Resolve publishing errors
- Bulk add or modify work items with Excel
- Create your backlog
- Requirements and compatibility

Add or modify Azure Boards work items in bulk with Microsoft Excel

10/4/2022 • 28 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you need to add or modify many work items, using Microsoft Excel can save you time. Excel supports adding work items, updating existing work items, adding links and attachments to multiple work items, and more. You can also use native Excel features to support other actions, such as summing a column, copy-and-paste rows, fill down data into cells, and more.

In this article you'll learn how to complete the following tasks:

- Choose the type of list or query to support your task
- Use select Excel features when connected to Azure Boards
- Import or update work items, either a flat list or hierarchy tree list
- Publish and refresh your work items
- Convert a flat-list to a tree-list, change your list type or query
- Add work items to your worksheet
- Add and remove work item fields from your worksheet
- Select user accounts for identity or person-named fields
- Link work items, find work items to link to, edit links, and more
- Add attachments to one or more work items
- Open a work item from Excel to add additional information (opens in web portal)
- Edit Area and Iteration Paths (opens in web portal)

For information about connecting to Excel, see Connect Azure Boards to an Office client. For answers to specific questions about the integration of Excel and Azure DevOps, see FAQs: Work in Excel connected to Azure Boards .

> **NOTE**
> If you don't have access to Excel, you can still perform bulk import and update using CSV formatted files. To learn more, see Bulk import or update work items using CSV files.

Prerequisites

> **NOTE**
> macOS isn't supported. Even if you've installed Visual Studio for Mac, connection to Azure DevOps from Excel isn't supported.

- Installed Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Installed Azure DevOps Office Integration 2019 (free).

> **NOTE**
> The only way to get the Azure DevOps Office Integration plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Office Integration installer. The plug-in supports connection to Azure Boards and Azure DevOps Server from Excel.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To view or modify work items, you must have these permissions set to **Allow**: **View work items in this node** and **Edit work items in this node**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To use the Select User feature, you need to install Visual Studio (at least VS 2015.1 or later version or Team Foundation Server Office Integration 2015 Update 2 or later version. You can download the free version of Visual Studio Community. Get this feature to avoid data validation errors by misspelling user names and when you must assign user names from a large group of user accounts.

- Installed Microsoft Excel 2010 or later version, including Microsoft Office Excel 365
- Installed Azure DevOps Office Integration 2019 (free).

> **NOTE**
> The only way to get the plug-in is by installing one of the latest editions of Visual Studio or the Azure DevOps Standalone Office Integration installer. The Azure DevOps Office Integration 2019 plug-in supports connection to Azure Boards and Azure DevOps from Excel, Project, and the PowerPoint-based storyboarding tool.

- To connect to an Azure Boards project, you need to be a member of the project. If you don't have an Azure Boards project yet, you can create one.
- To view or modify work items, you must have these permissions set to **Allow**: **View work items in this node** and **Edit work items in this node**. By default, the **Contributors** group has this permission set. To learn more, see Set permissions and access for work tracking.
- To add or modify work items, you must be granted **Stakeholder** access or higher. For details, see Stakeholder access quick reference.
- To use the Select User feature, install Visual Studio (at least VS 2015.1 or later version or Team Foundation Server Office Integration 2015 Update 2 or later version. You can download the free version of Visual Studio Community. Get this feature to avoid data validation errors by misspelling user names and when you must assign user names from a large group of user accounts.

To learn more about compatibility requirements, see Compatibility with Azure DevOps Server.

Choose list and query type

When you work in Excel connected to Azure Boards, you're always working with a query type and a list type. Queries correspond to the queries you create using the Query Editor.

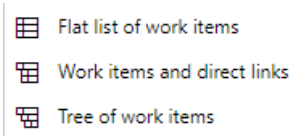
- **Query types**:
  - **None**: Indicates it's an **input list**.
  - **Query title**: Indicates the list of work items is tied to the specified **query**.
- **List types**:
  - **Flat list**: Simple list of work items that shows a single **Title** column. No link management is supported.

- **Tree list**: Hierarchical list of work items that support creating and updating tree topology links, such as Parent-Child links, between work items. These lists include two or more **Title** columns.

You can add, modify, publish, and refresh work items using any query type and list type.

Query types

Azure Boards supports three query types. The icon next to each query indicates the query type. The first two query types, **Flat list of work items** and **Work items and direct links** are imported as flat list queries.

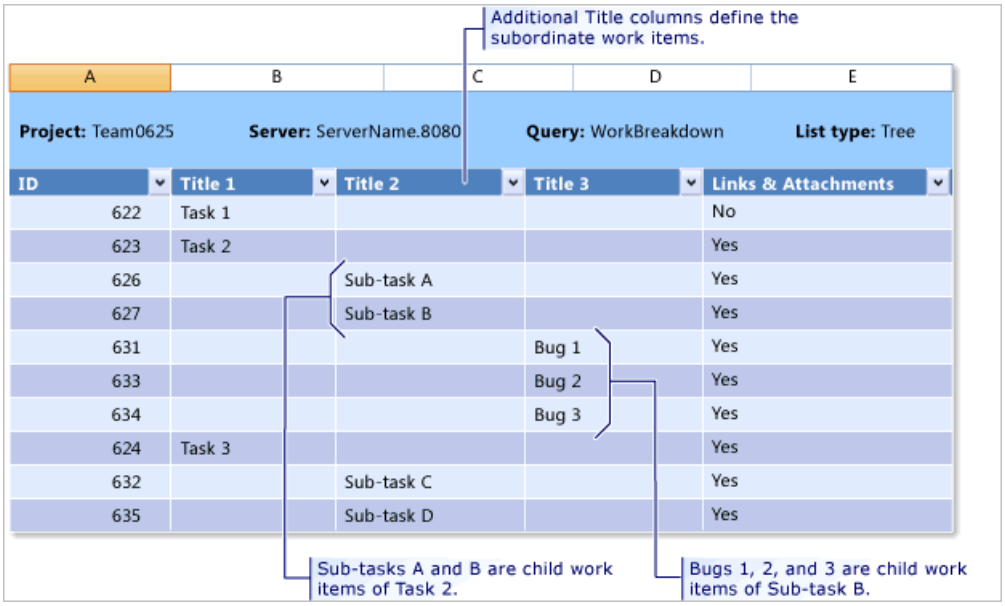


Only the **Tree of work items** queries import as a tree list. Direct links queries are imported as a flat list into Excel as modifying multiple types of links aren't a supported feature in Excel.

Tree lists

You can bulk add a nested list of work items, such as a work breakdown structure or a hierarchical set of user stories and customer experiences. For example, you can add a nested list of tasks, subtasks, and bugs, as shown in the following illustration, or linked tasks to product backlog items.

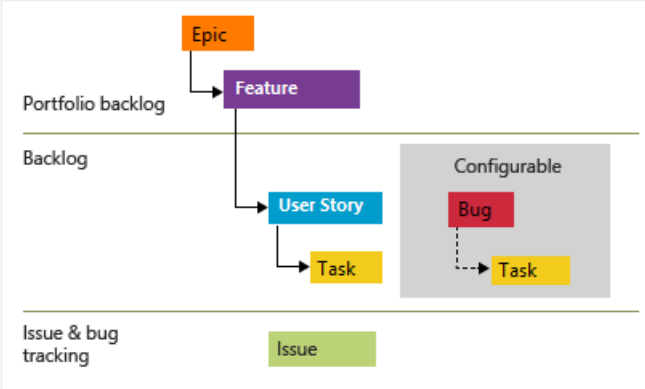
Here's how a three-level nested tree of items appears in Excel.



Parent-child links or other tree topology link types support creating a hierarchical backlog structure. The work item types that participate in the hierarchy differ with different processes and are shown in the following images.

- Agile process
- Basic process
- Scrum process
- CMMI process

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the Working with bugs setting. To learn more about using these work item types, see Agile process.

To import a hierarchical list, see Add or import a hierarchical list of work items later in this article.

My queries versus shared queries

You can open in Excel any query you've defined in Azure Boards. That includes queries defined under My Queries and Shared Queries. However, if you plan to share the workbook with other team members, then you should only work with a Shared Query. Other team members can't use the workbook or worksheet if it's based on a personal query stored under your My Queries folder.

Use list and query types

In general, you Use a flat list to bulk add or modify several types of work items at once, such as backlog items, tasks, bugs, or issues. Use a tree list to bulk add or modify work items and their tree-topology links.

Here's some more guidance:

- Use an **input list, flat list**: To import a list of work items or create new work items, no hierarchy
- Use an **input list, tree list**: To complete top down planning and import hierarchically linked work items
- Use a **query list, tree list**: To view and modify the hierarchy of link relationships of many existing work items.
- Use a **query list, flat list**: To bulk update a list of work items or create new work items, no hierarchy

- Use an input list, flat list: To import a list of work items or create new work items, no hierarchy
- Use an input list, tree list: To complete top down planning and publish parent-child linked work items
- Use a query list, flat list: To create an Excel report based on the query of work items

> **NOTE**
> To create an Excel report, you're project collection must be configured to support Analytics reporting. For more information, see Create Excel reports from a work item query.

- Use a query list, tree list: To view and modify the hierarchy and parent-child link relationships of many existing work items.

Use Excel features

You can use most Excel features when working with a list of work items. For example, you can use the following features:

- Format a cell or apply conditional formatting to a cell or column
- Cut and paste from one cell to other cells
- Cut and paste a single row

- Sum a column or add other formulas
- Fill down cells
- Filter
- Add multiple worksheets to your workbook

Each worksheet in Excel can contain a different input list or query. However, all worksheets within the workbook must connect to the same project within an organization or project collection.

The following features work slightly differently when working with a worksheet connected to Azure Boards.

- Each cell or column of cells corresponds to a work item field. Each field is associated with a data type. Excel won't allow you to enter data into a cell that doesn't meet the data type and requirements for that field.
- You can only insert a single row at a time within the worksheet.
- You can copy and paste multiple rows within the worksheet.
- To move a work item within a hierarchy, cut the entire row and paste it under the work item you want as its parent.
- Use **Outdent** and **Indent** to change the location of a work item within the tree.
- Undo (Ctrl Z) may not work. If you do something that you want to revert, you can refresh the worksheet.

We recommend you publish and refresh your worksheet often to make sure your local work remains in sync with Azure Boards data store.

To learn more about Excel, see Basic Excel tasks .

Sort work items

You can sort work item flat lists using the Excel sort feature.

However, if you're working from a tree list, you don't want to do any type of sort. Doing so changes the tree structure and as such, the links between work items.

If you want to use Excel to manage the order of your work items as they appear in a team backlog, you can do that by using the Stack Rank or Backlog Priority field (Agile or Scrum process). You can set values in these fields, publish your worksheet, and refresh your backlog. Your backlog items should appear reordered based on lowest to highest number. However, the next time the backlog is reordered from the backlog, the values you entered are subject to change.

If you want to maintain a certain order of work items, consider adding a custom field to manage the sort order, and then use that within Excel to sort your flat list of work items. This option, however, won't change the order that appears in your backlog.

Tasks you can and can't do with Excel

You can do the following tasks:

- Add tags and bulk update work items with tags as described in Add work item tags to categorize and filter lists and boards. Add the **Tags** field to your worksheet. Add multiple tags separated by a semicolon (;).
- You can add simple text to a rich-text field, but if you're updating several work items in bulk, you may lose formatting in existing work items.
- You can work offline and then reconnect and publish your changes. For details, see Connect Azure Boards to an Office client, Work offline, and reconnect.

You can't do the following tasks from an Excel worksheet:

- You can't delete work items
- You can't change the work item type of an existing work item
- You can't move work items to another project

- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread
- You can't link to a remote work item.

- You can't delete work items
- You can't change the work item type of an existing work item
- You can't move work items to another project
- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread.

- You can't delete work items
- You can't import or update test case steps or other test artifacts
- You can't add work items in any other State than the new State
- You can't add to a work item discussion thread

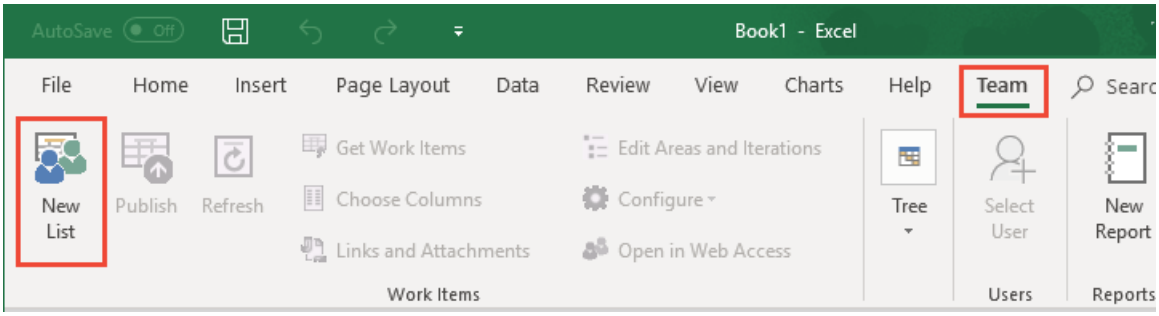
Import work items as a flat list

1. Open Excel and connect to your Azure Boards project. Use one of the four methods provided in Connect Azure DevOps project to Excel.

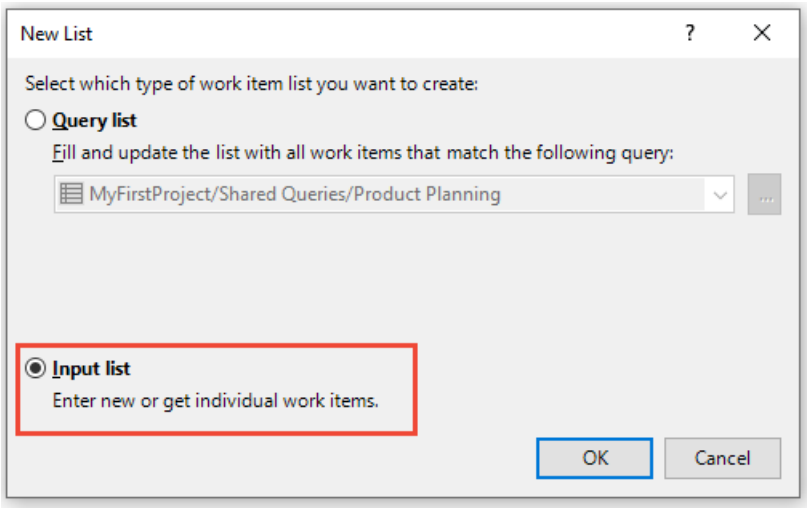
   > **NOTE**
   > When you connect to Azure Boards in the cloud, the **Team Project Collection** is automatically selected as there is only one collection associated with your Azure DevOps Services organization. When you connect to Azure Boards in an on-premises server, you choose the **Team Project Collection** prior to choosing the project.

2. In Excel, start with a blank worksheet. If you don't see the **Team** ribbon (or the **Team** menu if you use Excel 2007), see Azure DevOps Office integration issues.

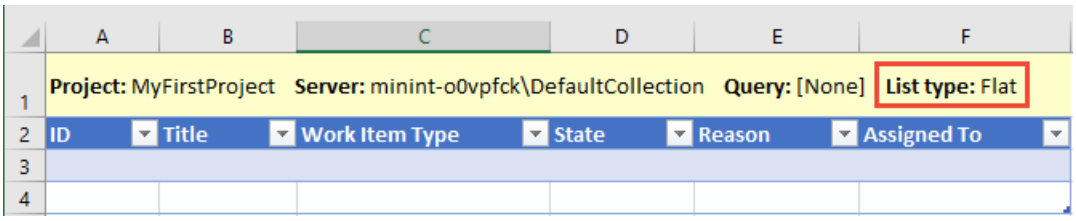
3. Choose **New List** from the Team ribbon.



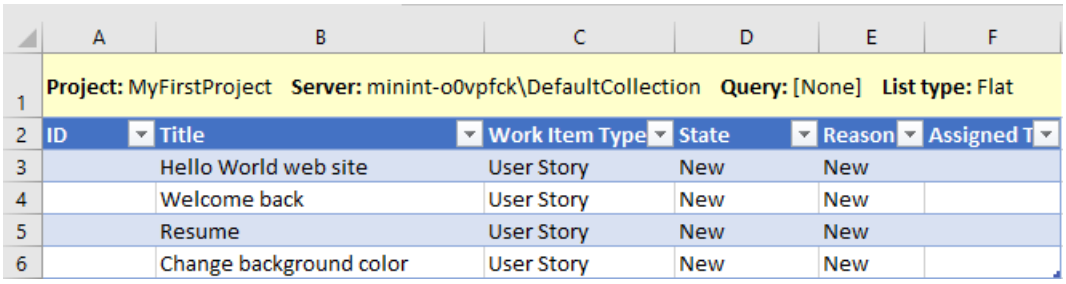
4. From the **New List** dialog, choose **Input list**.



5. Your worksheet is now bound to your project as an input list (Query[None]), flat list.

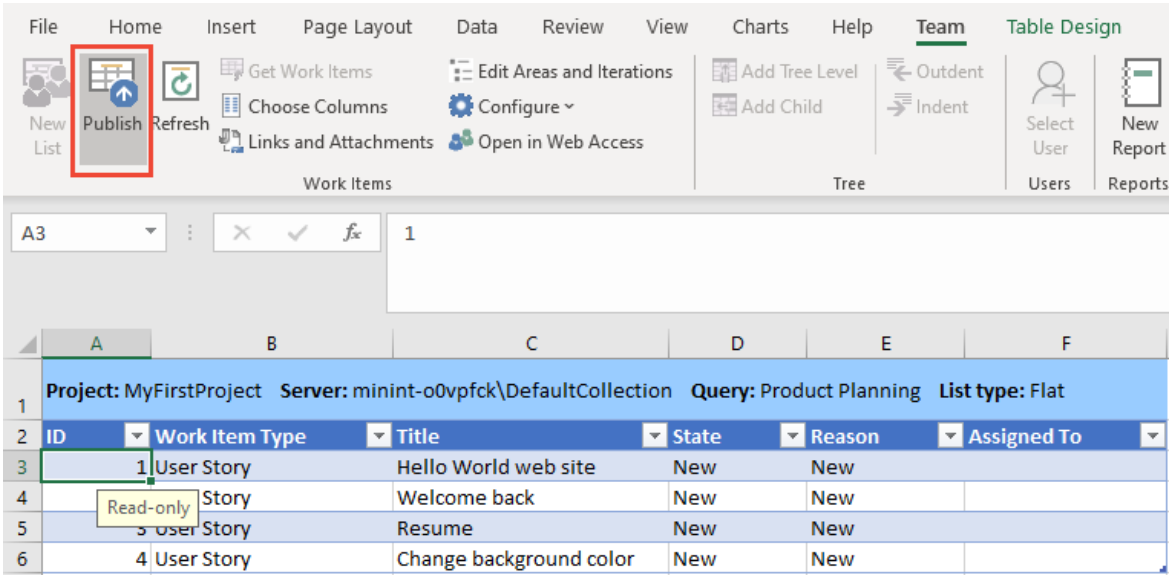


6. Specify the titles of the work items you want to add and their work item type.



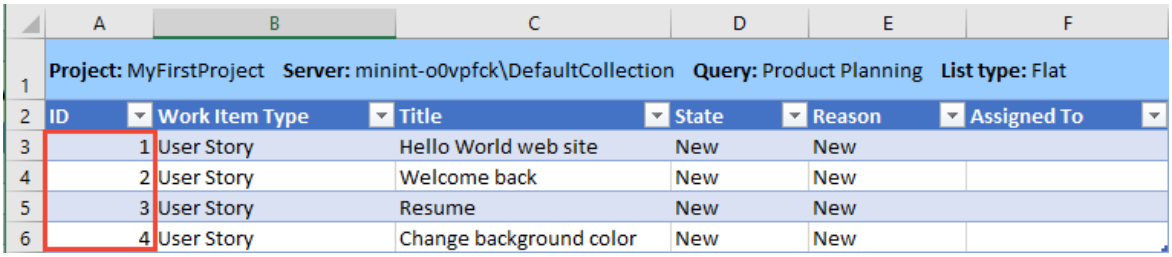
Notice how the **State** and **Reason** fields automatically fill in with default values once your select the work item type.

7. Publish your worksheet.



Make sure your cursor is in a cell that contains data. Otherwise, the **Publish** button might appear disabled.

Notice how IDs are now assigned to your work items.

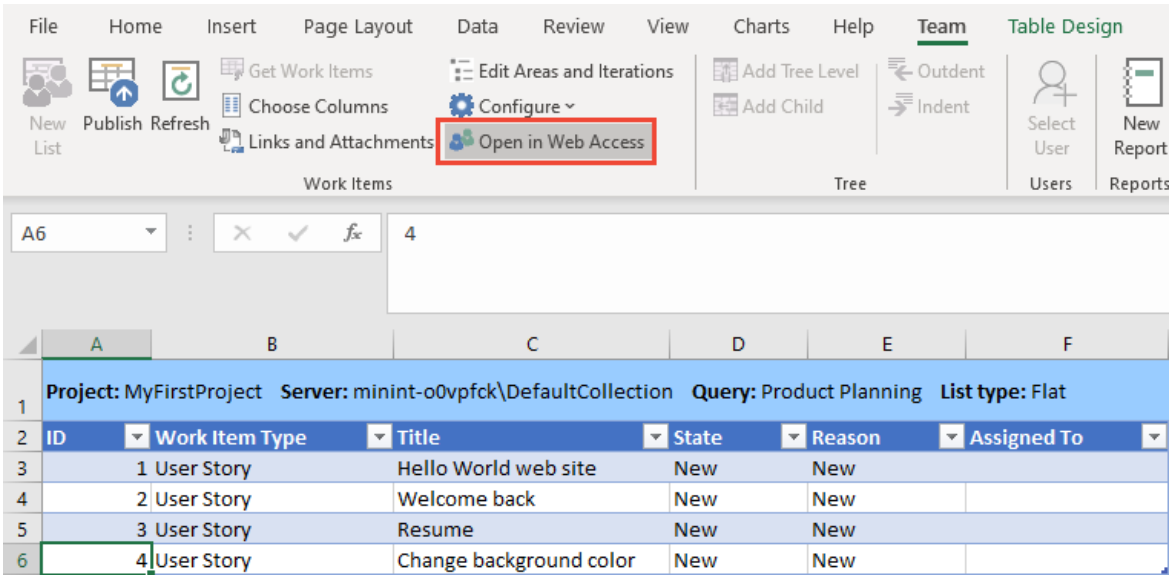


	A	B	C	D	E	F
1	**Project:** MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** Product Planning **List type:** Flat					
2	**ID**	**Work Item Type**	**Title**	**State**	**Reason**	**Assigned To**
3	1	User Story	Hello World web site	New	New	
4	2	User Story	Welcome back	New	New	
5	3	User Story	Resume	New	New	
6	4	User Story	Change background color	New	New	

8. To assign values to other fields, open **Choose Columns**, add the fields, make the assignments, and publish your changes.

> **TIP**
>
> If you're adding work items that you want to appear on a team backlog, make sure that you add and specify the team's Area Path and Iteration Path. If you need to add Area Paths or Iteration Paths, choose the **Edit Areas and Iterations** link. The link opens a web browser to the Project Settings page. To learn more, see Define area paths and assign to a team and Define Iteration Paths and configure team iterations.

9. To open a work item to add more information, Choose the work item you want to open and then choose **Open in Web Access**. Before you do, make sure you publish any changes you've made.



	A	B	C	D	E	F
1	**Project:** MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** Product Planning **List type:** Flat					
2	**ID**	**Work Item Type**	**Title**	**State**	**Reason**	**Assigned To**
3	1	User Story	Hello World web site	New	New	
4	2	User Story	Welcome back	New	New	
5	3	User Story	Resume	New	New	
6	4	User Story	Change background color	New	New	

A web browser opens and displays the work item.

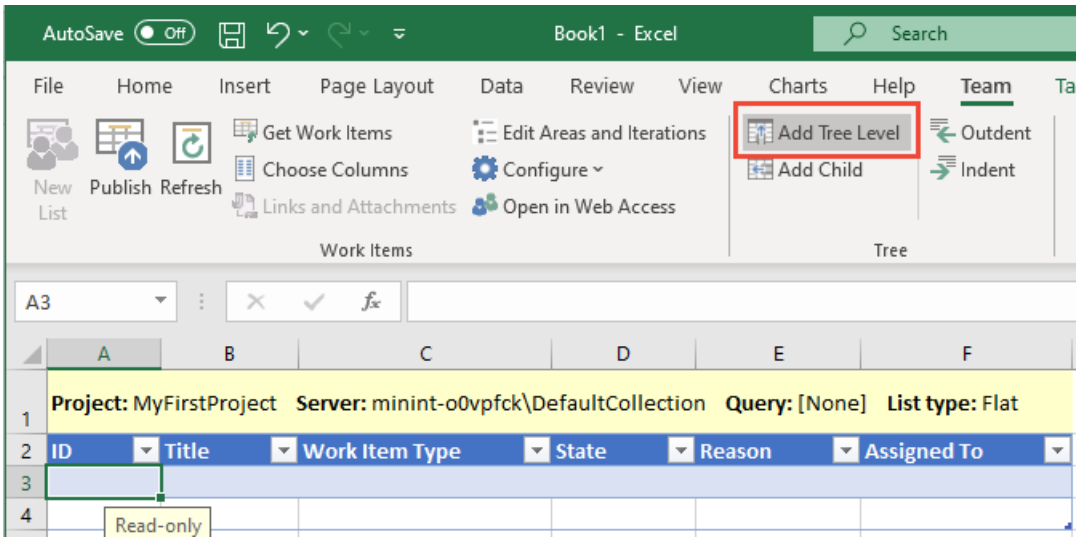
If you make changes to the work item, you should then immediately refresh your worksheet to capture the changes.

Import work items as a tree list

You can add a hierarchy of work items linked using parent-child links or other tree topology link type.

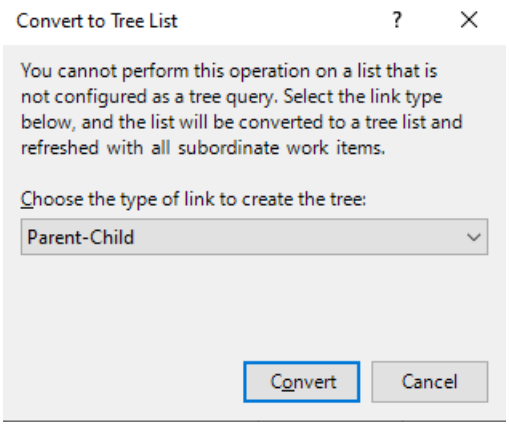
> **IMPORTANT**
>
> Don't sort a tree list. Sorting a tree list can change the hierarchical link relationships.

1. Starting from Step 5 from the previous procedure, convert your flat list, input list into a tree list. Choose a cell within the flat list and then choose **Add Tree Level**.

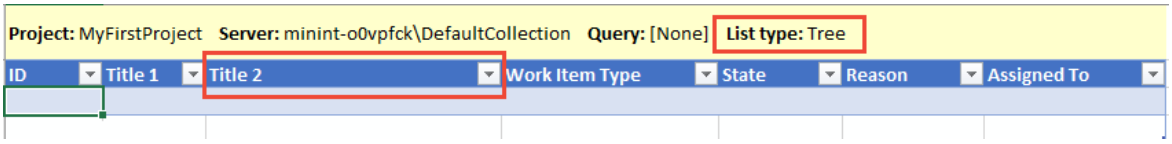


If the **Add Tree Level** is disabled, you're working from a query list. To convert your list to a tree list, you must first reconfigure your list to an input list.

2. Choose the link type to use when adding work items to a hierarchy, and then choose **Convert**. The most usual choice is **Parent-Child**. You can only select from tree topology link types. To learn more, see Link type topologies and restrictions.



Note the List type has changed to **Tree**, and a second **Title** column appears.



3. To add more levels to the hierarchy, choose **Add Tree Level** again. For example, if you want to add a hierarchy of Epics, Features, and User Stories, you'll want to have **Title 1**, **Title 2**, and **Title 3** columns.

   If you want to add tasks, add another tree level to have four title columns. To remove a column, see Remove a tree level.

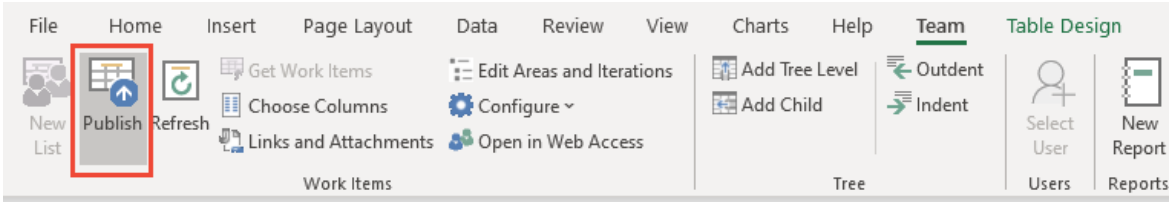
4. Save your Excel file.

5. Enter the **Work Item Type** and **Titles** for the hierarchy you want to import. The **State** fields automatically fill in with default values once you select the work item type.

Project: MyFirstProject **Server:** minint-o0vpfck\DefaultCollection **Query:** [None] **List type:** Tree

ID	Work Item Ty	Stat	Title 1	Title 2	Title 3
	Epic	New	Improve User Experience		
	Feature	New		Integrate client application with popular email clients	
	User Story	New			Implement a factory which abstracts the email client
	User Story	New			As a user, I can select a number of support cases and use cases
	User Story	New			Customer request support for GMail
	User Story	New			Customer request support for Yahoo mail
	User Story	New			Customer request support for Roundup
	Feature	New		Integrate client app with IM clients	
	Feature	New		Emoticon feedback enabled in client application	
	User Story	New			As a user, I can select an emoticon and add a short description
	User Story	New			Add animated emoticons
	Epic	New	Framework to port applications to all devices		
	Feature	New		Migrate legacy code to portable frameworks	
	User Story	New			Implement a factory that migrates legacy to portable frameworks
	User Story	New			As a developer, I can analyze a code base to determine compliance with
	Epic	New	Counter the Heartbleed web security bug		
	Feature	New		Scan all software for the Open SLL cryptographic code	
	User Story	New			Scan all code base and identify the affected code
	Feature	New		Update software to resolve the Open SLL cryptographic code	
	User Story	New			Update and retest suite code base affected by the vulnerability
	User Story	New			Update and re-test service code based affected by the vulnerability
	Epic	New	Shared Personalization and state		
	Epic	New	Service interfaces to support REST API		
	Epic	New	Convert legacy Odata service interfactes to REST API		
	Feature	New		Convert all services from using experiemental code	
	Feature	New		Convert all client service calls from using experimental code	

6. Publish your worksheet.

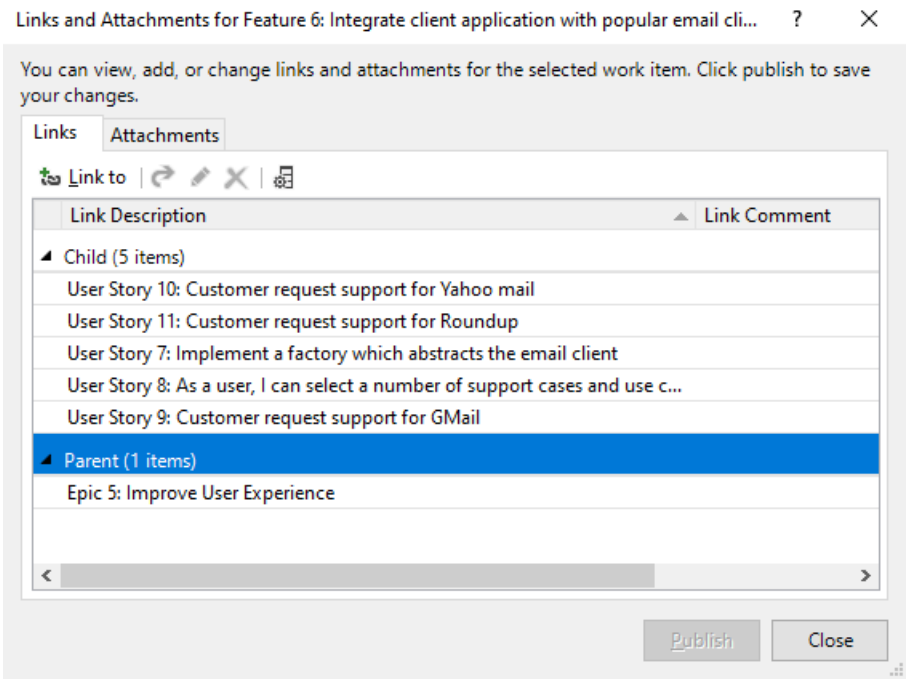


Make sure your cursor is in a cell that contains data. Otherwise, the **Publish** button might appear disabled.

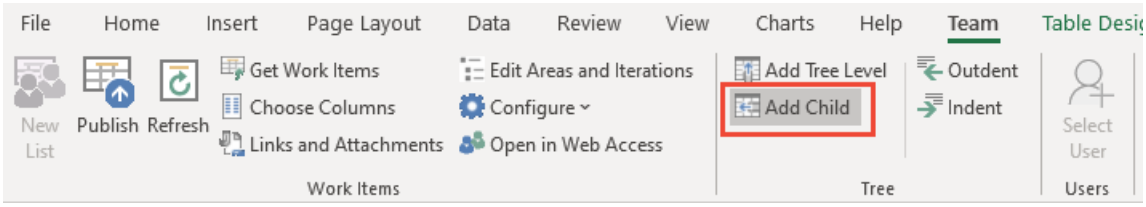
IDs are now assigned to your work items. In the background, the link type you selected is used to link each work item in the hierarchy. Epics are linked to Features, Features are linked to User Stories.

7. To check the links made, choose a work item and choose **Links and Attachments**.

For example, here we show the Child and Parent links created for a Feature that was imported.

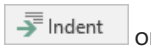


8. To enter a row under a work item where you want to add a child, choose the row and then choose **Add Child**.



9. To assign values to other fields, open Choose Columns, add the fields, make the assignments, and publish your changes.

10. To change the hierarchy, cut and paste the row of a work item to place it under the new parent. Make sure that you select the entire table row. When you publish the change, the old hierarchical links are deleted and the new hierarchical link are created.

    You can use the 

or 

indent/outdent icons to demote or promote a work item within the tree hierarchy. Verify that the column to the left or right of the parent work item's title is a **Title** column. The header at the top of the column should read **Title n**, if it doesn't, add a tree level.

Remove a tree level

1. First, publish changes that you've made to work items before you remove a tree level. Removing a tree level requires a refresh, which overwrites data in the work item list. You'll lose any data you haven't published.

2. Next, delete any content under the tree-level **Title** *number* column you want to remove—the highest numbered column—. This column should be the highest numbered column in the tree.

3. **Refresh** your worksheet. The column containing empty values for the Title is removed.

   You'll receive an error message if you attempt to delete the column.

Useful tips when working with a tree list

- Excel interprets the data in the **Title** columns to determine the pattern of links between work items. When you publish changes, any of the following conditions can result in an error, an invalid link, or a tree link to be created between incorrect work items:
  - A row between the work items is blank within the hierarchy

- The Title of a work item is in the wrong column. Make sure you enter a title for each child work item.
- Within a row, multiple **Title** columns contain data. Enter text in only one of the title columns within each row.
- The list was sorted. Don't sort a tree list. Sorting a tree list can change the hierarchical link relationships. If you do sort a tree list, you can recover from this operation by immediately refreshing.

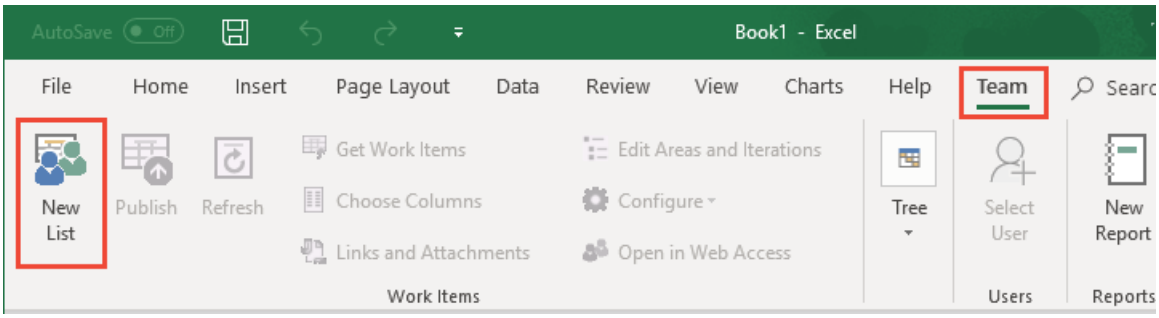
- To resolve an error, see Resolve invalid links.
- A parent-child linked work item can only have one parent. You can't add the same work item task to two backlog items. Instead, you need to define distinct work item tasks.

Update work items in bulk with a query list

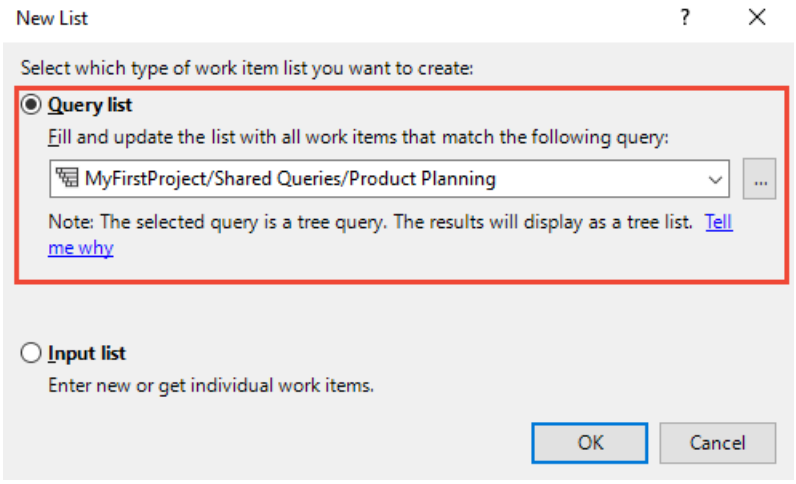
The easiest way to bulk update many work items is to create a query with the work items you want to update, and then open that query in Excel.

> **TIP**
> **Follow these tips to keep your work in sync:**
>
> - When you first open a saved worksheet, use (**Refresh**) to download the latest data from the data store.
> - Enter data for additional fields by adding columns to the worksheet using **Choose Columns**.
> - To avoid data conflicts, publish your additions and modifications often.
> - To prevent loss of data before you publish or refresh, save your workbook periodically.

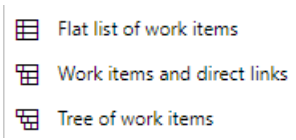
1. From the web portal or Visual Studio, create the work item query that contains the work items you want to update. For details, see Create and save managed queries with the query editor.
2. Open Excel and connect to your Azure Boards project. Use one of the four methods provided in Connect Azure DevOps project to Excel.
3. If you opened your query from the web portal or Visual Studio, you're done. Make any changes you want. Open **Choose Columns**, add fields, make assignments, and publish your changes.
4. If you start from Excel, open a blank worksheet. You can add a worksheet to an existing workbook, as long as you're choosing a query from the same project the workbook is bound to.
5. Choose **New List** from the Team ribbon.



6. From the **New List** dialog, choose **Query list**, and select the query you want from the drop-down menu.



The icon next to each query indicates the query type. The first two query types, **Flat list of work items** and **Work items and direct links** are imported as flat list queries. Only the **Tree of work items** queries import as a tree list.



7. With the work items imported to Excel, make the modifications you want and publish your changes.

   If you're working with a tree list, see also the information provided in Import a hierarchical list of work items.

Enable Tree commands

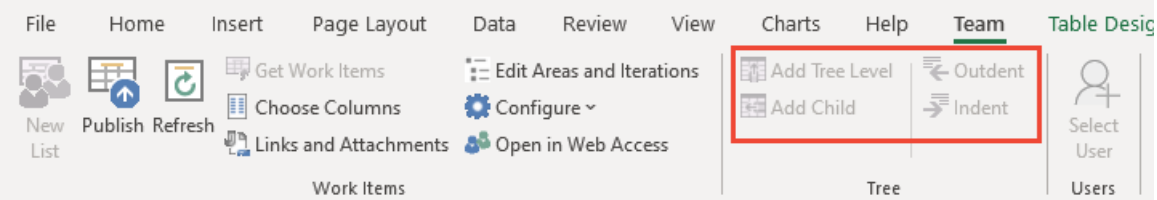
If the **Tree group** commands aren't available, your worksheet is configured as a flat list, query list. Convert the list to either an input list or a list based on a tree query to enable the Tree group commands. To learn how, see the next section on Change your list type or query.

Change your list type or query

You can change the work items listed in your worksheet. Specifically, you can:

- Change your flat list to a tree list
- Change from a query list to an input list
- Change from an input list to a query list
- Change the query your worksheet references

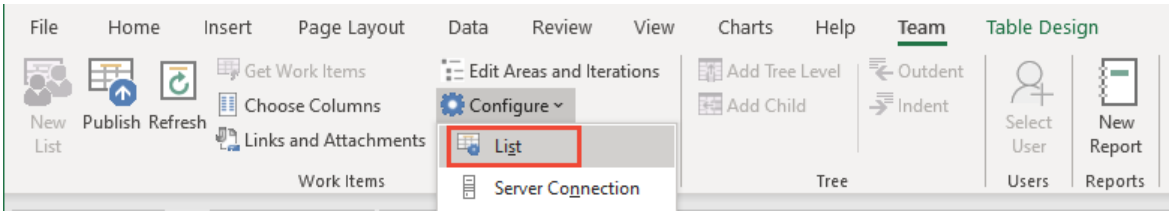
If you want to change your flat list to a tree list, you can. However, if your list is a query list, then you first need to reconfigure the list. You'll know that it's a flat list, query list as the **Tree group** commands are disabled.



To convert your query list to an input list, follow these steps.

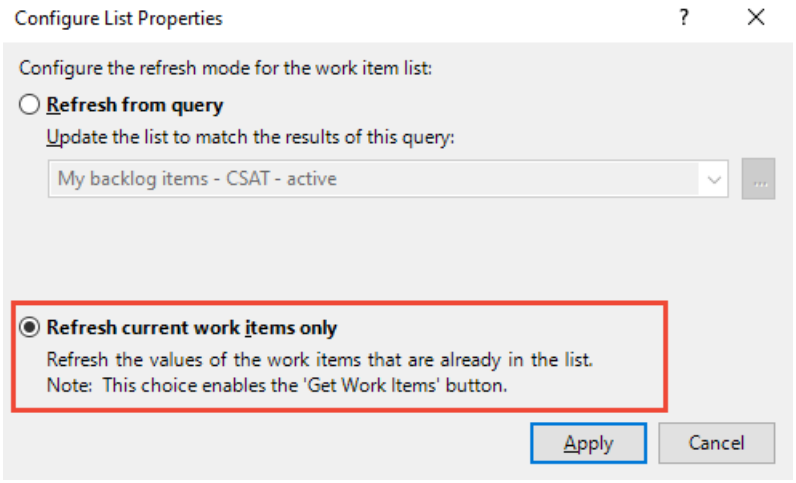
1. First, publish whatever changes you have made.

2. Next, on the **Team** ribbon, choose **Configure**, **List**.

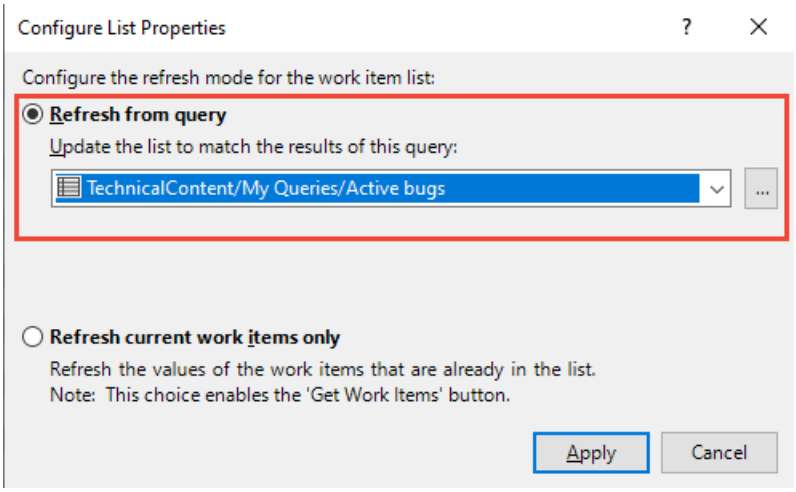


3. Choose **Refresh work items only** and then **Apply**.

   This choice changes the query list to an input list.



4. To convert from an input list to a query list, choose **Refresh from query**, select the query, and then **Apply**.

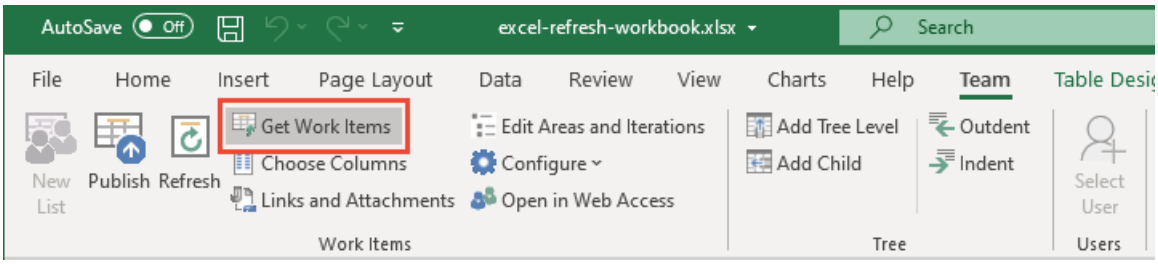


Add existing work items to your worksheet

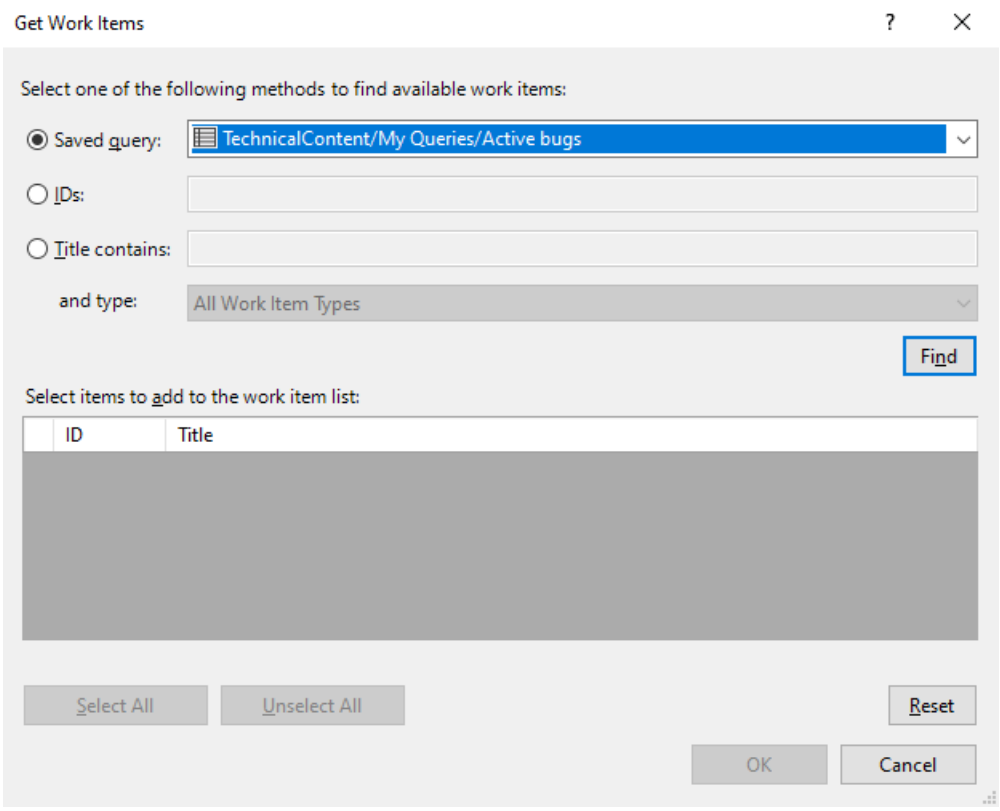
If you're working from a query, modify your query to contain the work items you want. Then refresh your list. The other work items will appear in your list.

If you're working with an input list, complete these steps.

1. From the **Team** ribbon, choose **Get Work Items**.



2. Choose the method you want from the three options available.



If the work items are defined in another project, then first select the Project. Then, make your selections:

- **Query**. Use this method when you've defined a query that contains the set or superset of work items you want.
- **IDs**. Use this method when you know the IDs of the work items that you want to link to. In the **IDs** box, type the IDs of the work items that you want to find, separated by commas or spaces.
- **Title contains**. Use this method to find work items that have a common word or phrase in the title field. In the **and type** list, select the type of work item that you want to retrieve.

> **NOTE**
> To minimize the time required to run the query, narrow the filter criteria of the search.

3. Choose **Find**.

   Only those work items defined for the selected project and specified work item type are listed. To sort on a column field, choose the column **Title**.

4. In the list of returned work items, select the check-box of one or more work items.

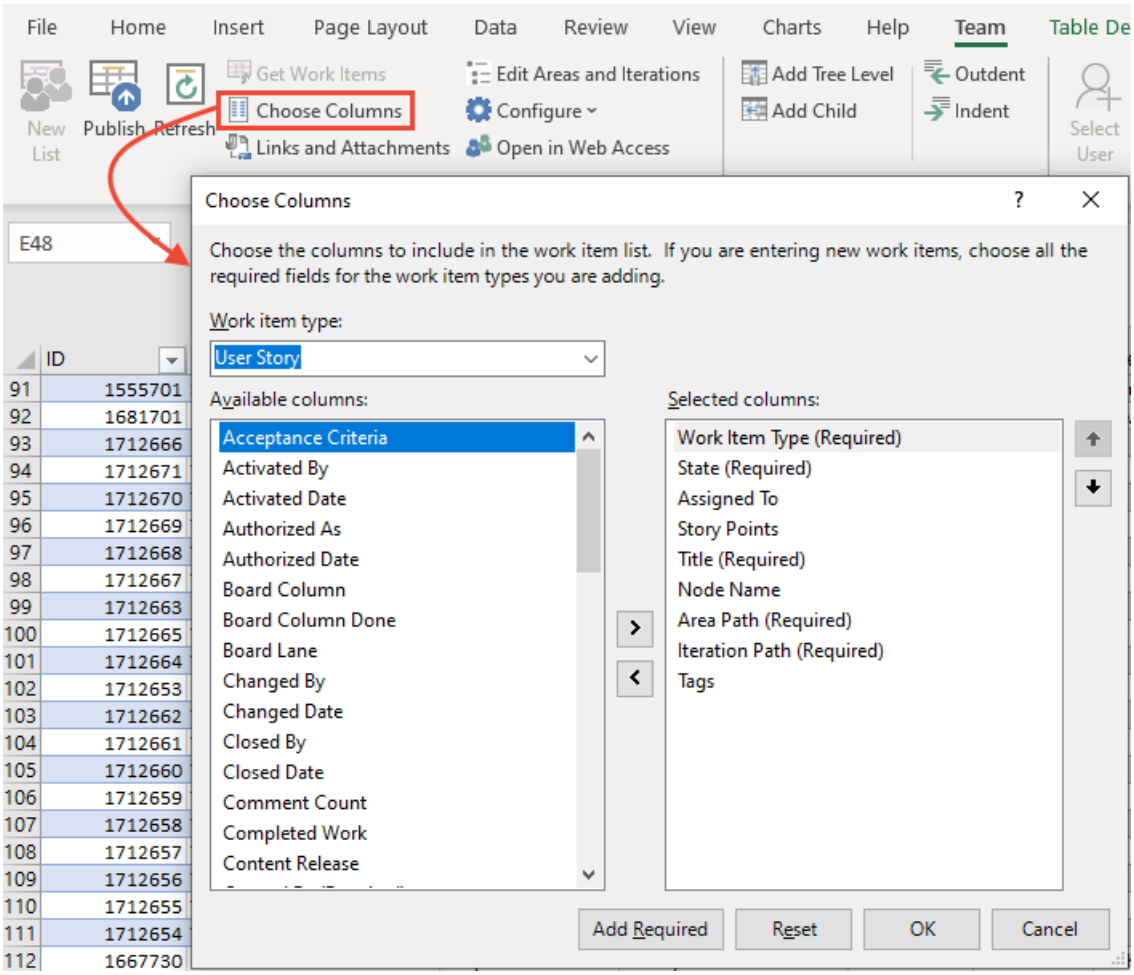
   - Select each work item that should link to the current work item. You can also press the SHIFT key while clicking to select a range of work items, or press the CTRL key while clicking to select multiple work items.
   - Choose **Select All** to select all work items in the list.

Add or remove column fields

If you start your worksheet with a **New List**, you'll see only a set of default field columns. You can add columns using the **Choose Columns** on the Team ribbon.

If you start your worksheet from an existing query, you'll see all the column fields defined for the query. From there, you can add columns using the **Choose Columns**. However, your additions don't modify the underlying query.

1. To assign values to other fields, choose **Column Options** to add the fields of interest.



   - To filter the fields based on work item type, select the **Work item type**.
   - To move or remove a field, choose the field and then select the > or < icons.
   - To change the field sequence, move the field up or down in the list using the up and down arrows.
   - You can add a rich-text field, such as the **Description** field, however you may lose some of the formatting upon publish.
2. Once the fields appear in the worksheet, assign values and publish your updates. When working with identity fields, ones that accept user accounts, see the next section, Select user accounts.
3. Save your worksheet.

Select user accounts

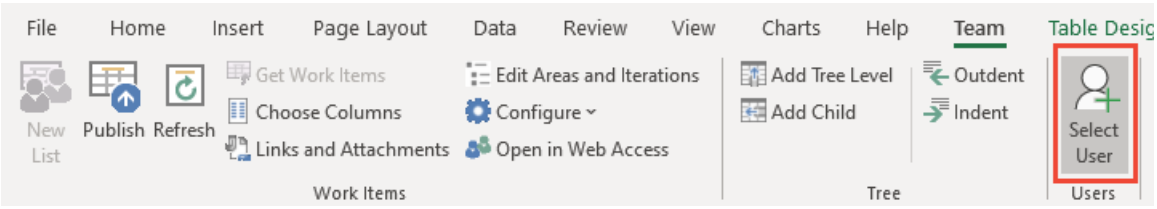
You can use the Select User feature to find user accounts and assign values to person named fields. Also, this feature provides access to the most recently used (MRU) values. If your team contains several hundreds or thousands of user accounts, you'll want to use this feature.

TIP

Without the **Select User** feature, you must enter user names exactly as they are in the database, or you'll receive data validation errors upon trying to publish.

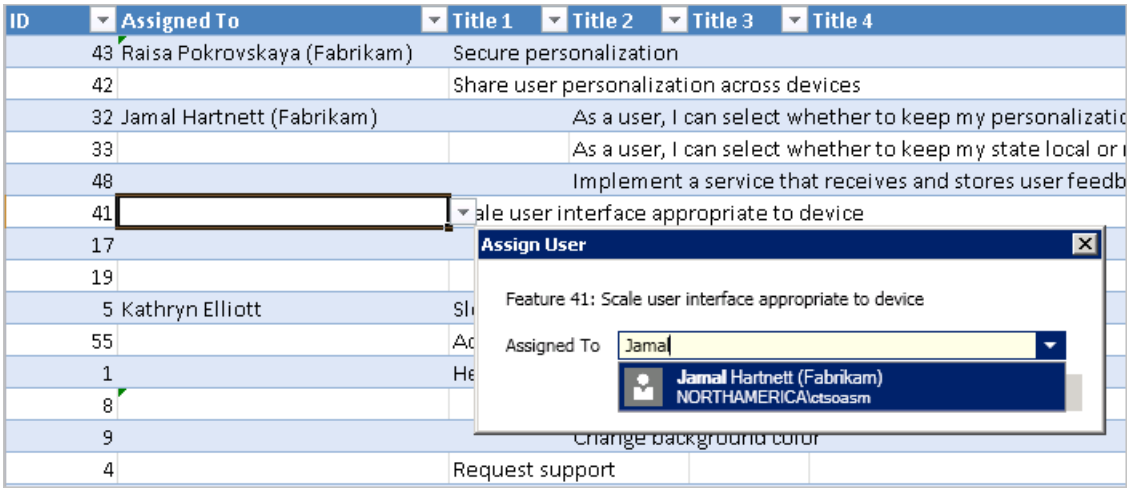
1. If you haven't installed or updated to the latest version of Visual Studio (at least VS 2015.1 or later version, do that now. You need the latest update to access the Select User feature.

2. Choose an identity or person-named field to activate the **Select User** feature in the Team ribbon.



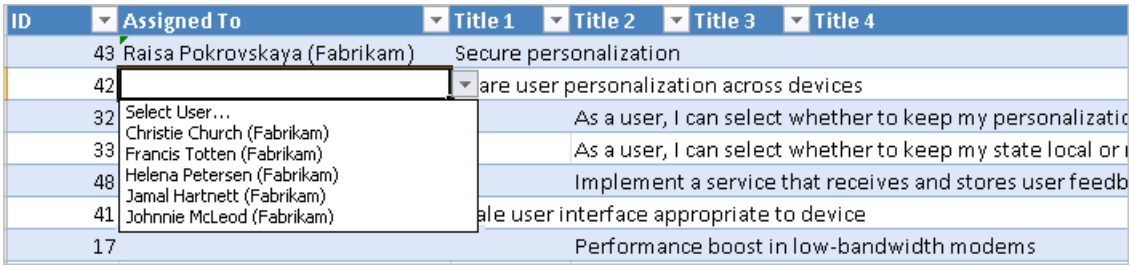
An identity or person-named field is a field that contains a user identity. These fields are typically synchronized to a database of user accounts, such as Azure Active Directory, Active Directory, or a Workgroup.

3. Begin typing the name of the user account and the Assign User dialog will automatically filter the results until you can select the account of interest.



Enter a letter to tab to the start of names beginning with that letter. Enter only user names as account aliases aren't recognized.

You'll notice that as you select user names, Excel remembers your recent selections and you can select the user accounts directly from the field.



Link work items

You can complete many actions from the **Links** tab of the **Links and Attachments** dialog. Specifically, you can:

- Review the existing links defined for the selected work item

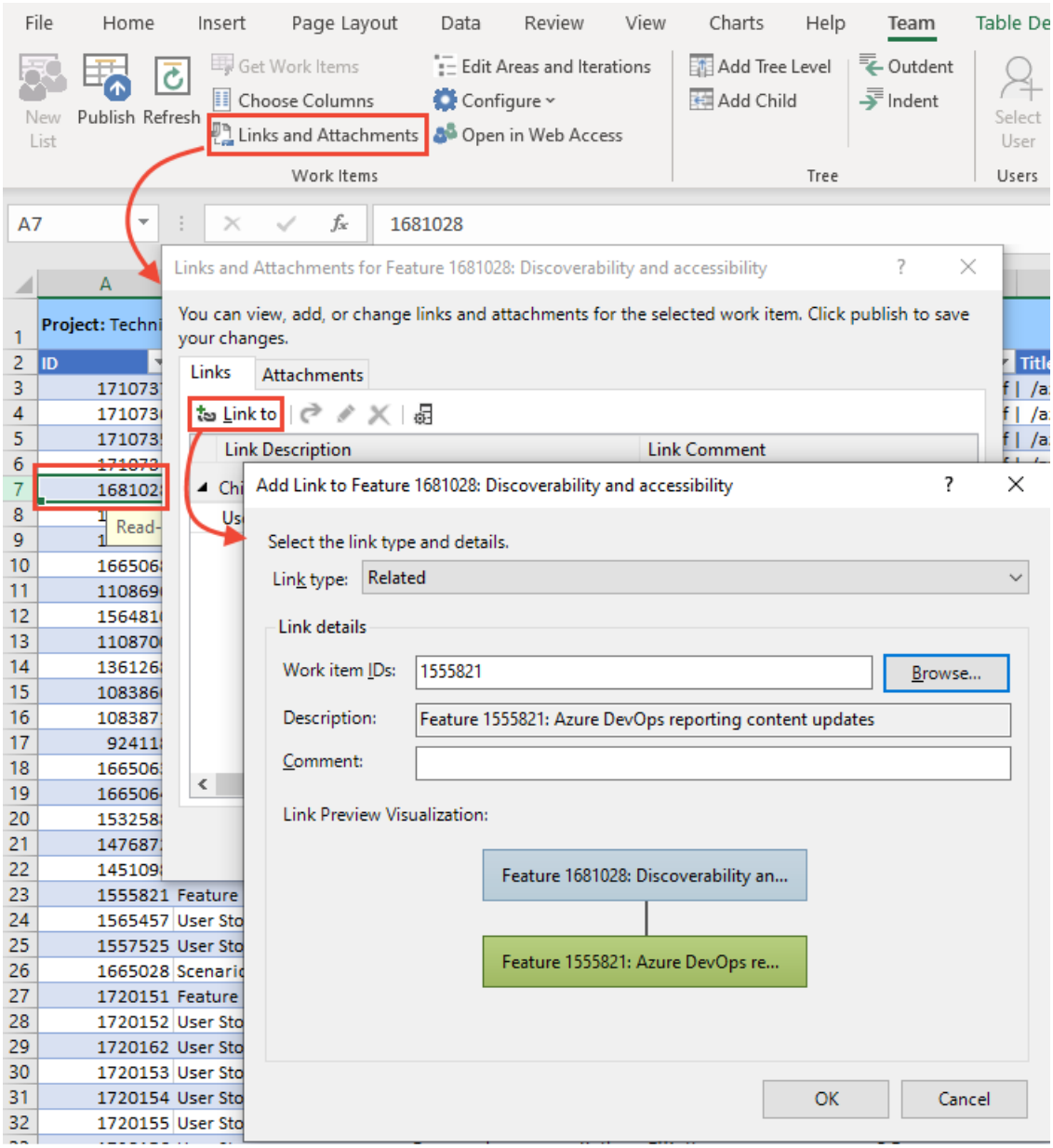
- Add links to selected work items to one or more work items or select objects
- Delete links
- Open a linked work item (opens in the web portal)
- Edit the link type of an existing link
- Add columns to the Link list and sort on that list

For more information on linking work items, see Link user stories, issues, bugs, and other work items.

View and add links

You can't use the Links and Attachments dialog to bulk update work item links. You can only bulk update tree-topology link types using a tree list.

1. To link a work item to other work items, choose the work item and then choose **Links and Attachments**. From the Links tab, choose **Link to** and then choose the **Link Type** and work item(s) you want to link to. Choose **OK** and then **Publish**.



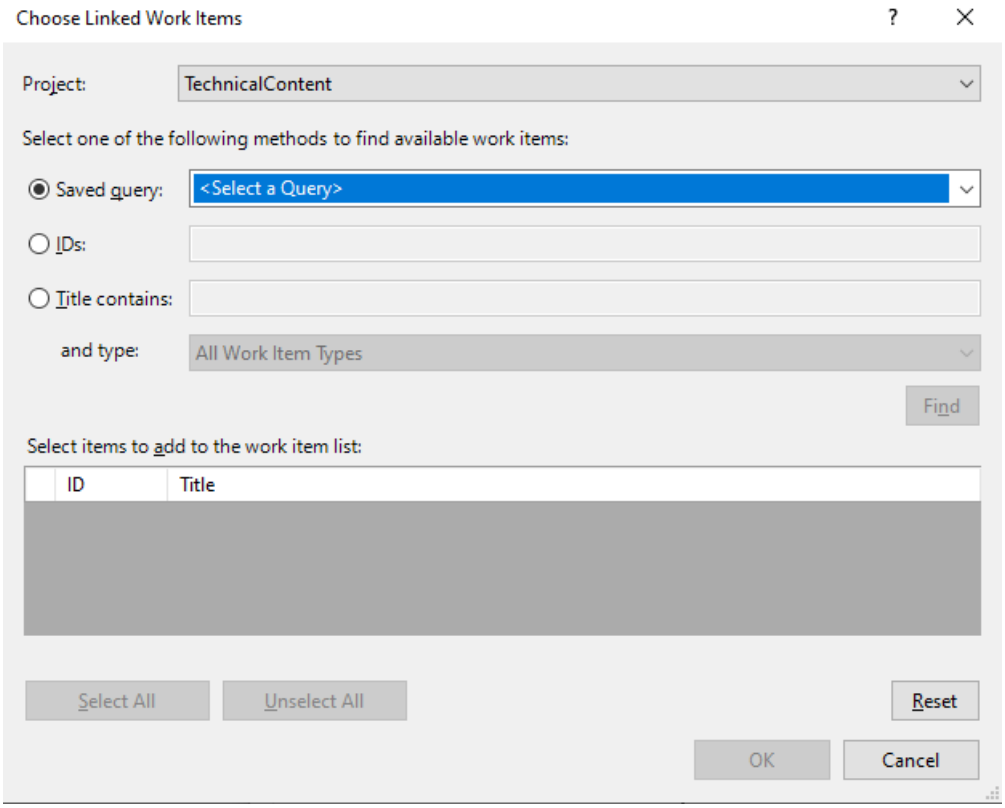
When finished, choose **Close** to dismiss the dialog.

2. To link several work items to the same work item(s), multi-select them by using **Ctrl-click** for consecutive rows, or **Shift-click** for non-consecutive rows.

Find work items to link to

From the Add link dialog, you can open a secondary dialog to choose one or more work items to link to. If you're going to find and list work items to link to by using a saved query, first define the query to use.

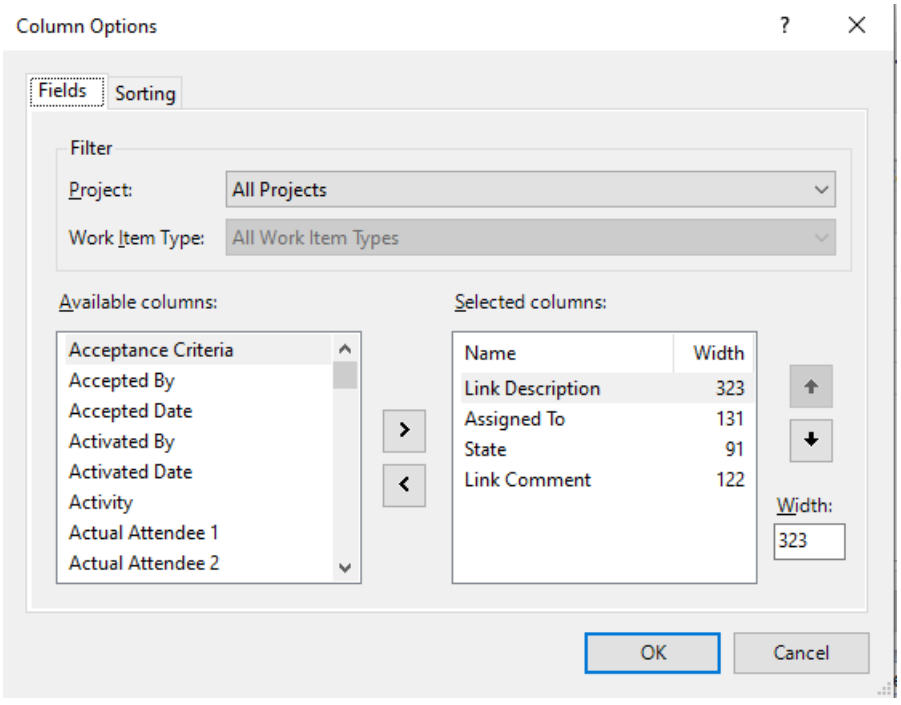
From the Add link dialog, choose the **Browse** button (Visual Studio) to open the following dialog.



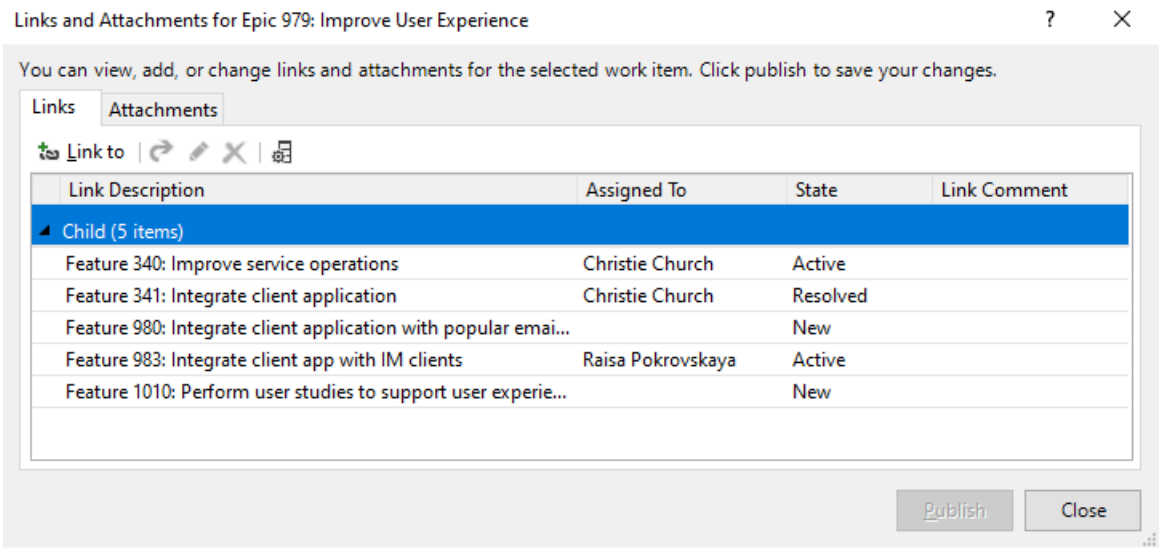
The **Choose Linked Work Items** dialog works in the same way as the **Get Work Items** dialog. To learn more, see Add existing work items to your worksheet described earlier in this article.

Add columns to the links list

1. From the **Links** tab, choose the **Columns** icon, and add the fields you want displayed. Here we add the Assigned to and State fields.



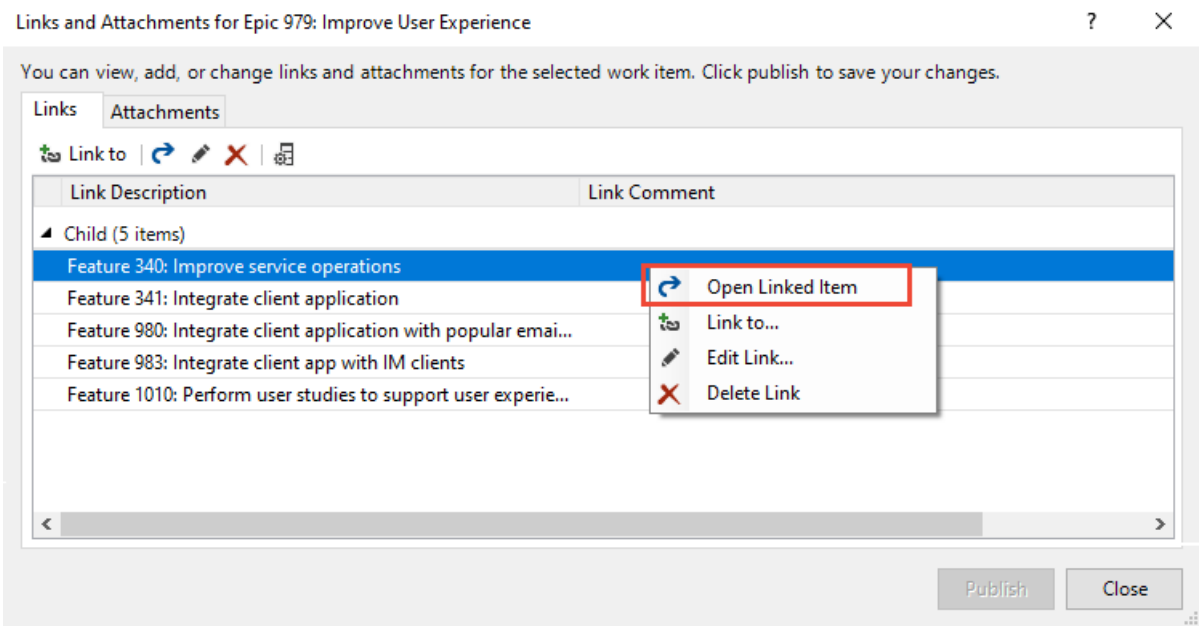
2. To reorder the links, choose the field to sort the list on that field.



This dialog works in the same way as the **Get Work Items** dialog. See Add existing work items to your worksheet described earlier in this article.

Open a linked work item

From the **Links** tab, choose the linked work item, right-click to open the context menu, and choose **Open Linked Item**.

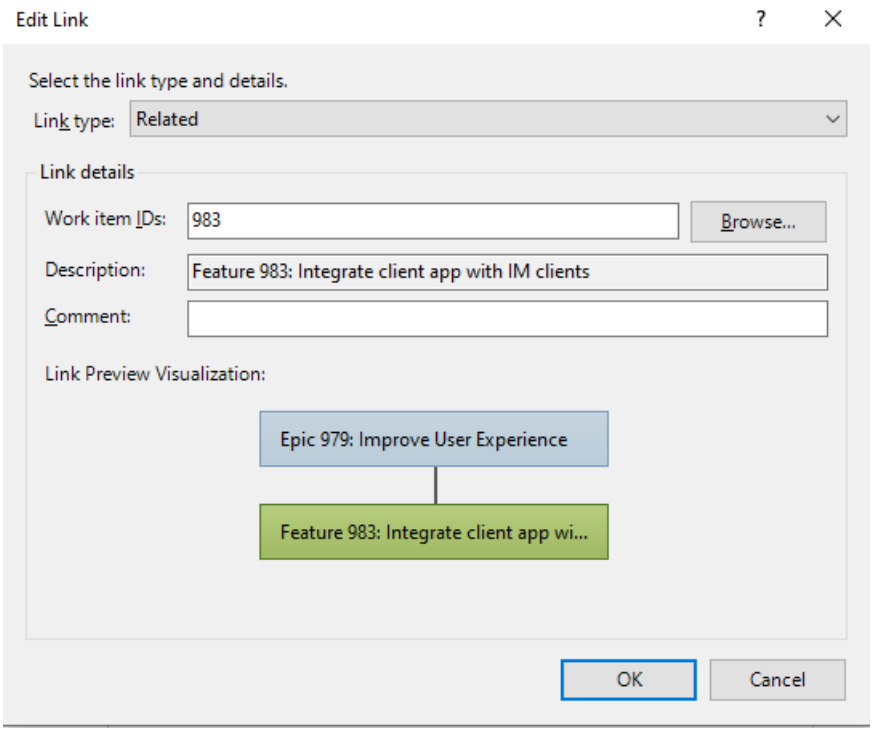


The work item opens in your web portal.

Edit the link and change the link type

You can edit any link listed. You can change the link type and the work items linked to.

1. Choose the link and choose the **Edit** icon.
2. Change the link type as needed.



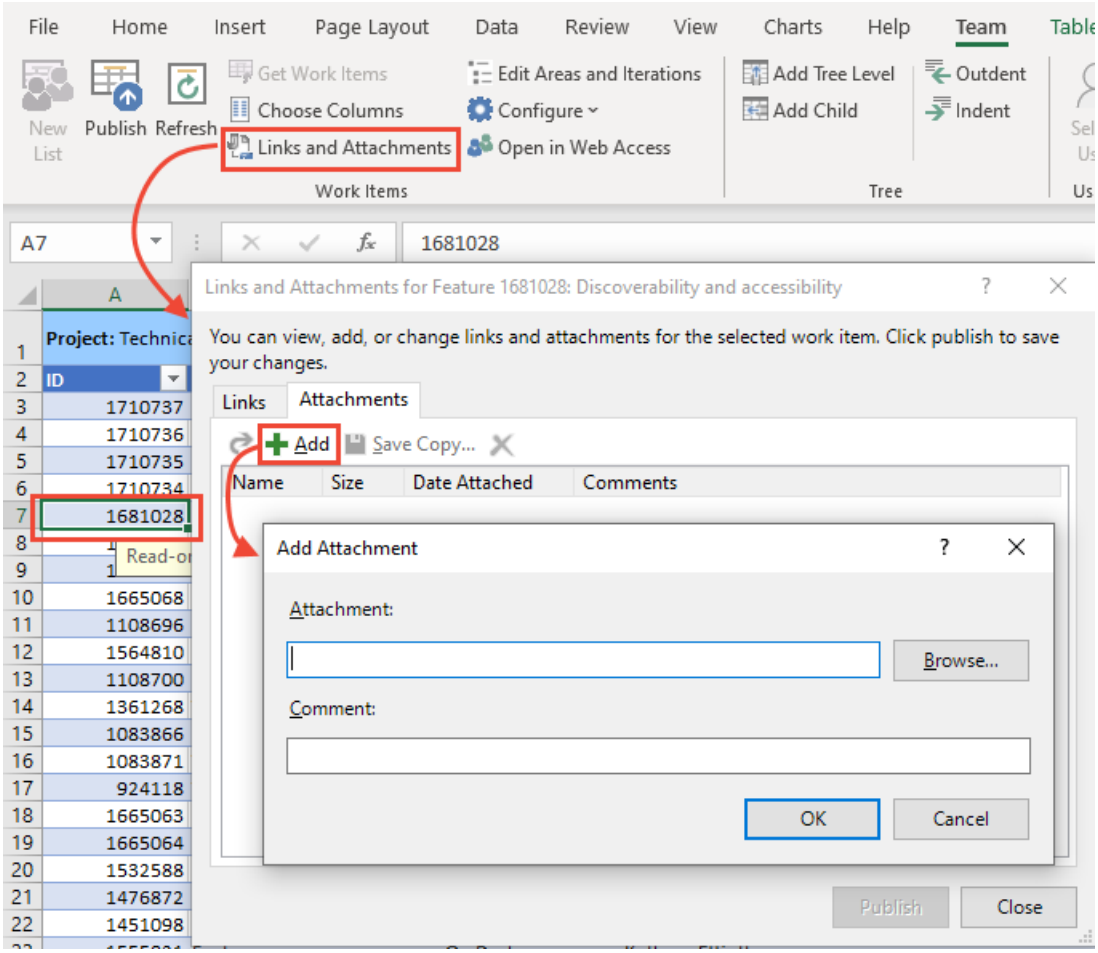
3. To change the work item linked to, enter the ID of the work item, or choose **Browse** to find the work item(s) to link to.

   The **Choose Linked Work Items** dialog works in the same way as the **Get Work Items** dialog. To learn more, see Add existing work items to your worksheet described earlier in this article.

Add attachments

- To add attachments, choose the work item, then **Links and Attachments**, and then the **Attachments** tab.

  Choose the file you want to attach, then choose **OK** and then **Publish**.



When finished, choose **Close** to dismiss the dialog.

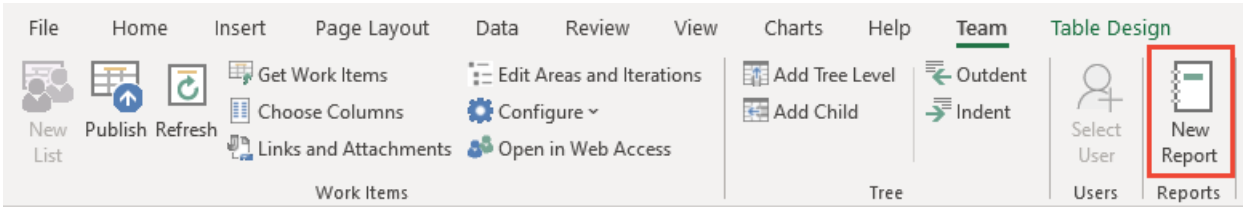
- To add the same attachment(s) to several work items, multi-select them by using **Ctrl-click** for consecutive rows, or **Shift-click** for non-consecutive rows.

Create a report

You can create a report or chart from the web portal for flat-list queries. See Track progress by creating status and trend query-based charts.

> **IMPORTANT**
> You can only create an Excel report using **New Report** from an on-premises Azure DevOps Server. These reports require that your project's project collection is configured to support SQL Server Analytics Server.

You can create a report using the **New Report** feature based on a flat list of work items.



To learn more, see Create Excel reports from a work item query.

Resolve publishing errors

To resolve publishing errors that arise when working in Excel, see one of the following articles:

- Resolve data conflicts: A data conflict occurs when a field value is changed in Azure Boards since the last time you published from Excel.
- Resolve data validation errors: A data validation error occurs if a field value violates the rules for that field and work item type.
- Resolve invalid links in a tree hierarchy: An invalid link occurs if a team member view work items in Excel as a hierarchy or tree list, and then moves a work item or sorts the list such that it breaks the dependencies between work items. You can resolve this error by reviewing the error message and repositioning work items to reflect the work item structure.
- Address Error TF208104: Hierarchical Link Relationship Is Locked:
  If you receive error TF208104, the changes you made to the fields are published. But your changes to the link hierarchy aren't published. At least one of the link relationships defined for the work item is locked by another process, such as Project Server integration.

Related articles

- Bulk modify work items (web portal)
- Azure DevOps Office integration issues
- FAQs: Work in Excel connected to Azure Boards
- Bulk import or update work items using CSV files
- View and add work items
- Basic Excel tasks

- Bulk modify work items (web portal)
- Azure DevOps Office integration issues
- FAQs: Work in Excel connected to Azure Boards

- Create Excel reports from a work item query

- Basic Excel tasks

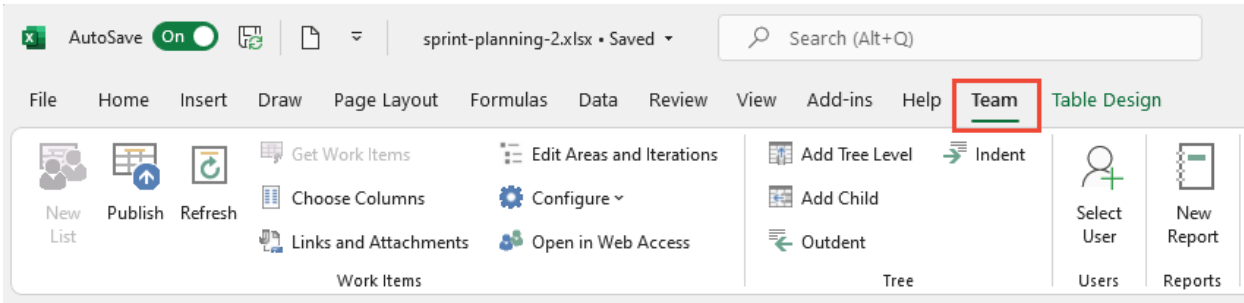
Resolve Azure DevOps Office integration issues

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

All Office integration tasks require that you have installed a version of Visual Studio or the free Azure DevOps Office Integration 2019. These software installs the Azure DevOps Office Integration Add-in or Team Foundation Office Integration Add-in For a list of prerequisites, see Azure Boards and Office integration.

If you don't see the **Team** ribbon in Microsoft Excel, as shown in the image below, you may want to resolve the issue with the procedures provided in this article.



IMPORTANT

Microsoft Project Integration and the **TFSFieldMapping** command are not supported for:

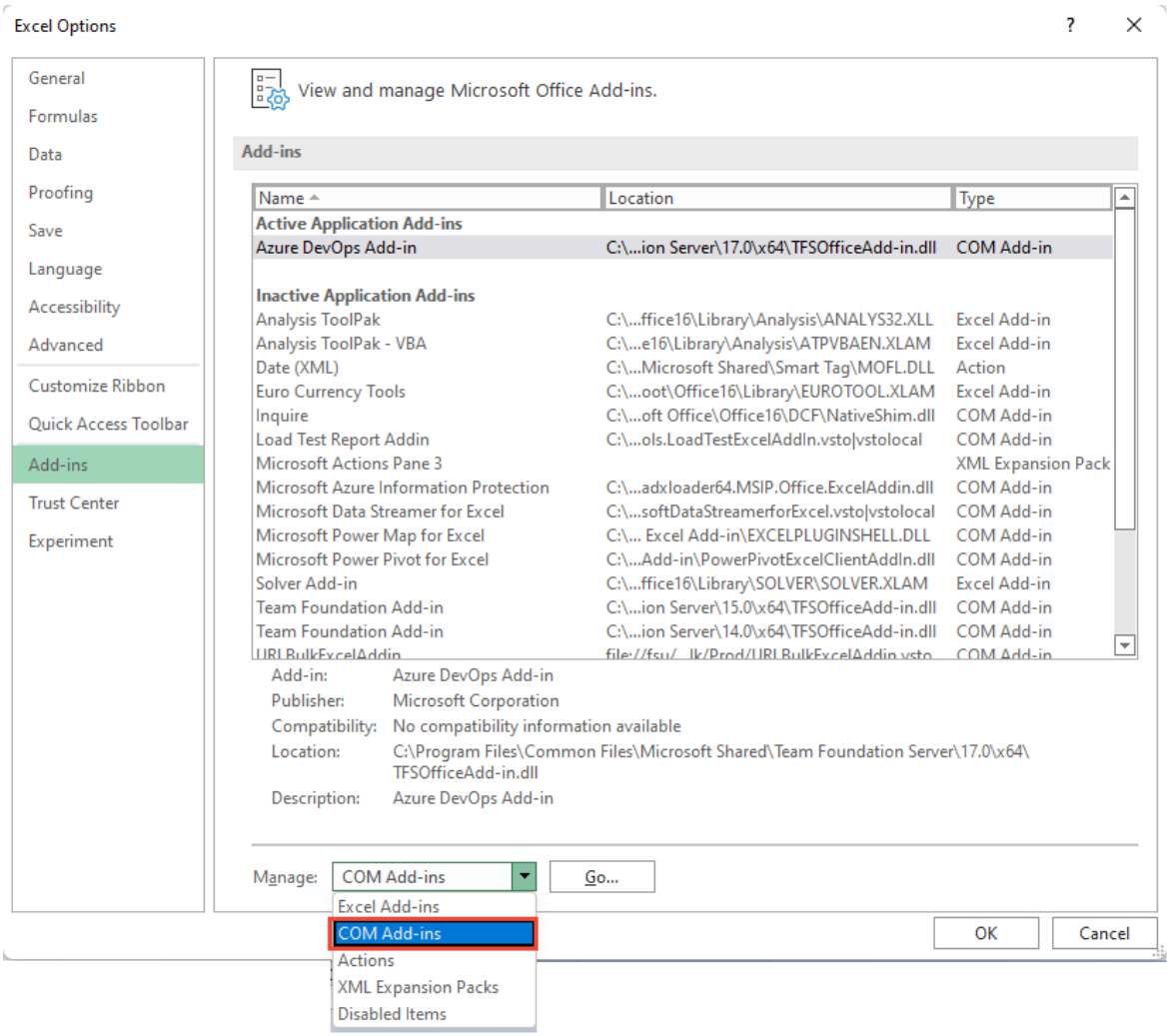
- Visual Studio 2019 and Azure DevOps Office® Integration 2019
- Azure DevOps Server 2019 and later versions, including Azure DevOps Services.

However, full support for Microsoft Excel integration is maintained and supports bulk import and update of work items. Alternatives to using Microsoft Project include the following:

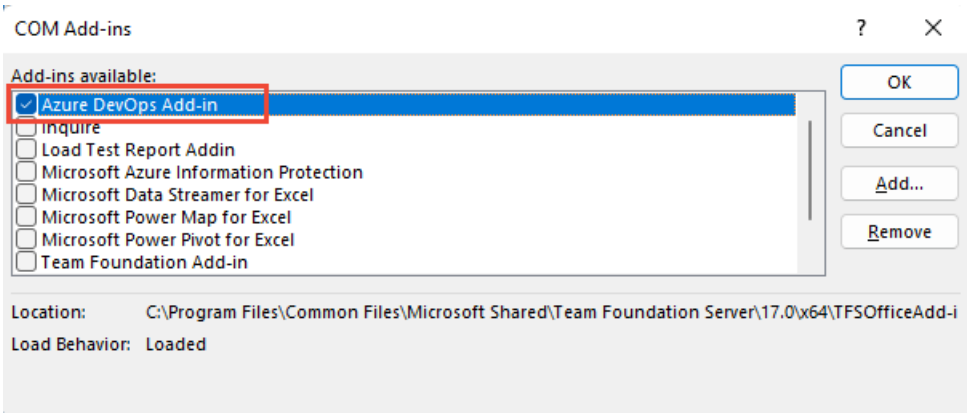
- Delivery plans
- A Marketplace extension such as Project Connect or GANTT chart.

Enable the Azure DevOps add-in

1. From the Excel **File** menu, choose **Options**.
2. Choose Add-ins and from the **Manage** picklist, choose **COM Add-ins**, and then choose **Go**.



3. Make sure that a check is placed in the **Team Foundation Add-in** checkbox.

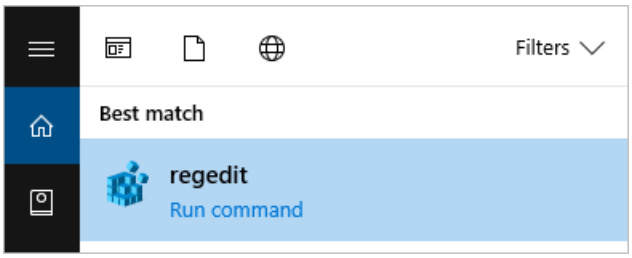


4. Restart Excel. You should now see the **Team** ribbon.

If the **Team** ribbon doesn't appear at next launch, the load behavior of the add-in may have changed and you'll need to complete the following steps:

Update the registry

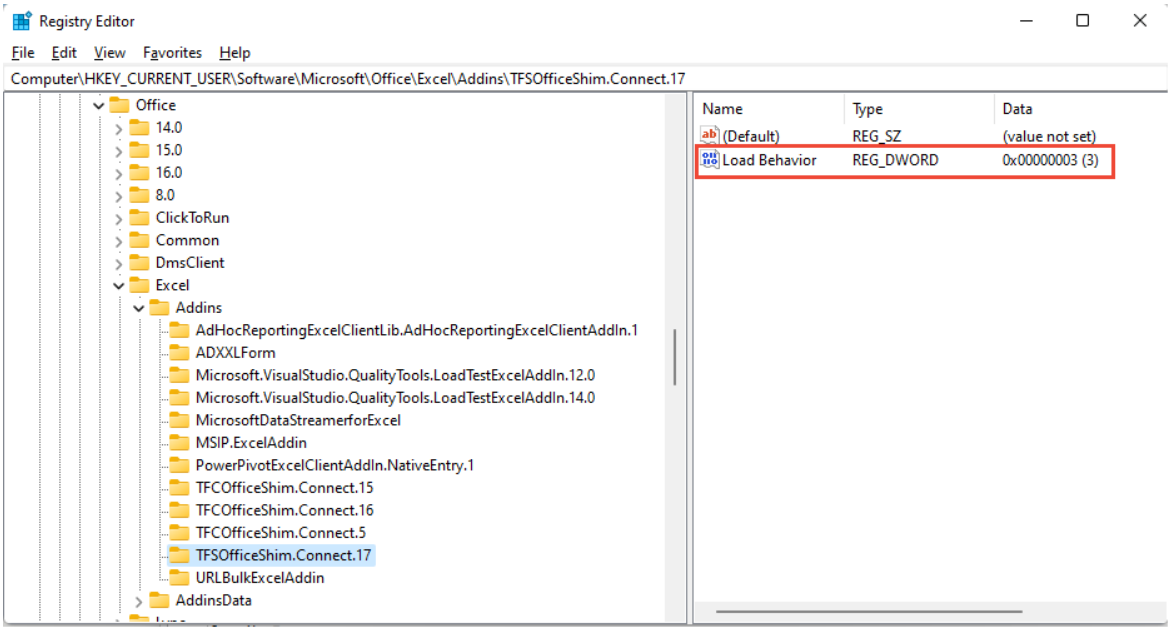
1. Launch the Registry Editor from your Windows Start Menu by typing `regedit` in the Search or Run box.



2. Navigate to one of the following paths containing the **TFCOfficeShim.Connect.[version]** folder:

> **NOTE**
> If there are multiple folders with the same name, select the one with the highest version number.

- `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\Excel\Addins` (if this key doesn't exist, try one of the options below)
- `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\Excel\Addins`
- `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Office\Excel\Addins`



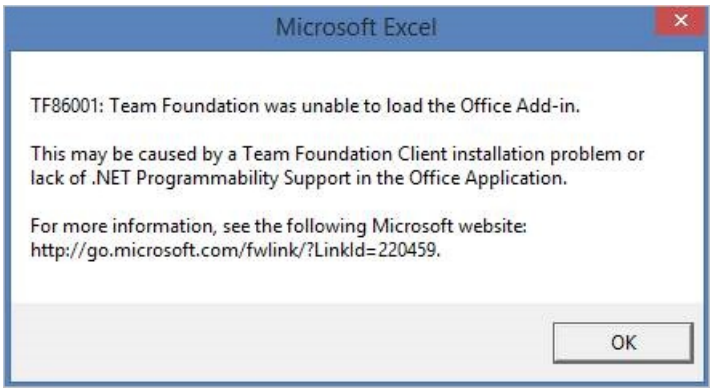
3. Double-click to open **LoadBehavior** and set the value data field to **3** (if the value is **0**, the **Team** ribbon won't load).

4. Press **OK** and restart Excel.

   To learn more about the LoadBehavior entry, see Registry Entries for VSTO Add-ins, LoadBehavior values.

Office add-in doesn't load or open in Excel when Visual Studio fails

To connect to Azure Boards, go to the **Team** ribbon and choose **New List**. If the New List dialog fails to open, or you receive TF86001 or similar error message, then you may need to repair Visual Studio.



This error is typically caused when you install Visual Studio before you install Office Excel or Project. In this instance, the Visual Studio Tools for Office Run Time aren't correctly configured. To correct this error, you must repair Visual Studio.

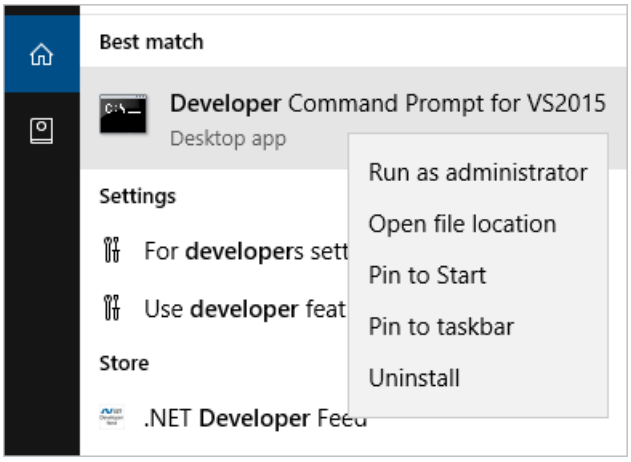
> **NOTE**
> For authentication issues, such as `TF31003` and `TF30063`, please refer to User account does not have permission.

Prerequisites

Install Visual Studio to ensure that you have access to the Visual Studio Command Prompt and the Gacutil.exe (Global Assembly Cache Tool). If you don't have Visual Studio, you can install the Visual Studio Community edition for free.

Run the Gacutil tool

1. Open the Visual Studio Command Prompt and choose to run it as an administrator.



2. **For Microsoft 365**, run the following commands:

```
GACUTIL /I
C:\Windows\assembly\GAC_MSIL\Policy.14.0.Microsoft.Office.Interop.Excel\15.0.0.0__71e9bce111e9429c\Policy.14.0.Microsoft.Office.Interop.Excel.dll
```

```
GACUTIL /I
C:\Windows\assembly\GAC_MSIL\Policy.14.0.office\15.0.0.0__71e9bce111e9429c\Policy.14.0.Office.dll
```

For Office 2016 and Office 2013, run the following commands:

```
GACUTIL /I
C:\Windows\assembly\GAC_MSIL\Policy.12.0.Microsoft.Office.Interop.Excel\15.0.0.0__71e9bce111e9429c\Po
licy.12.0.Microsoft.Office.Interop.Excel.dll
```

```
GACUTIL /I
C:\Windows\assembly\GAC_MSIL\Policy.12.0.office\15.0.0.0__71e9bce111e9429c\Policy.12.0.Office.dll
```

For Office 2010, run the following commands:

```
GACUTIL /I
C:\Windows\assembly\GAC_MSIL\Policy.12.0.Microsoft.Office.Interop.Excel\14.0.0.0__71e9bce111e9429c\Po
licy.12.0.Microsoft.Office.Interop.Excel.dll
```

```
GACUTIL /I
C:\Windows\assembly\GAC_MSIL\Policy.12.0.office\14.0.0.0__71e9bce111e9429c\Policy.12.0.Office.dll
```

3. Once you've successfully run the `GACUTIL` commands, restart Excel and look for the Azure DevOps Integration Tool for Office add-in.

If the above steps are unsuccessful, try the following steps:

1. Run a full repair of Office.
2. Uninstall Office and Reinstall Office.
3. Contact the Microsoft support team.

Related articles

- FAQs: Work in Excel connected to Azure Boards
- Add or remove add-ins

Resolve data conflicts when you publish or refresh Excel data

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

A data conflict occurs when you try to publish a work item from Excel and the version of that work item differs from the version in the work item database. The following example shows how two team members can create such a conflict.

1. A team member opens a copy of a work item in a work item list in Excel or Project.
2. Team member A edits the work item and makes one set of changes.
3. Team member B edits that same work item and makes a different set of changes, and publishes those changes.
4. Team member A finishes editing the work item and tries to publish the changes to the work item.
5. Excel or Project displays the **Work Item Publishing Errors** dialog box, which shows items that it couldn't publish.

To resolve a data conflict

1. In the **Work Item Publishing Errors** dialog box, for each work item in the **Unpublished work items** box that has **Conflict** in the **Issue** column, follow these steps.

   a. In the **Unpublished work items** box, select the work item.

      The **Details** area shows a list of conflicts for the selected work item. The **Conflicting field** column shows the name of the field in which the conflict occurs. The **Local version** and **Server version** columns show the local and server data, respectively, and a check box appears next to the data in each of these columns.

   b. For each row in the **Details** box, select the check box next to the correct value.

      When you select the local version, the data in Office Excel or Office Project overwrites the data on the server. If you select the server version, the server data overwrites the data in Office Excel or Office Project.

2. Select **Publish**.

> **NOTE**
> This step publishes only the work items that you corrected. If you do not resolve all data validation errors related to a work item, that work item is not published.

Related articles

- Resolve invalid links
- Resolve data validation errors
- Connect Azure Boards to an Office client](track-work.md)

Required permissions

To update work items, you must be a member of the **Contributors** group or have your **View work items in this node** and your **Edit work items in this node** permissions set to **Allow**. For more information, see Change project-level permissions.

Resolve data validation errors that occur when you publish from Excel

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

A data validation error occurs when a change in the work item list or project plan violates a work item type's rule. The following examples show common data validation errors:

- Someone assigns a work item to a team member whose name isn't included in the list of allowed values
- Someone creates a work item but forgets to complete a required field, such as the work item type.

  If a data validation error occurs when you try to publish changes, the **Work Item Publishing Errors** dialog box appears, and in the **Unpublished work items** list the **Issue** column shows **Validation error** or another phrase that contains **Invalid**.

Prerequisites

To update work items, you must be a member of the **Contributors** group or have your **View work items in this node** and your **Edit work items in this node** permissions set to **Allow**. For more information, see Change project-level permissions.

Resolve a data validation error

You can use the **Work Item Publishing Errors** dialog box to resolve a data validation error.

1. In the **Work Item Publishing Errors** dialog box, for each work item that appears with a data validation error, follow these steps.

   > **NOTE**
   > If the data validation error is an invalid work item type, the **Edit Work Item** button is not visible, and a work item form does not appear. You must correct the error in the Office Excel worksheet or the Office Project plan. For information about how to resolve an error in Office Excel, see the next procedure in this article.

   a. In the **Unpublished work items** box, select the work item, and then select **Edit Work Item**.

      A work item form appears.

   b. In the work item form, review the information and correct the value.

   c. Select **Close** to save your changes and close the work item form.

2. After you correct the data validation errors, select **Publish** to publish the corrected work items.

   > **NOTE**
   > This step publishes only the work items that you corrected. If you do not resolve a data validation error, that work item is not published.

3. Select **Close** to close the **Work Item Publishing Errors** dialog box.

Resolve a data validation error by using error checking in Excel

You can use the Office Excel tools to find and resolve an error in a work item list. For more information about how to use Office Excel error checking tools, see the Office Excel Help.

To resolve a data validation error by using error checking in Excel 2007

1. Start Excel, and select the **Formulas** tab.
2. In the **Formula Auditing** group, select **Error Checking**.

   If the error checking tool finds an error, the **Error Checking** dialog box appears.
3. For basic information about the error, see the text that describes the error in the **Error Checking** dialog box. For more information about the error, select **Help on this error**.
4. In the work item list, select the cell that contains the error, and then correct the value.
5. In the **Error Checking** dialog box, select **Resume** to find the next data validation error.

   - If the **Error Checking** dialog box shows another error, repeat the previous two steps to resolve the error.
   - If a message appears that indicates error checking has completed, select **OK** to close both this message and the **Error Checking** dialog box.

Related articles

- Resolve data conflicts
- Resolve invalid links
- Connect Azure Boards to an Office client](track-work.md)

Resolve invalid links in an Excel tree list in Azure Boards

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

If you try to publish a tree list that contains an invalid link, the **Work Item Publishing Errors** dialog box appears and displays an error message that states why the tree is invalid. When you work with work items in a tree in Excel, the tree must be in a valid state before it can be published. In Excel, an invalid link occurs in a tree list of work items. It occurs if the title of a work item title is missing or occurs in the wrong title column.

You can resolve most errors using the procedures provided in this article.

Prerequisites

To update work items, you must be a member of the **Contributors** group or have your **View work items in this node** and your **Edit work items in this node** permissions set to **Allow**. For more information, see Change project-level permissions.

Error message TF208000: Duplicate titles

If you add a value to multiple Title columns of a work item, when you try to publish the tree, the error message **TF208000** appears in the **Work Item Publishing Errors** dialog box. The error message specifies the row number of the invalid link.

1. Note the row number that appears in the dialog box.
2. Close the **Work Item Publishing Errors** dialog box.
3. In the work item list, find the row that the error message specified.
4. Delete the duplicate title or titles so that only one title column has a value.
5. On the **Team** tab, in the **Work Items** group, choose **Publish**.

Error message TF208001: Child work item disconnected from parent

If you remove the title of a child work item, when you try to publish the work item list the error message **TF208001** appears in the **Work Item Publishing Errors** dialog box. The error message specifies the row number of the invalid link. This error message also appears if you create an invalid link structure by putting the title of a child work item into the wrong column.

> **NOTE**
> If you put a title in the wrong column, the resulting tree structure might be valid but not match your intent. The system cannot detect this problem, therefore, an error message does not appear.

To resolve an orphaned work item

1. Note the row number that appears in the dialog box.
2. Close the **Work Item Publishing Errors** dialog box.

3. Find the row that the error message specified.

> **NOTE**
> The misplaced title might be in this row, or it might be in the next row.

4. Move the title to the correct column to fix the invalid link.
5. On the **Team** tab, in the **Work Items** group, choose **Publish**.

Error message TF208017: Missing Title 1 in the first row

If the first work item in the list has no value in the **Title 1** column, the error message **TF208017** appears in the **Work Item Publishing Errors** dialog box when you try to publish the work item list.

1. Close the **Work Item Publishing Errors** dialog box.
2. Determine why the first work item lacks a **Title 1** value.

   The analysis of the cause of the error determines what you must do to repair the work item list.

   a. If the first work item should be at the top of the list, move its title value into the **Title 1** column.

   b. If the first work item should be lower in the list, move the correct work item to the top of the list.

      For more information about how to move work items, see Bulk add work items with Excel.
3. On the **Team** tab, in the **Work Items** group, choose **Publish**.

Error message TF208022: Sorted tree list

If you haven't installed Service Pack 1 for Visual Studio 2010 or later version, the following error message appears when you choose **Publish** after you've sorted the work items in a tree list:

> TF208022: Cannot publish a sorted tree list. Before you can publish, you must clear any sort criteria applied to this work item list. Be aware that the order of work items has changed. Removing sort criteria will not return the list to its original order. Verify that all of the parent-child relationships in the tree are correct before you publish.

You can't publish your changes until you re-establish the tree hierarchy. You can resolve this error either by discarding your changes and refreshing the list or by manually restoring the hierarchy and then publishing the list.

To resolve sorted tree list issues

- Choose **Refresh** to discard your changes and restore the tree hierarchy.

  > **NOTE**
  > If you refresh the tree list, you remove all your changes other than the sort. To refresh the tree list, on the **Team** tab, in the **Work Items** group, choose **Refresh**.

- Manually restore the tree hierarchy by moving the row entries of child items under their parent items. Then, on the **Team** tab, in the **Work Items** group, choose **Publish**.

Error message TF208102: Excel sort on a tree list

The following error message appears if you sort the work items in a tree list in Excel:

> TF208102: You have performed an Excel sort on a tree list. This action removed the modified or newly introduced hierarchical link relationships of the tree.
>
> You can still publish the changes you have made to individual work items. After you publish, the list will restore to previous hierarchy.
>
> In general, you should not sort a tree list whose hierarchy has been modified.

This message indicates that you can publish the changes that you made to the fields, but all changes that you made to the link hierarchy have been discarded. The tree hierarchy automatically reverts to its original structure.

To publish changes and retrieve the tree hierarchy

1. On the **Team** tab, in the **Work Items** group, choose **Publish**.
2. Choose **Refresh**.

Error message TF208104: Hierarchical link Relationship is locked

If you publish a worksheet that contains work items that are synchronized with Project Server and whose hierarchical link relationships are locked (), the following error message may appear:

> TF208104: You have modified one or more hierarchical link relationships that may have been locked by other processes, such as Project Server.
>
> Changes that you made to individual work items were published. Changes that you made to locked links were auto-corrected.

This error appears when you change the link hierarchy that contains locked links. This message indicates that the changes that you made to the fields are published. All changes you made to the link hierarchy, whether locked or not locked, aren't published and were reverted to their original assignments.

To modify locked hierarchical, make your changes in the enterprise project plan mapped to the project. For more information, see Manage project details.

To publish changes to links that aren't locked

- For work items that aren't synchronized, you can modify the hierarchical link relationship from Team Explorer or the web portal. For more information, see Bulk add or modify work items with Excel.
- To modify unlocked hierarchical link relationships in Excel, revise the query that you use to export the work items to exclude all work items with locked links. For example, you can add a clause to the filter criteria to omit items whose **Project Server Is Linked** field is set to **Yes**.

Related articles

- Resolve data validation errors
- Resolve data conflicts
- Connect Azure Boards to an Office client](track-work.md)

Configure or add a project portal

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The project portal is a site associated with a team project for the purposes of sharing information.

> **IMPORTANT**
>
> Configuring the project portal and process guidance features have been deprecated. You can only set them from Visual Studio 2017 or earlier versions and when connected to TFS 2017 or earlier versions. SharePoint integration with Azure DevOps (formerly named Team Foundation Server) was deprecated with the release of TFS 2018. For TFS 2018 and later versions, you can use the built-in wiki feature to share information, guidance, and documents. To learn more, see About Wikis, READMEs, and Markdown.

Related articles

- Choose a process
- About SharePoint integration
- Discontinue SharePoint integration: TFS 2017 and earlier versions

Overview of extensions for Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Azure DevOps Marketplace offers a wide variety of extensions to customize or enhance the default experience. You can learn more about those extensions developed by Microsoft from the following articles and links. For information on developing your own extension, see Develop a web extension.

> **NOTE**
> Most extensions, such as Feature Timeline and Epic Roadmap and Delivery Plan, are not supported features of Azure Boards and therefore not supported by the product team. For questions, suggestions, or issues you have when using these extensions, visit their corresponding extension page.

Integrate with Azure Boards

- Azure Boards Microsoft Teams
- Azure Boards Slack

Product planning

> **NOTE**
> A new version of Delivery Plans is available. This feature is now part of Azure Boards and replaces the Delivery Plans extension. Delivery Plans provides support for the following tasks:
>
> - Add up to 15 team backlogs
> - Add custom portfolio backlogs as well as Epics
> - View work items that span several iterations
> - Reset Start Date and Target Date through drag and drop borders
> - Add backlog items to a team from a plan
> - View rollup progress of Features, Epics, and other portfolio items
> - View work item dependencies
> - Stakeholders can view plans

To learn more, see Review team Delivery Plans

- Azure DevOps Open in Excel
- Feature Timeline
- Epic Roadmap
- Retrospectives
- Split!
- Team Calendar
- Reactivations Report

- Azure DevOps Open in Excel
- Delivery Plans
- Feature Timeline

- Epic Roadmap
- Retrospectives
- Split!
- Team Calendar
- Reactivations Report

- Azure DevOps Open in Excel
- Delivery Plans
- Feature Timeline
- Epic Roadmap
- Retrospectives
- Split!
- Team Calendar
- Reactivations Report

- Delivery Plans
- Feature Timeline
- Epic Roadmap
- Retrospectives

Customizing work item types

- Cascading Lists
- Clickable Links
- Color picklist control
- Multi-value control
- Work item form control library Work item form one click actions
- Work Item Visualization
- WSJF (Weighted Shortest Job First)

- Color picklist control
- Multivalue control
- WSJF (Weighted Shortest Job First)

In addition to the above extensions, you can

Query and reporting

- Open in Power BI
- Query Based Boards
- Tags Manager
- Wiql Editor
- Wiql to OData

Dashboards

- Azure Application Insights Widgets
- Countdown Widget
- GitHub Stats Widget
- Work Item Details Widget

- Roll-up Board Widget
- Roll-up Board Widget

DevOps integration

- Creates a new work item from a build or release

Command-line interface

- Azure DevOps CLI
- Azure Boards Teams Tool CLI
- Azure Boards Teams Tool CLI

Automation and miscellaneous tools

- Azure Boards Kanban Tools
- Migration Tools for Azure DevOps
- Microsoft Flow, Power Apps and Power Automate
- Power Automate, Azure DevOps

Related articles

- Cross-service integration overview
- Power Platform, Connectors, Azure DevOps
- Azure DevOps Work Items Microsoft Graph connector
- Connect to Azure DevOps from Power Apps

Migrate and integrate work tracking data in Azure Boards

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You have a choice of tools to help you migrate your work tracking data to the Azure DevOps platform. This article provides an overview of what's available and links to tools that support migration of work tracking data and processes. You can also integrate Azure Boards with many third-party tools.

Migrate from an on-premises Azure DevOps server

The data migration tool for Azure DevOps provides a high fidelity way to migrate collection databases from Azure DevOps Server to Azure DevOps Services. To learn more, see Migrate data from Azure DevOps Server to Azure DevOps Services.

Migrate data between Azure DevOps Projects

The Azure DevOps Migration Tools marketplace extension allows you to bulk edit and migrate data between Azure DevOps Projects on both on-premises and the cloud. This tool supports the following tasks:

- Migrate work items from one project to another project and synchronize changes after a migration
- Merge many projects into a single project
- Split one project into many projects
- Assist changing process templates
- Edit work items in bulk
- Migrate test suites and test plans

To learn more, see Azure DevOps Migration Tools.

OpsHub Azure DevOps Migrator (OADOM) formerly known as OpsHub Visual Studio Migration Utility(OVSMU) enables a seamless migration between Azure DevOps Server(TFS) and Azure DevOps Services(VSTS) or Azure DevOps and Azure DevOps. Migrate projects from DevOps Services to a collection on DevOps Server including:

- All version control information and history
- All work items and history
- All test cases and test results
- Supports the 2010, 2012, 2013, 2015, 2018, and 2019 versions of Azure DevOps Server
- Supports all versions of Azure DevOps Services

OADOM covers the following scenarios:

Supports data migration from Azure DevOps Server (TFS) to Azure DevOps Server (TFS), Azure DevOps Services (VSTS) to Azure DevOps Server (TFS) and Azure DevOps Services (VSTS) to Azure DevOps Services (VSTS).

Migrate process models between Azure DevOps organizations

The Process Tools for Azure DevOps marketplace extension provides support to complete the following tasks:

- Migrate Inherited Processes between Organizations
- Upload an On-premises XML process model from an export

For constraints on process template definitions that you can import, see Resolve validation errors for process import.

Migrate between legacy tools and Azure DevOps

The OpsHub Migration Manager marketplace extension supports full fidelity, large-scale data migration projects out of the box between legacy or ALM tools and Azure DevOps. OMM offers the following benefits:

- Zero downtime migration or non-disruptive migration
- Factory approach for large-scale migration projects
- Full fidelity migration with history preservation for all artifacts, comments, attachments, and relationships

Export and import work tracking data

The main tool you can use to import work tracking data you've exported from elsewhere is Microsoft Excel. Excel supports publishing a flat list of work items or a hierarchical tree of parent-child linked work items. To learn more, see Bulk add or modify work items with Excel.

If you want to migrate from Jira to Azure Boards, consider using the Marketplace extension, Jira to Azure DevOps work item migration tool. This tool provides support for the following tasks:

- Export Jira issues from Jira queries
- Map users from Jira to users in Azure DevOps
- Migrate work item field data
- Migrate links and attachments
- Migrate history

Integrate with GitHub

Azure Boards and Azure DevOps Server 2019 integrate with GitHub. By connecting Azure Boards with GitHub repositories, you enable linking between GitHub commits, pull requests, and issues to work items. You can use GitHub for software development while using Azure Boards to plan and track your work.

To learn how, see Connect Azure Boards to GitHub.

Integrate with third-party tools using service hooks

You can integrate Azure Boards with other third-party tools such as Jenkins and Trello. Using service hooks you can generate actions based on Azure DevOps events. To learn more, review one of the following articles:

- Create a service hook for Azure DevOps with Jenkins
- Create a service hook for Azure DevOps with Trello

Project and portfolio management

The following tools support integration and synchronization with Azure DevOps with one or more project and portfolio management tools. These tools also support integration between Azure Boards and Microsoft Project Online or Microsoft Project Server.

- OpsHub Integration Manager(OIM) is an enterprise class integration platform. It enables bi-directional synchronization for Azure DevOps(Server and Services) with 50+ ALM, DevOps, Project, and Portfolio Management tools out of the box. It enables synchronization of all work items including test artifacts with an

easy to use mapping UI and is scalable for any number of projects and teams.

- ConnectAll helps organizations achieve effective Value Stream Management by connecting the applications used to collaborate, drive decisions, and manage artifacts that are used during the software delivery process, like ALM, Agile, and DevOps.
- Project Connect offers a seamless approach to exchange critical information between development teams and project teams using Microsoft Project Online and Azure DevOps Services.
- Tasktop provides enterprise-grade integration between Microsoft Project Server, Azure DevOps, and to most other popular Agile, DevOps and Systems Development Lifecycle (SDLC) tools.

Related articles

- Migrate data from Azure DevOps Server to Azure DevOps Services
- Integrate with service hooks

Delivery Plans 1.0 Marketplace extension

10/4/2022 • 10 minutes to read • Edit Online

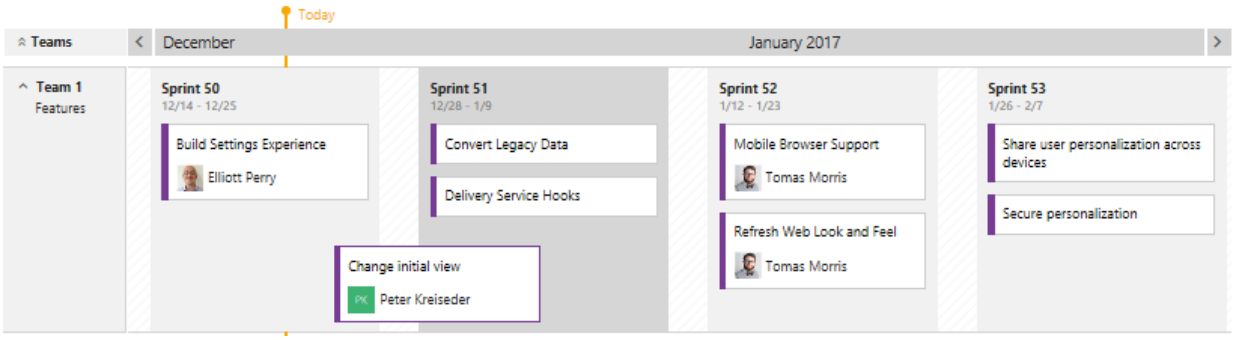
Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use the visualization options that Delivery Plans provide to review the schedule of stories or features your teams plan to deliver. Delivery Plans show the scheduled work items by sprint (iteration path) of selected teams against a calendar view.

> **NOTE**
> Delivery Plans 2.0 is available for Azure DevOps Services and Azure DevOps Server 2022 and later versions. The new version of Delivery Plans supports several new features and is a supported feature of Azure Boards. It isn't an extension. The plans you have already defined with the extension will open using Delivery Plans 2.0. For a summary of the differences between the two versions, see Delivery Plans FAQs.

Use Delivery Plans to ensure your teams are aligned with your organizational goals. You can view multiple backlogs and multiple teams across your whole account. You can interact with the plan with simple drag-and-drop operations to update or modify the schedule, opening cards, expanding and collapsing teams, and more.

You can change the assigned sprint of a work item by dragging it to a new sprint as shown in the following image.



In this article you'll learn:

- How to review a plan with your teams
- How to add and edit a plan
- How to add field criteria, customize cards, and add markers
- How to open a plan from the list of defined plans
- How to work with the interactive elements of plans and change the plan view
- What permissions are required to create and view plans

Prerequisites

- To add and configure a Delivery Plan, you must have the following criteria in place:
  - Installed the Delivery Plans extension from the Visual Studio Marketplace.
  - Be a member of a project and granted Basic access or greater access level. Users granted **Stakeholder** access can't add nor view plans.
  - Configured teams

    - Define area paths and assign to a team
    - Define iteration (sprint) paths and configure team iterations
    - Teams have defined user stories, features, or other product or portfolio backlogs and assigned those items to iterations.
    - Team Backlog settings have enabled the backlogs to show in the delivery plans. To learn more, see Select backlog navigation levels for your team.
- To view a Delivery Plan, you must be a member of the Project Collection Valid Users group. Members of the project's Readers group are valid users. Users with Stakeholder access for a private project can't view or add plans.
- To manage permissions for a Delivery Plan, to edit, or to delete a plan, you must:
    - Be the creator of the plan
    - Be a member of the Project Administrators or Project Collection Administrators group
    - Be granted explicit permission through the plan's Security dialog.

  For details, see Edit or manage Delivery Plan permissions.

Review a plan with your teams

It takes several teams to develop large software projects. Large projects require multiple autonomous teams. Review Agile culture for a discussion of autonomous teams and organizational alignment. Autonomous teams manage their own backlog and priority, which contributes to a unified direction for that project.

Regular reviews of the project schedule with these teams help ensure that the teams are working toward common goals. Delivery Plans provide the needed multi-team view of your project schedule.

Some questions you might address during the review:

- *How confident are the teams in meeting the deliverables scheduled for each sprint?*
- *Are dependencies across teams adequately addressed via the planned deliverables?*
- *Are there gaps in the schedule, where no deliverables are scheduled? What's the cause? Can this risk be mitigated?*

For example, we use Delivery Plans internally to share the schedule of Features. By seeing the work many teams have planned for the next three sprints, we can easily discuss whether these priorities are the right priorities and if dependencies exist.

In this way, a Delivery Plan is a driver of alignment while allowing each team to keep a strong sense of autonomy. Individual teams can work to different sprint cadences, if needed, and manage different work item types—stories, features, or epics. Their work is all visible with the same plan view. Teams can even be part of different projects if they use different processes. You can also customize the card fields so that you only see the data fields of interest and applicable per work item type.

Learn best practices

- Use a consistent sprint schedule across your project teams and organization. Use the same sprints for backlogs, features, and epics. Don't create specific sprints for epics or other portfolio backlogs.
- Minimize the number of fields you choose to display on your cards.
- Eliminate cross-team ownership of area paths. Cross-team area path ownership isn't recommended and can lead to undesirable edge cases.

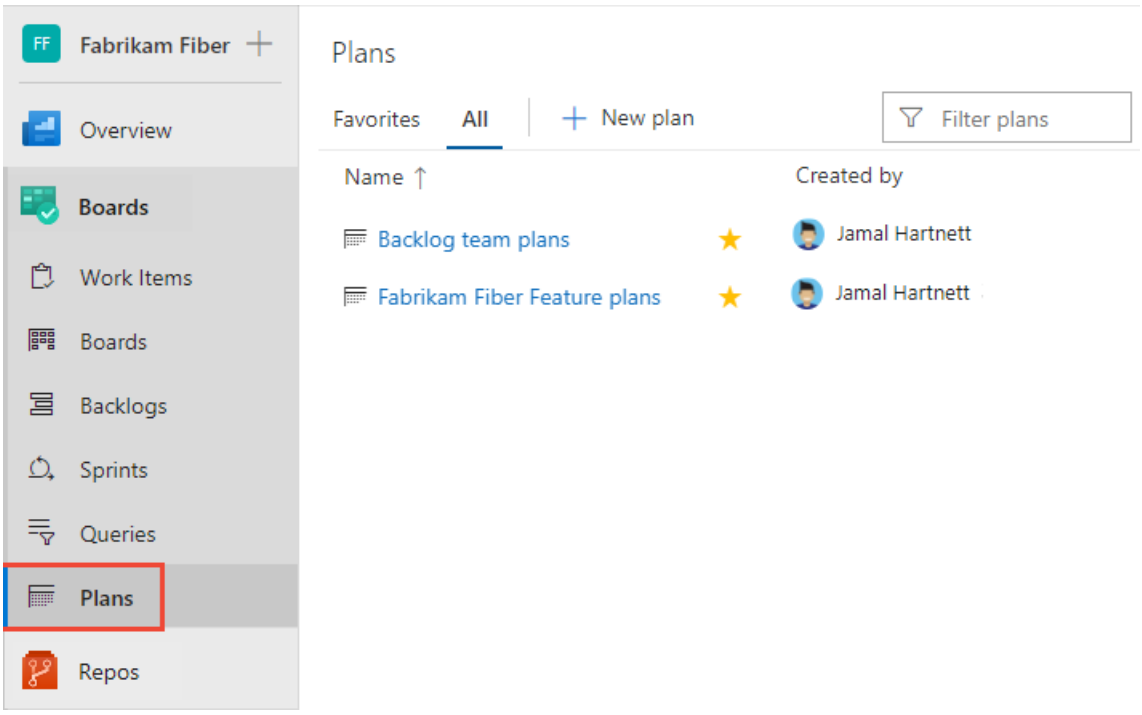
Note the following criteria:

- Plan views are limited to 12 to 13 months.
- Plan views are limited to a maximum of 10 teams/backlogs

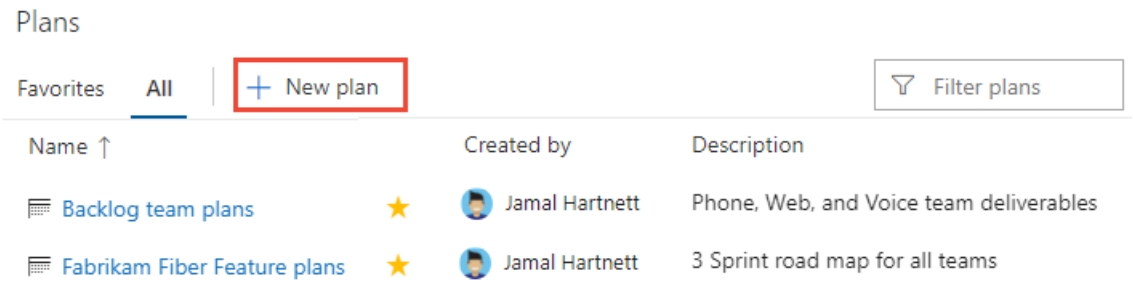
- Zooming out can cause fields and tags to disappear from the cards. The further you zoom out, the harder it's to fit items on a card. By design, we hide select items depending on the zoom level.

Add a plan

1. Open **Boards > Plans**.

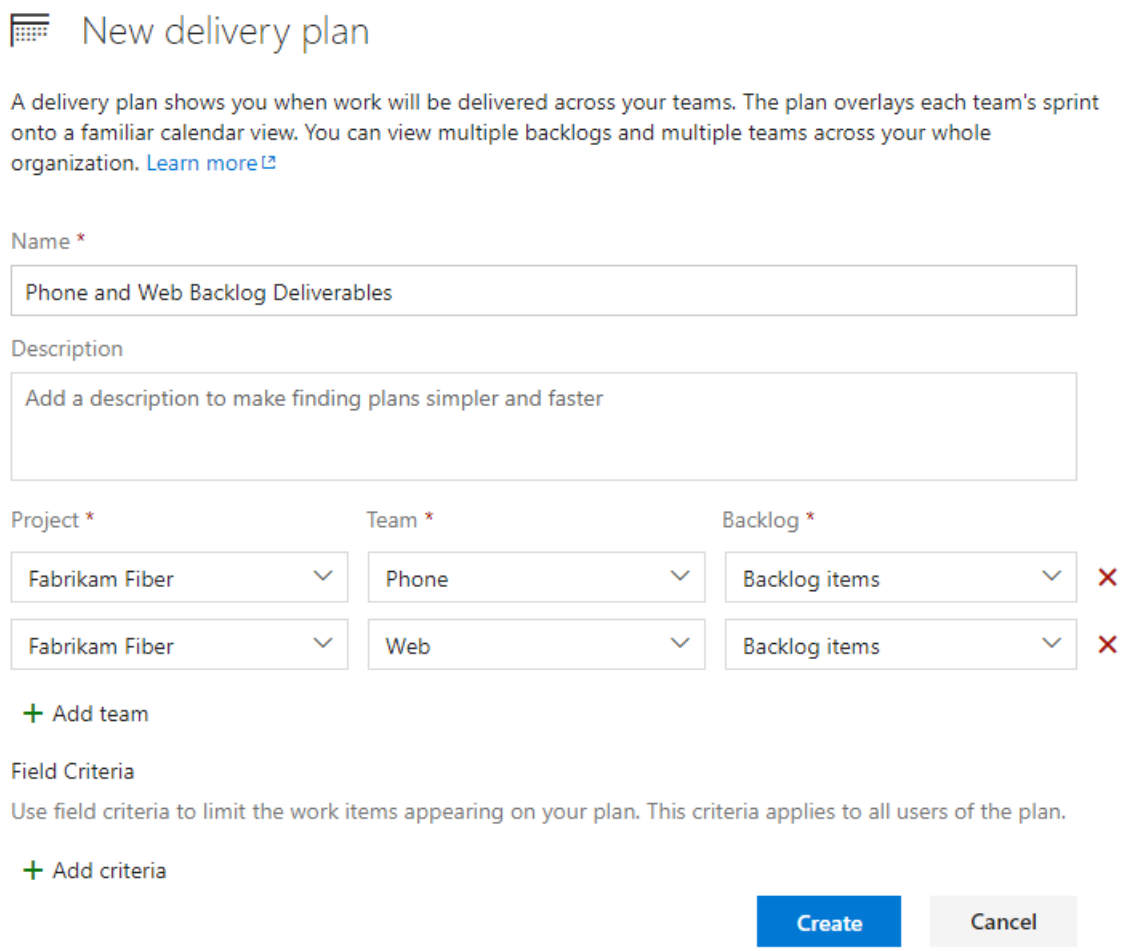


2. To add a plan, choose **New Plan**.

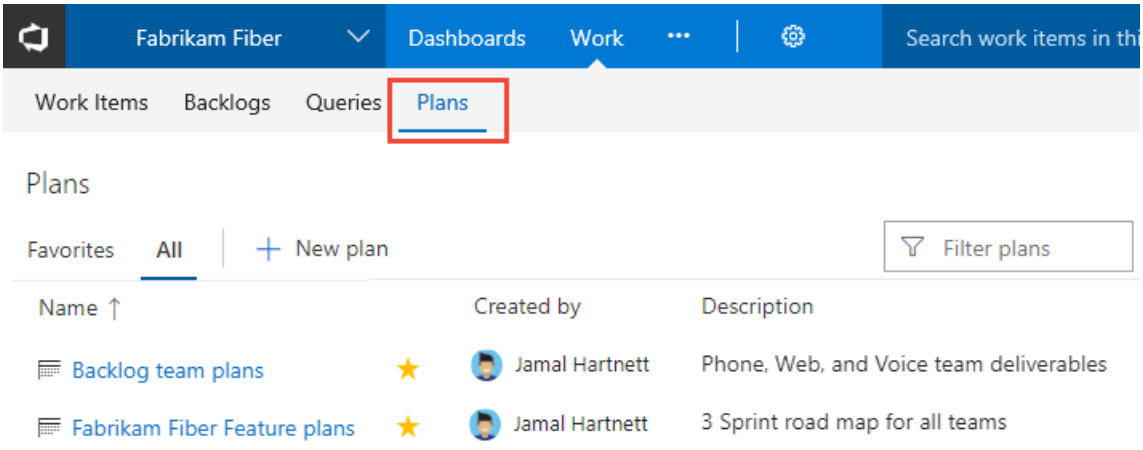


   All users, except users assigned Stakeholder access, have permissions to create a plan and manage the plans they create. To manage permissions for a plan, see Set permissions and access for work tracking, Manage or edit Delivery Plans.

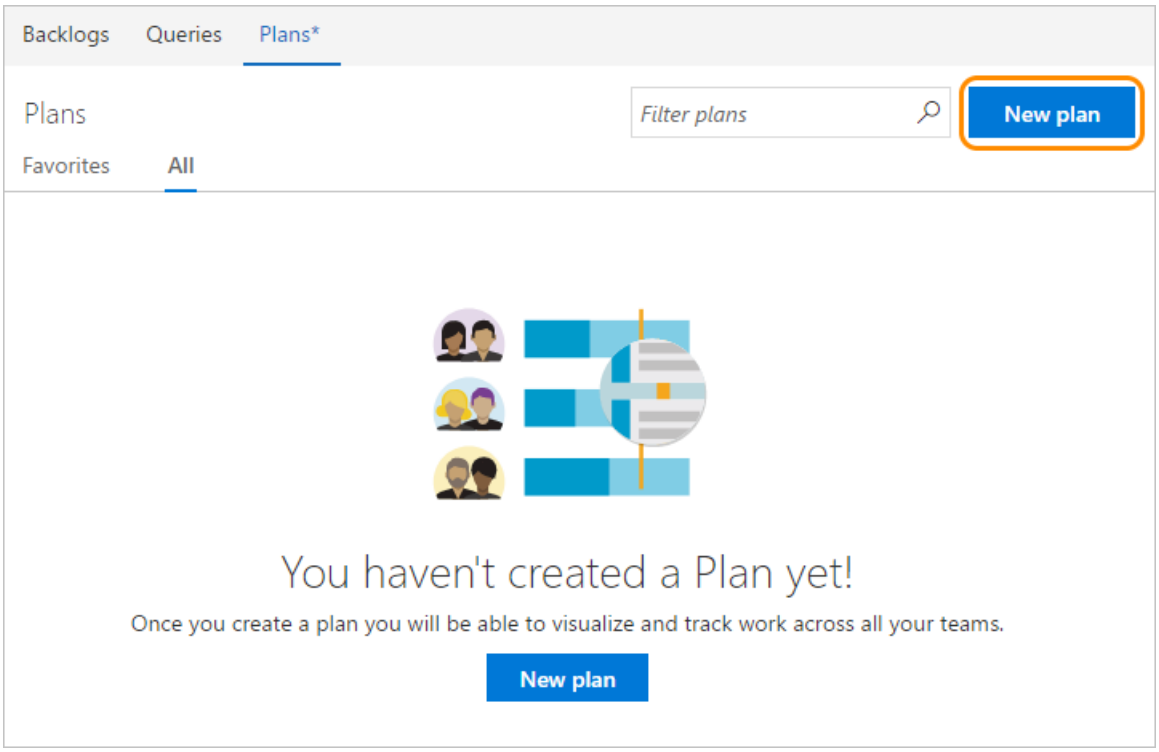
3. Fill in the form to name, describe, and specify the team backlogs that you want to appear within your plan.



1. Open **Boards > Plans**.

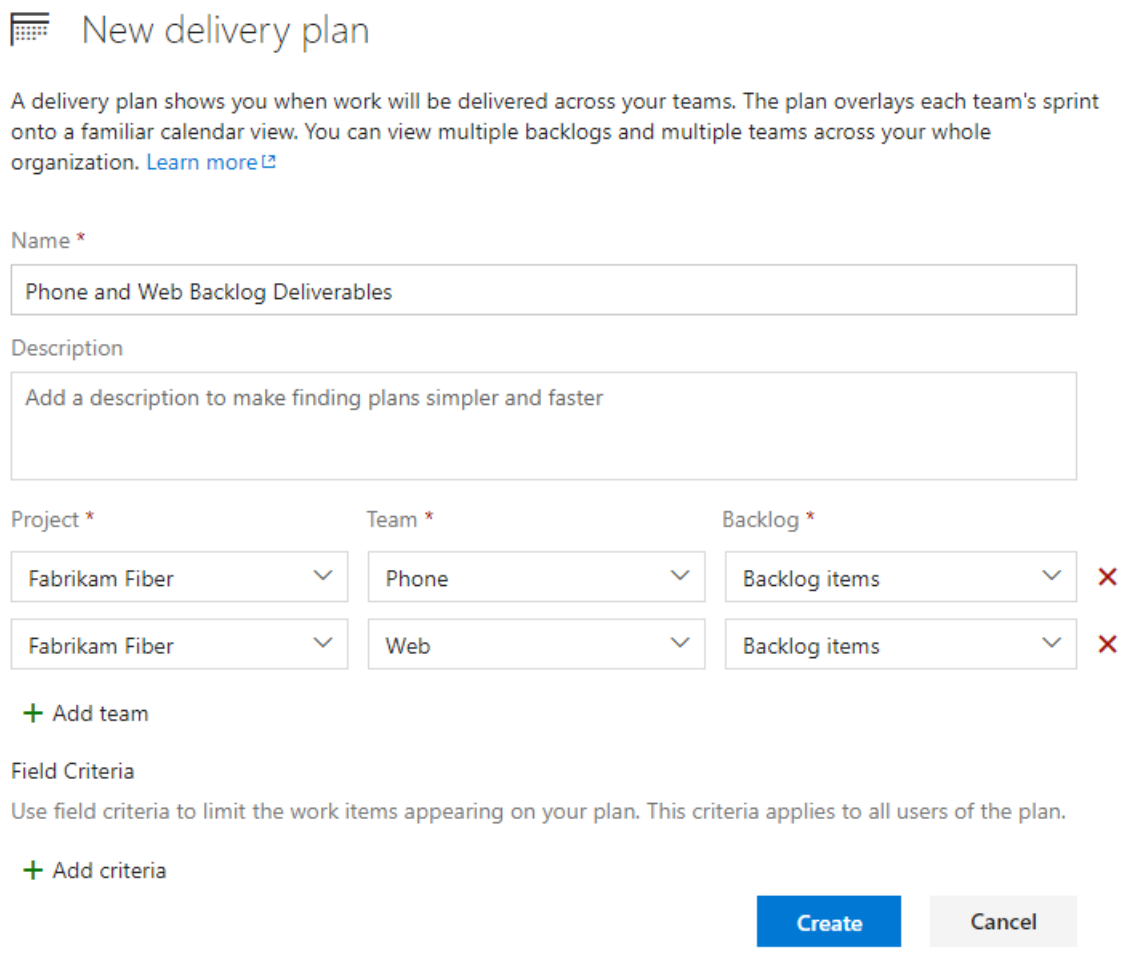


2. To add a plan, choose **New Plan**.



All users, except users assigned Stakeholder access, have permissions to create a plan and manage the plans they create. To manage permissions for a plan, see Set permissions and access for work tracking, Manage or edit Delivery Plans.

3. Fill in the form to name, describe, and specify the team backlogs that you want to appear within your plan.



When defining a plan, note the following criteria:

- Use the name and description field to clearly identify your plan within the project
- You can choose one or more teams from any project defined in the organization or collection. Choose up to a maximum of 10 teams.
- You can choose one or more active backlogs for a team

> **NOTE**
> If you aren't able to select a backlog level, check the Team Backlog settings to ensure the backlog level is enabled for the team. To learn more, see Select backlog navigation levels for your team.

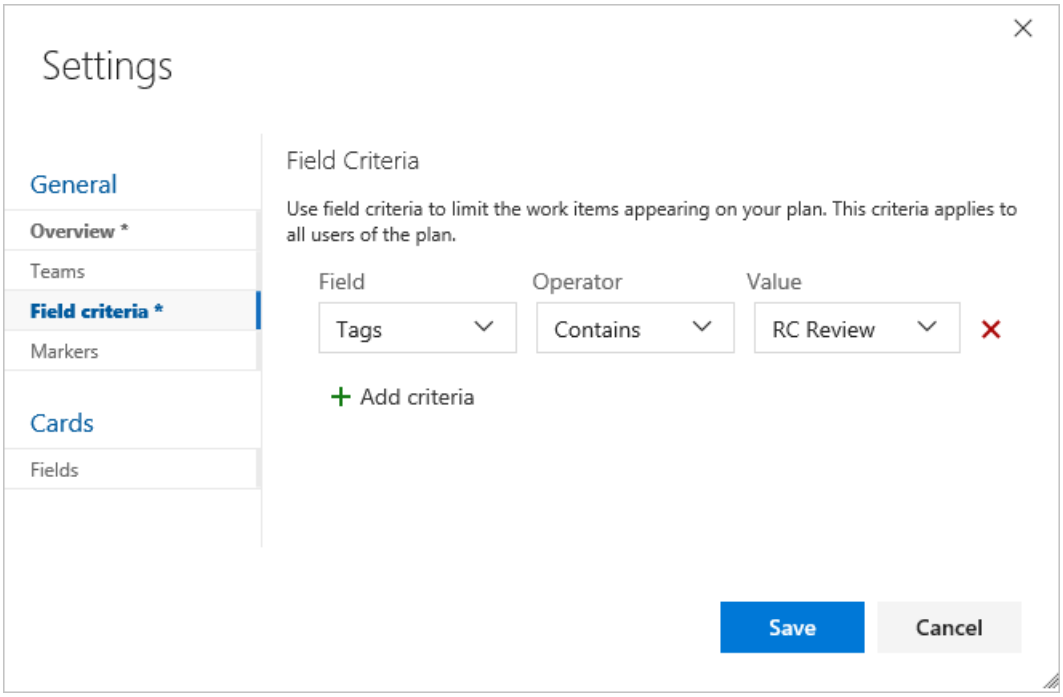
- You can reorder the team backlogs by dragging and dropping them into the sequence you want
- To filter for specific work items, specify the field criteria. For example, to exclude bugs from the view, add the following criteria: `Work Item Type <> Bug`.

Edit a plan

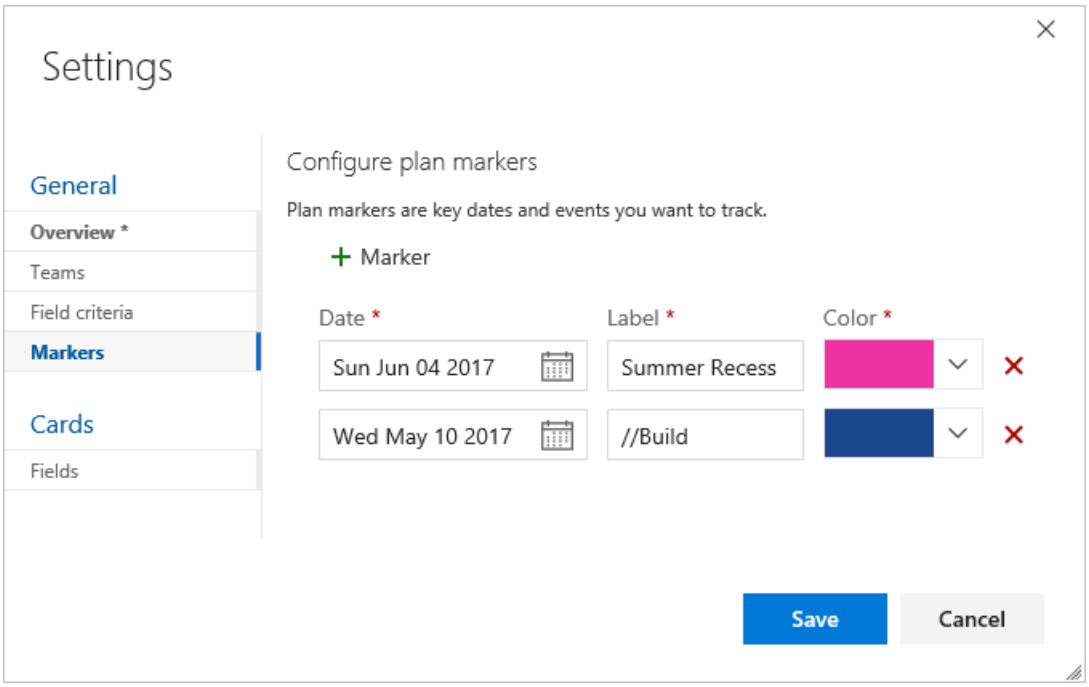
Once you've defined a plan, you can further customize it.

1. To open the Settings dialog, choose **Configure plan settings** .
2. Then, choose the page you want to edit. You can customize the plan in the following ways:
   - Edit the teams you've selected and their backlog level
   - Set field criteria to further limit the work items that will appear on the plan
   - Add markers to show important upcoming events on your timeline
   - Customize the fields that display on the cards, similar to how you customize them for your Kanban or taskboard.

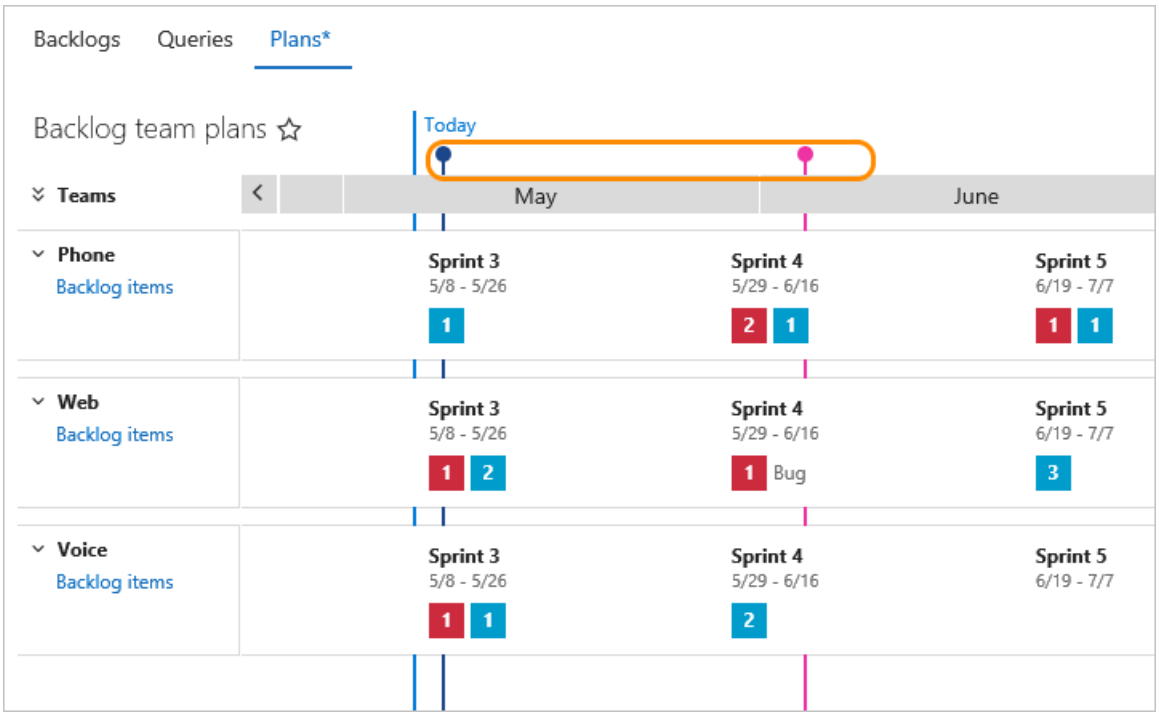
     Here, we add the Tags field criteria. Only work items that contain the *RC Review* tag will appear in the Delivery Plan.



3. To set a marker, open the Markers page, specify a date, and select a color.



Markers appear on the plan as shown:



Open a plan

Once you've defined a few plans, you'll see them listed from the **Plans** page under **All**, or the ones you've favorited (**Add to favorites** ☆) under **Favorites**. You can see their title, description, and their most recent creator/editor.

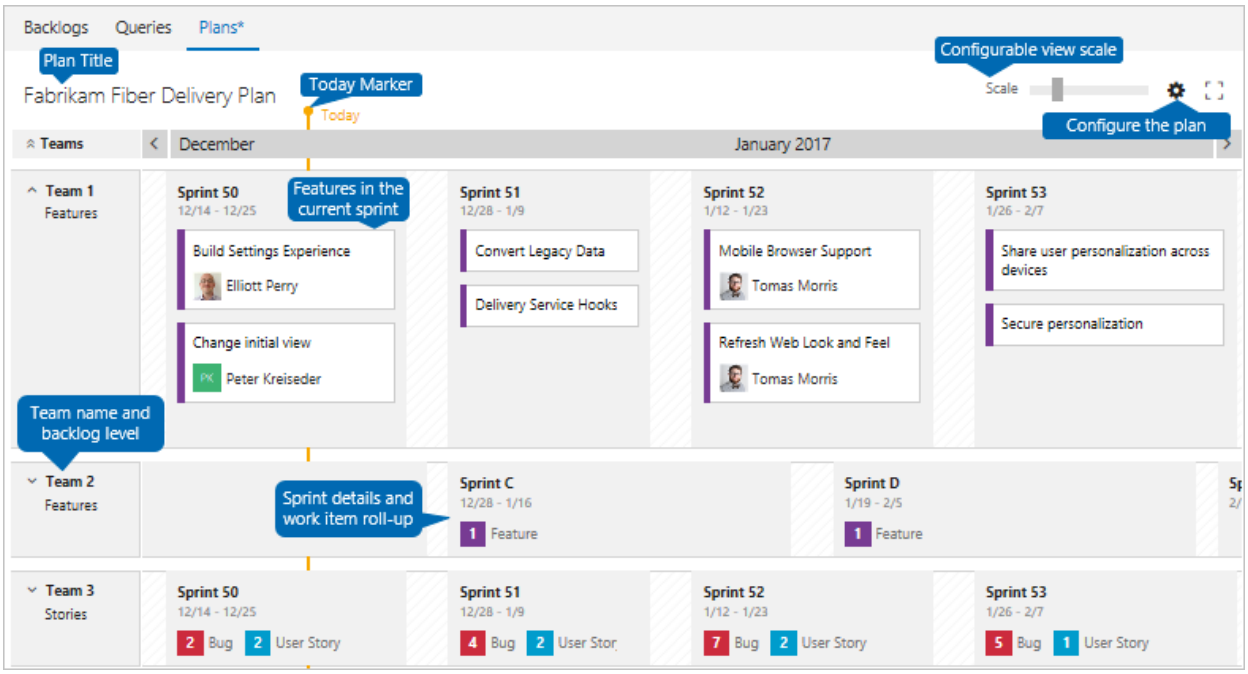
Use the favorite's star to favorite a plan so that you can quickly return to that plan. You can also search for other plans in the project.

To open a plan, choose the plan name.

Interact with a plan

Each team's backlog specified in a Delivery Plan appears as a row within the plan view. When collapsed, a roll-up of the backlog items shows. When expanded, a card for each backlog item appears, organized by their assigned

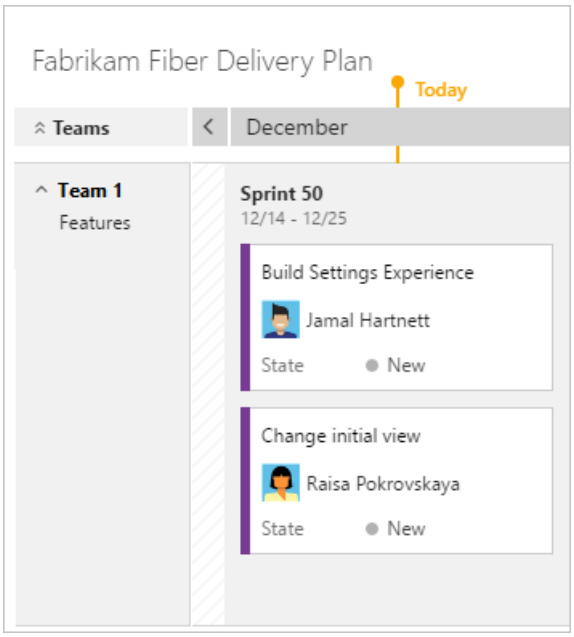
iteration.



You can interact with the plan in the following ways:

- Filter the plan by choosing Choose **Filter** . For more information, see Interactively filter your backlogs, boards, and plans.
- Scale the size of the cards (enter **+** or **-** to also scale)
- Scroll the view horizontally via the calendar to view previous months or future months (Enter **Shift-left** or **Shift-right** to scroll)
- You can also scroll the plan via click and dragging your mouse.
- Scroll the view vertically to view teams that appear lower within the plan view. You can also scroll the plan by clicking the plan and dragging your mouse vertically.
- Select a card title to open the backlog item and view details. Close the work item to return to the plan.
- To view details for a team, expand the team row by choosing **Expand team row**.
- To expand and collapse all team rows, choose **Expand all team rows** or **Collapse all team rows** next to **Teams**. (Enter **u** to collapse, **o** to expand all rows).
- Enter **t** to quickly toggle fields shown on cards to just the Title or all other card field settings
- To edit the plan and change the fields displayed on the cards, choose **Configure plan settings** .

In the following example, Team 1 features expanded, two items are scheduled for delivery in Sprint 50.



You can also quickly see that:

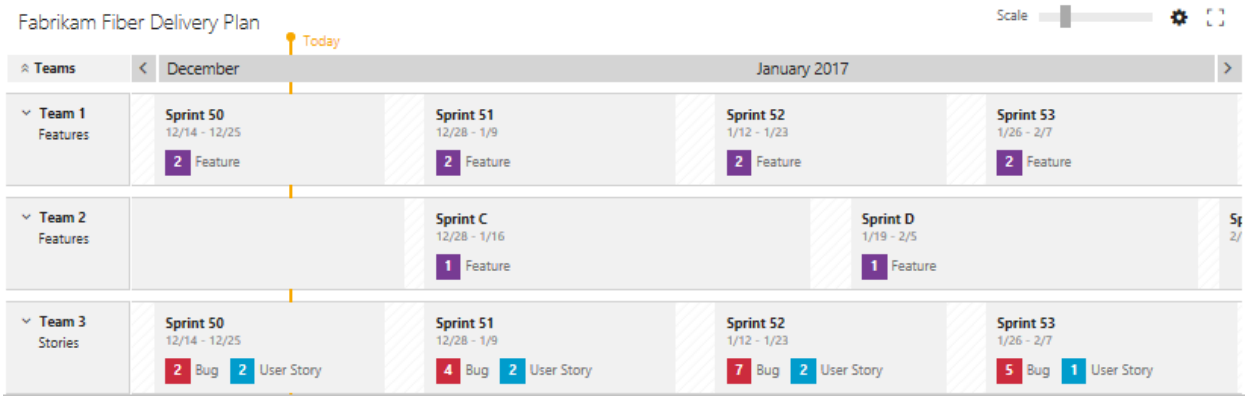
- Team 1 is currently working on Sprint 50 deliverables
- Sprint 50 began on December 14 and is scheduled to end December 25.
- The team expects to deliver the two features shown by the cards.

Collapse teams for summary information

A benefit of Delivery Plans is to view multiple teams across your projects that you care about. Two main ways to view more teams within the plan view is to collapse all teams to focus on summary data. You can also minimize the number of fields displayed on cards.

To gain a summary view of work that is schedule, collapse all teams. You can then more easily look for gaps in the forward forecast.

For example, here you can see the count of Features for Team 1 for the next three sprints. You can collapse/expand team rows by choosing **Expand team row** or **Collapse team row** next to the team name.

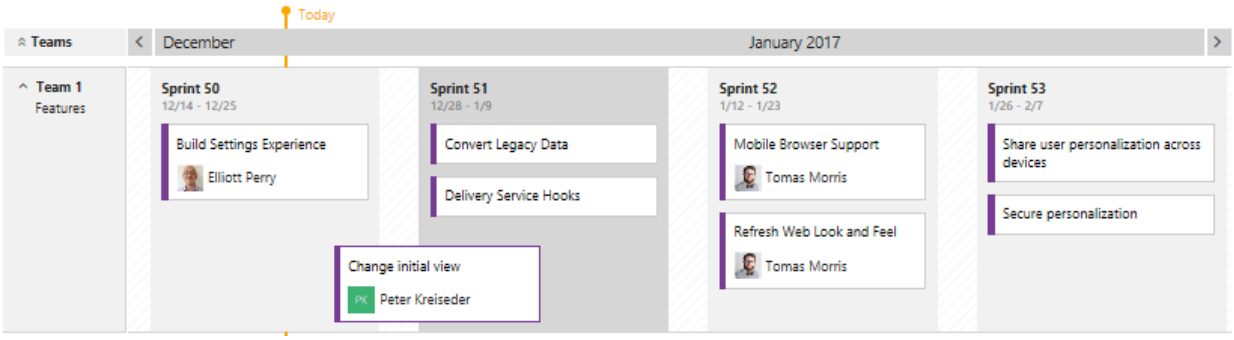


View titles only in the collapsed card view

The collapsed card view allows you to quickly switch back and forth between cards showing titles only and cards showing all fields configured for the plan. To toggle between titles only and show all fields, enter "**t**" from your keyboard.

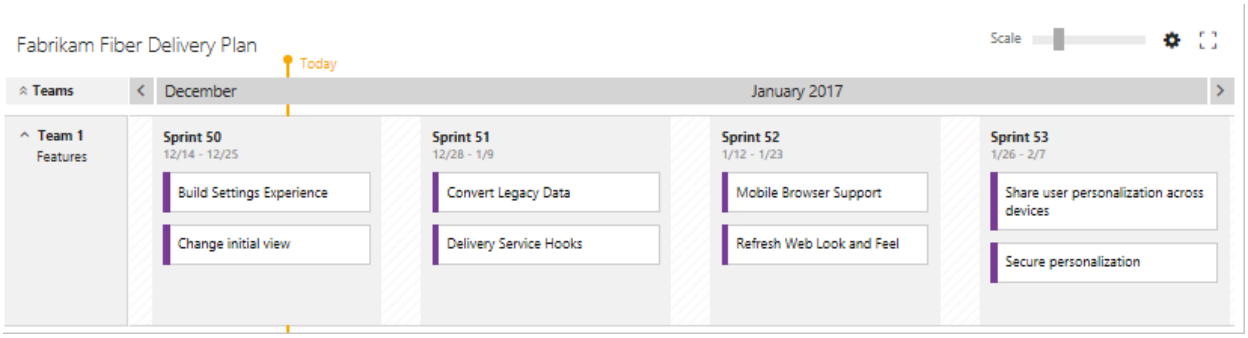
Update the iteration for a backlog item

As changes occur to the schedule, you can update the iteration for a backlog item. Update the iteration by moving a card to a different iteration. This action will help to drive alignment across your organization.



Minimize the fields displayed on cards

To quickly change the cards to only show their Title, enter the keyboard shortcut '**t**'. This will hide all other fields, as shown in the following image, the Assigned to field no longer appears. To persist this view, edit the plan's settings for card fields.

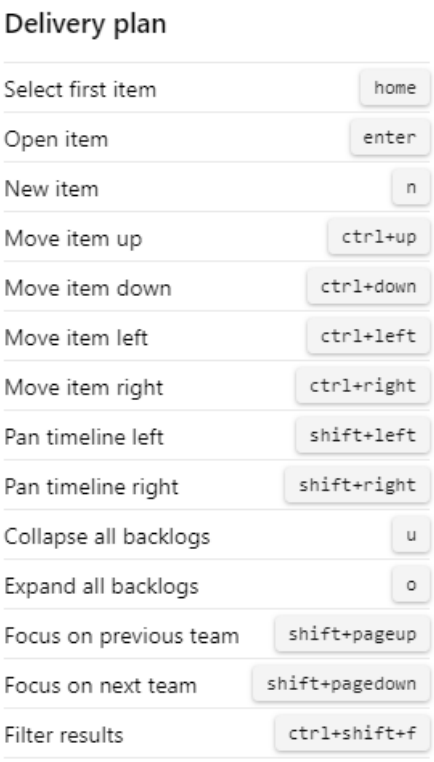


Learn keyboard shortcuts

You can use the following keyboard shortcuts when interacting with a delivery plan. To view the valid shortcuts, enter **?** when viewing a plan from the **Boards > Plans** or **Work > Plans** page.

To toggle between card titles only and card details, enter **t**.

> **NOTE**
> Type **?** to access Global and service-specific shortcuts.



Home Select first item
Enter Open item
n New item

Ctrl+↑ Move item up
Ctrl+↓ Move item down
Ctrl+← Move item left
Ctrl+→ Move item right
Shift+← Pan timeline left
Shift+→ Pan timeline right

u Collapse all backlogs
o Expand all backlogs

Shift+pageup Focus on previous team
Shift+pagedown Focus on next team
Ctrl+Shift+f Filter results

Related articles

For more resources to help work with multiple teams, see these articles:

- Interactively filter your backlogs, boards, and plans
- Backlogs, boards, and plans
- Add teams
- Portfolio management
- Manage teams and configure team tools
- Keyboard shortcuts

Programmatically manage Delivery Plans

You can manage plans using the REST API, Plans.

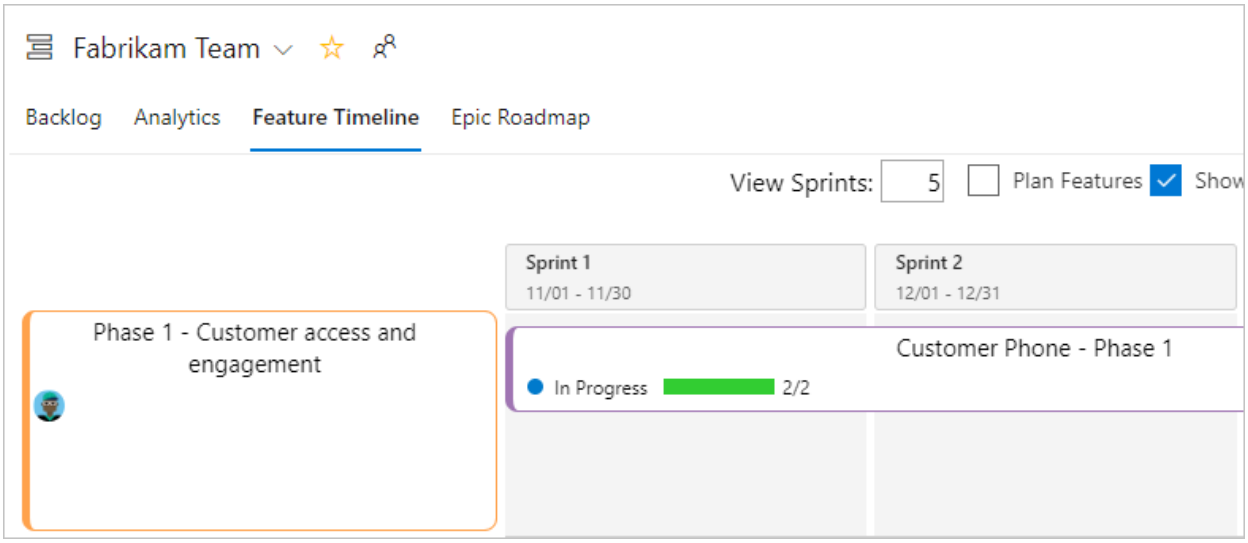
View portfolio progress with the Feature Timeline

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The Feature Timeline supports portfolio management. It provides an all-up progress view of features grouped by their epic parents. This view provides you with calendar views of feature progress with the ability to drill down to see details at the requirements level.

To see the full image, click the image to expand. Choose the close icon to close.



NOTE

The Feature Timeline and Epic Roadmap extension is not a supported feature of Azure Boards and therefore not supported by the product team. For questions, suggestions, or issues you have when using the extension, visit the extension page.

Use the Feature Timeline to support the following tasks:

- Support roadmap planning
- Produce reports at each business level to show high and low-level progress views
- Adjust sprint assignments to child work items

NOTE

The Feature Timeline and Epic Roadmap extension is available on TFS 2017 Update 2 and later versions. You can install it from the Marketplace for Azure DevOps, Feature Timeline and Epic Roadmap.

Portfolio backlog requirements

Both Feature Time and Epic Roadmap require at least two levels of portfolio backlogs to be useful. These levels typically correspond to Features and Epics, as defined by the Agile, Scrum, and CMMI default processes. The Basic process only defines the Epics portfolio backlog.

If you customize your portfolio backlogs to remove or disable a portfolio backlogs, then you may find that you

have limited usage of the Feature Time and Epic Roadmap views.

If you rename or edit the work items defined for a portfolio backlog, the Feature Timeline and Epic Roadmap views still work. Note the labels on each view remain Features and Epics.

- The label for the first portfolio backlog level remains as Features, for example, **Closed Features**.
- The label for the second portfolio backlog level remains as Epics, for example **Epic** drop-down menu.

In order to define Epics and Features from the team backlogs or boards, they must be selected through the team settings. For details, see Select backlog navigation levels for your team.

Prerequisites

- Install the Feature Timeline and Epic Roadmap extension for the organization or collection for which you want to track progress at the epic and feature level. To install an extension, you must be a member of the Project Collection Administrator Group. To learn more, see Install extensions.
- To view the Feature Timeline, you must be a member of the project. You must also have view permissions to work items under the area path they're assigned to.
- To modify work items, you must have permissions to edit work items under the area path they're assigned to.

Define your features and requirements

The Feature Timeline is designed to display features and the progress made to their child requirements. The Feature Timeline displays features grouped under epics when they're linked to parent epics. Ungrouped features appear at the bottom of the view.

> **NOTE**
> *Requirements* specify expectations of users for a software product. In Azure Boards, requirements are defined by work items that appear on your product backlog. They correspond to User Stories (Agile), Product Backlog Items (Scrum), Issues (Basic), or Requirements (CMMI) based on the process selected for your project. They also belong to the Requirements Category which manages which work item types appear on the product backlog.

To gain the most from the Feature Timeline view, make the following definitions:

- Define teams and area paths to support the rollup of the team's work into features and epics.
- Define sprints with dates for the project. Select sprints for the team.

  > **NOTE**
  > Make sure you assign work items to a **flat set of sprints**. Assigning features to one hierarchy of sprints and child items to another won't display correctly in the Feature Timeline view.

- For work to be completed in some future iteration, you can leave the dates unset for the iteration and it will appear as the last sprint in the roadmap.
- Make sure the team is subscribed to the sprints of interest.
- Define features and child work items. If no child work items are defined, then assign the feature to a sprint.
- When child work items are defined, assign the child items to sprints.
- To view progress by Effort, Story Points, or Size, assign values to those fields to the child requirements.
- Once all child requirements are completed, set the State of the parent feature or epic to Done or Completed. Closed epics and features no longer appear in the Feature Timeline.

TIP

To support roadmap planning, make sure your team has subscribed to several future iterations.

To learn more, review the following articles:

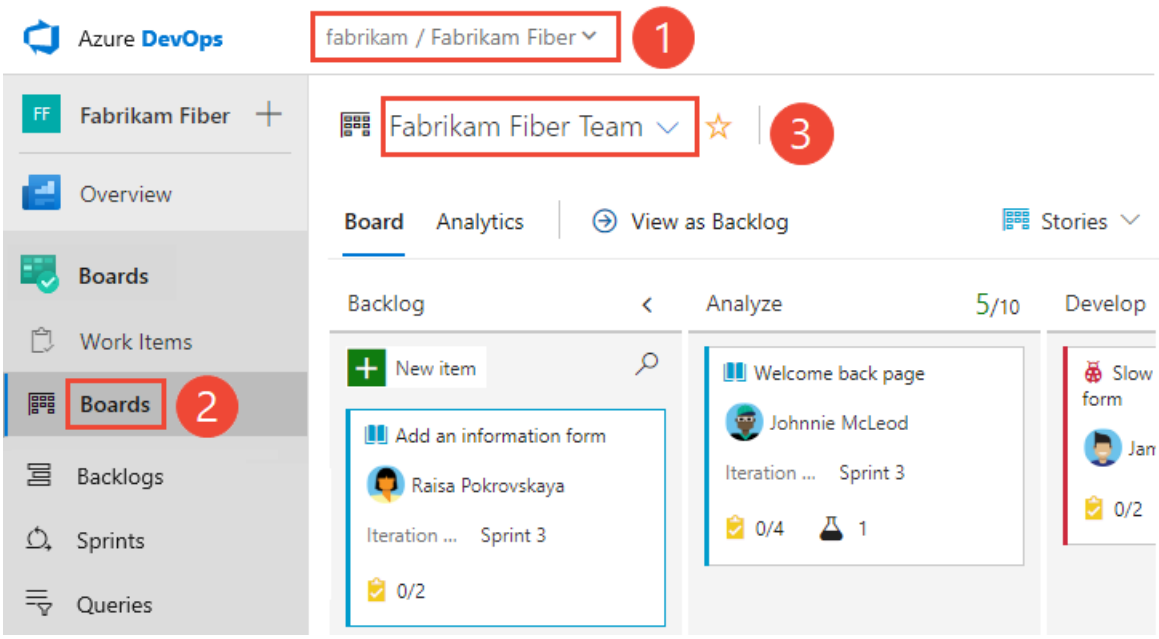
- Create your backlog (Requirements)
- Define features and epics
- Configure a hierarchy of teams
- Define area paths & assign to a team
- Define iteration paths (sprints) & assign team iterations

Open boards or backlogs

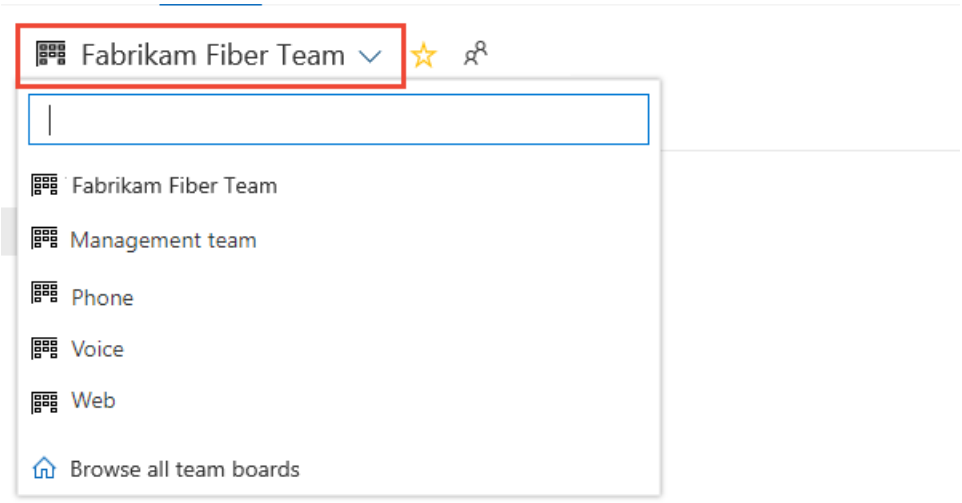
You can access the Feature Timeline from either your team's Kanban board or backlog.

Here we open Boards.

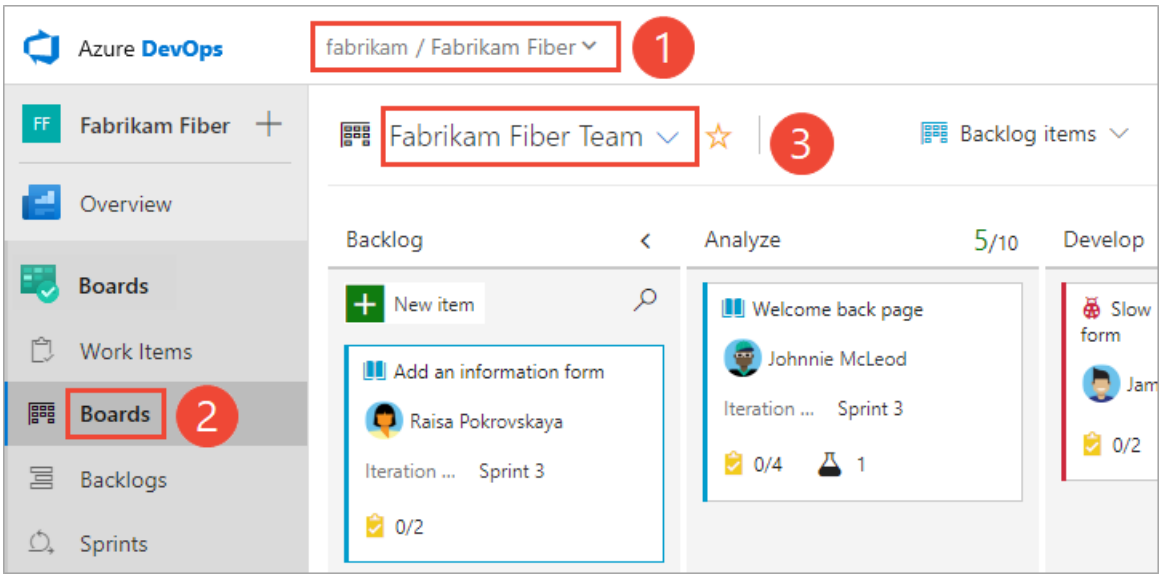
1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



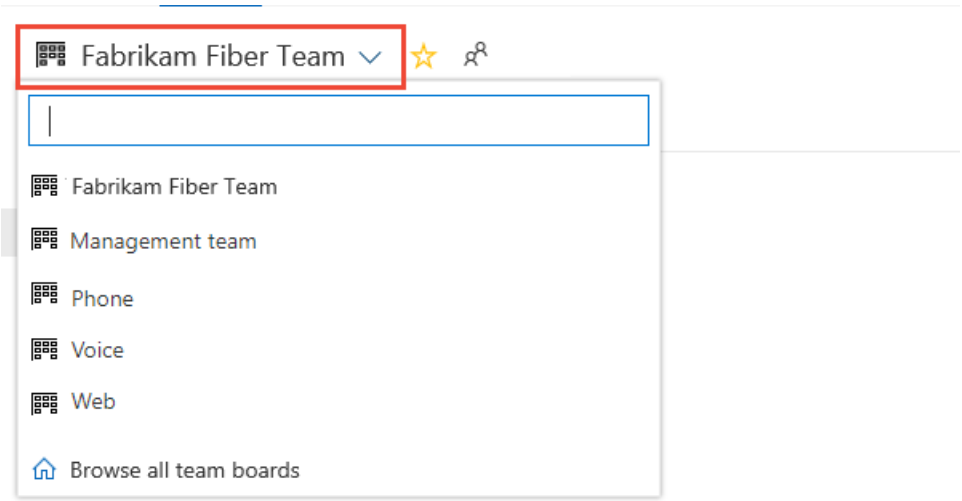
   To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



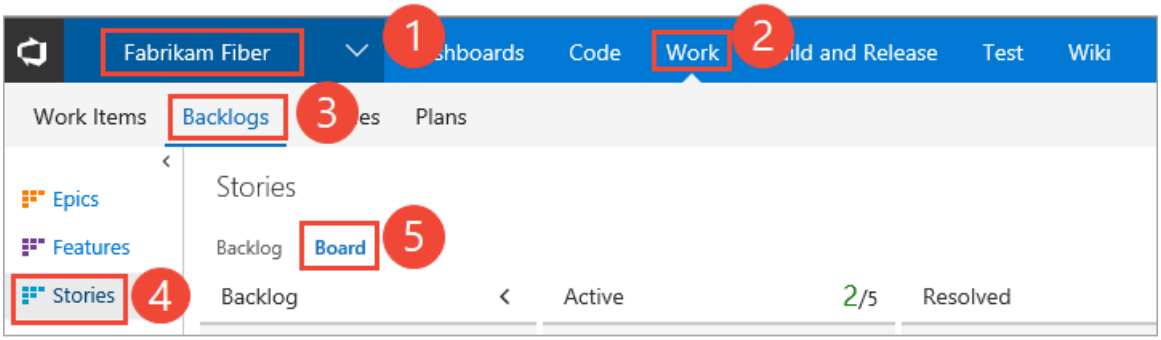
1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.



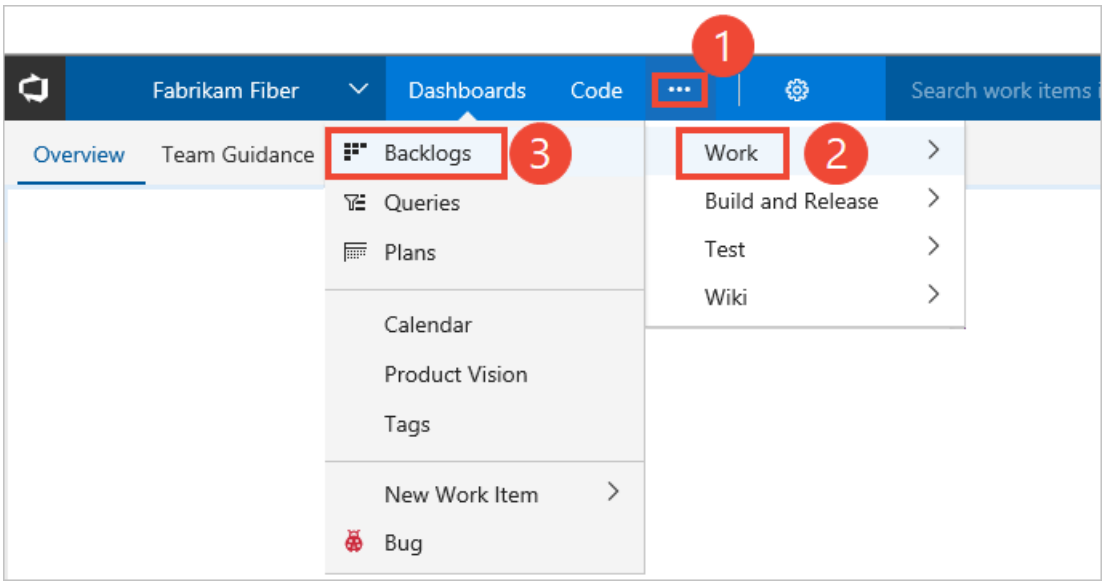
To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



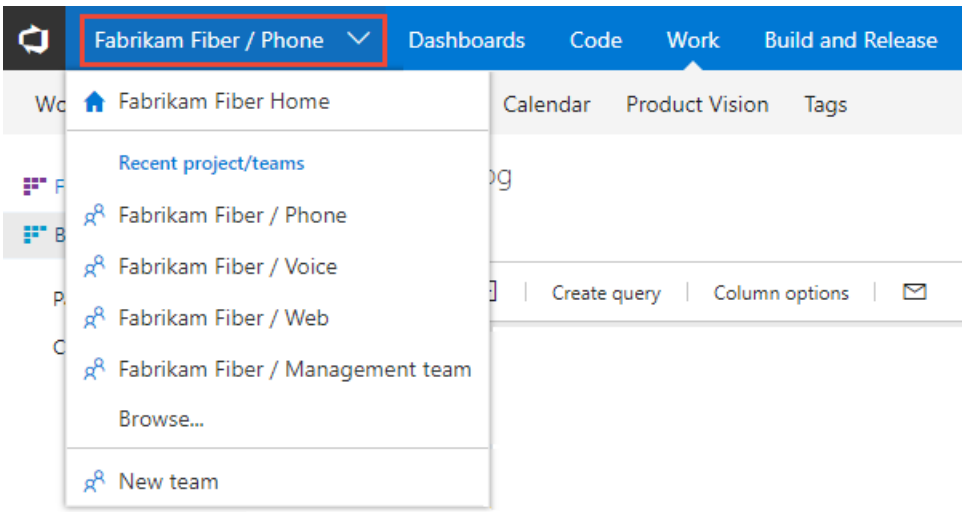
1. To view your Kanban board, open your project from a web browser. Select **Work** > **Backlogs** > **Stories**, and then select **Board**.



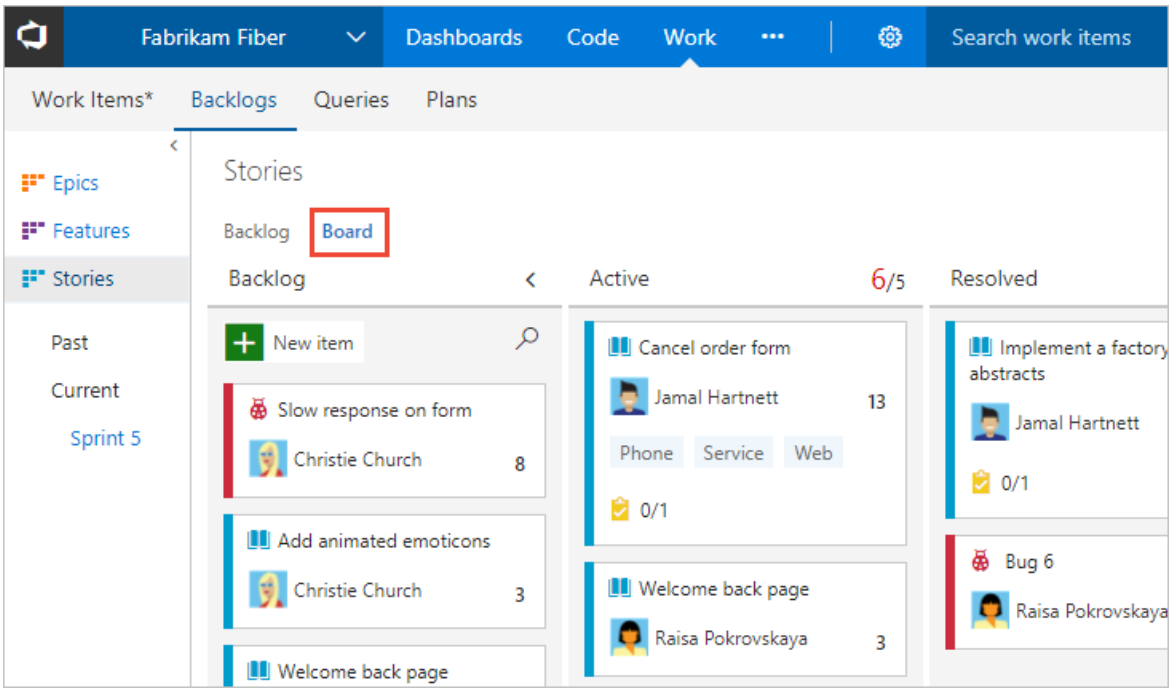
If you don't see **Work**, your screen size might be reduced. Select the three dots ( ) icon. Then select **Work** > **Backlogs** > **Board**.



2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.

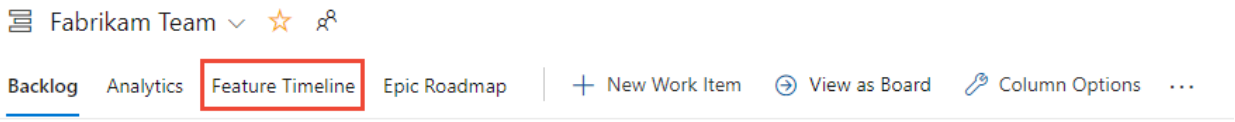


Your Kanban board appears.



Open the Feature Timeline

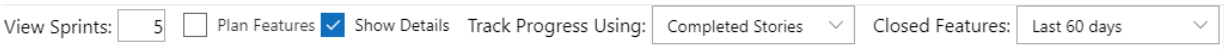
Choose **Feature Timeline**.



If you don't see the **Feature Timeline** link, then the Feature Timeline and Epic Roadmap extension isn't installed or enabled. Check with your Project Collection Administrator to request to have it installed. To learn more, Request and approve extensions.

Customize your view

You can customize your view of the Feature Timeline with the controls shown in the following image:



- **View Sprints**: Enter the number of iterations to show. The maximum number is 11.

  NOTE

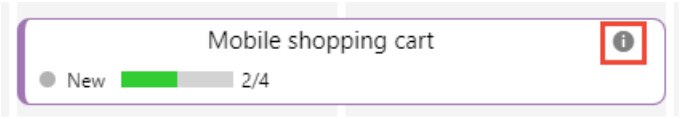
  Sprint labels may not display for iterations above six, however, the calendar view represents those iterations.
- **Plan Features**: Opens a side panel of other features participating in a Portfolio Plan.
- **Show Details**: Displays progress bars of Feature child items
- **Track Progress Using**: Progress bars indicate completion based on child requirements or overall total effort.
- **Closed Features**: Filters the Features based on those features closed within the selected time frame.

NOTE

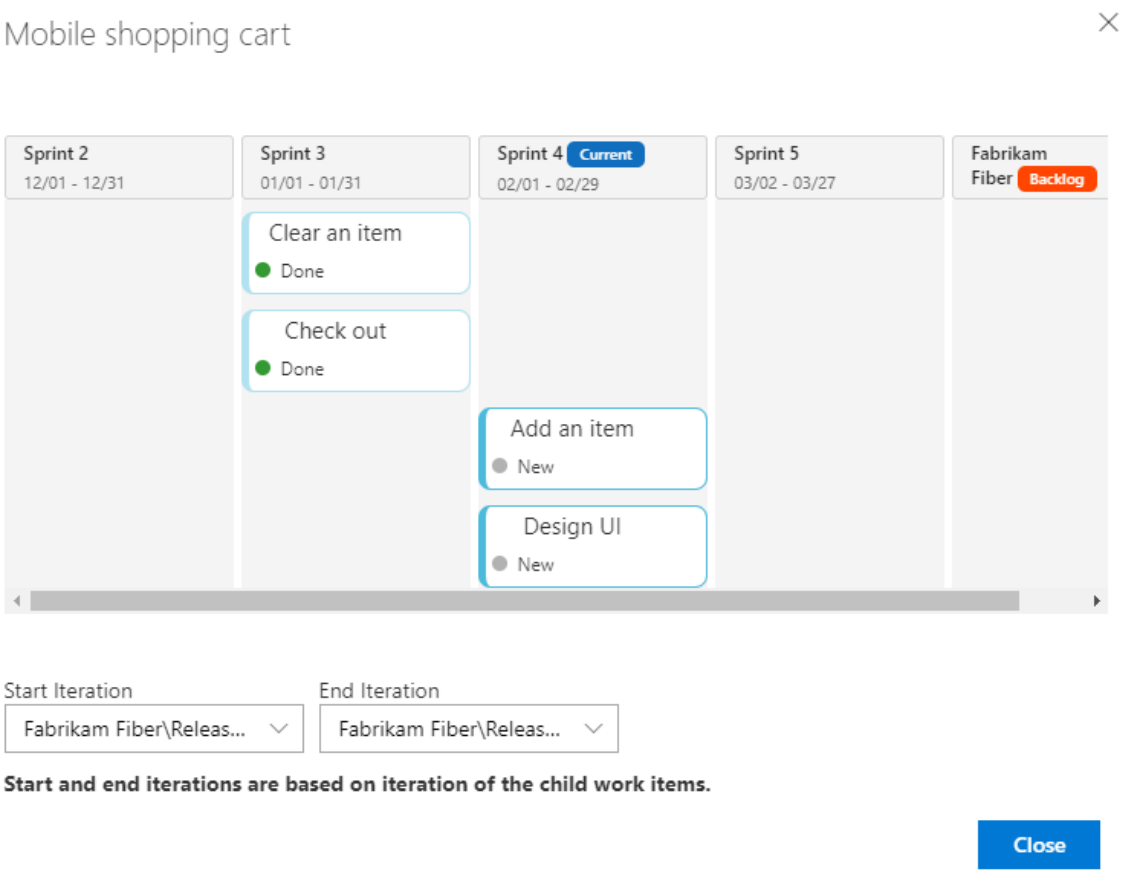
The Plan Features is in preview and available for Azure DevOps Services only at this time. Use of this feature integrates with Portfolio plans. Portfolio Plans are not yet documented.

Drill down and change Start and End iterations

1. To view the requirements linked to a feature, choose the info icon for that feature.

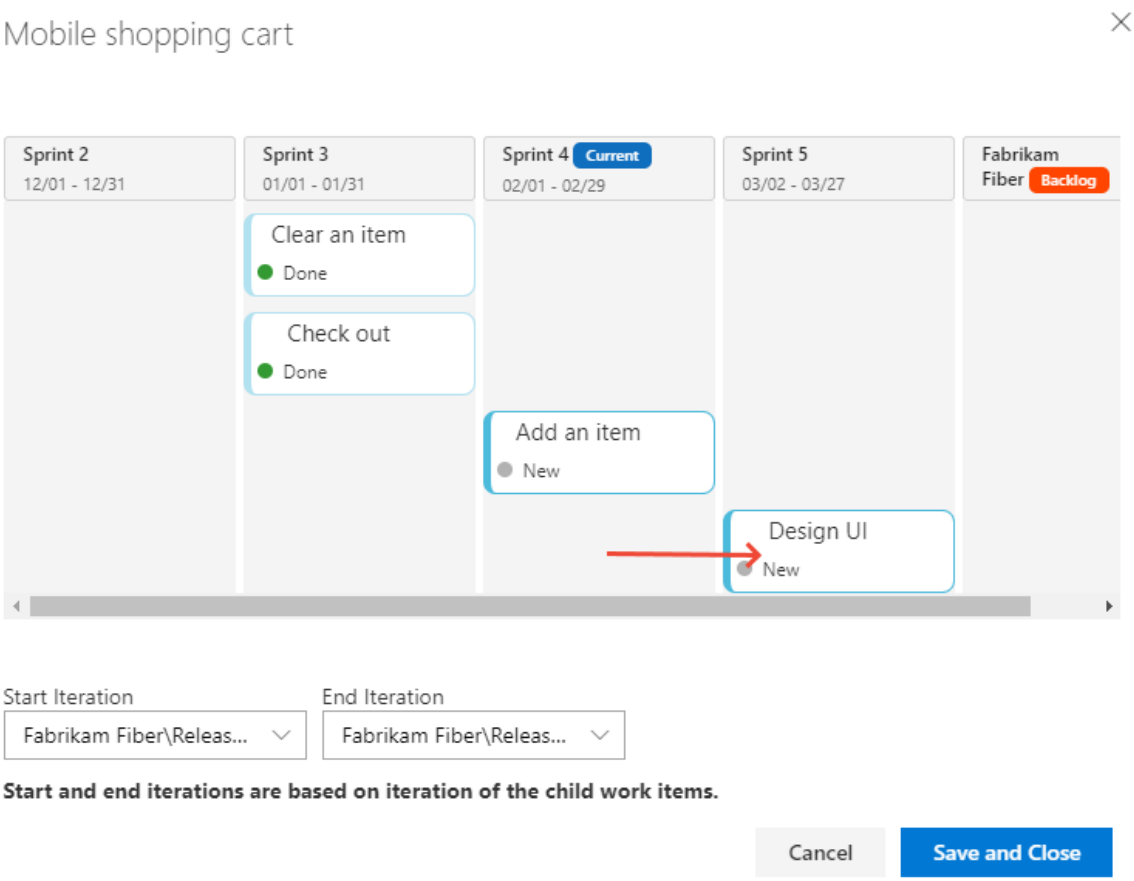


A dialog opens showing the child items of the feature.



The Start and End iterations are derived from the iteration paths assigned to the child work items. You can change those values by selecting new Start and End iterations from the drop-down path.

2. Or, you can also drag and drop a child item to a new iteration.



Q & A

How can I view the roadmap, the next three sprints?

A: Make sure you enter sufficient number of sprints to display the next three sprints, and assign work items to the next three sprints. Use the Planning pane to assign work items to sprints through drag-and-drop. To learn how, see Assign backlog items to a sprint.

Related articles

- Configure and customize Azure Boards
- Review team delivery plans
- View progress using the Epic Roadmap

Related Marketplace extensions

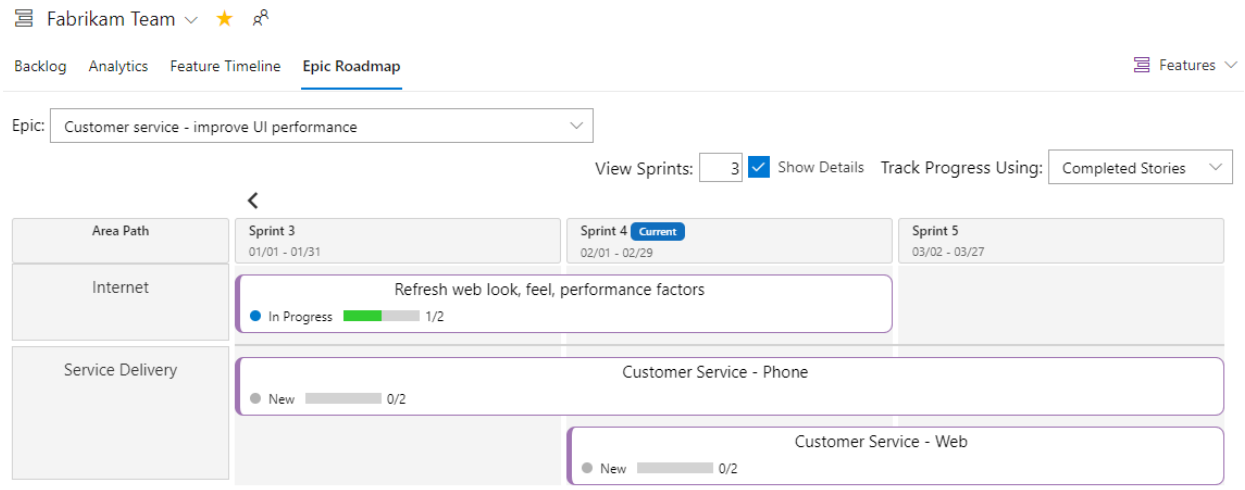
- Work Item Visualization

View progress using the Epic Roadmap

10/4/2022 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Similar to the Feature Timeline, the Epic Roadmap supports portfolio management. It provides a calendar view of a single epic and its child features. Within each epic roadmap view, you can drill down to see details at the feature and requirements level.



> **NOTE**
> The Feature Timeline and Epic Roadmap extension is not a supported feature of Azure Boards and therefore not supported by the product team. For questions, suggestions, or issues you have when using the extension, visit the extension page.

Use the Epic Roadmap to focus on a single epic and to support the following tasks:

- Support roadmap planning
- View work that spans several iterations
- Produce reports at each business level to show high and low-level progress views
- Adjust sprint assignments to child work items
- View dependencies linked to features

> **NOTE**
> The Feature Timeline and Epic Roadmap extension is available on TFS 2017 Update 2 and later versions. You can download it from the Marketplace for Azure DevOps, Feature Timeline and Epic Roadmap.

Minimum Viable Product (MVP) overview

Agile methodologies favor defining a Minimum Viable Product (MVP) over Critical Path Management (CPM). An MVP defines a product with enough features to satisfy early adopters. You then define and complete other features after receiving feedback from the early adopters users. By gleaning insights from an MVP, you're more likely to reduce development costs and risks.

By using MVP, you identify the shortest path and dependencies by completing these tasks:

- Define requirements, features, and epics that support MVPs
- Organize your requirements, features, and epics to work on the most important work to deliver your MVPs
- Group features under epics, and requirements under features.

> **NOTE**
> *Requirements* specify expectations of users for a software product. In Azure Boards, requirements are defined by work items that appear on your product backlog. They correspond to User Stories (Agile), Product Backlog Items (Scrum), Issues (Basic), or Requirements (CMMI) based on the process selected for your project. They also belong to the Requirements Category which manages which work item types appear on the product backlog.

For more context, see Running a lean startup on Azure DevOps.

Portfolio backlog requirements

Both Feature Time and Epic Roadmap require at least two levels of portfolio backlogs to be useful. These levels typically correspond to Features and Epics, as defined by the Agile, Scrum, and CMMI default processes. The Basic process only defines the Epics portfolio backlog.

If you customize your portfolio backlogs to remove or disable a portfolio backlogs, then you may find that you have limited usage of the Feature Time and Epic Roadmap views.

If you rename or edit the work items defined for a portfolio backlog, the Feature Timeline and Epic Roadmap views still work. Note the labels on each view remain Features and Epics.

- The label for the first portfolio backlog level remains as Features, for example, **Closed Features**.
- The label for the second portfolio backlog level remains as Epics, for example **Epic** drop-down menu.

In order to define Epics and Features from the team backlogs or boards, they must be selected through the team settings. For details, see Select backlog navigation levels for your team.

Prerequisites

- Install the Feature Timeline and Epic Roadmap extension for the organization or collection for which you want to track progress at the epic and feature level. To install an extension, you must be a member of the Project Collection Administrator Group. To learn more, see Install extensions.
- To view the Epic Roadmap, you must be a member of the project and have view permissions to work items under the area path they're assigned to.
- To modify work items, you must have permissions to edit work items under the area path they're assigned to.

Define your epics, features, and stories

The Epic Roadmap displays individual epics, and the features and requirements mapped to them. Features are grouped under the Area Path they're assigned to. The Area Path generally corresponds to a feature team. Only those features that are children of an epic appear in the Epic Roadmap view.

To use the Epic Roadmap, make the following definitions:

- Define teams and area paths to support the rollup of the team's work into features and epics.
- Define sprints with dates for the project. Select sprints for the team.

> **NOTE**
> Make sure you assign work items to a **flat set of sprints**. Assigning features to a hierarchy of sprints and child items to another set of sprints won't display correctly in the Epic Roadmap view.
> or work to be performed in some future iteration, you can leave the dates unset for the iteration and it will appear as the last sprint in the roadmap.

- Make sure the team is subscribed to the sprints of interest.
- Define epics and their child features. If no child work items are defined, then assign the feature to a sprint.
- When child work items are defined, assign the child items to sprints.
- To view progress by Effort, Story Points, or Size, assign values to those fields to the child requirements.
- Once all child requirements are completed, set the State of the parent feature or epic to Done or Completed. Closed epics no longer appear in the drop-down list of the Epic Roadmap.

> **TIP**
> To support roadmap planning, make sure your team has subscribed to several future iterations.

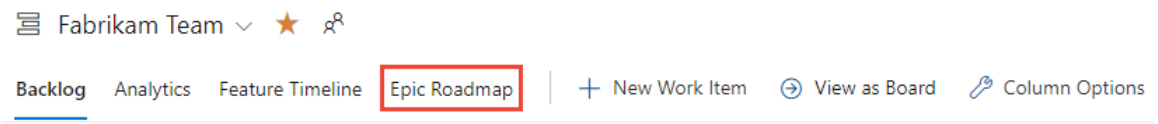
To learn more, review the following articles:

- Create your backlog (Requirements)
- Define features and epics
- Define area paths & assign to a team
- Define iteration paths (sprints) & assign team iterations

Open Epic Roadmap

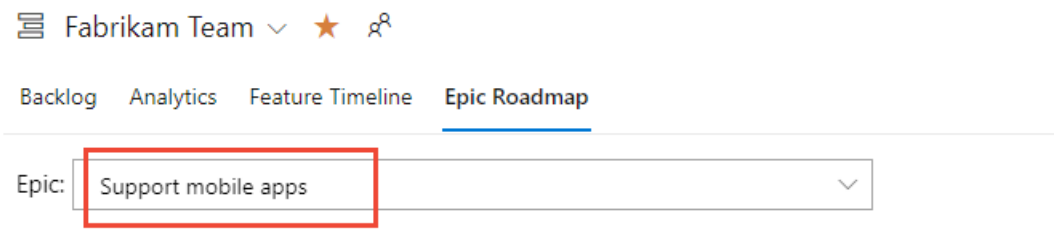
You can access the Epic Roadmap from either your team's Kanban board or backlog.

1. Open the Board or Backlog for your team, see View portfolio progress with the Feature Timeline, Open Boards, or Backlogs.
2. Choose **Epic Roadmap** and then choose the Epic you want to view.



   If you don't see the **Epic Road** link, then the Feature Timeline and Epic Roadmap extension isn't installed or enabled. Check with your Project Collection Administrator to request to have it installed. To learn more, Request and approve extensions.

3. Choose the Epic you want to view from the drop-down menu.



4. Customize your view.

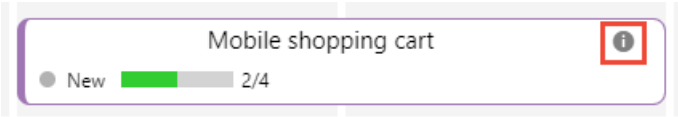
- **View Sprints**: Enter the number of iterations to show. The maximum number is 11.

  > **NOTE**
  > Sprint labels may not display for iterations above six, however, the calendar view displays the iterations.

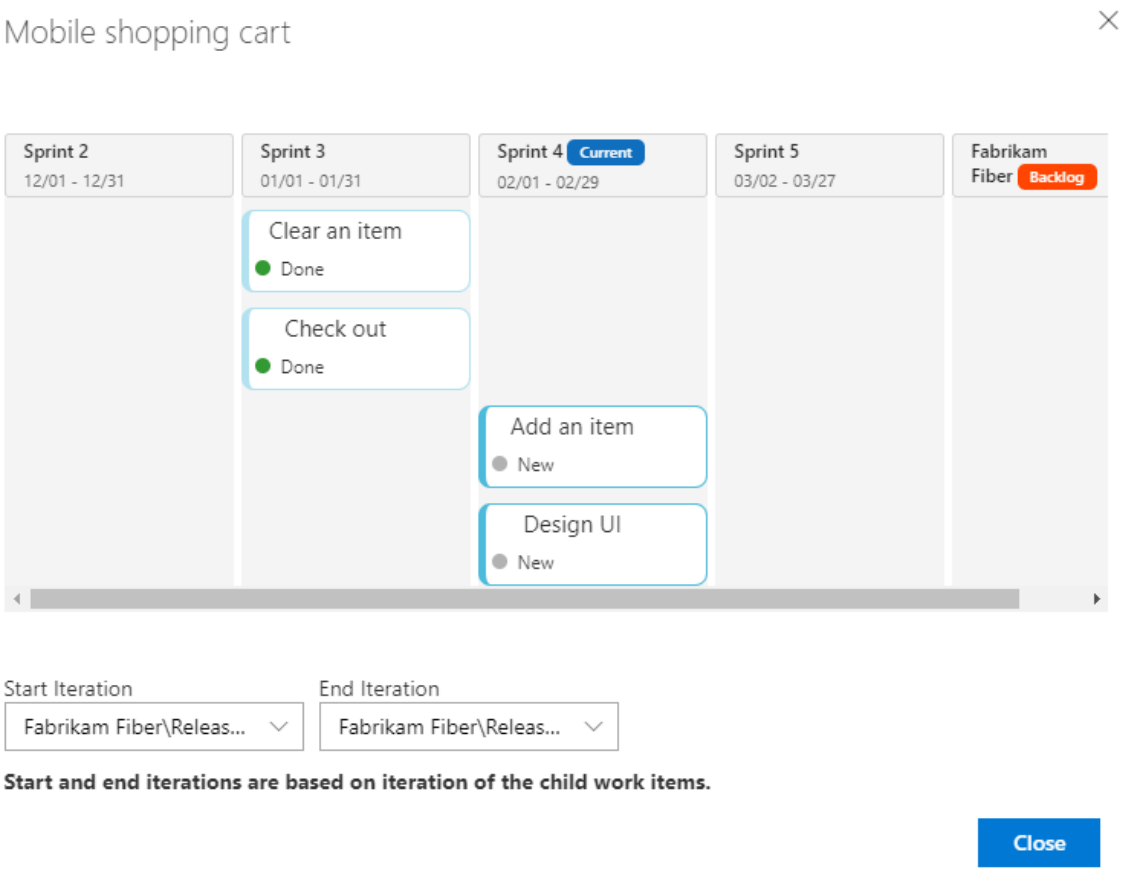
- **Show Details**: Displays progress bars of Feature child items
- **Track Progress Using**: Progress bars indicate completion based on child requirements or overall total effort.

Drill down to view details

1. To view the requirements linked to a feature, choose the info icon for that feature.

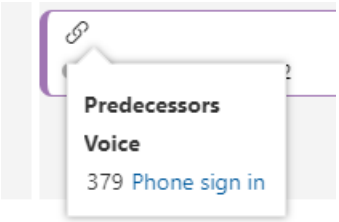


A dialog opens showing the child items of the feature.



The Start and End iterations are derived from the iteration paths assigned to the child work items. You can change those values by selecting new Start and End iterations from the drop-down path.

2. To view the dependency linked to a feature, choose the link icon for that feature.



> **NOTE**
> Another tool that supports dependency views is Delivery Plans.

Related articles

- Review team delivery plans
- View portfolio progress with the Feature Timeline
- Track dependencies by using Delivery Plans

Related Marketplace extensions

- Work Item Visualization

Plan and track dependencies using the Dependency Tracker

10/4/2022 • 18 minutes to read • Edit Online

Azure DevOps Services

> **NOTE**
> We recommend that you use Delivery Plans to track dependencies instead of Dependency Tracker. The Dependency Tracker extension is not a supported feature of Azure Boards and isn't supported by any product team. For questions, suggestions, or issues you have when using the extension, visit the Marketplace for Azure DevOps, Dependency Tracker extension page. The Dependency Tracker extension is only available on Azure DevOps Services.

The Dependency Tracker extension enables management of dependencies across teams, projects, and organizations. It provides filterable views to show all dependencies a team is consuming and producing. These views allow you to track the state and schedule of dependencies to support you in assessing the risk of dependencies to product deliverables.

You use the Dependency Tracker to plan dependencies at the beginning of an iteration or release, and to track the status during development. For any given dependency, there are two parties involved:

- **Consumer**: Feature team who has a need and starts a request for work
- **Producer**: Feature team who makes a commitment to deliver work

Each work request and work deliverable is defined as a work item. The work items are linked by the Successor-Predecessor link type or other directional link type. For details about link types, see Link type reference Producing for/Consuming from link.

> **TIP**
> While any work item type can participate in dependency tracking, you may want to decide if you want to limit dependencies to specific types, such as Features, Epics, User Stories, or Bugs. You can create that restriction through Configuration of Dependency Tracker.

From the Dependency Tracker, you can choose different views and filters, and drill down to obtain specific details. These views and options are described in the following sections:

- Filter options
- Drill-down
- Consuming Dependencies
- Producing Dependencies
- Timeline
- Risk Graph

Recommended use and key terms

You can use Dependency Tracker to visualize and track the following work items:

- Dependencies on deliverables for work that your team is delivering

- Dependencies you have on other teams for work that your team is delivering
- Dependencies that other teams have on work your team is delivering

All teams across organizations can participate in tracking dependencies.

> **NOTE**
> Dependency Tracker doesn't replace the in person interactions required to agree to doing the work. It provides easier planning and tracking capabilities. Dependencies should be agreed upon by all parties before they are entered in to the Dependency Tracker.

Key terms

- **Dependency**: work that Team A requires from Team B to do the work Team A is trying to do
- **Consumer**: the team that wants work done
- **Producer**: the team that is being asked to do the work
- **Sequencing**: when a producing team's work is needed before the consuming team can start their work

Recommended practices

- The consumer is the team that asks for the work – they start all discussions on the work they require
- The consumer owns the engagement and tracking of that work – since it's the work their scenario requires, the burden is on the consumer to file, monitor, and track the status of the work
- The consumer owns entering the work into Azure Boards and submitting that work request to the producer
- Once the work has been submitted to the producer, the producer owns the work item,
  - The producer is responsible for maintaining the work item in Azure Boards
  - The producer owns the state of the work item (is it going to be done) and iteration (when it will be done).
  - The consumer shouldn't touch these values, once the work item has been handed off
- The consumer is in charge of managing the work they requested so that they're aware of any material changes and adjustments.

Prerequisites

- Install the Dependency Tracker extension for the organization(s) for which you want to track dependencies.
- To view dependencies, you must be a member of the Project Valid Users group for the project.
- To create a dependency, you must be a member of the Contributors group for both projects that participate in the dependency linking.
- To support cross-organization participation, all organizations must authenticate users through the same Azure Active Directory.
- Azure Boards must be enabled as a service. If it's disabled, then you'll need to have it reenabled. For details, see Turn a service on or off.
- To modify the configuration, you must be a member of the Project Collection Administrator Group.

> **IMPORTANT**
> The default configuration for Dependency Tracker supports the Agile process. If your project(s) are based on a different process or you have customized your process, you may need to modify the configuration. See Configure the Dependency Tracker later in this article.

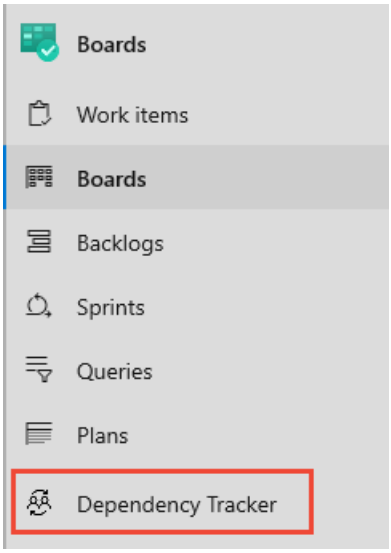
Also, the following configuration or customization tasks should be performed:

- Set up the area paths and teams that will participate in dependency tracking.

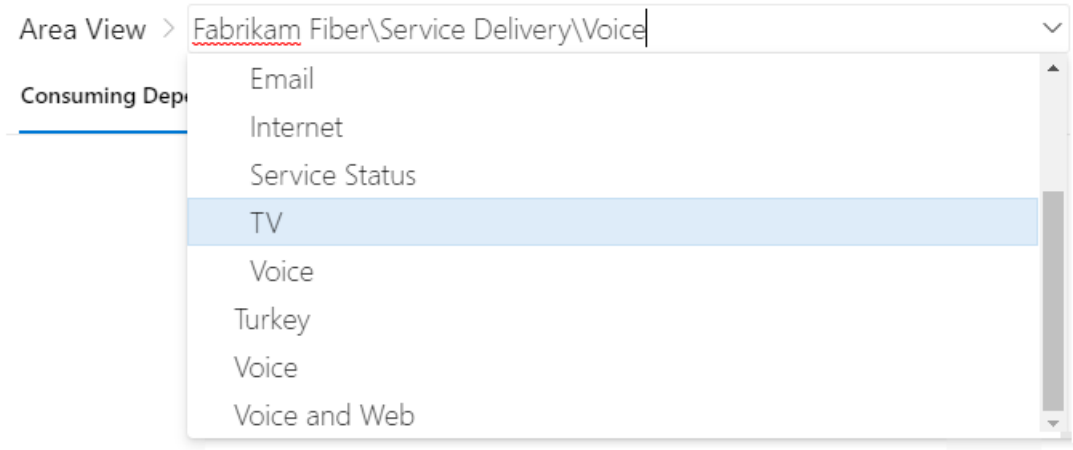
- Configure iteration paths/sprints for the project and assign them to work items participating in dependency tracking. This task is essential for the Timeline view to yield meaningful data.
- Customize your process as needed to support any other work items or fields.
- Configure the Dependency Tracker to support your business needs and address any customizations you've made.

Open the Dependency Tracker

1. Open the web portal for the project where your team is defined.
2. Choose **Dependency Tracker** from under the Boards group.



3. To focus on your area of ownership, choose the **Area** that corresponds to the team you want to view dependencies for.

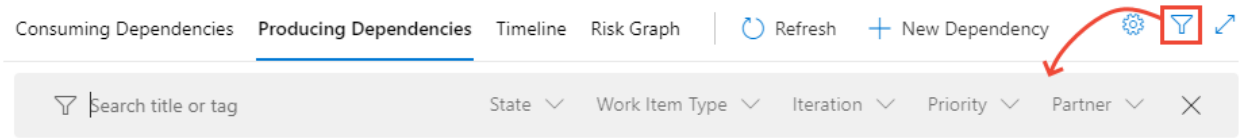


You can only filter on those Area paths defined for the project.

Filter options

You can filter each supported view by typing a keyword or using one or more of the fields. Provided fields include State, Work item type, and Iteration Path. Based on the keyword that you enter, the filter function lists work items based on any displayed column field.

To show the filter toolbar, choose the filter icon.



You can toggle filters on and off by choosing the filter icon. To see more filters use the arrows at the end of the list of filters.

Choose one or more values from the multi-select drop-down menu for each field. The values for these fields are populated as follows:

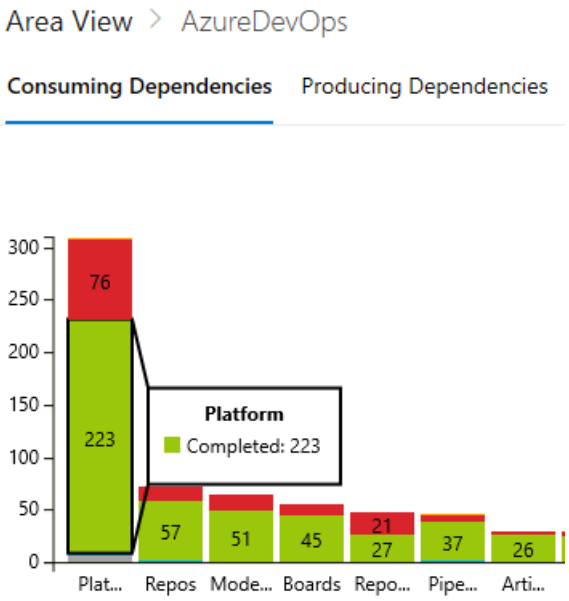
- **State**: Check one or more check boxes for the work item states you want to view. The drop-down list should include all workflow States defined for all work item types shown in the selected view.
- **Work item type**: Check one or more check boxes for the Work item types you want to view. Work item types configured to participate in dependency tracking. The default work item types are: Epic, Feature, User Story, and Bug. To modify the configuration, see Configuration of Dependency Tracker.
- **Iteration**: Check one or more check boxes for the Iteration Paths you want to view. The drop-down list should include all Iteration Paths configured for the project and for which there are work items listed in the current view.
- **Priority**: Check one or more check boxes for the Priorities you want to view. The priority values assigned to work items
- **Partner**: The partner organization for which the work item is defined.

> **NOTE**
> Filter options are dependent on the configuration defined for the Dependency Tracker. Also, only those options that correspond to work items shown in the selected view that meet the filter criteria. For example, if you don't have any work items assigned to Sprint 4, then the Sprint 4 option won't appear in the filter options for the Iteration Path.

Ability to drop dependencies within the selected area (used for excluding dependencies inside my team)

View drill-down options

Several views provide interactive visualizations through drill-downs. These features are addressed in the tabbed views descriptions later in this article.



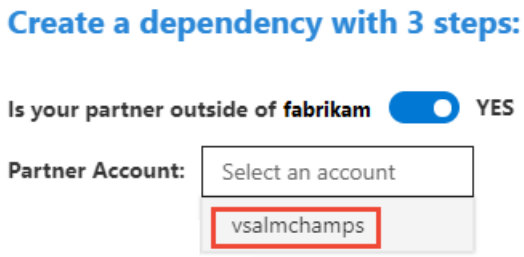
Create a dependency

A dependency represents work where one team is dependent on another team. Both teams should track their own work in their own area path. By linking the work that is dependent on the other teams work, the dependencies can be visualized and tracked.

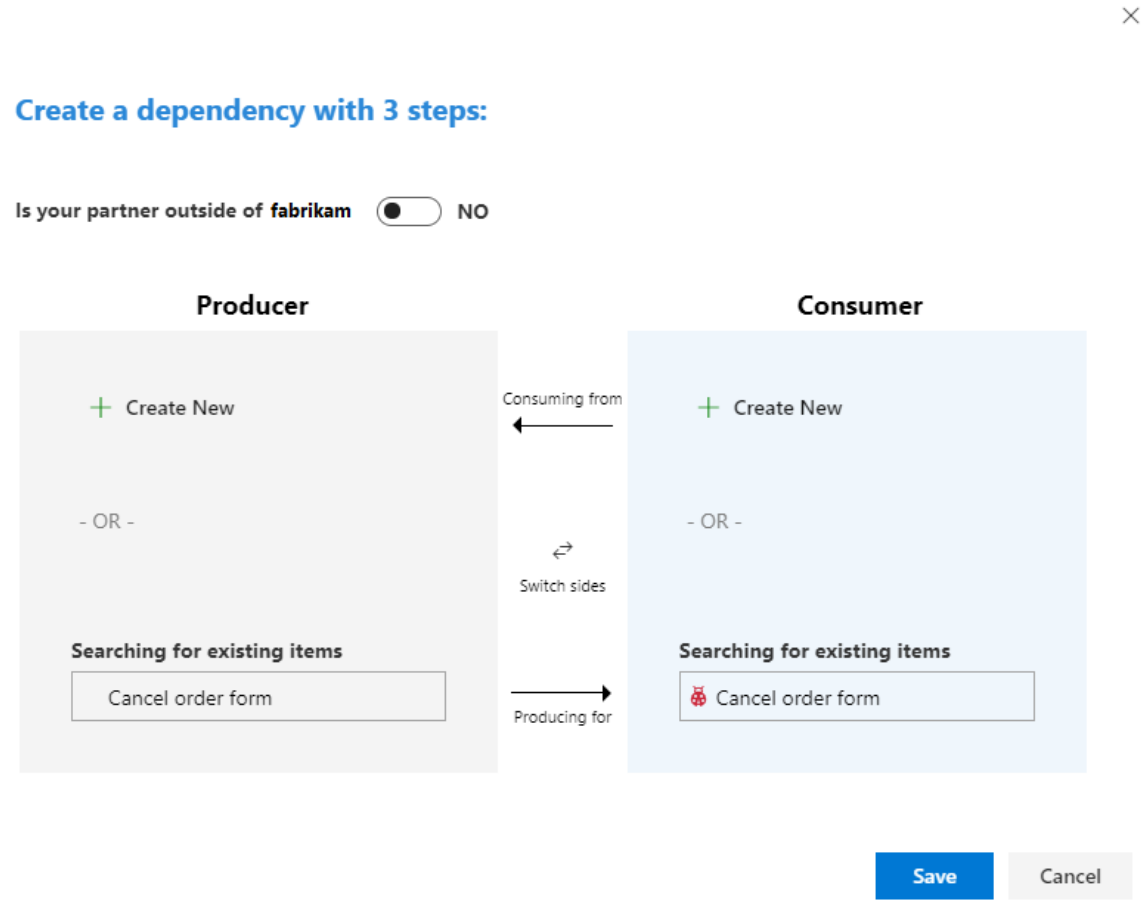
1. Choose **New Dependency**.



If the partner team is in a different organization, then first choose the **Partner Account**. The Partner Account option can be enabled or disabled by configuring the Dependency Tracker.



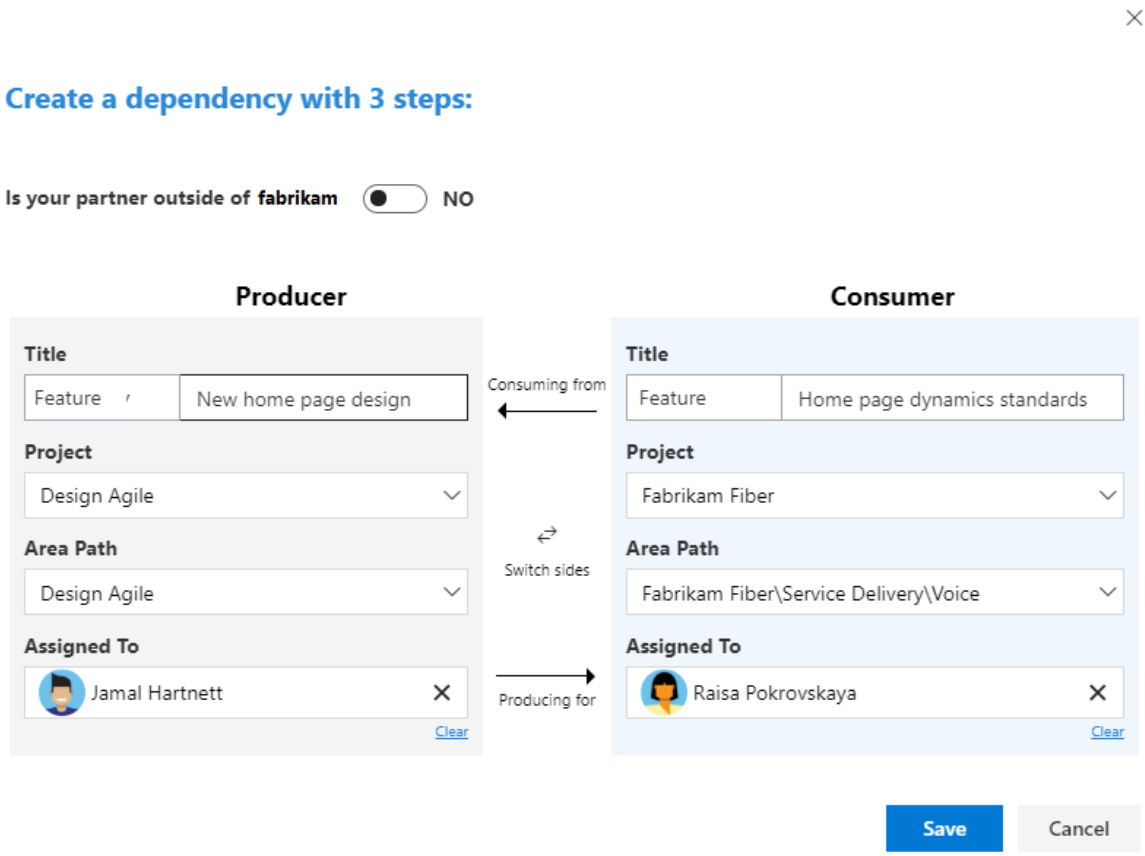
2. You can search for work items by ID or by entering a keyword contained within the work item title. Here, we link a user story and a bug.



- The Producer is the team that commits to delivering the work.
- The Consumer is the team that needs or is dependent on the work.
- The fastest way to create a dependency link is to type the Producer and Consumer work item IDs in the search boxes and then choose **Save**.

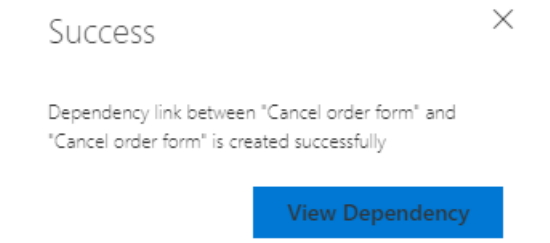
Optionally, you can choose **Create New** to add work items that you then link as dependent upon each

other. Here we create two new features and link them.



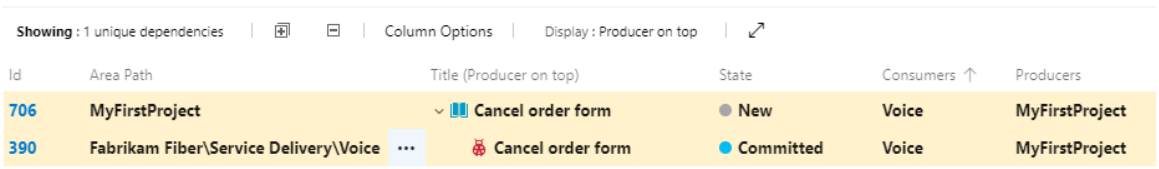
If no work items exist for one half of the dependency, you can create a new work item as needed.

3. Choose **Save**. The **Save** button becomes available only after you've chosen two work items to link.
4. From the success confirmation dialog, choose **View dependency**.



5. The work items that you linked are highlighted.

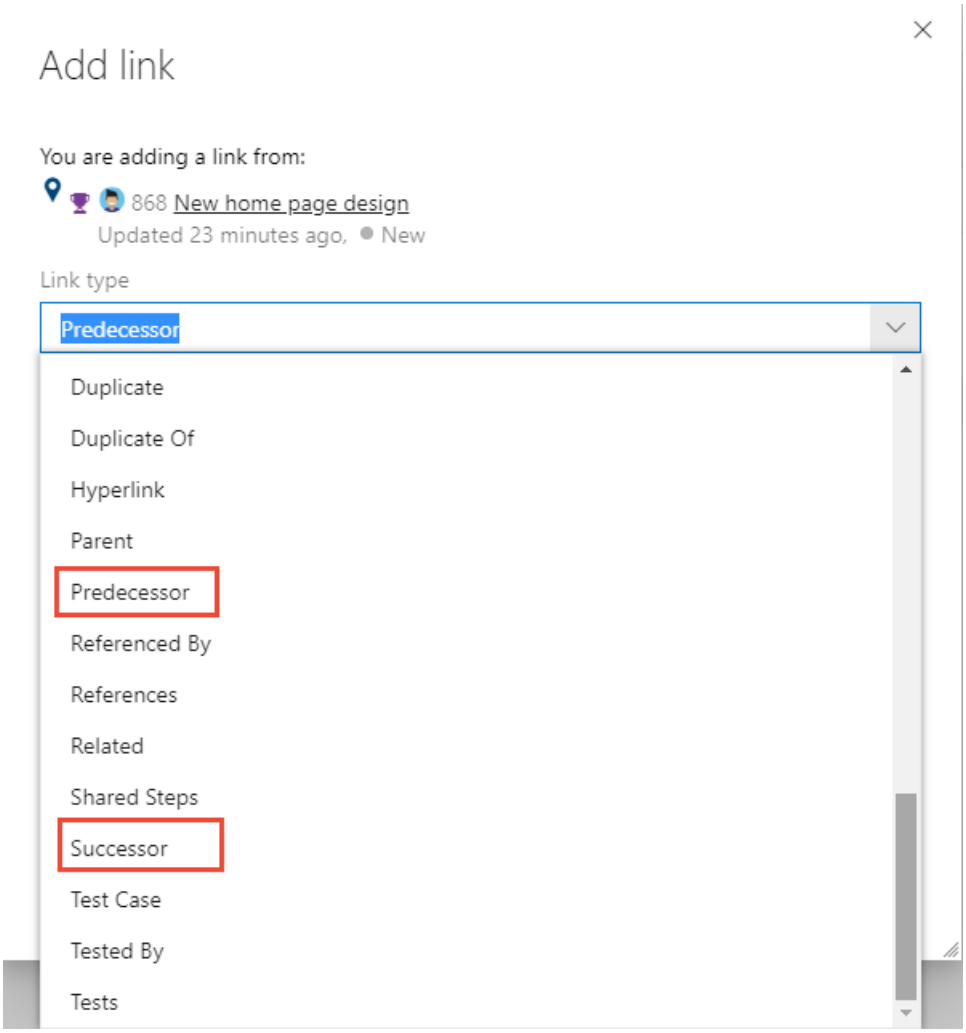
   As shown in this example, the Fabrikam Fiber/Service Delivery/Voice team is dependent on the MyFirstProject team to deliver their `User Story 706: Cancel order form` to complete `Bug 390: Cancel order form`.



Create links manually

You can also link work items using the **Links** tab to create Successor/Predecessor links. A Predecessor is the

producer of the work item, or the item that must come first. A Successor is the consumer of the work item, or the item that is dependent on the first item.

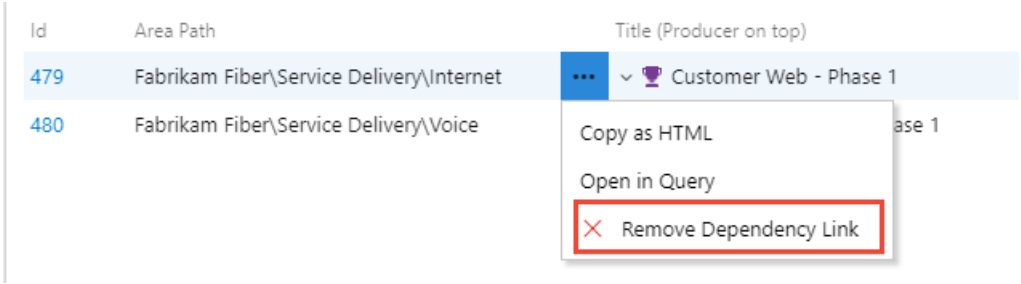


> **NOTE**
> The Successor/Predecessor (consumes/produces) link types are the default link types used by the Dependency Tracker. If you're projects are customized using a Hosted XML process model, it's possible to specify different link types in the Dependency Tracker configuration. See Configure the Dependency Tracker later in this article.

To learn more, see Link user stories, issues, bugs, and other work items.

Remove dependency links

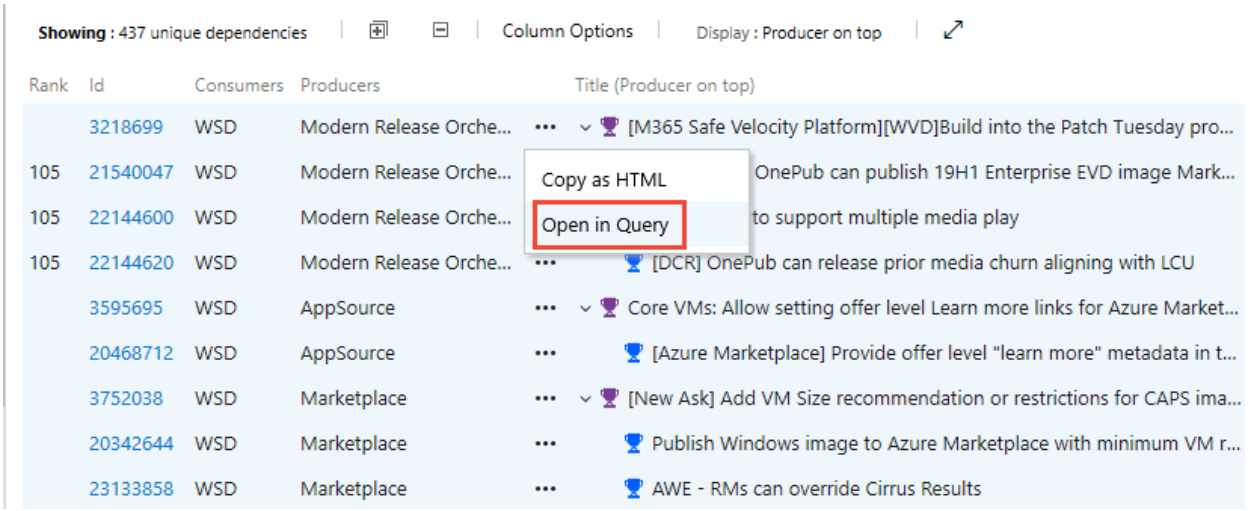
You can remove a dependency by choosing the ··· actions icon from the linked work item and choose **Remove Dependency Link** option from the menu.



Optionally, you can remove the link from the work item's **Links** tab.

Create a query of dependencies

To open a set of dependent work items, select them in the same way you would via a bulk edit, choose the ··· actions icon from one of the selected linked work items and choose **Open in Query** option from the menu.



A new tab opens to the Query Results page.

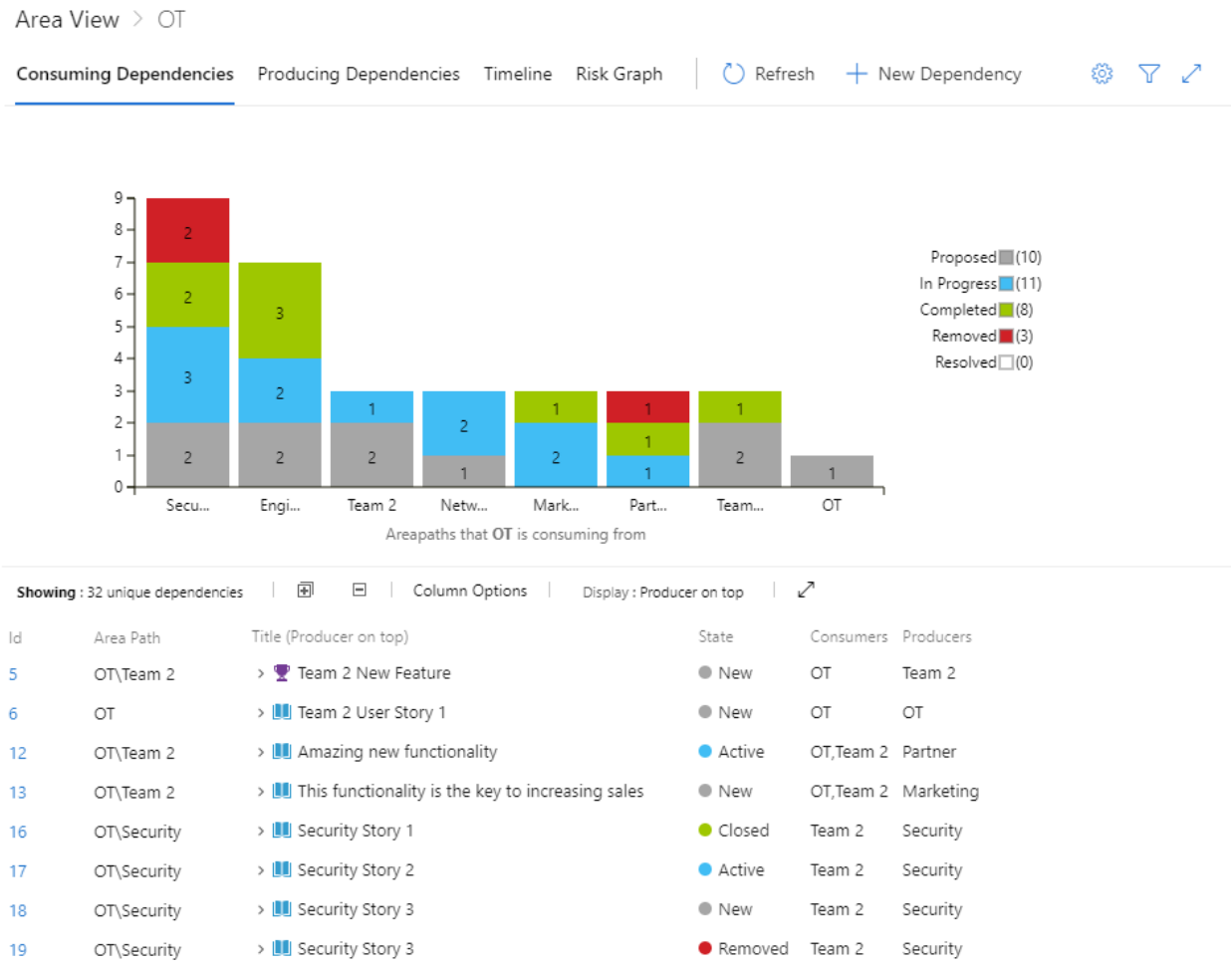
You can also create a custom query by selecting the Work items and direct links query type and choose any work item type.

Choose **Copy to HTML** to copy the selected work items to the clipboard as a formatted table.

Consuming Dependencies view

The **Consuming Dependencies** view shows work that a team is dependent upon other teams/area paths. It's useful for answering the following questions:

- *Which dependencies am I consuming as the owner of the selected areas and sub areas?*
- *How many dependencies per producer team (by area level 3)?*
- *What are the workflow states of my consumer dependencies?*



Each column on the bar chart represents another area path that is producing dependencies by workflow state for the selected **Area View**. The table shows the number of unique dependencies. It also lists all work items participating in the filtered view of tracked dependencies.

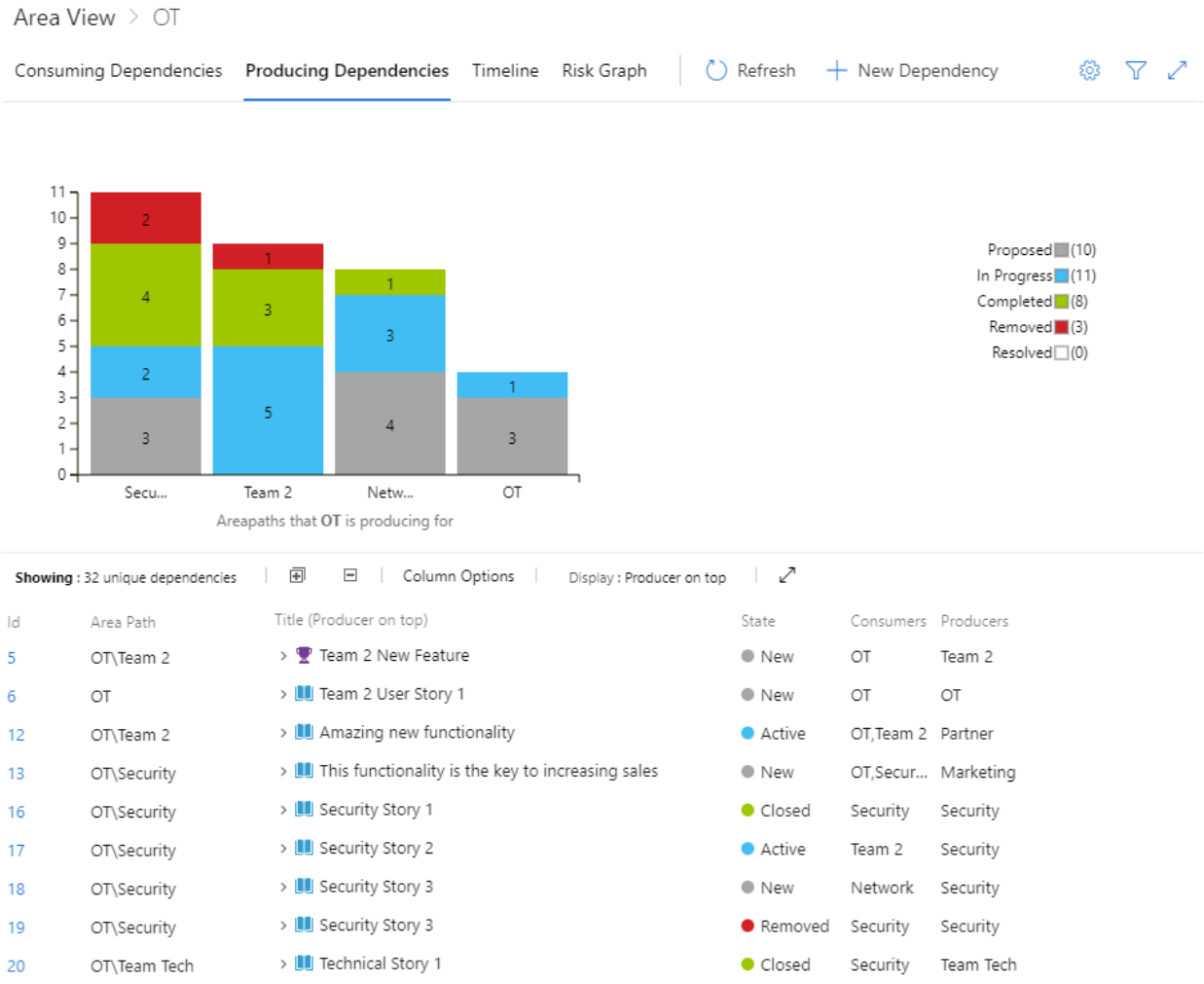
Within the table, you can complete the following actions.

- Filter the list of work items by choosing one of the area path bars and progress states in the bar chart graph
- Expand or collapse the list of work items to show the full list of dependent work items using the ⊞ expand and ⊟ collapse icons
- Add or remove column fields by opening Column Options
- Switch the sequence of work items by choosing the Display: **Consumer on top** or **Producer on top**

Producing Dependencies view

The **Producing Dependencies** view shows work that other teams/area paths are dependent on per the selected area. It's useful for answering the following questions:

- *Which dependencies is my team responsible for delivering as the owner of the selected area(s)?*
- *How many dependencies exist per consumer team (by area level 3)?*
- *What are the workflow states of my producer dependencies?*



Each column on the bar chart represents another area path that is consuming dependencies by workflow state for the selected **Area View**. The table shows the number of unique dependencies and lists all work items included in the filtered view of tracked dependencies.

Within the table, you can complete the same actions as in the Consuming Dependencies view.

Timeline tab

The **Timeline** tab provides a calendar view of dependencies. The Timeline view is in Beta. The Timeline view helps answering the following questions:

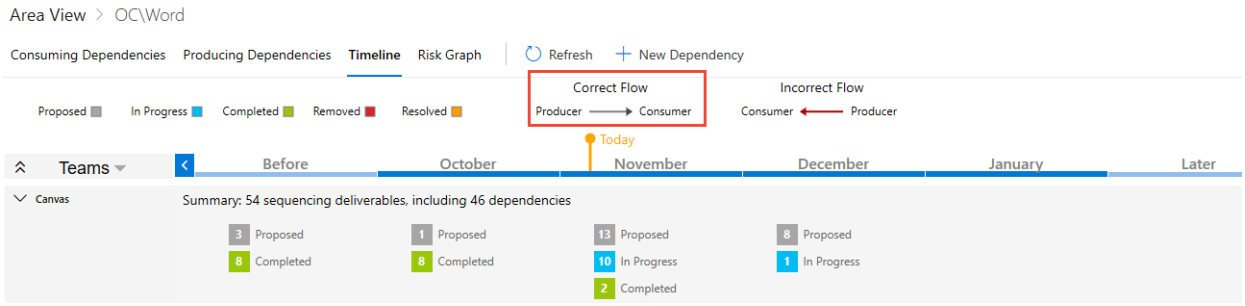
- *What is the sequence of dependencies within the time window.*
- *What are all the deliverable dependencies against within the three-month time window for a given team?*

> **IMPORTANT**
> In order for the Timeline to show meaningful data, you must have assigned the dependent work items to Iteration Paths, and the Iteration Paths must have start and end dates assigned.

There are two versions of the Timeline view: **Correct Flow** and **Incorrect Flow**. Each version shows the color-coded workflow state. Color codes can be customized within the Dependency Tracker configuration.

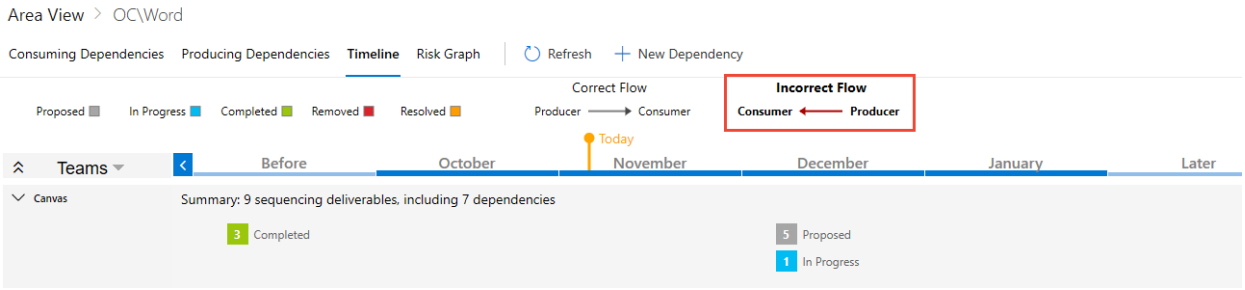
Correct Flow view

The Correct Flow view shows those dependencies that are in the correct sequence. Successor work items are scheduled to be completed after their predecessor work item.



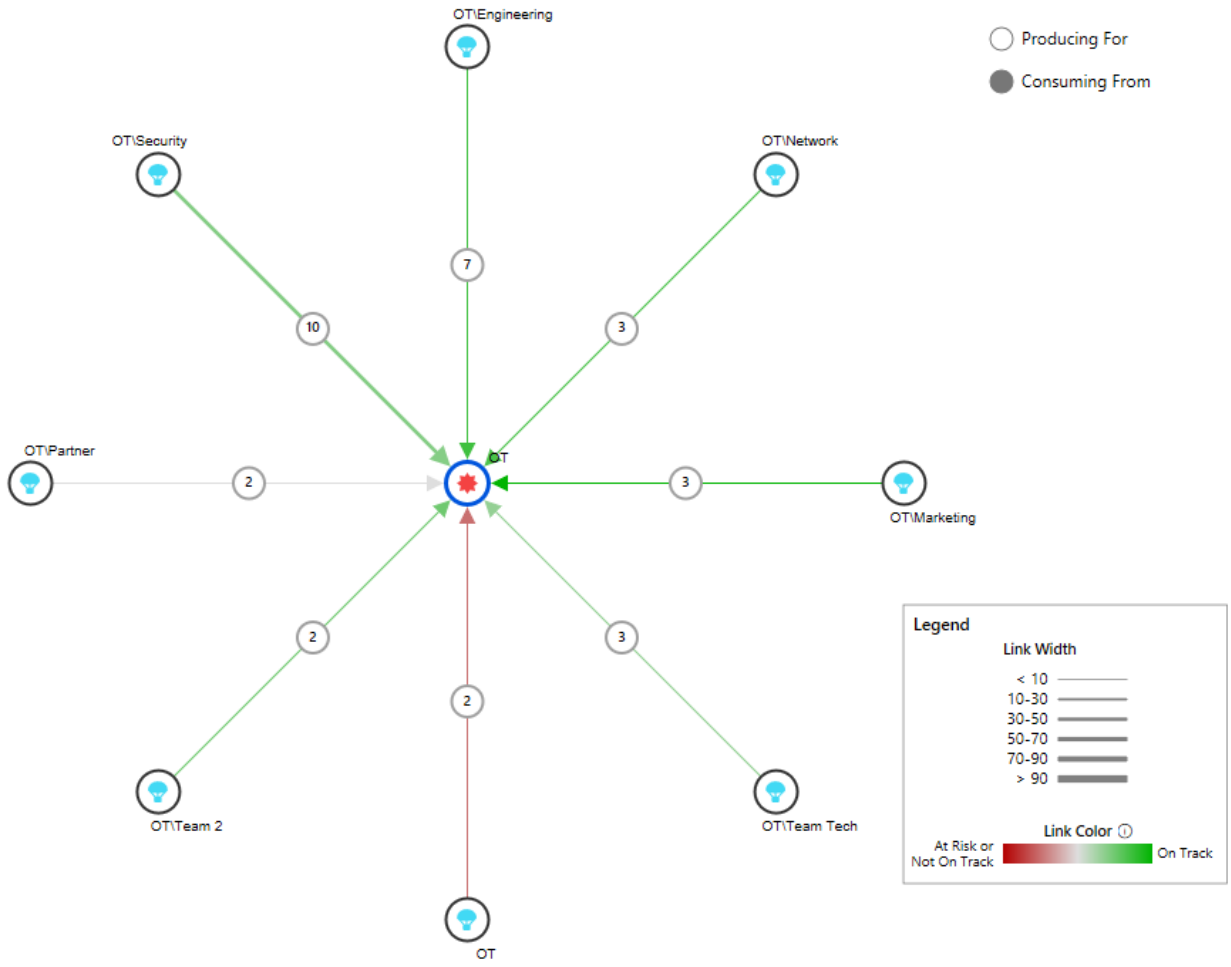
Incorrect Flow view

The Incorrect Flow view shows those dependencies that are out of order. At least one predecessor work item is scheduled to be completed after its successor work item.



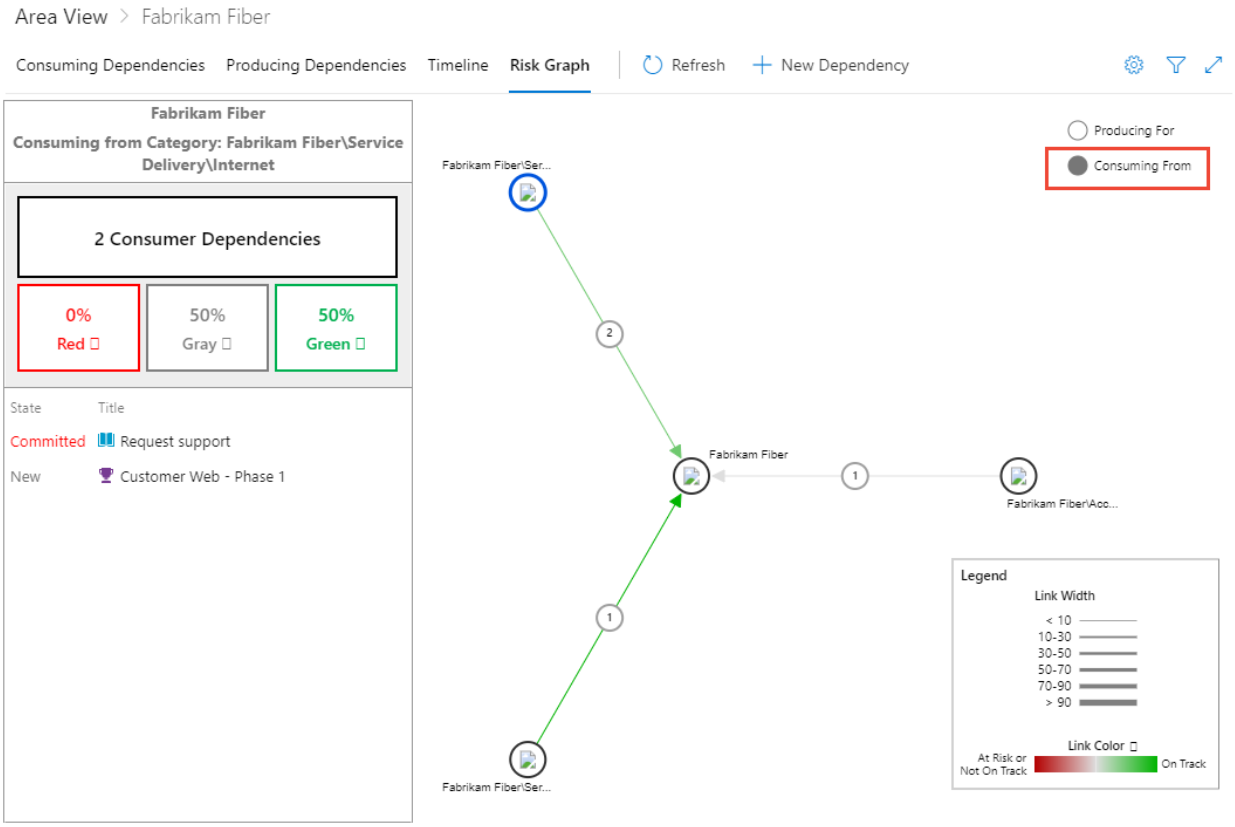
Risk Graph

The Risk Graph provides a visualization of how dependencies flow from Consumer team to Producer team, or from Producer to Consumers. The graph allows a team to, at a glance, understand the number of dependencies and level of risks associated. Also, the risk graph view demonstrates the value of linking dependencies and laddering them up to Stories.

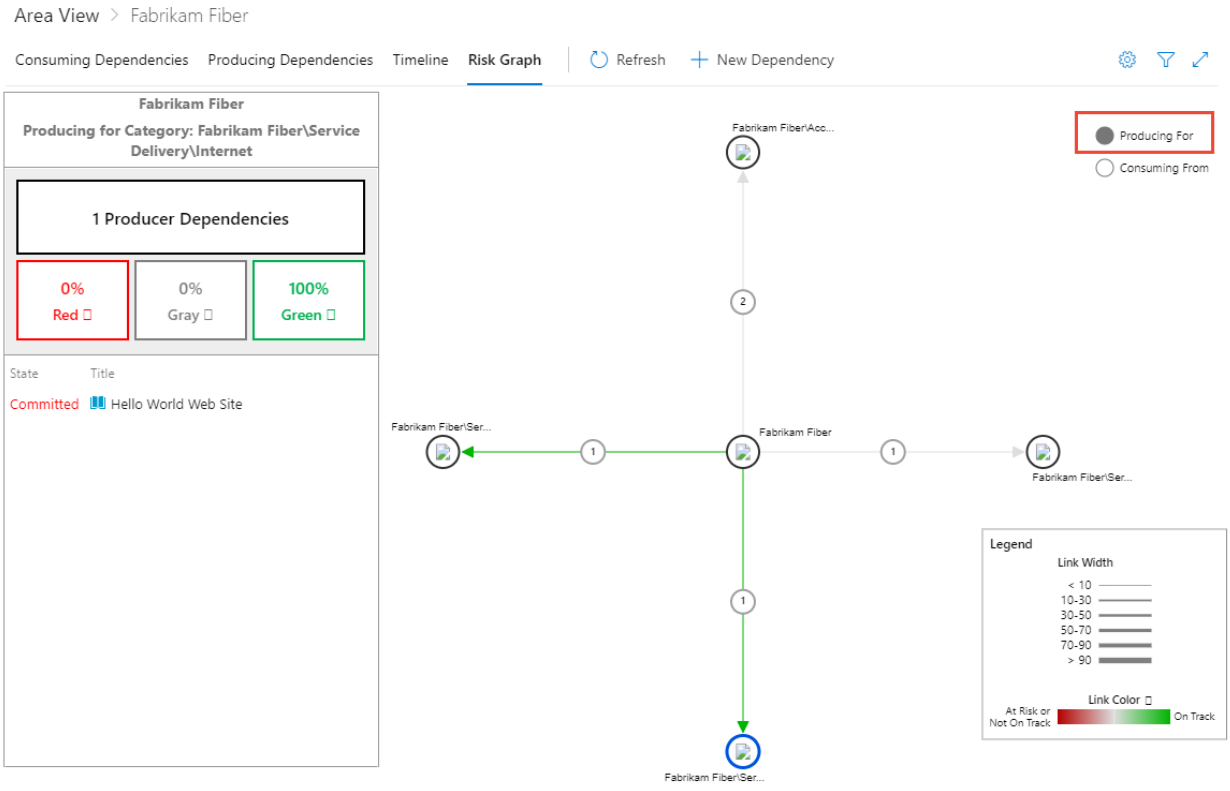


There are two views: **Consuming From** and **Producing For**. The workflow state color coding is configurable. The width of the lines indicates how many dependencies exist in that area, the thicker the link the more dependencies as indicated in the legend.

Consuming From

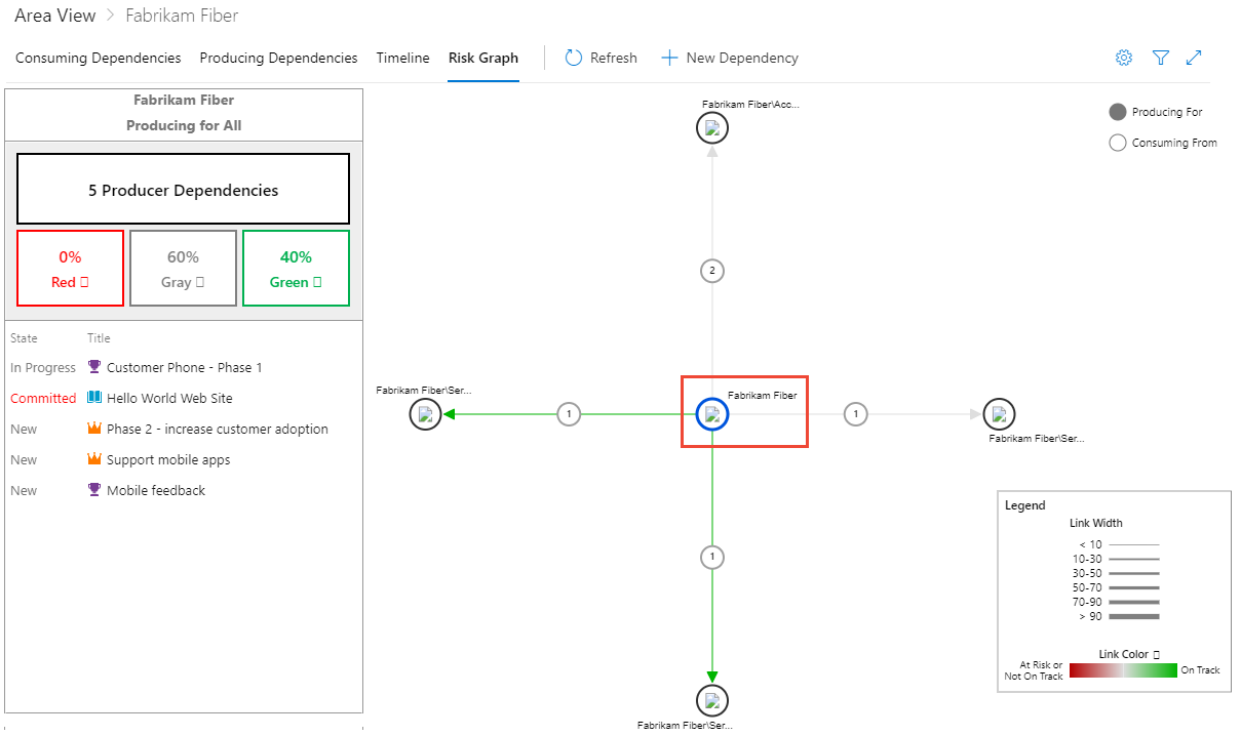


Producing For



Filtered on a specific dependency

You can drill down into specifics by choosing one of the dependencies.



Configure the Dependency Tracker

You must be a member of the Project Collection Administrator Group to modify the configuration. All changes to the configuration apply to all projects defined in the organization.

To change the configuration, choose the gear icon and modify the syntax listed. Choose **Save** when done.

The main properties you can modify are summarized as follows:

- The link types to use to create dependency links. Defaults are the Successor/Predecessor link types. Only customize when you use the Hosted XML process model to customize work tracking.
- Work items and work item types
  - Work item types to participate in dependency tracking
  - Mapping of work item category states to colors
  - Mapping of work item workflow states and colors
- Default field columns in dependency list tables
- Default filter selections:
  - Selected dependency work item types
  - Selected Iteration Paths
- Enabled options:
  - Timeline
  - New Dependency link
  - Cross account (organization) dependencies
  - Cross account dependency toggle default state
- Risk graph configuration:
  - Work item state(s) associated with at risk (Red color) work items
  - Work item state(s) associated with neutral (Gray color) work items
  - Work item state(s) associates with on track (Green color) work items

For a full list and description, see the Property descriptions provided later in this section.

Enable or disable the New Dependency option

The `newDependencyButtonEnabled` property enables or disables the **New Dependency** link option. When enabled, the link appears on the Dependency Tracker page. When disabled, users can't create dependencies from the tracker, only review the dependencies that have been created through other means. The default value is set to `true` (enabled).

Enable or disable cross-organization linking

The `crossAccountConfigs` property enables or disables cross-organization dependency linking from the New dependency dialog. The default value is set to `true` (enabled).

To disable, set the following syntax in the JSON configuration to `false`.

```
{
"crossAccountConfigs": {
  "crossAccountDependencyEnabled": false,
  "crossAccountDependencyToggleDefaultState": false, //default state for cross account toggle
  "crossAccountDependencyToggleOnText": "Cross-account dependencies on",
  "crossAccountDependencyToggleOffText": "Cross-account dependencies off"}
}
```

Cross account linking requires the use of a special link type and should only be used in coordination with the **New Dependency** option.

Property descriptions

The following table describes each of the property items specified in the configuration file.

Property/Description

Default/Example

consumesLinkName

> Specifies the link type used to create the link from producer to consumer.

System.LinkTypes.Dependency-Reverse

producesLinkName

> Specifies the link type used to create the link from consumer to producer.

System.LinkTypes.Dependency-Forward

queryFields

> Specifies the custom fields to use in place of the system fields used by the dependency tracker to return linked work item results. By default. system reference names are used to return values for the following fields:
>
> - areaPath - Area Path
> - assignedTo - Assigned To
> - id - ID
> - areapath - IterationID
> - areapath - Iteration Path
> - areapath - Priority
> - areapath - State
> - areapath - Tags

- teamProject - Team Project
- title - Title
- workItemType - Work Item Type

If a custom field is used in place of one of the system fields, you specify the substitution by entering:

```
{
title: "Custom.Title",
assignedTo: "Custom.AssignedTo"
}
```

dependencyWorkItemTypes

> Specifies the work item types that participate in dependency tracking. From the Create a dependency dialog, only those work item types listed can be created.

Default:

```
[
"Epic",
"Feature",
"User Story",
"Bug"
]
```

If using the Scrum process, you would change the entry to:

```
[
"Epic",
"Feature",
"Product Backlog Item",
"Bug"
]
```

selectedDependencyWorkItemTypes

> Restricts the initial focus to just those work item types that the dependency tracker displays or lists. Based on the default "Any", any work item type that contains a dependency link type is displayed or listed. Users can change the focus through filtering.

Default:

```
Any
```

To restrict the work item types to just Epics and Features, specify:

```
[
"Epic",
"Feature"
]
```

selectedReleases

> Restricts the initial focus to just those work items that are assigned to those Iteration Paths equal to or under the specified releases. Based on the blank default, no restrictions are applied. Users can change the focus through filtering.

Default:

```
[]
```

To restrict the work item types to just Release 1 and Release 2 for the Fabrikam project, specify:

```
[
"Fabrikam/Release 1",
"Fabrikam/Release 2",
]
```

workItemCategoriesAndColors

> Specifies the colors used to represent work items based on their category and workflow state. For more information, see How workflow states and state categories are used in Backlogs and Boards.

Default:

```
{
"Proposed": {
"displayName": "Proposed",
"color": "#a6a6a6"
},
"InProgress": {
"displayName": "In Progress",
"color": "#00bcf2"
},
"Completed": {
"displayName": "Completed",
"color": "#9ac70b"
},
"Removed": {
"displayName": "Removed",
"color": "#d9242c"
},
"Resolved": {
"displayName": "Resolved",
"color": "#ff9d00"
}
}
```

workItemDislayStatesAndDisplayColors

> Maps the workflow states to colors used to display them. If you customize the workflow states, or use a process that uses different workflow states, you must update this property.

Default:

```
{
"New": {
"textColor": "rgb(112, 112, 112)",
"chartColor": "rgb(112, 112, 112)",
"states": [
"New"
]
},
"Active": {
"textColor": "rgb(0, 122, 204)",
"chartColor": "rgb(0, 122, 204)",
"states": [
"Active",
"Resolved"
]
},
"Closed": {
"textColor": "rgb(16, 124, 16)",
"chartColor": "rgb(16, 124, 16)",
"states": [
"Closed"
]
},
"Removed": {
"textColor": "rgb(204, 41, 61)",
"chartColor": "rgb(204, 41, 61)",
"states": [
"Removed"
]
},
"Other": {
"textColor": "rgb(178, 178, 178)",
"chartColor": "rgb(178, 178, 178)",
"states": []
}
}
```

riskAssessementValues

> Specifies the Risk field values. The Risk field specifies a subjective rating of the relative uncertainty around the successful completion of a user story. It is defined for the Agile process, but can be added to work item types used in other processes.

Default:

```
["1-High", "2-Medium", "3-Low"]
```

partnerAccounts

> Optional configuration that specifies which Azure DevOps organizations are selectable from the Dependency dialog when creating a Cross account dependency. If not specified it will generate a list based on previous organizations that the user has visited.

Default:

```
[]
```

Example:

```
["account-1", "account-2"]
```

timelineEnabled

> Enables or disables the Timeline view.

Default:

```
true
```

newDependencyButtonEnabled

> Enables or disables the **New Dependency** link to create a new linked dependency.

Default:

```
true
```

crossAccountConfigs

> (1) Enables or disables the support of creating new dependencies to work items in other partner accounts, and (2) specifies the default state of the Partner account options in the Create a dependency dialog.

Default:

```
{
"crossAccountDependencyEnabled": true,
"crossAccountDependencyToggleDefaultState": false
}
```

If you don't want any dependencies created that belong to other organizations, then change this configuration to:

```
{
"crossAccountDependencyEnabled": false,
"crossAccountDependencyToggleDefaultState": false
}
```

priorityValues

> Specifies the Priority field values. The Priority field specifies a subjective rating of a bug, issue, task, or user story as it relates to the business. It is defined for most backlog work item types and processes, but can be added to work item types used in other processes.

Default:

```
["0","1","2","3","4","(blank)"]
```

defaultColumns

> Specifies the field columns and order used to display dependency lists.

Default:

```
[
"Id",
"Area Path",
"Dependency Title",
"State",
"Consumers",
"Producers"
]
```

riskAnalysisEnabled

> Specifies whether or not Risk functionality is enabled. If set to true, then the riskAssessmentValues property must be defined.

Default:

```
False
```

riskAssessmentValues

Default:

```
[]
```

riskGraphConfig

> Maps the workflow States to one of the three Risk areas displayed on the Graph: `atRisk` is Red, `nuetral` is Gray, and `onTrack` is Green.

Default: 8

```
{
"atRisk": [
"Removed"
],
"neutral": [
"New"
],
"onTrack": [
"Active",
"Resolved",
"Closed",
"Other"
]
}
```

Add or remove workflow states used in work item types participating in dependency tracking.

iterationDepth

> Specifies the hierarchical depth of Iteration Paths that the Dependency Tracker queries to build the Timeline view.

Default: 8A depth of 3 would correspond to: Fabrikam/Release 1/Sprint 20.

Default configuration syntax

```
{
    "consumesLinkName": "System.LinkTypes.Dependency-Reverse",
    "producesLinkName": "System.LinkTypes.Dependency-Forward",
    "queryFields": {},
    "dependencyWorkItemTypes": [
        "Epic",
        "Feature",
        "User Story",
        "Bug"
    ],
    "selectedDependencyWorkItemTypes": "Any",
    "selectedReleases": "",
    "workItemCategoriesAndColors": {
        "Proposed": {
            "displayName": "Proposed",
            "color": "#a6a6a6"
        },
        "InProgress": {
            "displayName": "In Progress",
            "color": "#00bcf2"
        },
        "Completed": {
            "displayName": "Completed",
            "color": "#9ac70b"
        },
        "Removed": {
            "displayName": "Removed",
            "color": "#d9242c"
        },
        "Resolved": {
            "displayName": "Resolved",
            "color": "#ff9d00"
        }
    },
    "workItemDislayStatesAndDisplayColors": {
        "New": {
            "textColor": "rgb(112, 112, 112)",
            "chartColor": "rgb(112, 112, 112)",
            "states": [
                "New"
            ]
        },
        "Active": {
            "textColor": "rgb(0, 122, 204)",
            "chartColor": "rgb(0, 122, 204)",
            "states": [
                "Active",
                "Resolved"
            ]
        },
        "Closed": {
            "textColor": "rgb(16, 124, 16)",
            "chartColor": "rgb(16, 124, 16)",
            "states": [
                "Closed"
            ]
        },
        "Removed": {
            "textColor": "rgb(204, 41, 61)",
            "chartColor": "rgb(204, 41, 61)",
            "states": [
                "Removed"
            ]
        },
```

```
        "Other": {
            "textColor": "rgb(178, 178, 178)",
            "chartColor": "rgb(178, 178, 178)",
            "states": []
        }
    },
    "riskAssessmentValues": [],
    "releases": [],
    "partnerAccounts": [],
    "timelineEnabled": true,
    "newDependencyButtonEnabled": true,
    "crossAccountConfigs": {
        "crossAccountDependencyEnabled": true,
        "crossAccountDependencyToggleDefaultState": false
    },
    "priorityValues": [
        "0",
        "1",
        "2",
        "3",
        "4",
        "(blank)"
    ],
    "defaultColumns": [
        "Id",
        "Area Path",
        "Dependency Title",
        "State",
        "Consumers",
        "Producers"
    ],
    "riskGraphConfig": {
        "atRisk": [
            "Removed"
        ],
        "neutral": [
            "New"
        ],
        "onTrack": [
            "Active",
            "Resolved",
            "Closed",
            "Other"
        ]
    },
    "iterationDepth": 8
}
```


Related articles

- Work item field index
- Review team delivery plans
- Inheritance process model
- Hosted XML process model
- How workflow states and state categories are used in Backlogs and Boards

Related Marketplace extensions

- Work Item Visualization

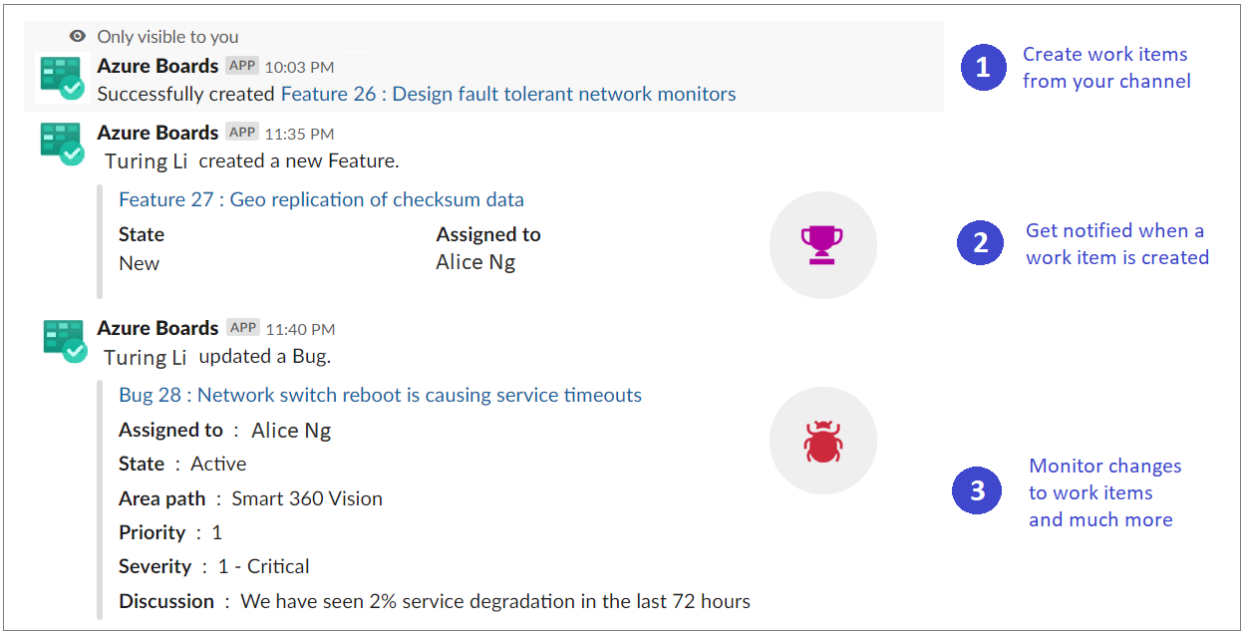
Use the Azure Boards app with Slack to manage work items

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services

If you use Slack, you can use the Azure Boards app for Slack to create work items and monitor work item activity in your Azure Boards project from your Slack channel.

The Azure Boards app for Slack allows users to set up and manage subscriptions in their Slack channel. They can manage subscriptions for create, update, and other work item events. Users can also get notifications for these events in their Slack channel. Conversations in the Slack channel can be used to create work items. Previews for work item URLs help users to start discussions around work.



Read this article to learn how to:

- Add the Azure Boards app to your Slack workspace
- Link and unlink your Azure Boards project to the Azure Boards app
- Set up subscriptions to work item related events in your Slack channel
- Create work items from your Slack channel
- Monitor work item activity in your Slack channel
- Get notifications in private Slack channels

NOTE

- Azure Boards and Slack integration is only supported for Azure DevOps Services.
- Notifications are currently not supported inside direct messages.

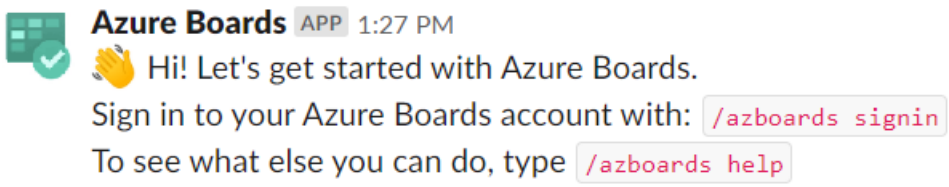
Prerequisites

- To create a work item, you must be a contributor to the Azure Boards project. If you don't have a project yet, you can sign up and create a project. For details, see Start using Azure Boards.

- To create subscriptions in a Slack channel for work item events, you must be a member of the Azure Boards Project Administrators group or Team Administrators group. To get added, see Change project-level permissions or Add Team Administrator.
- To receive notifications, the **Third party application access via OAuth** setting must be enabled for the organization. See Change application access policies for your organization

Add the Azure Boards app to your Slack workspace

1. To install the Azure Boards app to your Slack workspace, open a web browser, sign into Slack, and open the Azure Boards app.
2. Once added, you'll see a welcome message from the app as shown in the following image.

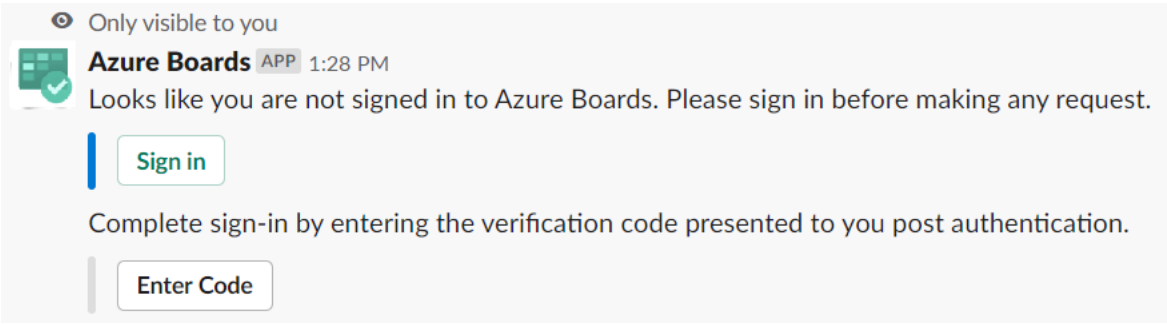


3. Use the `/azboards` Slack handle to interact with the app. A list of commands is provided later in this article, Command reference.

Link your Azure Boards project to the Azure Boards app

To use the app, you must first link your Azure Boards project to your Slack channel.

1. Once the app has been installed in your Slack workspace, connect and authenticate yourself to Azure Boards.



2. After signing in, use the following slash command inside a Slack channel to link to the Azure Boards project that you specify with the URL:

```
/azboards link [project url]
```

   For example:

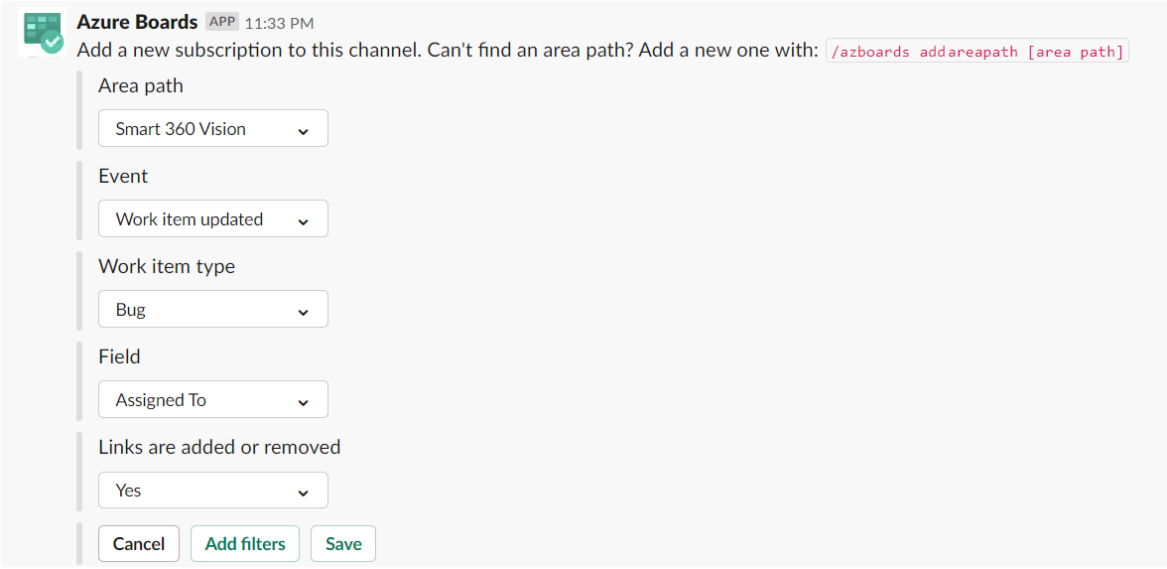
```
/azboards link https://dev.azure.com/myorg/myproject
```

Once the project is linked, you can create work items using `/azboards create` command or use message actions.

Set up subscriptions to monitor work items

You can create subscriptions to monitor work items at any time using the `/azboards subscriptions` command. You have an option of setting up subscriptions just after linking a project.

1. Select area path you want, the event that you're interested in, and use the associated filters to customize your Slack channel. To easily set up subscriptions, your recently accessed area paths are shown in the area path dropdown.



   In case your team's area path doesn't appear in the Area path dropdown menu, follow the instructions mentioned in the next section, Add area paths. Area paths added using the `/azboards addAreapath` command and area paths for which subscriptions are created in the Slack channel always appear in the Area path dropdown along with recently accessed area paths.

Add area paths

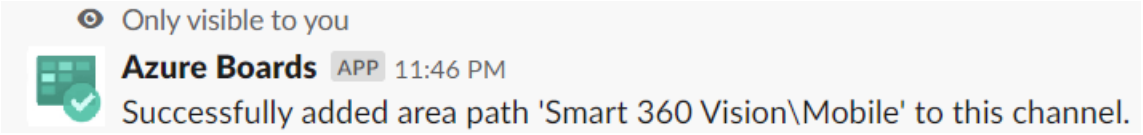
You can add areas that your team works on to the channel so that they're always available for creating work items and subscriptions. This is important mainly for the teams with more than 100 area paths.

- Use the following command to add area paths from your project to the Slack channel.

  ```
  /azboards addAreapath [area path]
  ```

  For example:

  ```
  /azboards addAreapath myproject\fabrikam
  ```

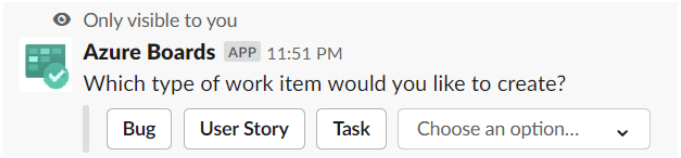


- If you choose project name as your area path, then you'll receive notifications for all the area paths in the project. It's logically equivalent to choosing 'Any' area path.

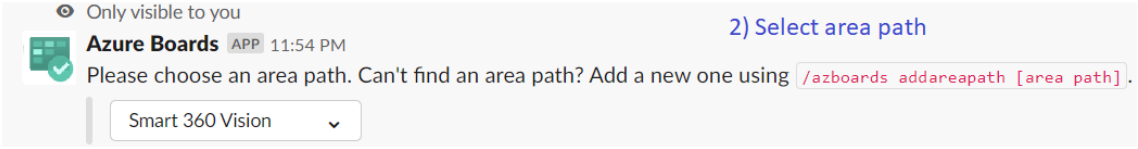
Create a work item with a command

1. With the Azure Boards app, you can create work items from your channel. The app supports custom work items as well.

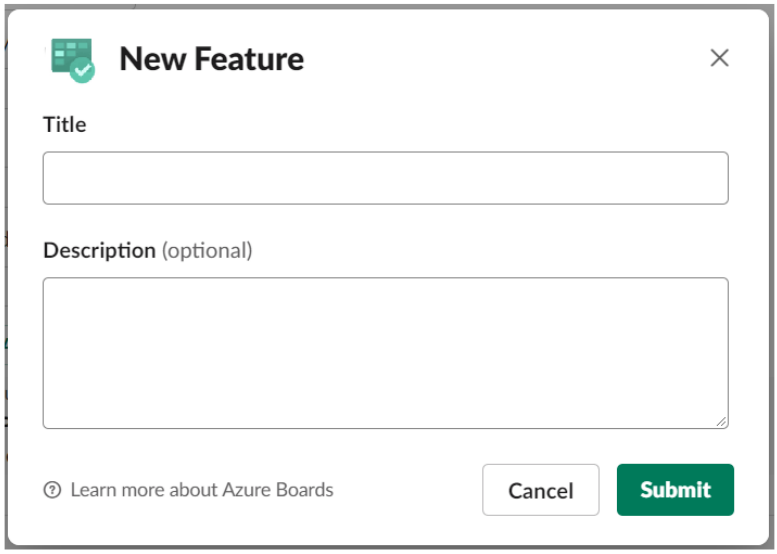
- To create a work item, use `/azboards create`.



1) Choose work item type



2) Select area path



3) Fill necessary details

2. You can create work items directly from a command by passing work item type and title as parameters. Work items will be created only if they don't have any fields to be mandatorily filled.

```
/azboards create [work item type] [work item title]
```

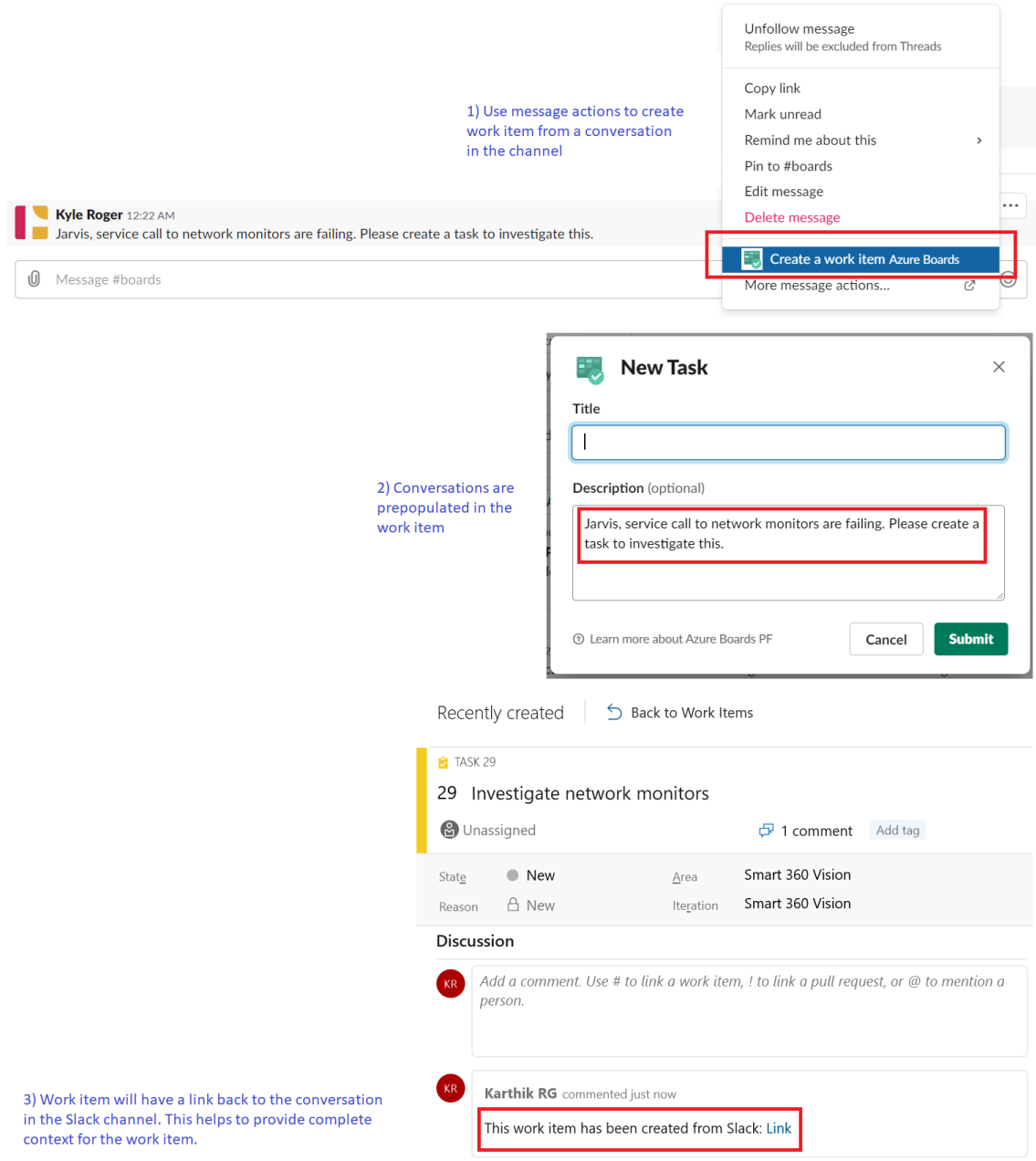
For example:

```
/azboards create 'user story' Push cloud monitoring alerts to mobile devices
```

Create a work item from message actions

Often, discussions in a channel call for creation of work items. You can use message actions to create a work item. The selected message is prefilled in the description section of the work item. A link back to the conversation in the channel is stored in the Discussion section of the newly created work item, giving users access to the discussion that led to the creation of the work item.

- To create work items using message actions



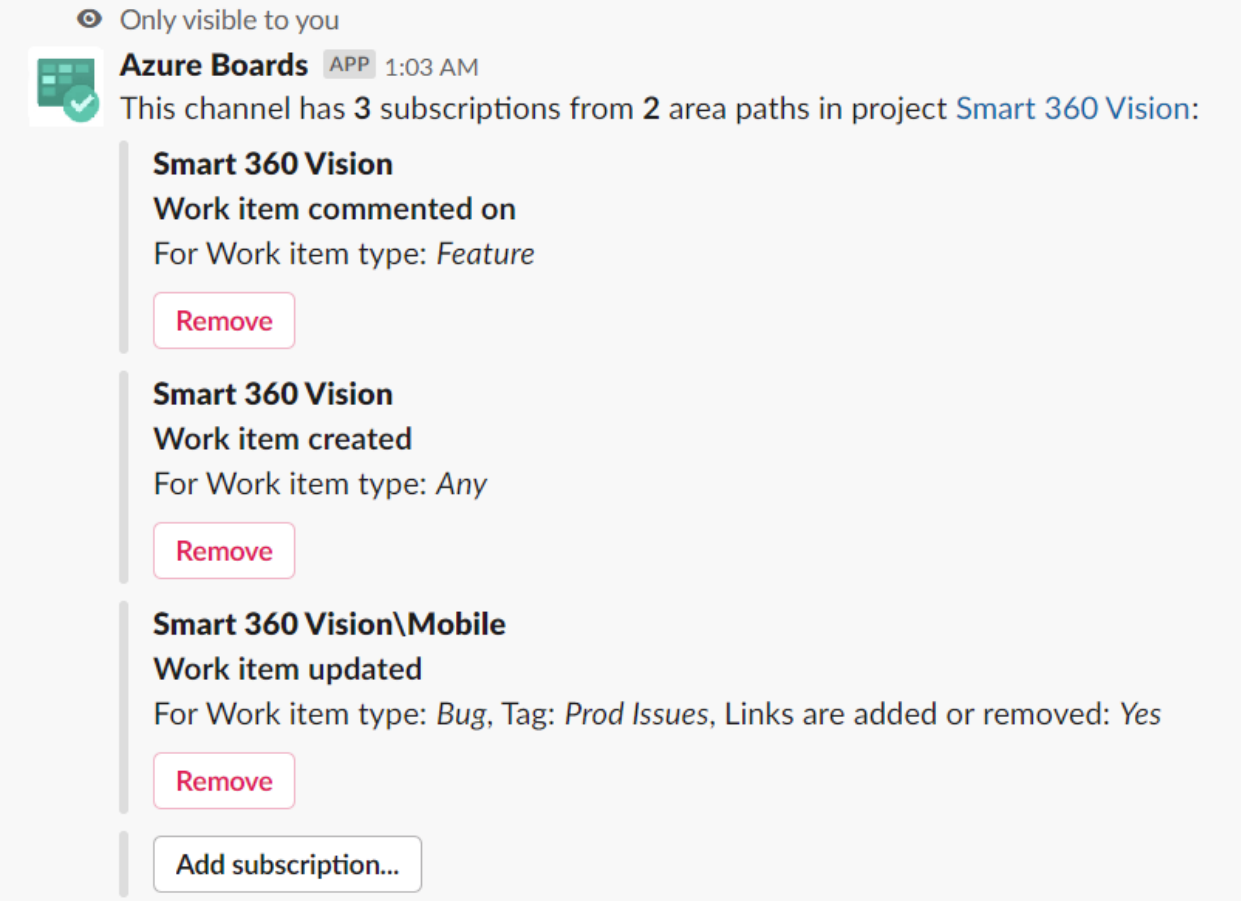
Manage Azure Boards subscriptions

1. To view, add and remove subscriptions for a channel, use the `/azboards subscriptions` command:

   ```
   /azboards subscriptions
   ```

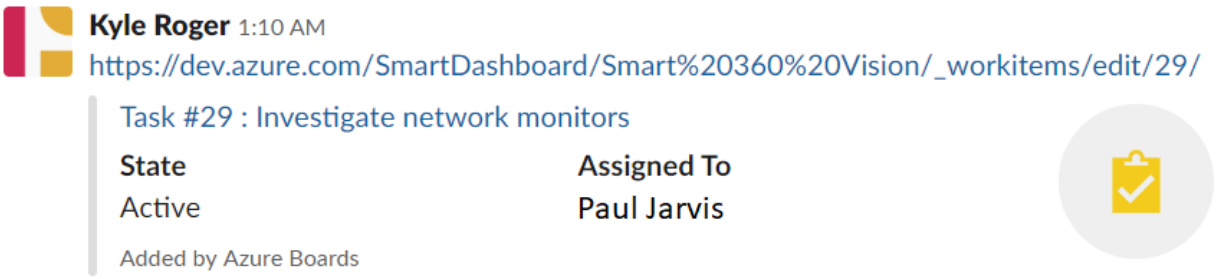
   This command lists all the current subscriptions for the channel and allows you to add new subscriptions and remove existing ones. As part of adding subscriptions, you can also customize what you get notified on by using various filters.

[!NOTE]Team administrators aren't able to remove or modify subscriptions created by Project administrators.



Previews of work item URLs

To support collaboration around work items discussed within a channel, a preview of work items referenced in the channel is displayed. When a user pastes the work item URL, a preview is shown similar to the following image. This preview helps to keep work-item-related conversations relevant and correct.



For this feature to work, users have to be signed-in. Once they're signed in, this feature will work for all channels in a workspace.

Unlink a project from a channel

A Slack channel can only link to one Azure Boards project at a time. To link to a different project, you must first unlink the current project using `/azboards unlink` command.

Unlinking a project deletes all the subscriptions along with added area paths from the channel. If the channel has no subscriptions, any user can unlink a project. However if a channel has subscriptions, only project admins can unlink a project from a channel.

Command reference

The following table lists all the `/azboards` commands you can use in your Slack channel.

SLASH COMMAND	FUNCTIONALITY
/azboards link [project url]	Link a project to this channel to create work items and receive notifications
/azboards subscriptions	Add or remove subscriptions for this channel
/azboards create or /azboards create [work item type] [title]	Create a work item
/azboards addAreapath [area path]	Add an area path from your project to this channel
/azboards signin	Sign in to your Azure Boards organization
/azboards signout	Sign out from your Azure Boards organization
/azboards unlink	Unlink a project from this channel
/azboards feedback	Report a problem or suggest a feature

Manage work in private channels

The Azure Boards app for Slack can help you create work items and monitor the work item activity in your private channels as well. To invite the bot to your private channel, enter `/invite @azboards`. After you post that, you can create work items and manage your notifications in the same way as you would for a public channel.

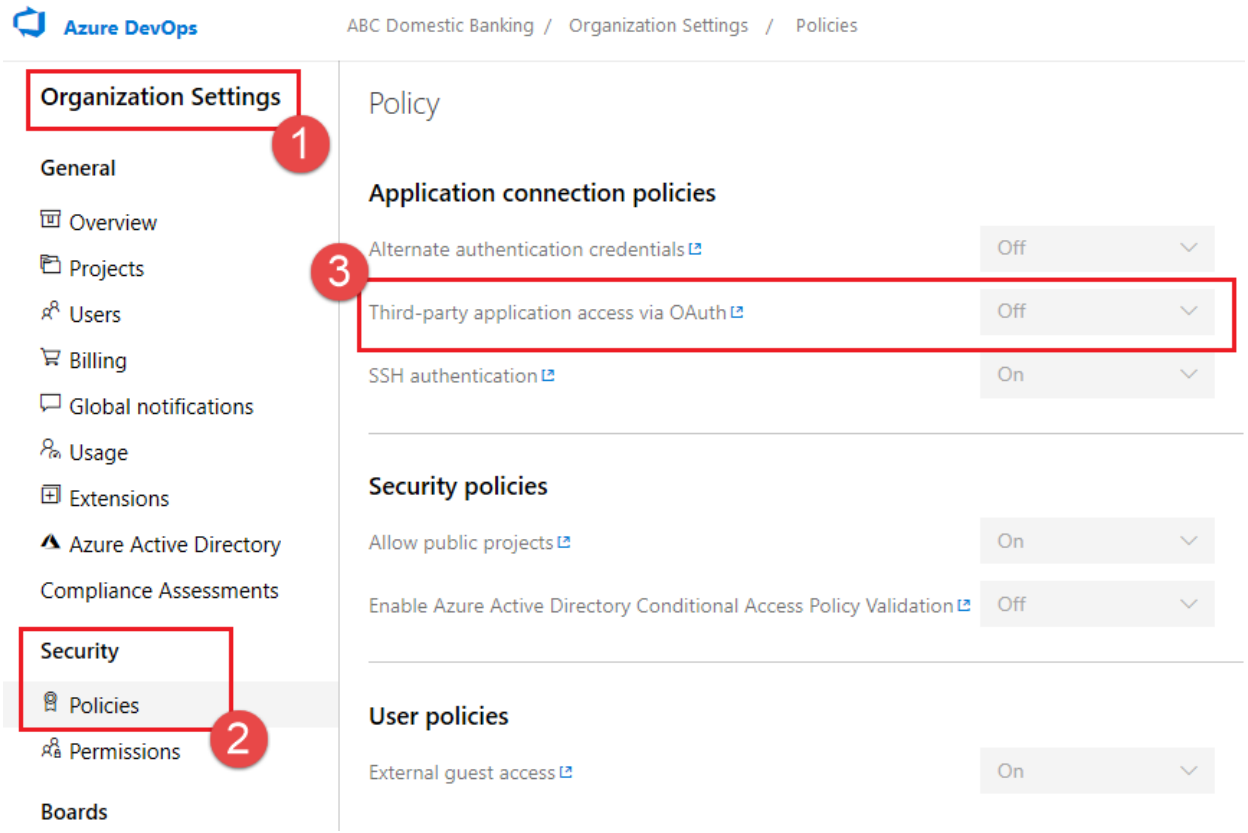
Troubleshoot errors

If you're experiencing the following errors when using the Azure Boards App for Slack, follow the procedures in this section.

- Sorry, something went wrong. Please try again.
- Configuration failed. Please make sure that the organization '{organization name}' exists and that you have sufficient permissions.

Sorry, something went wrong. Please try again.

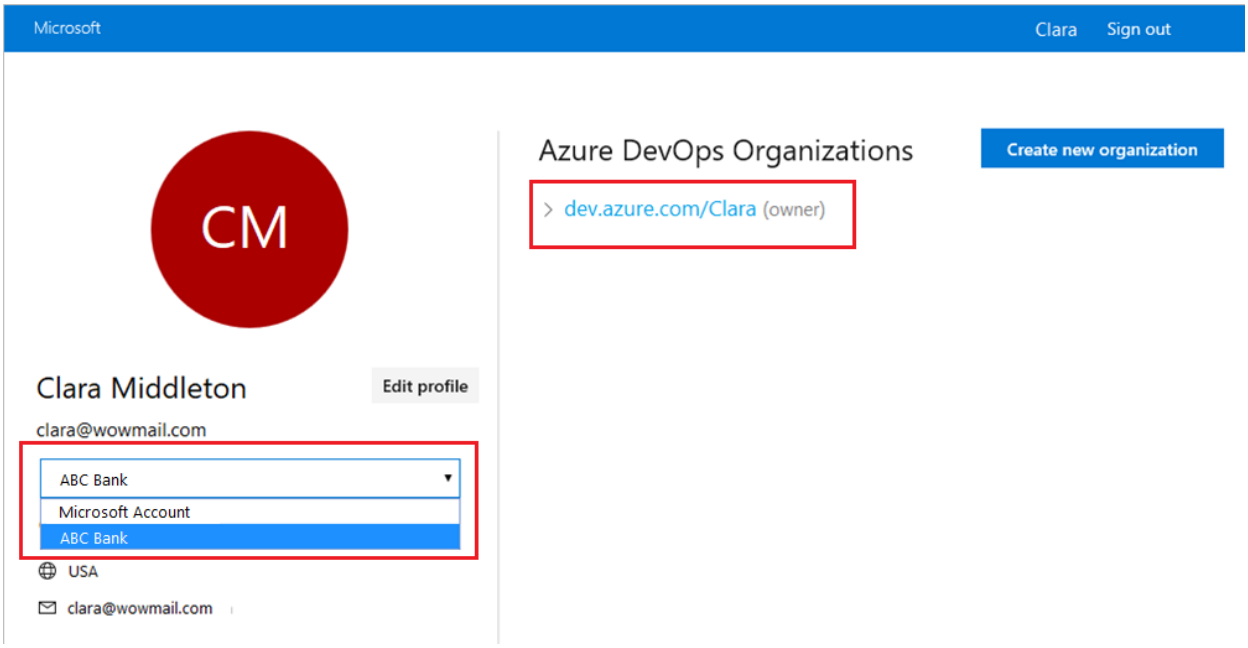
The Azure Boards app uses the OAuth authentication protocol, and requires Third-party application access via OAuth for the organization to be enabled. To enable this setting, navigate to **Organization Settings** > **Security** > **Policies**, and set the **Third-party application access via OAuth for the organization** setting to **On**.



Configuration failed. Please make sure that the organization '{organization name}' exists and that you have sufficient permissions.

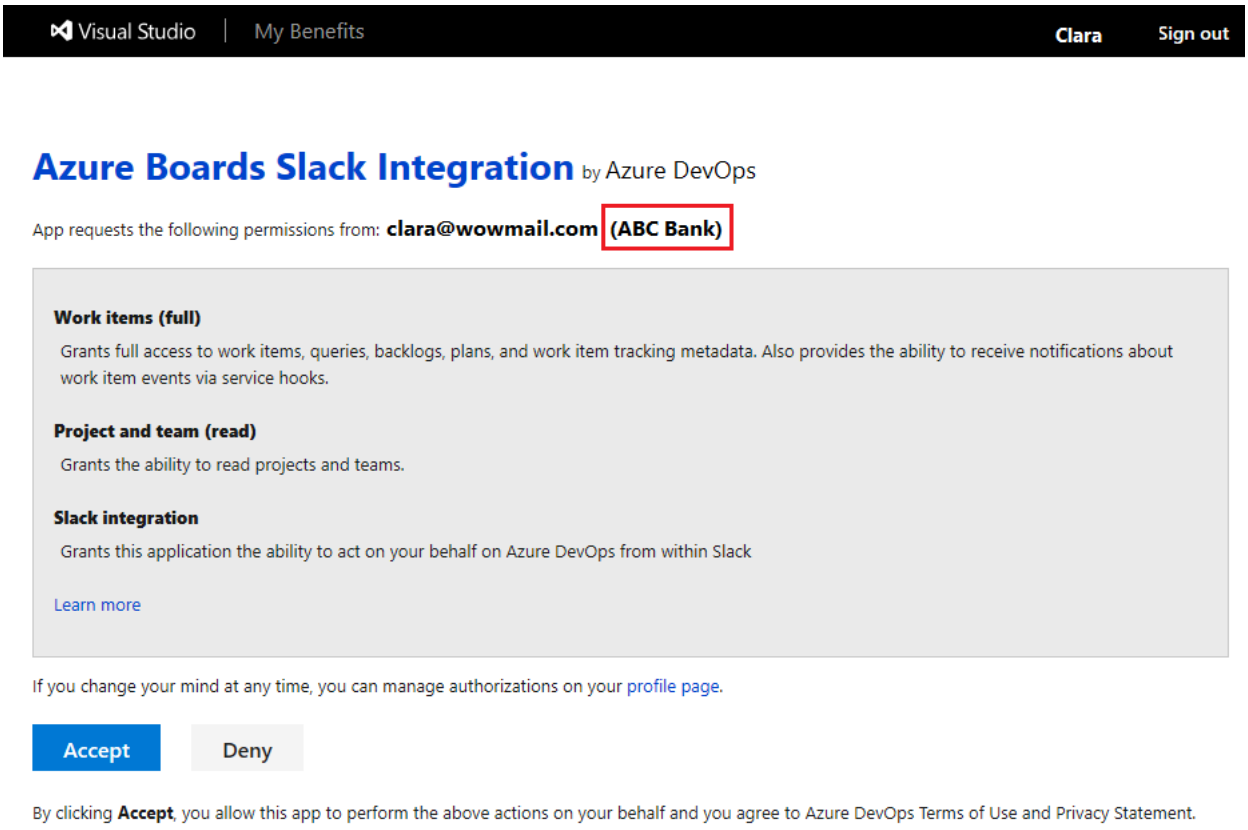
Sign out of Azure DevOps by navigating to `https://aka.ms/VsSignout` using your browser.

Open an **In private** or **incognito** browser window and navigate to `https://aex.dev.azure.com/me` and sign in. In the dropdown under the profile icon to the left, select the directory that contains the organization containing the project that you want to link.



In the **same browser**, start a new tab, navigate to `https://slack.com`, and sign in to your work space (**use web client**). Run the `/azboards signout` command followed by the `/azboards signin` command.

Select the `Sign in` button and you'll be redirected to a consent page like the one in the following example. Ensure that the directory shown beside the email is same as what was chosen in the previous step. Accept and complete the sign-in process.



If these steps don't resolve your authentication issue, reach out to us at Developer Community.

Related articles

- Define area paths and assign to a team
- Azure Pipelines with Slack
- Azure Repos with Slack
- Create a service hook for Azure DevOps with Slack

Use the Azure Boards app in Microsoft Teams

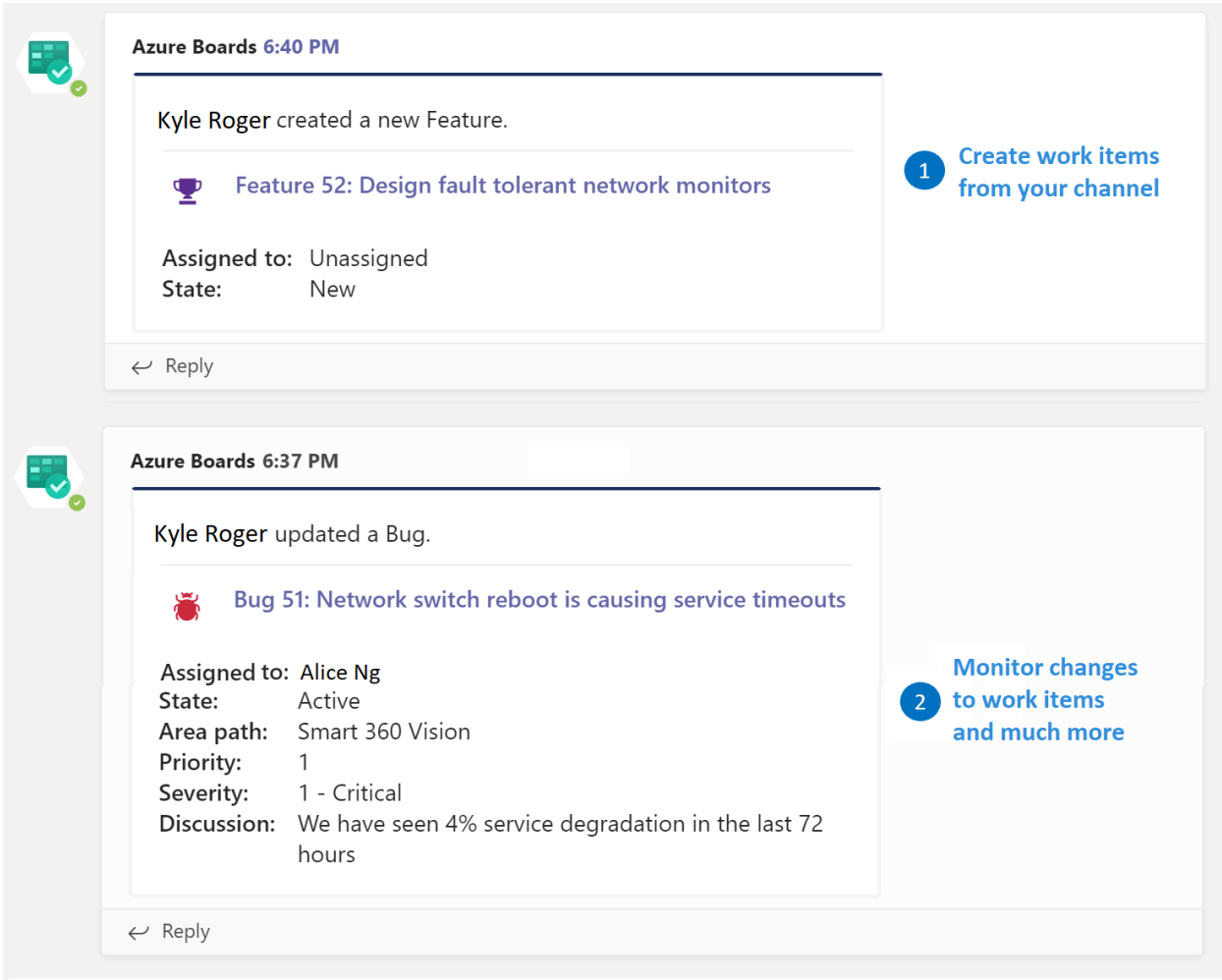
10/4/2022 • 9 minutes to read • Edit Online

Azure DevOps Services

If you use Microsoft Teams, you can create work items and monitor work item activity in your Azure Boards project from your Teams channel. You accomplish this by adding the Azure Boards app for Microsoft Teams to your Teams channel.

The Azure Boards app for Microsoft Teams enables users to complete the following tasks:

- Set up and manage subscriptions for creating and updating work items
- Manage other work item events
- Receive and manage notifications for work item events in their Teams channel
- Create work items from conversations in the channel
- Search and share work items with other members in the channel using the messaging extension
- View work item previews from their URLs to start discussions and keep the conversations contextual.



Read this article to learn how to:

- Add the Azure Boards app to your team in Microsoft Teams
- Link and unlink your Azure Boards project to the Azure Boards app
- Set up subscriptions to work item related events in your Teams channel
- Create work items from your Teams channel

- Monitor work item activity in your Teams channel

> **NOTE**
> Azure Boards and Microsoft Teams integration is only supported for Azure DevOps Services.

Prerequisites

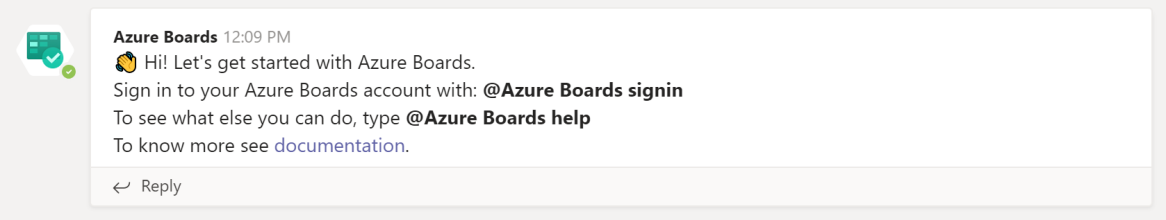
- To create a work item, you must be a contributor to the Azure Boards project. If you don't have a project yet, you can sign up and create a project. For details, see Start using Azure Boards.
- To create subscriptions in a Teams channel for work item events, you must be a member of the Azure Boards Project Administrators group or added to the team administrator role for the team. To get added, see Change project-level permissions or Add team administrator.
- To receive notifications, you must enable the **Third-party application access via OAuth** setting for the Azure DevOps organization. See Change application access policies for your organization.

> **NOTE**
> - You can link the Azure Boards app for Microsoft Teams only to a project hosted on Azure DevOps Services at this time.
> - Notifications are currently not supported inside direct messages.
> - Only public channels are supported.

Add the Azure Boards app to Microsoft Teams

You add the app to your Teams channel in Microsoft Teams.

1. Visit the App store in Microsoft Teams and search for the Azure Boards app. Upon installing, a welcome message from the app displays as shown in the following image.

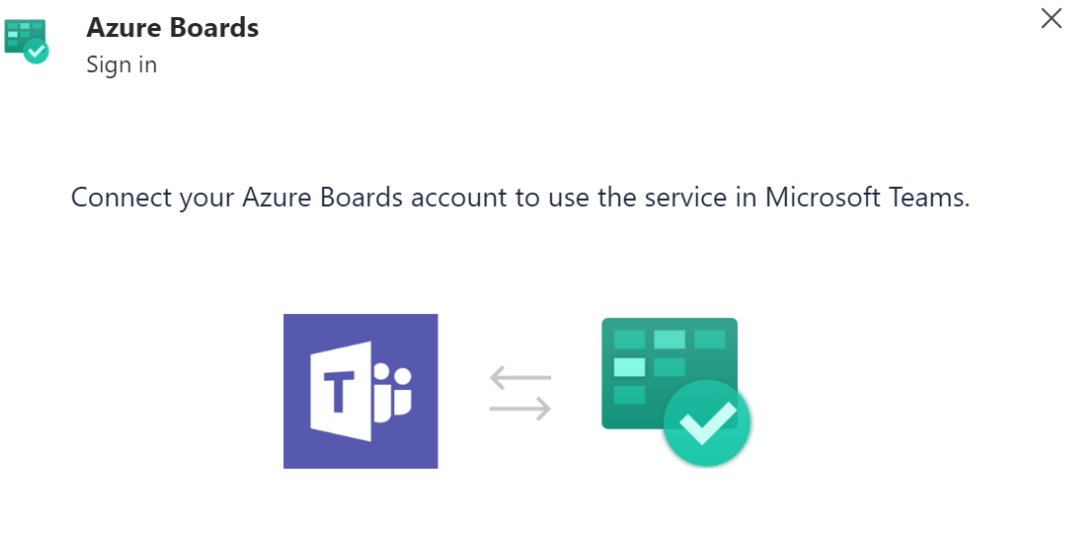


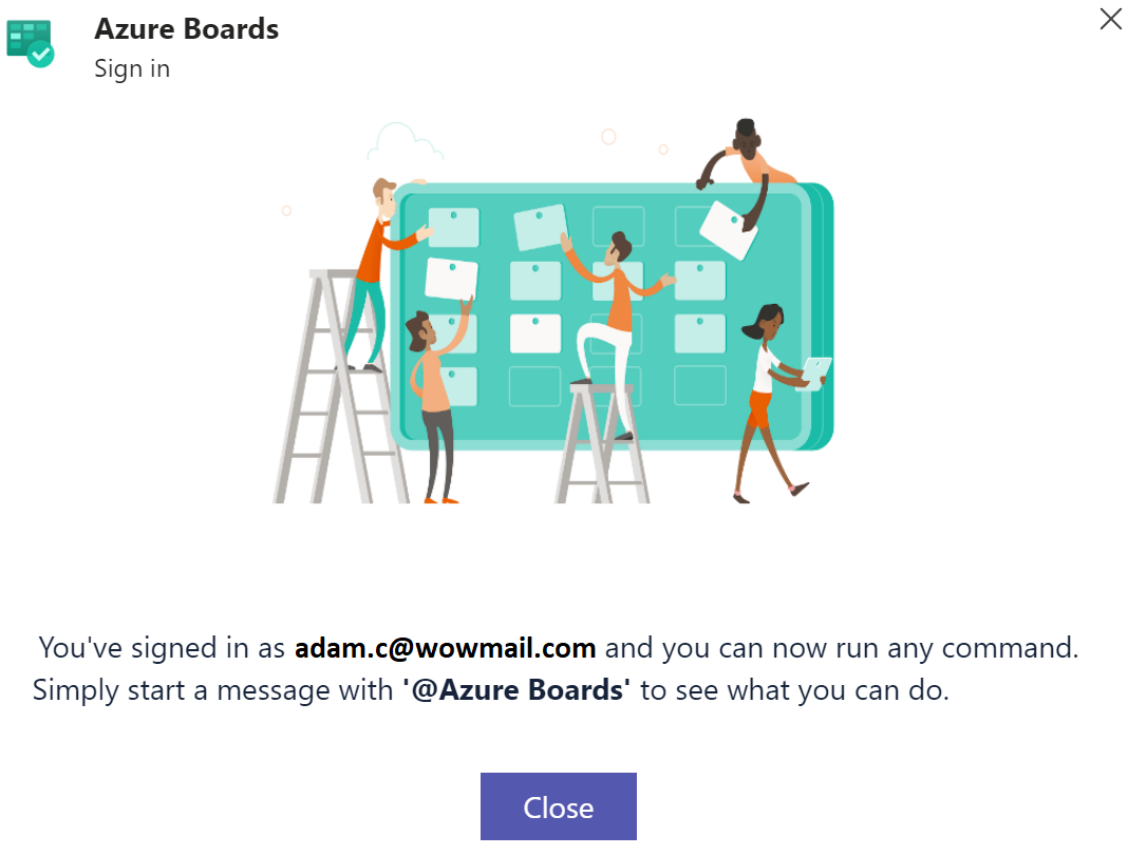
2. Use the `@azure boards` handle to interact with the app. For a list of commands, see Command reference provided later in this article.

Link your Azure Boards project to the Azure Boards app

To use the app, you must first link your Azure Boards project to your Teams channel.

1. Once the app has been installed in your team, connect and authenticate yourself to Azure Boards. Use **Sign in with different email** if your Microsoft Teams and Azure Boards are in different tenants.





2. After signing in, use the following command inside a Teams channel to link to the Azure Boards project that you specify with the URL:

```
@azure boards link [project url]
```

For example:

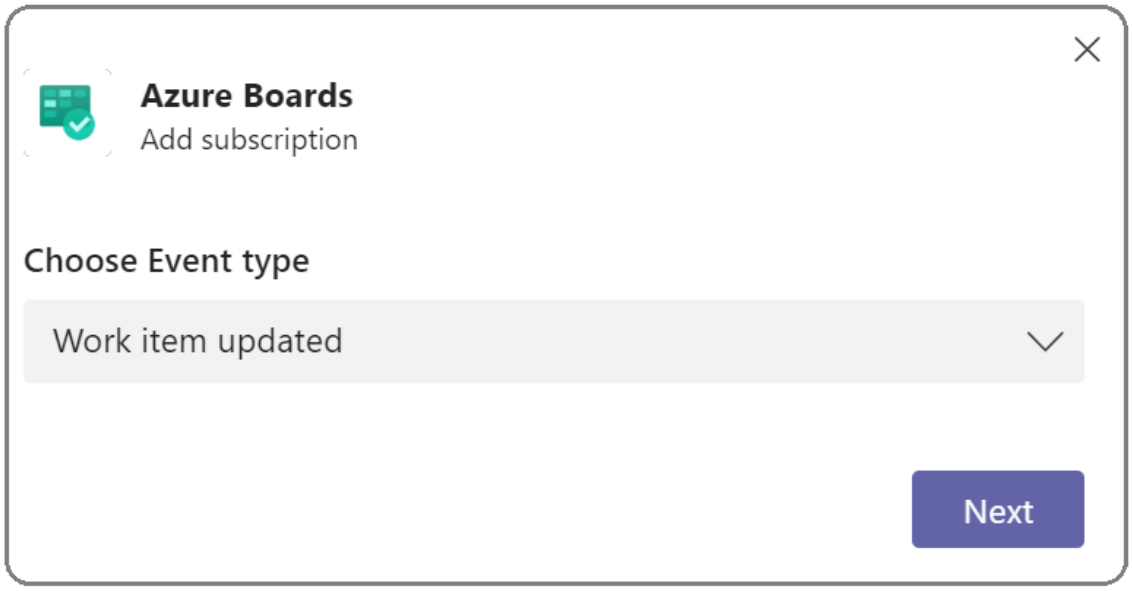
```
@azure boards link https://dev.azure.com/myorg/myproject
```

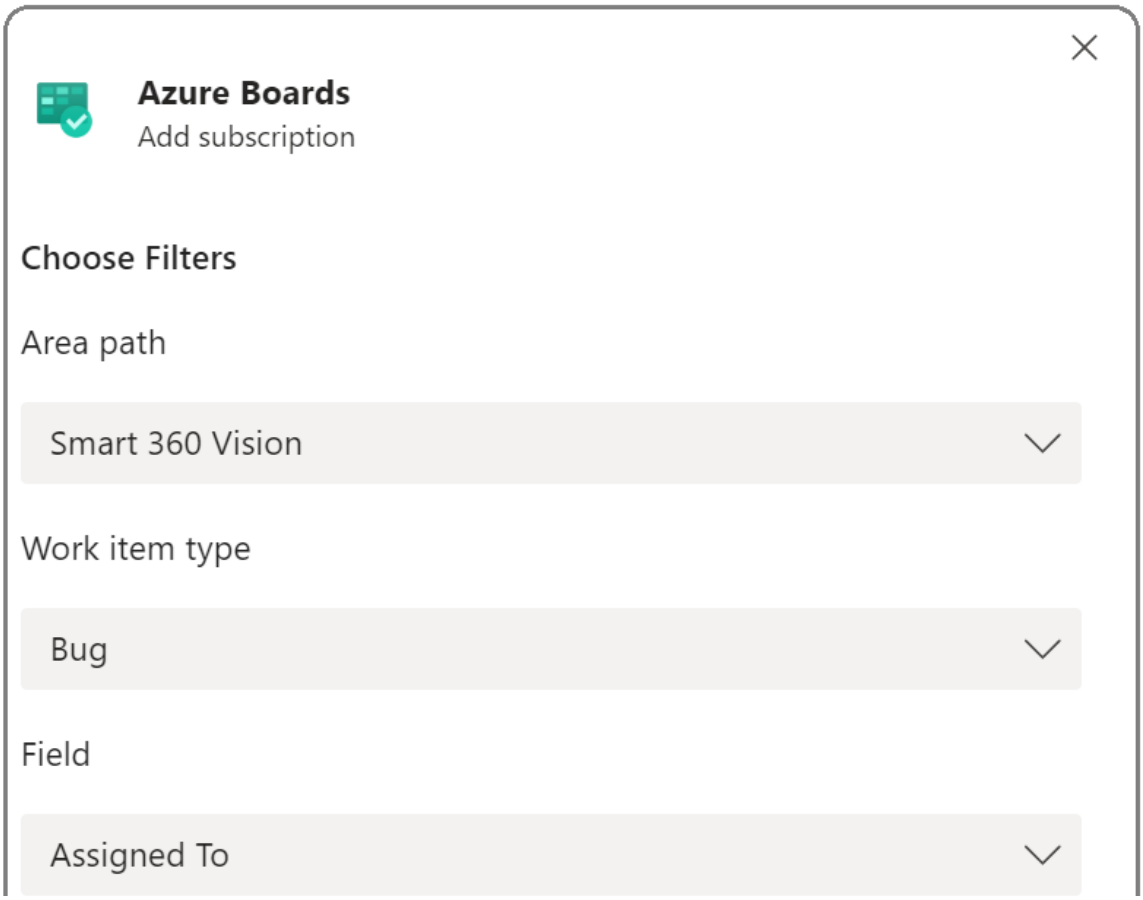
Once the project is linked, you can create work items using `@azure boards create` command or use message actions.

Set up subscriptions

You can create subscriptions to monitor work items at any time using the `@azure boards subscriptions` command.

1. Select the area path you want and event that you're interested in. Use the associated filters to customize what you get notified on in your Teams channel. To help easily set up subscriptions, your recently accessed area paths are shown in the area path dropdown.







In case the area path you want doesn't appear in the Area path dropdown menu, follow the instructions mentioned in the next section, Add area paths. Area paths added using the `@azure boards addAreapath` command and area paths for which subscriptions are created in the channel always appear in the Area path dropdown along with recently accessed area paths.

Add area paths

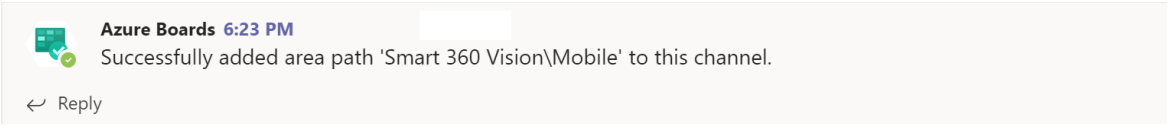
You can add areas that your team works on to the channel so that they're always available for creating work items and subscriptions. This feature is useful for teams with more than 100 area paths in their project.

- Use the following command to add area paths from your project to the Teams channel.

  ```
  @azure boards addAreapath [area path]
  ```

  For example:

  ```
  @azure boards addAreapath myproject\fabrikam
  ```

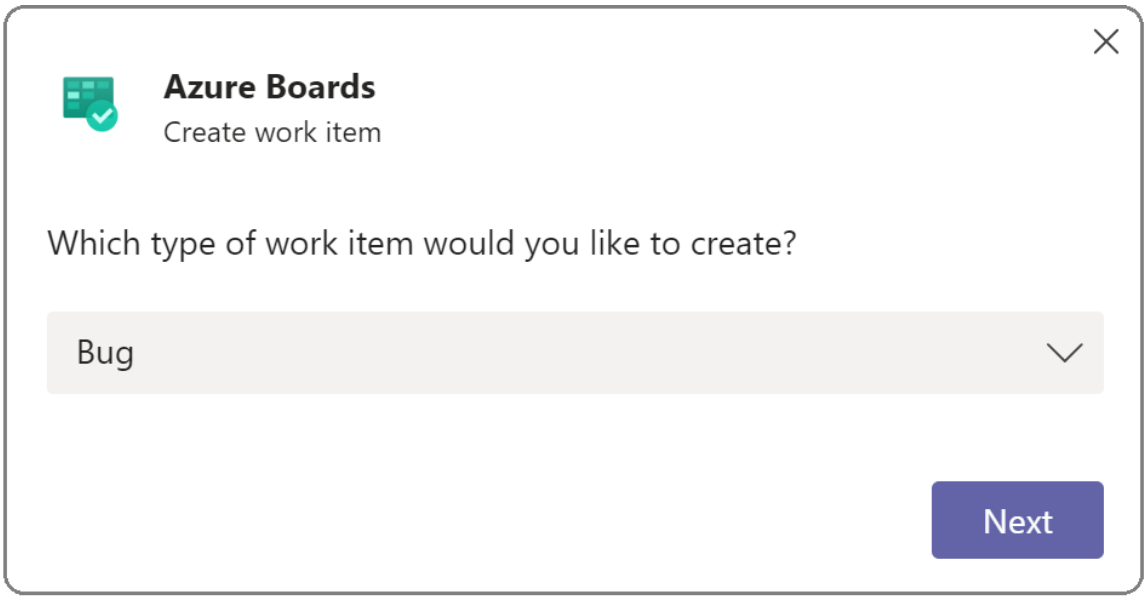
  

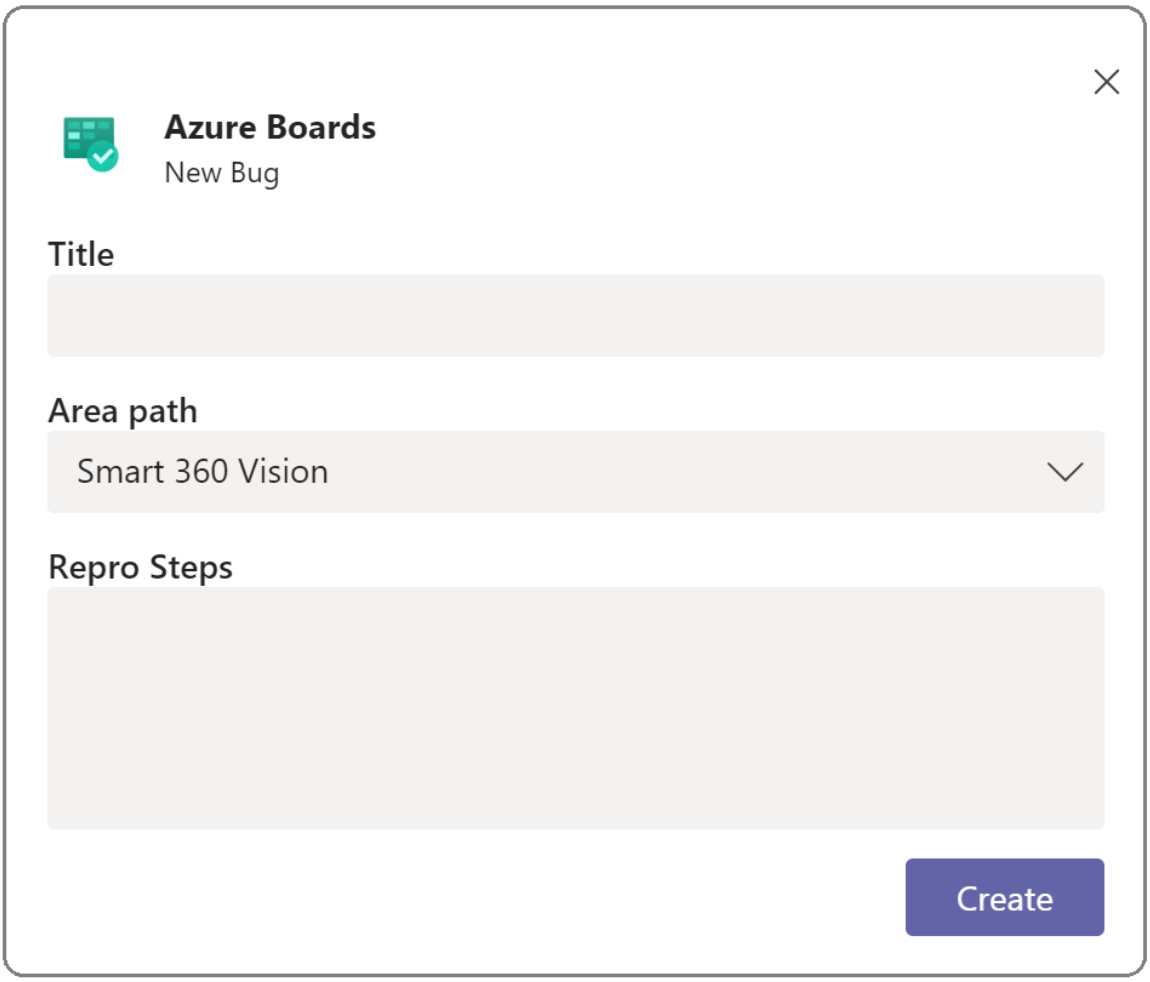
- If you choose project name as your area path, then you'll receive notifications for all the area paths in the project.

Create a work item with a command

With the Azure Boards app, you can create work items from your channel. The app supports custom work items as well.

- To create a work item, use `@azure boards create`.

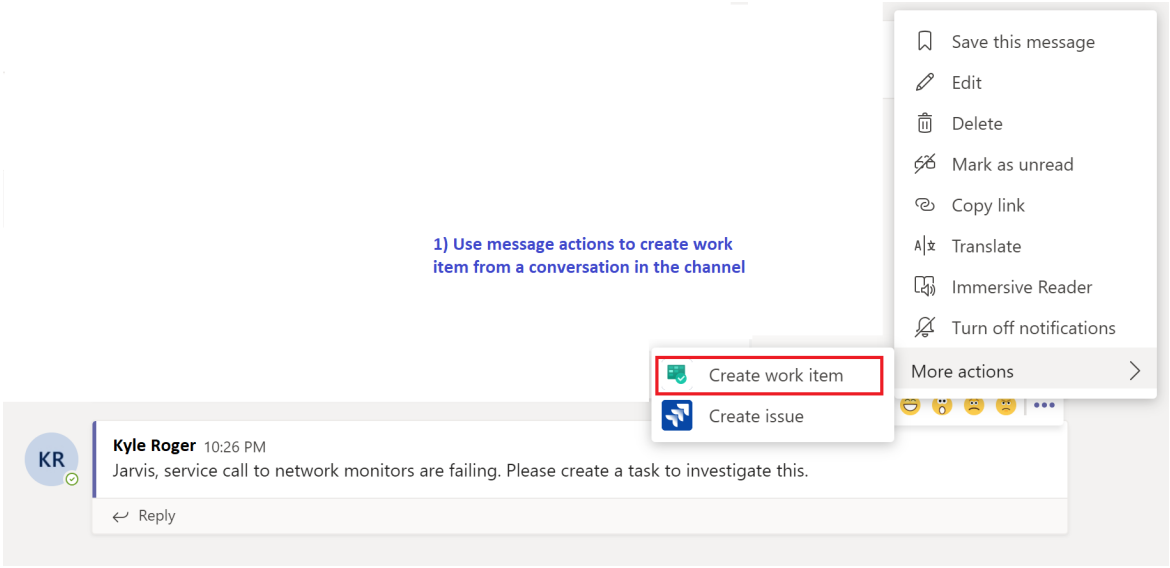


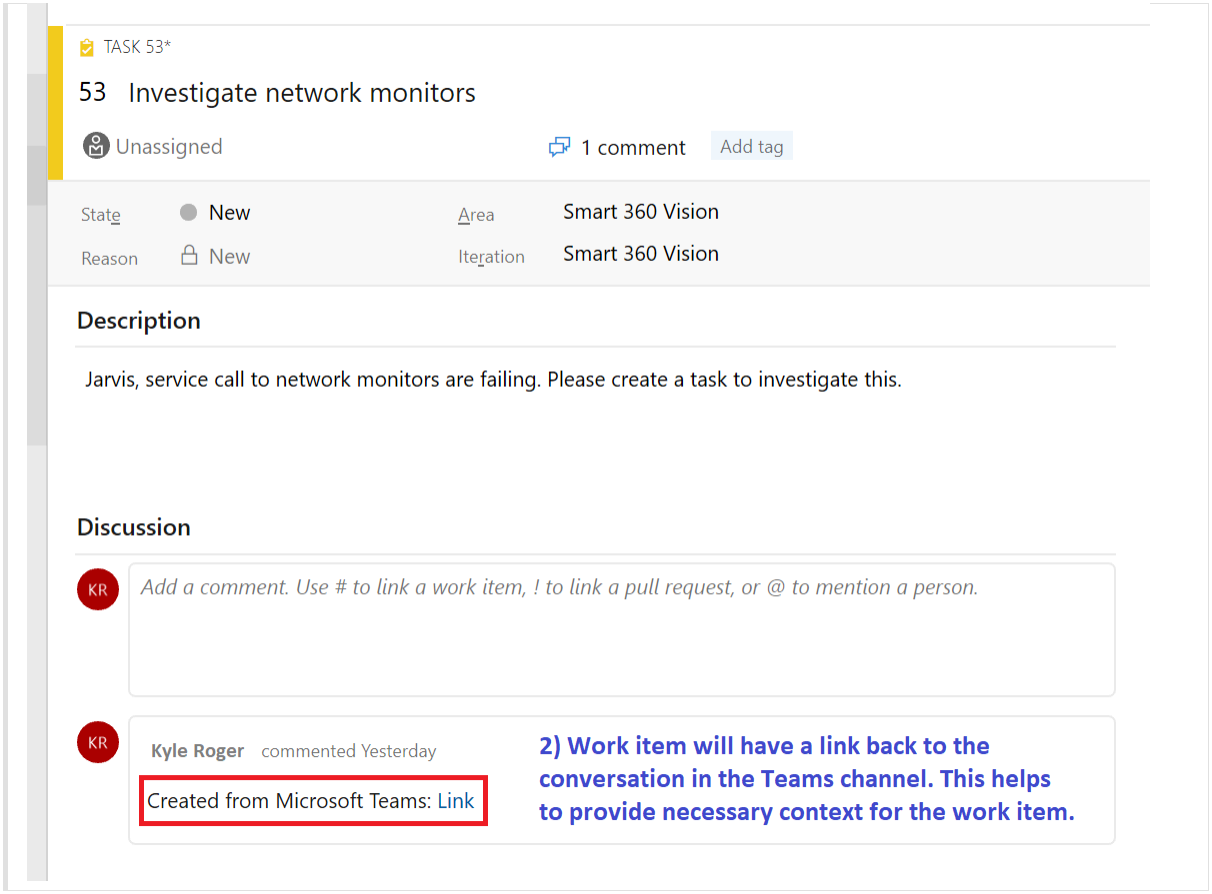


Create a work item from message actions

Often, discussions in a channel require creation of work items. You can use message actions to create a work item. The selected message is pre-filled in the description section of the work item. The Discussion section of the newly added work item stores a link back to the conversation in the channel.

- To create work items using message actions



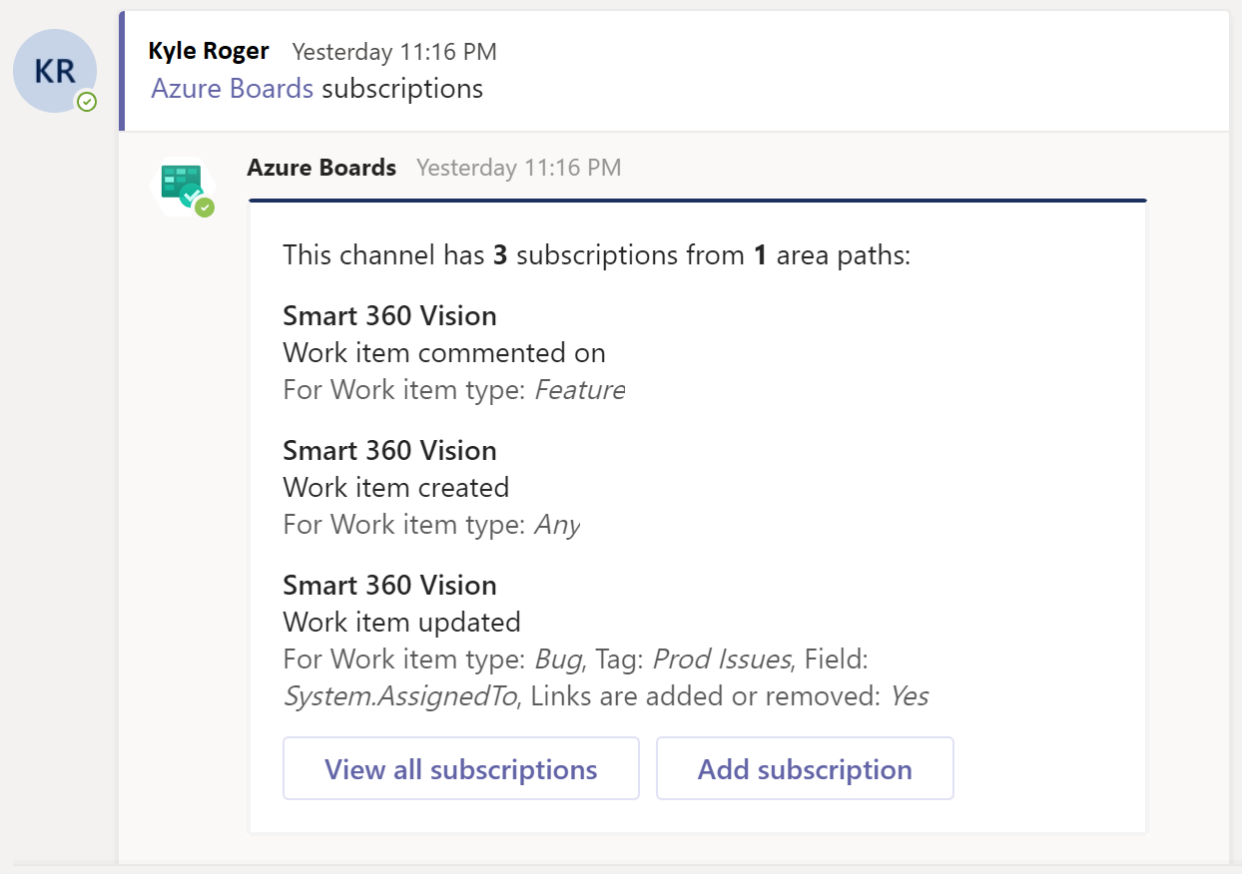


Manage Azure Boards subscriptions

1. To view, add and remove subscriptions for a channel, use the `@azure boards subscriptions` command:

```
@azure boards subscriptions
```

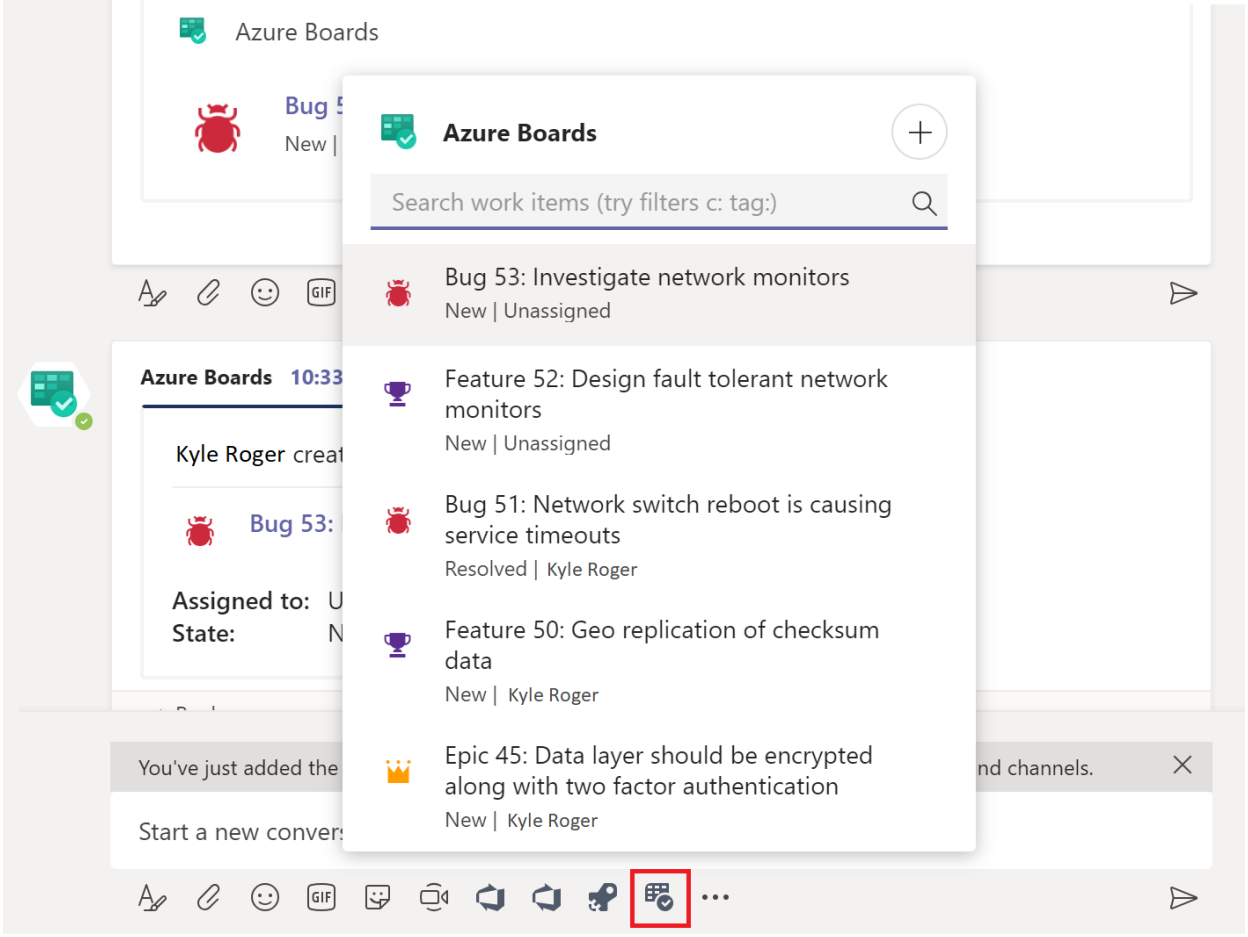
This command lists all the current subscriptions for the channel and allows you to add new subscriptions and remove existing ones. As part of adding subscriptions, you can also customize what you get notified on by using various filters.



> **NOTE**
> Team administrators aren't able to remove or modify subscriptions created by Project administrators.

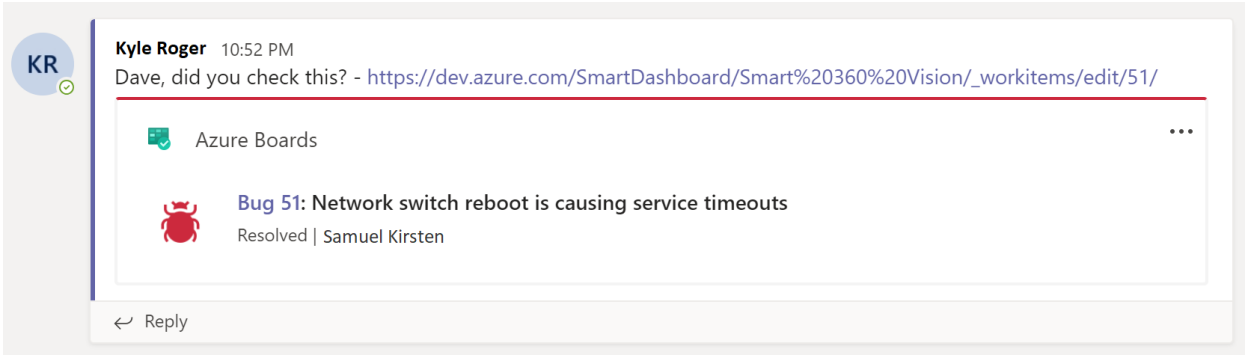
Search and share work items using compose extension

To help users search and share work items, the Azure Boards app for Microsoft Teams supports compose extension. You can search for work items by work item ID, title, or supported functional command. For a list of commands, see Functional work item search. To use the compose extension, users must sign in to Azure Boards app either by running `@azure boards signin` command or by signing into the compose extension directly.



Preview work item URLs

To support collaboration around work items discussed within a channel, the channel displays a preview of work items referenced. When a user pastes the work item URL, a preview is shown similar to the following image. This preview helps to keep work item-related conversations relevant and correct.



For this feature to work, users must be signed in. Once signed in, this feature works for all channels in a team in Microsoft Teams.

Unlink a project from a channel

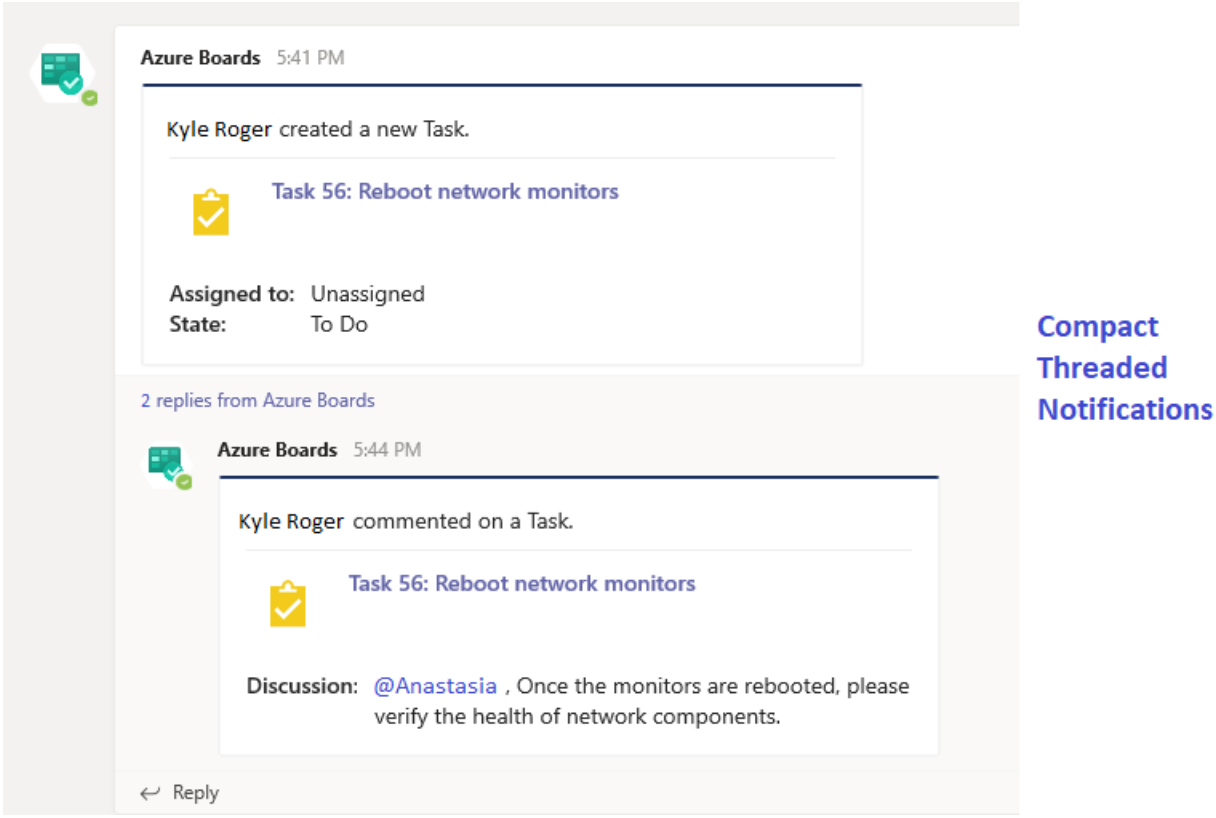
A Teams channel can only link to one Azure Boards project at a time. To link to a different project, you must first unlink the current project using `@azure boards unlink` command.

Unlinking a project deletes all the subscriptions along with added area paths from the channel. If the channel has no subscriptions, any user can unlink a project. However if a channel has subscriptions, only project admins can unlink a project from a channel.

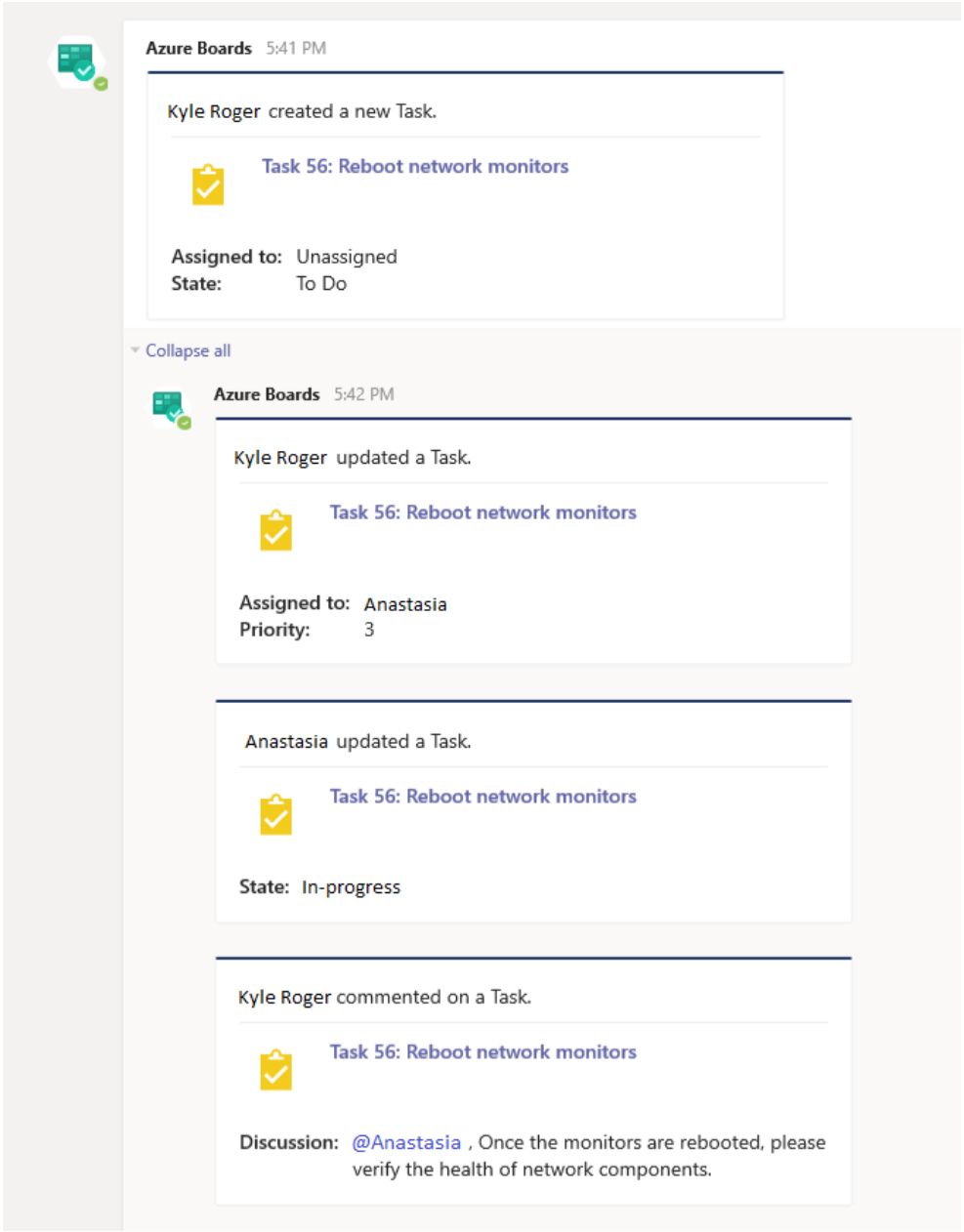
Threaded notifications to link and reduce notifications

The Teams channel threads notifications so as to logically link and reduce related notifications in the channel. All notifications linked to a particular work item are linked together.

Compact view of threaded notifications



Expanded view of threaded notifications



Expanded view of Threaded Notifications

Azure Boards command reference

The following table lists all the `@azure boards` commands you can use in your Microsoft Teams channel.

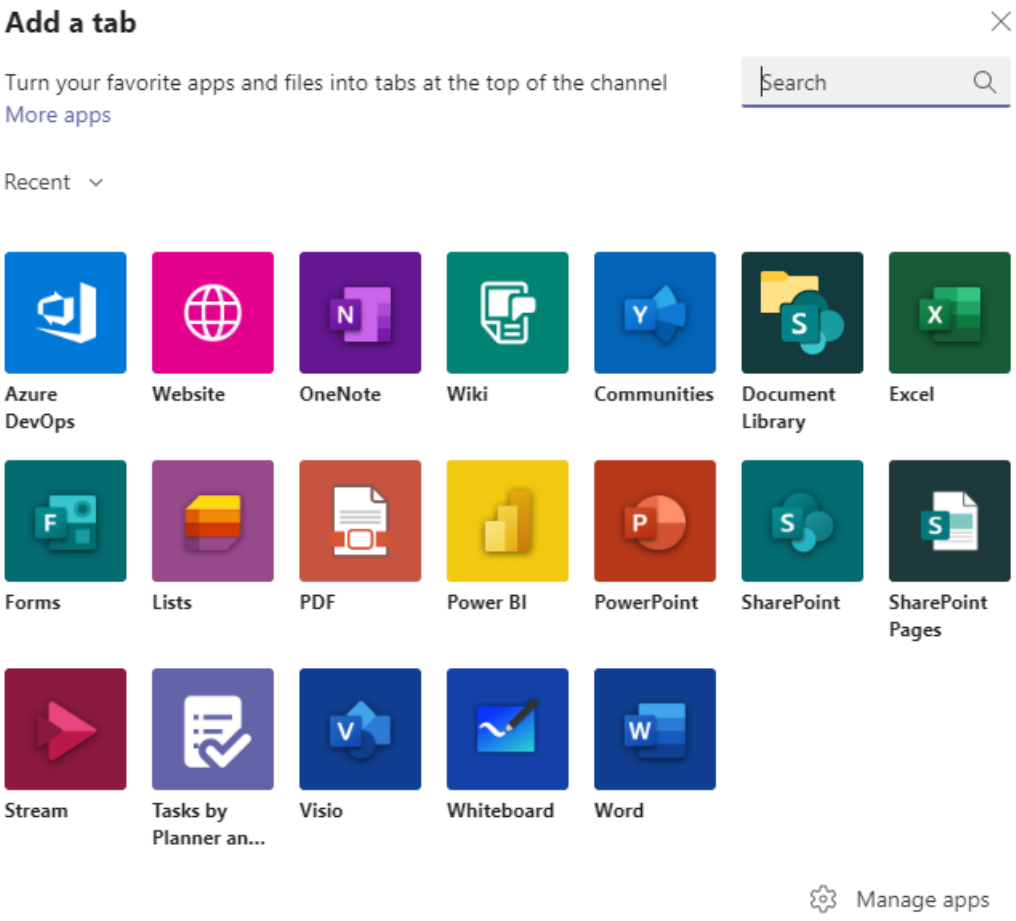
COMMAND	FUNCTIONALITY
@azure boards link [project url]	Link a project to this channel to create work items and receive notifications
@azure boards subscriptions	Add or remove subscriptions for this channel
@azure boards create	Create a work item
@azure boards addAreapath [area path]	Add an area path from your project to this channel
@azure boards sign in	Sign in to your Azure Boards organization
@azure boards sign out	Sign out from your Azure Boards organization

COMMAND	FUNCTIONALITY
@azure boards unlink	Unlink a project from this channel
@azure boards feedback	Report a problem or suggest a feature

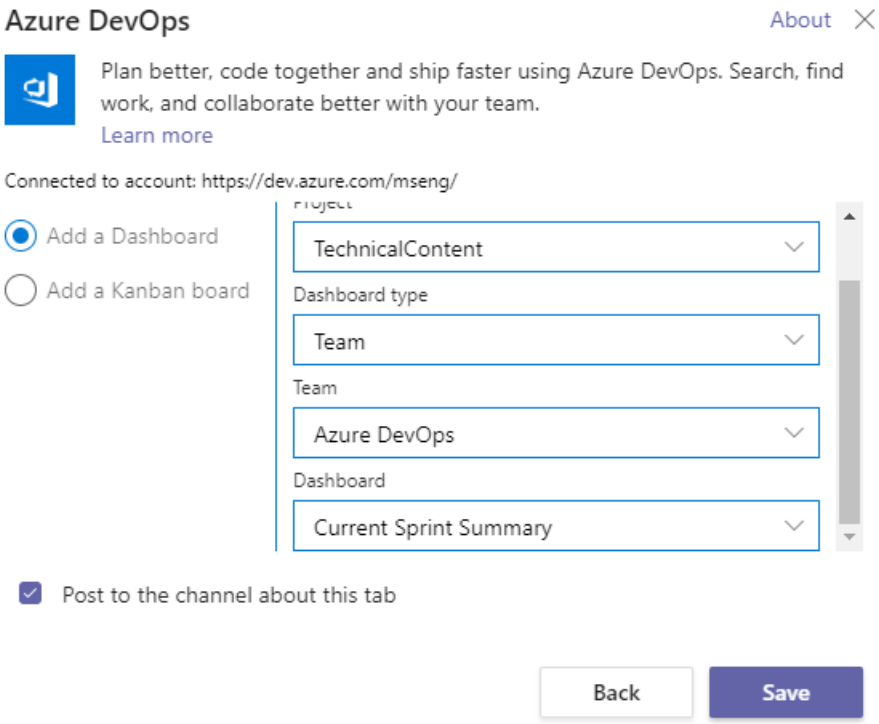
Configure Azure DevOps Services tabs in Microsoft Teams

1. To bring your Kanban board or Dashboard into Microsoft Teams, select the '+' ('add new tab') button on the top nav of your team channel.

   The **Add a tab** dialog displays. Icons are arranged typically according to most recent access. You can select A-Z for an alphabetized list.



2. Choose the **Azure DevOps** icon and authenticate your identity. Optionally, you can choose the **Website** icon and add the URL of your Kanban board or dashboard to the channel.
3. Choose the organization whose board or dashboard you want to add.
4. Fill out the form presented. For example, here we add a dashboard for the Azure DevOps team for the TechnicalContent project.



5. The Kanban board or dashboard you selected displays.

Multi-tenant support

In your organization if you're using a different email or tenant for Microsoft Teams and Azure DevOps, complete the following steps to sign in and connect based on your use case.

Case

Email ID and tenant in Microsoft Teams

Email ID and tenant in Azure DevOps

Steps to take

1

email1@abc.com
(tenant 1)

email1@abc.com
(tenant 1)

Sign in using **Sign in** button.

2

email1@abc.com
(tenant 1)

email1@abc.com
(tenant 2)

- Sign in the Azure DevOps account
- In the same browser, start a new tab, navigate to https://teams.microsoft.com
- Run the `signin` command and choose the **Sign in** button.

3

email1@abc.com
(tenant 1)

email2@pqr.com
(tenant 2)

Sign in using **Sign in with different email address**, in the email ID picker use the email2 to sign in to Azure DevOps.

4

email1@abc.com
(tenant 1)

email2@pqr.com
(non default tenant 3)

This scenario isn't supported today

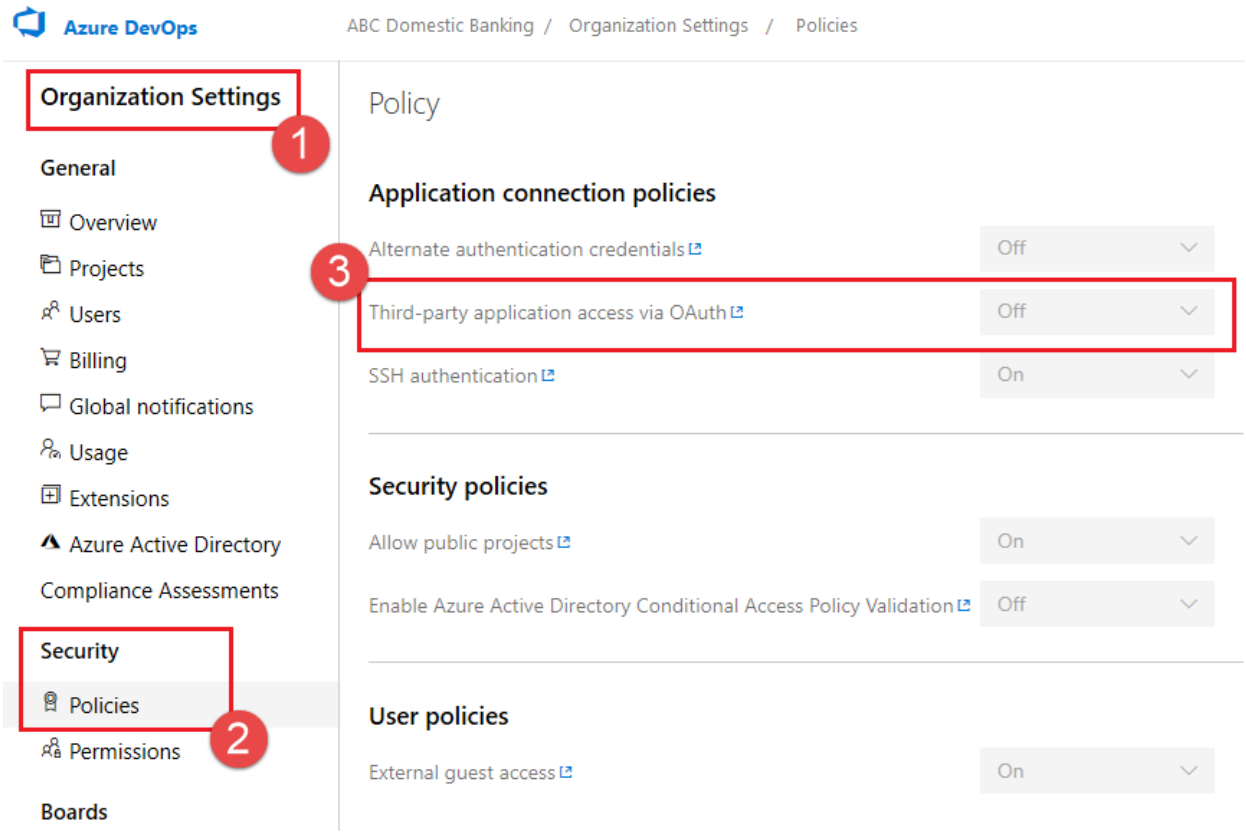
Troubleshoot errors

If you're experiencing the following errors when using the Azure Boards App for Microsoft Teams, follow the procedures in this section.

- Sorry, something went wrong. Please try again.
- Configuration failed. Please make sure that the organization '{organization name}' exists and that you have sufficient permissions.

Sorry, something went wrong. Please try again.

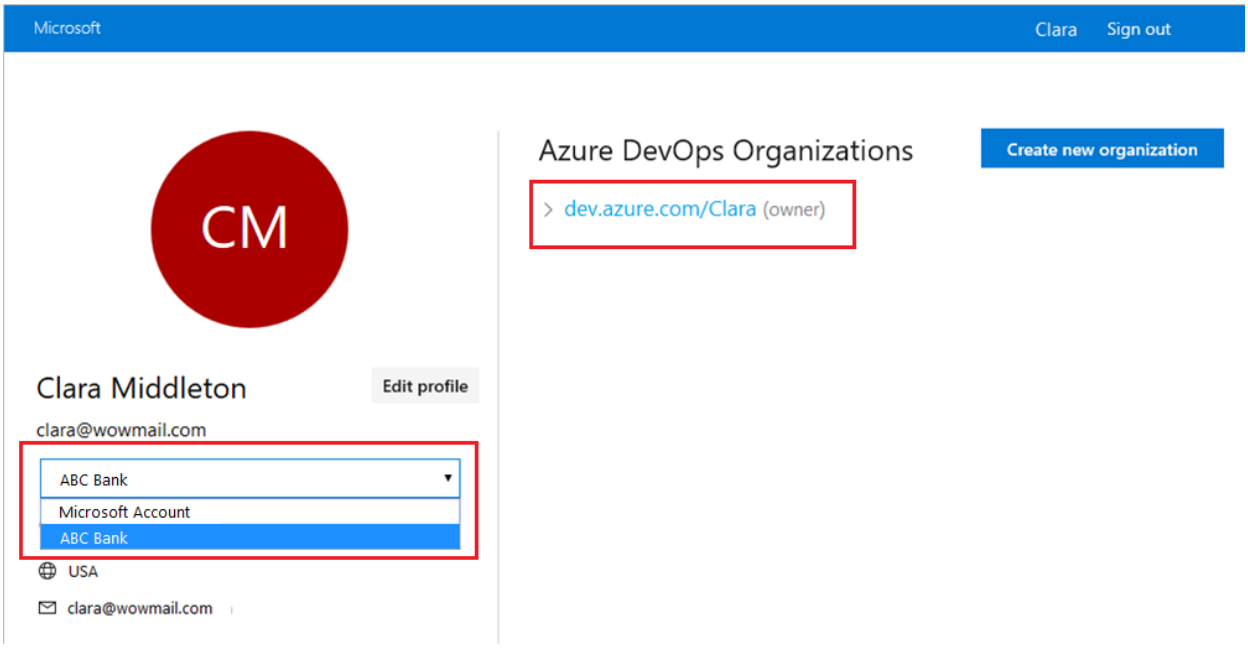
The Azure Boards app uses the OAuth authentication protocol, and requires Third-party application access via OAuth for the organization to be enabled. To enable this setting, navigate to **Organization Settings** > **Security** > **Policies**, and set the **Third-party application access via OAuth for the organization** setting to **On**.



Configuration failed. Please make sure that the organization '{organization name}' exists and that you have sufficient permissions.

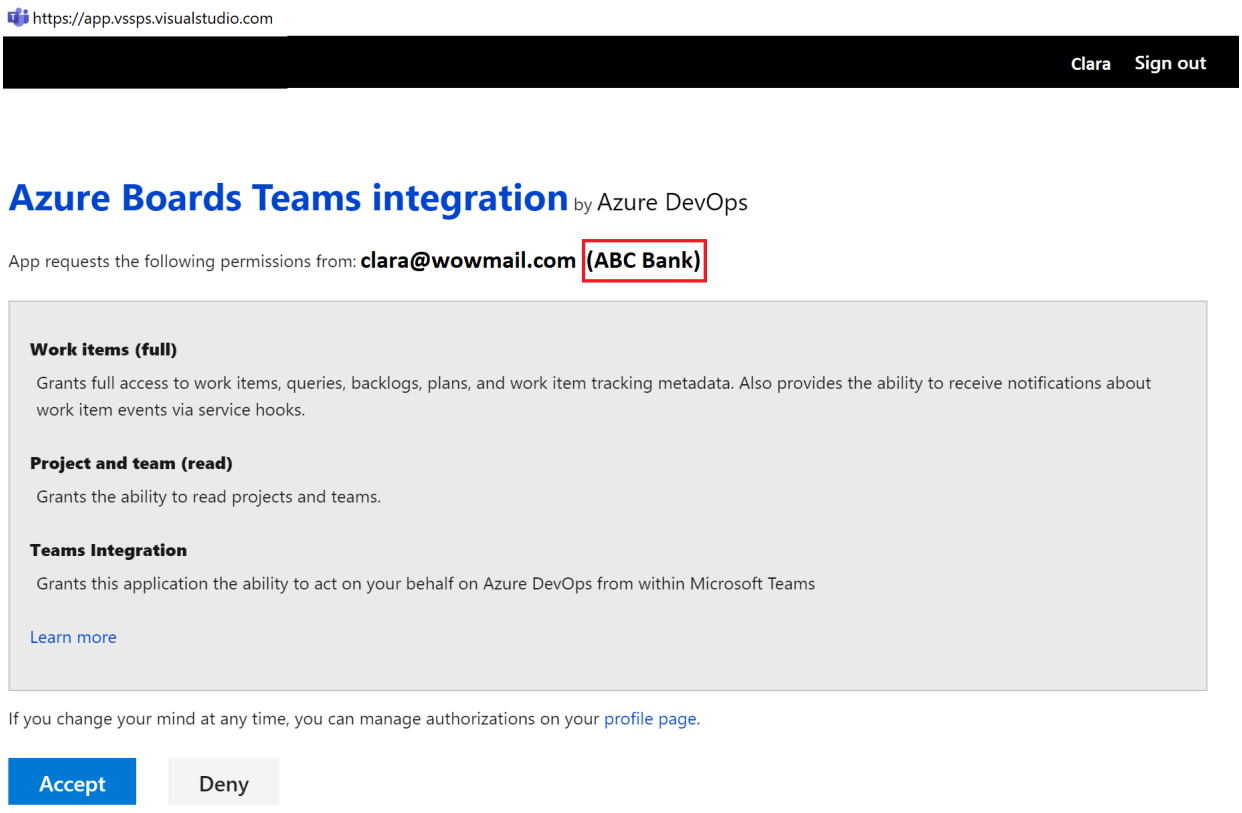
Sign out of Azure DevOps by navigating to `https://aka.ms/VsSignout` using your browser.

Open an **In private** or **incognito** browser window and navigate to `https://aex.dev.azure.com/me` and sign in. In the dropdown under the profile icon to the left, select the directory that contains the organization containing the project that you want to link.



In the **same browser**, start a new tab, navigate to `https://teams.microsoft.com/`. Run the `@azure boards signout` command and then run the `@azure boards signin` command in the channel where the Azure Boards app for Microsoft Teams is installed.

Select the `Sign in` button and you'll be redirected to a consent page like the one in the following example. Ensure that the directory shown beside the email is same as what was chosen in the previous step. Accept and complete the sign-in process.



If these steps don't resolve your authentication issue, reach out to us at Developer Community.

Related articles

- Define area paths and assign to a team
- Azure Pipelines with Microsoft Teams
- Azure Repos with Microsoft Teams

Set up your project's backlogs and boards in Azure Boards

10/4/2022 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

In most cases, you can start using your product and portfolio backlogs once your project is created. A default team is created along with associated backlogs and boards. You can start adding work items to your product backlog using the Backlog or Board.

However, you may need to ensure you've configured your backlogs and boards correctly. Ensure the configuration if you've added a team and want to start using the team backlogs and boards. Changes may be made to a project or team configuration over time. These changes can influence the work items that appear on your backlog and boards.

For an overview of the tools associated with your team, see Manage and configure team tools.

Default backlog and board work items

The first thing you need to understand is that your product **Backlog** and **Board** display work items that meet the following criteria:

- Work item type belongs to the Requirements category. The types differ depending on the process selected for your project:
  - Basic : Issue, Backlog name=**Issues**
  - Agile: User Story, Backlog name=**Stories**
  - Scrum: Product Backlog Item, Backlog name=**Backlog items**
  - CMMI: Requirement, Backlog name=**Requirements**
- Work item **Area Path** matches one of the selected team's Area Paths
- Work item **Iteration Path** is under the team's Default Iteration Path

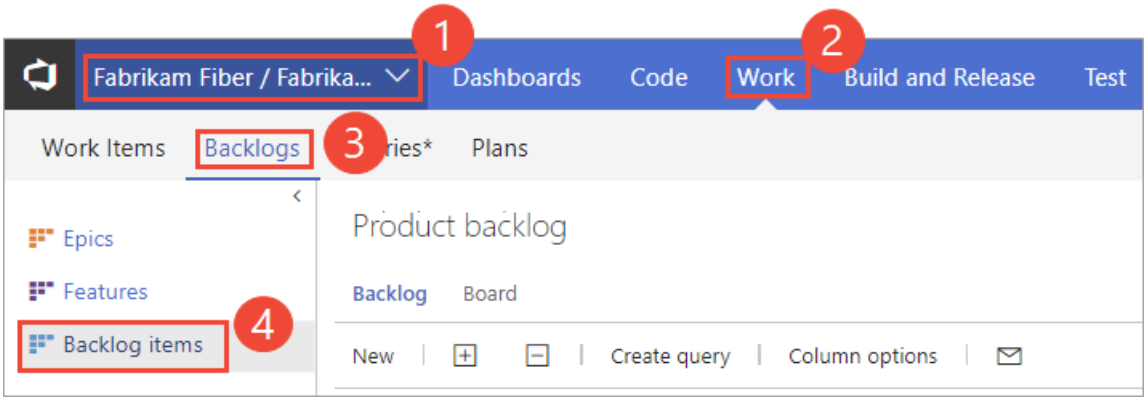
> **NOTE**
> The Basic process is available when you add a project to Azure DevOps Services or Azure DevOps Server 2019 Update 1. For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

- Work item type belongs to the Requirements category. The types differ depending on the process selected for your project:
  - Agile: User Story, Backlog name=**Stories**
  - Scrum: Product Backlog Item, Backlog name=**Backlog items**
  - CMMI: Requirement, Backlog name=**Requirements**
- Work item **Area Path** matches one of the selected team's Area Paths
- Work item **Iteration Path** is under the team's Default Iteration Path

You can determine the work item types that belong to your Requirements category. Determine the items by opening your product Backlog and checking the product backlog name.



As shown in the following image, (1) choose the team, (2) **Work**, (3) **Backlogs**, and then the product backlog.



Look up your team's Area Path(s) and Iteration Paths. For more information, see Define area paths and assign to a team and Define sprint paths and configure team iterations.

Default sprint backlog and Taskboard work items

Your sprint backlog and Taskboard apply the filters associated with your team's default backlog and board work items along with the Iteration Path you select.

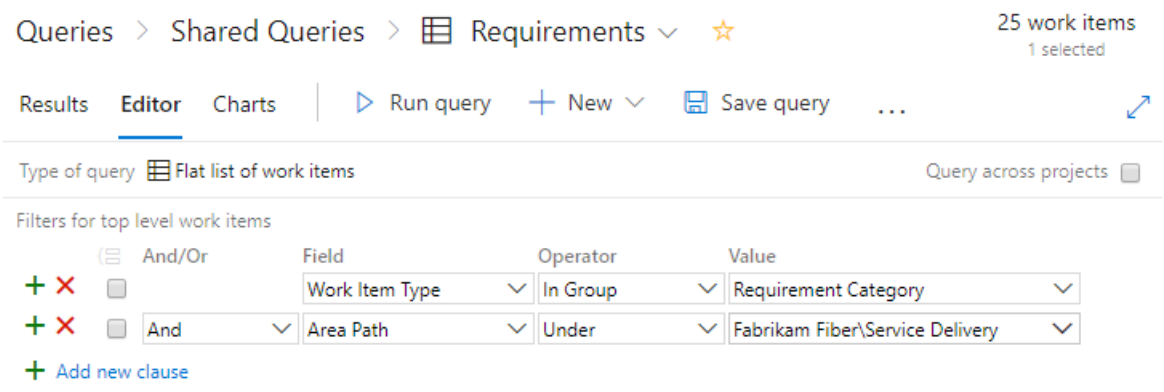
You can only select Iteration Paths that have been preselected by your team.

Your sprint backlog displays only those work items assigned to the selected sprint. Child tasks assigned to other sprints aren't displayed.

Review checklist for work items, backlogs, and boards

If you don't see the work items you expect on your product Backlog or Kanban board, complete the following checks:

1. Make sure you've selected the team backlog or board of interest. To learn how, see Use breadcrumbs and selectors to navigate and open artifacts.
2. Create a query of your backlog items, specifying the work item types that belong to your Requirements category and the Area Path associated with your team, for example:



3. Add the **State**, **Area Path**, and **Iteration Path** fields to the column options.

4. Check the query results and that the values of the work items you expect to show up on your backlog meet these criteria:

   - **Area Path** belongs to your team's area path(s)
   - **Iteration Path** belongs under your team's default iteration path
   - **State** isn't Closed, Completed, Done, or Removed.

> **NOTE**
> You can also filter your product backlog to show or hide work items that are in an **In Progress** state category, corresponding to an Active, Resolved, Committed, Doing workflow state.

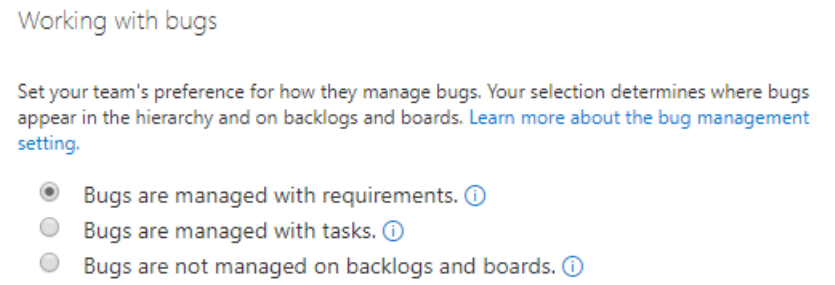
Add bugs to your backlogs and boards

For all processes except the Basic process, each team manages the way bugs are tracked. Track bugs in the Requirements category because they show up on the Backlog and Kanban board or the Tasks category. They can also show up on the Taskboard or the Bugs category where they don't appear on either backlogs or boards.

> **NOTE**
> Bug work item types aren't available with the Basic process. The Basic process tracks bugs as Issues and is available when you create a new project from Azure DevOps Services or Azure DevOps Server 2019.1 or later versions.

Each team can manage the way bugs are tracked. You can track bugs as belonging to the Requirements category. Those bugs show up on the Backlog and Kanban board or the Tasks category. They can show up on the Taskboard or the Bugs category where they don't appear on either backlogs or boards.

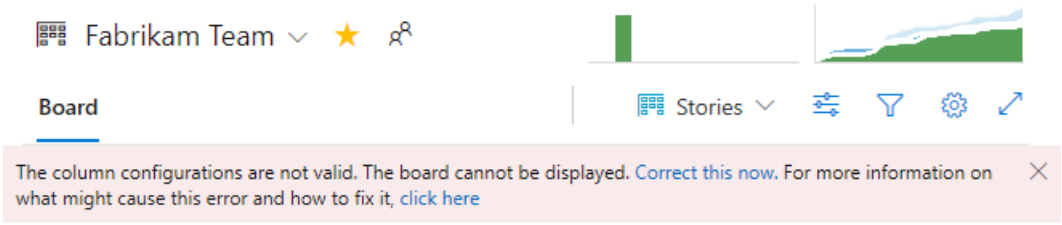
If you want bugs to show up on your Backlog and Board, choose **Bugs are managed with requirements**.



For details, see Show bugs on backlogs and boards.

Correct your Kanban board configuration

If you see the following error when you open your Kanban board, you need to correct the configuration. The main reason for this error is that the workflow states of work item types that have been added to the Requirements category aren't mapped to the column.

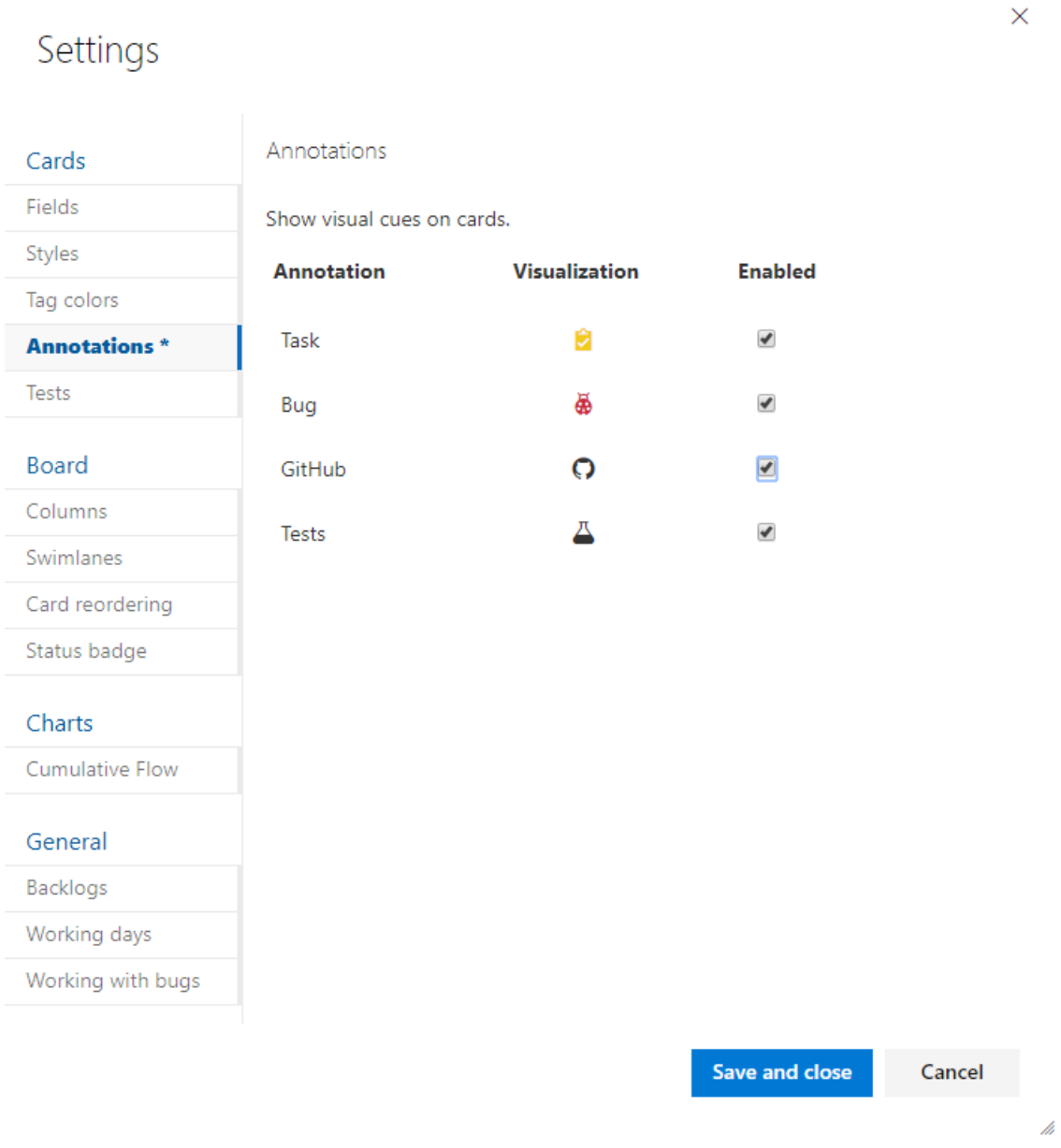


Choose **Correct this now** to open the Settings dialog. To map the workflow states, refer to Add columns to your Kanban board, Update Kanban column-to-State mappings.

Customize your Kanban board checklist items

Checklists are a great way to create work items that are automatically linked with a parent-child link to another work item on a Kanban board. You can customize the work item types that you can add as a checklist by opening the Board Settings, choose **Annotations**, and enable the work item types you want to appear on the board. For details, see Customize cards.

For example, here we've chosen to track bugs along with tasks, and enable Task, Bug, GitHub objects, and Tests to appear within checklists.



NOTE

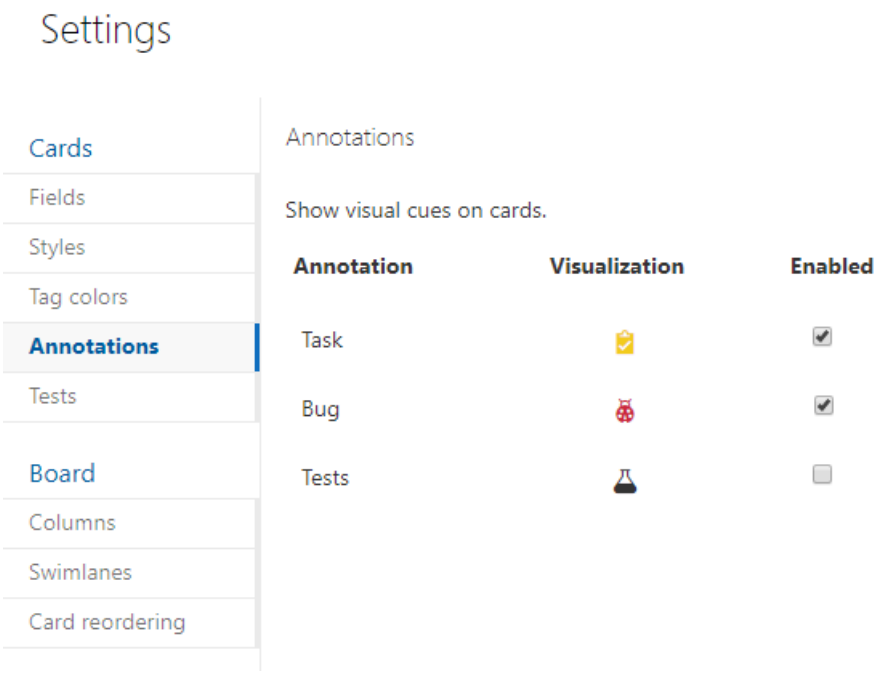
The GitHub annotations require Azure DevOps Server 2019 update 1 or later version.

To learn more about checklists, see the following articles:

- Add tasks or child items as checklists
- Add, run, and update inline tests

- Link GitHub commits, pull requests, and issues to work items

For example, here we've chosen to track bugs along with tasks, and enable Task and Bug to appear within checklists.



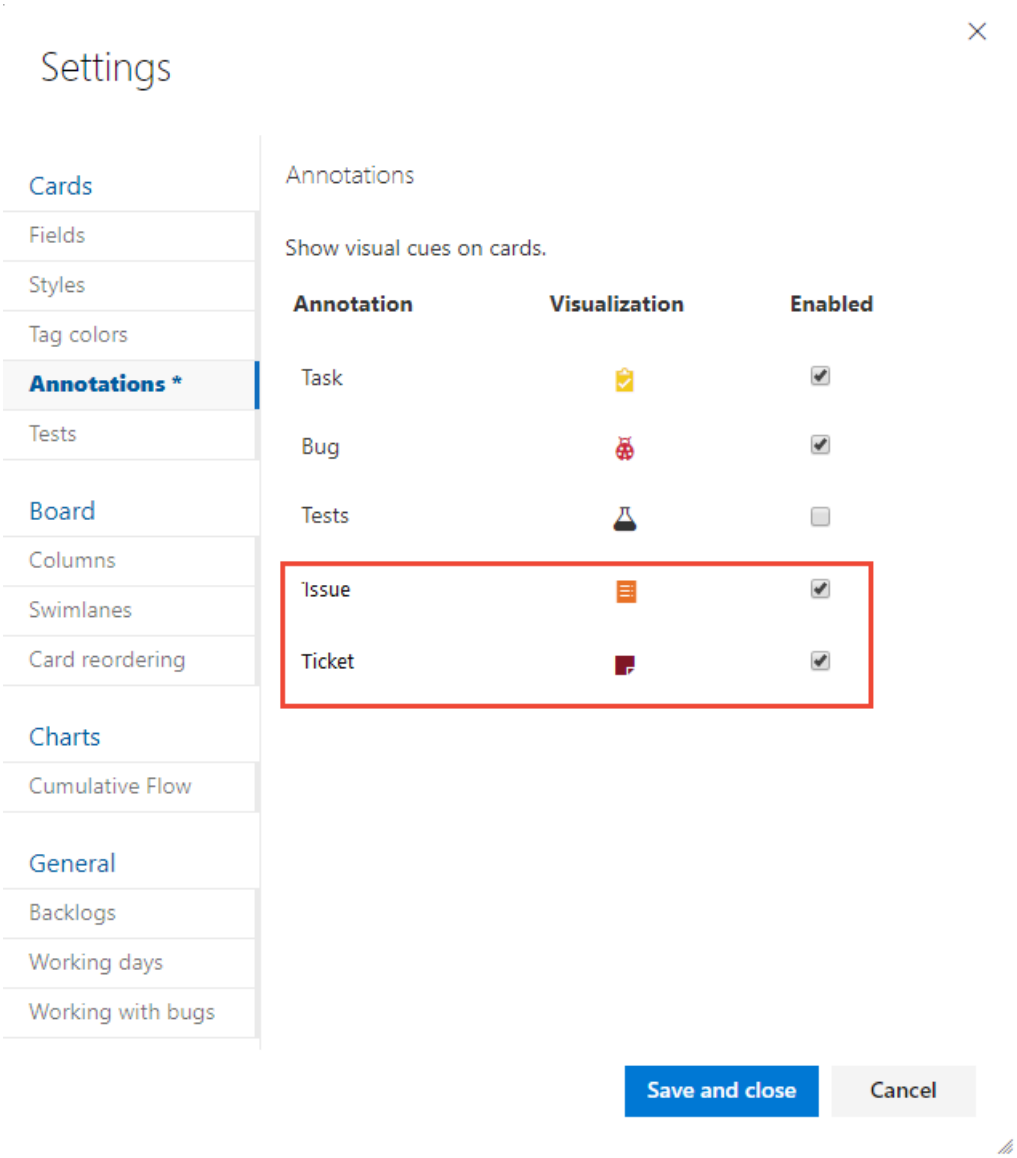
To learn more about checklists, see the following articles:

- Add tasks or child items as checklists
- Add, run, and update inline tests

Add other work item types to your Kanban board checklist

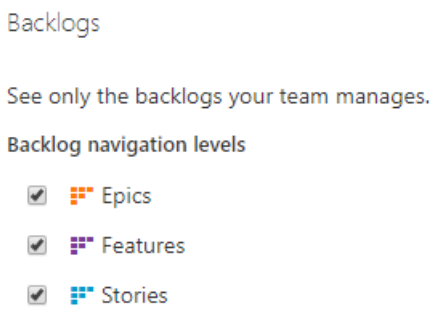
If you added work item types to the Task Category as described in Add custom work item types to your Taskboard later in this article, you can choose if these types appear within a checklist on your product Kanban board. You make this choice by opening **Board Settings**, choose **Annotations**, and enable the work item types you want to appear on the board. You can enable up to five annotations. For details see Customize cards.

For example, here we've chosen to track bugs along with tasks, and we enable Issue and Ticket and Task and Bug. To learn more about checklists, see Add tasks or child items as checklists and Add, run, and update inline tests.



Hide or show backlog levels

Your team can also choose to hide or show one or more backlog levels. Feature teams often manage backlog items, while management teams manage features and epics. In this situation, you can enable or disable a backlog level.



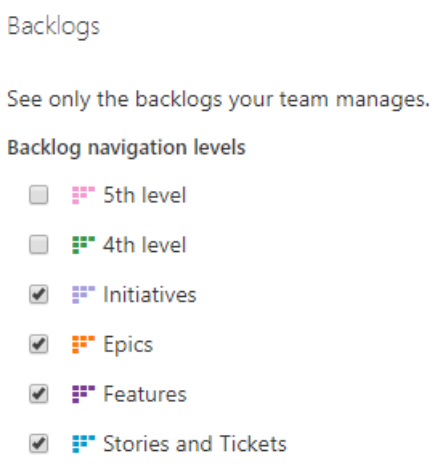
For details, see Select backlog navigation levels for your team.

Add custom work item types to your backlogs and portfolio backlog levels

If you want to track different work item types on your product backlog, you can do that by adding custom work item types and adding them to a specific backlog level.

You can also add custom work item types and add them to portfolio backlogs. You can add up to five portfolio backlogs.

For example, here we've added Initiatives, fourth level, and fifth level work item types to support five levels of portfolio backlogs. We've also added a custom work item type named Ticket and added that to the product backlog.



For details, see the following resources:

- Add and manage work item types (Inherited process)
- Customize your backlogs or boards (Inherited process)
- Customize an inheritance process

Inheritance process model:

- Add and manage work item types (Inherited process)
- Customize your backlogs or boards (Inherited process)
- Customize an inheritance process

On-premises XML process model:

- Add a work item type to a backlog and board
- Add portfolio backlogs
- Customize the On-premises XML process model

- Add a work item type to a backlog and board
- Add portfolio backlogs
- Customize the On-premises XML process model
- Add portfolio backlogs

Add custom work item types to your Taskboard

To add custom work item types to appear on your sprint Taskboard, follow the steps outlined next based on the process model your project uses.

> **NOTE**
> You can enable work item types that you add to your Iteration backlog to appear as a checklist on your product Kanban board. To learn how, see Customize your Kanban board checklist items provided earlier in this article.

Track custom work items with the Inherited process model

For example, if you want to track a custom work item type, Tickets, along with Tasks and Bugs, you do the following tasks:

1. Define the Ticket custom work item type. See Add and manage work item types.
2. Add the Ticket work item types to the Iteration backlog. For details, see Customize your backlogs or boards for a process.

Add existing and custom work item types with the On-premises XML process model

For on-premises deployments that use the On-premises XML process model to customize work tracking, you can add existing and custom work item types to the sprint Taskboards. For example, if you want to track Issues (or Impediments for the Scrum process) and a custom work item type, Tickets, along with Tasks and Bugs, you would do the following tasks:

1. Define the Ticket custom work item type. See Add or modify a work item type.
2. Add Issue and Ticket work item types to the Task Category by modifying the Categories XML file. For details, see Categories XML element reference.

   For example, here we add Issue and Ticket to the Task Category.

   ```
   <CATEGORY name="Task Category" refname="Microsoft.TaskCategory">
     <DEFAULTWORKITEMTYPE name="Task" />
   <WORKITEMTYPE name="Issue" /
   <WORKITEMTYPE name="Ticket" /
   </CATEGORY>
   ```

3. Make sure that the Issue and Ticket workflow states are mapped to category states. As needed, modify the ProcessConfiguration XML file to add Issues and Tickets to the `TaskBacklog` section.

   For example, here the New, Active, and Closed states are mapped for the Task Category.

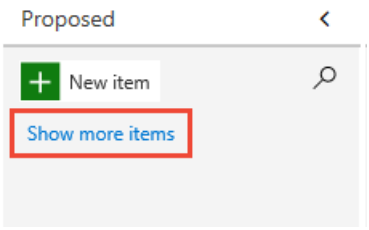
   ```
     <TaskBacklog category="Microsoft.TaskCategory" pluralName="Tasks" singularName="Task"
   workItemCountLimit="1000">
       <States>
         <State value="New" type="Proposed" />
         <State value="Active" type="InProgress" />
         <State value="Closed" type="Complete" />
       </States>
   . . .
     </TaskBacklog>
   ```

4. To verify your changes, open a Sprint backlog and make sure you can add an Issue or Ticket in the same way you add a Task. See Add tasks.

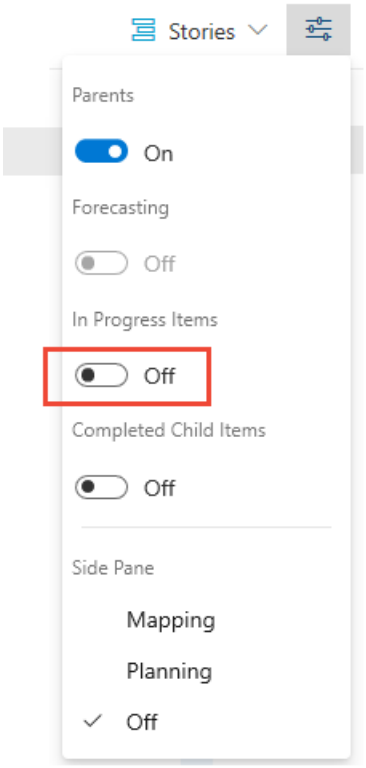
Other factors that can affect work items in your backlogs and boards

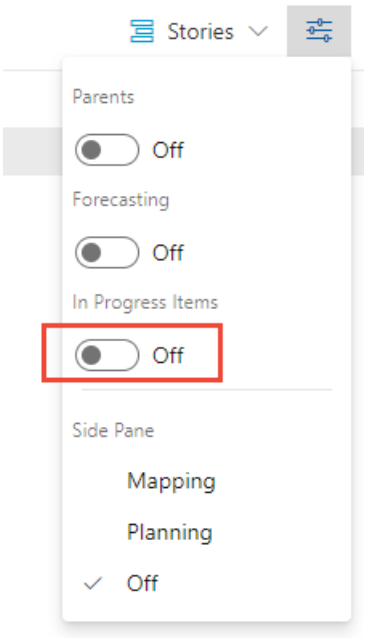
The following settings can influence the type and number of work items that will appear in your backlogs and boards.

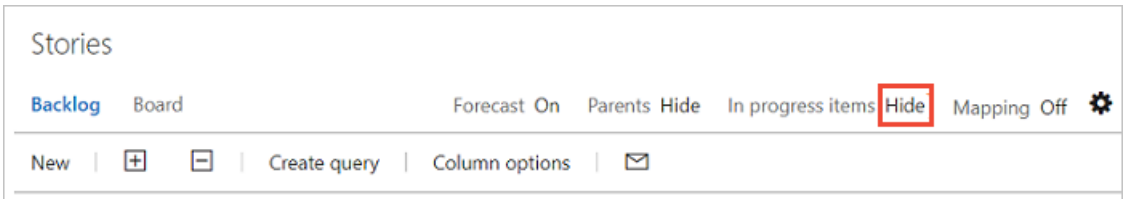
- In your Kanban board, newly added work items may not appear if they're stack ranked lower within the product backlog. By choosing **Show more items**, you can cause the board to refresh and display more work items.



- If you have nested work items that belong to the same category, only leaf nodes may appear on the Kanban board (for TFS 2018.1 and earlier versions). For this reason, we recommend that you don't nest work items of the same work item type or belonging to the same category. To learn more, see Fix reordering and nesting issues, How backlogs and boards display hierarchical (nested) items.
- If you've turned off the **In Progress** view, then those work items where work has started won't appear in the backlog list.







- Work items appear in the priority order in which they're added or moved to. This order or sequence is managed by the **Stack Rank** (Basic, Agile, and CMMI processes) or **Backlog Priority** (Scrum) field. To learn more, see the Stack rank section in Backlogs, portfolios, and Agile project management.
- Each backlog can display up to 999 work items. If your backlog exceeds this limit, then you may want to consider adding a team and moving some of the work items to the other team's backlog.
- Sprint backlogs show only those work items that meet the team's area path and the **Iteration Path** defined for the sprint.
- Inheritance process model: If an administrator disables or deletes a work item type, it will no longer appear on backlogs and boards.
- On-premises XML process model: If an administrator deletes or destroys a work item type, it will no longer appear on backlogs and boards.

Related articles

- Add a team, move from one default team to several teams
- Refine your backlog
- Create your backlog
- Backlog priority or stack rank order
- Use categories to group work item types
- Workflow states & state categories

Fix issues in Azure Boards with displaying, reordering, and nesting work items

10/4/2022 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure Boards backlogs display a natural hierarchy of work items. When you add parent-child links to work items not in the natural hierarchy, you'll receive a message that indicates reordering is disabled. Some items may not display. Also, the system may disable the drag-and-drop reorder feature.

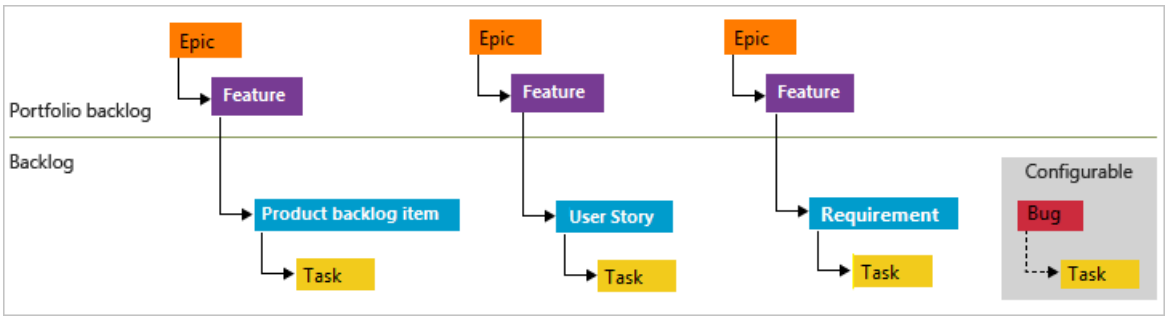
Use this article to fix the issues that occur and that display one of the following messages:

- `You cannot reorder work items and some work items may not be shown.`
- `You cannot reorder work items and some work items may not be shown. See work item(s) 7 to either remove the parent to child link or change the link type to 'Related'." or "Work item 3 can't be reordered because its parent is on the same category".`
- `Items added to the backlog may disappear on a refresh because your team project marks them as "in progress". Those items will appear when you change the "In progress" filter to Show.`

> **NOTE**
> This article addresses issues that arise when you create parent-child links that don't obey the natural hierarchy defined for backlogs. For other issues that may occur with multi-team ownership, see Configure a hierarchy of teams, Exercising select features with shared area paths.

Natural hierarchy for work item types

The following image indicates the natural hierarchy for the Agile, Scrum, and Capability Maturity Model Integration (CMMI) processes. Along with these work item types, other custom work item types may be added to backlogs and boards. Also, custom backlog levels may be added.



You break this natural hierarchy when you create same-category links between work items.

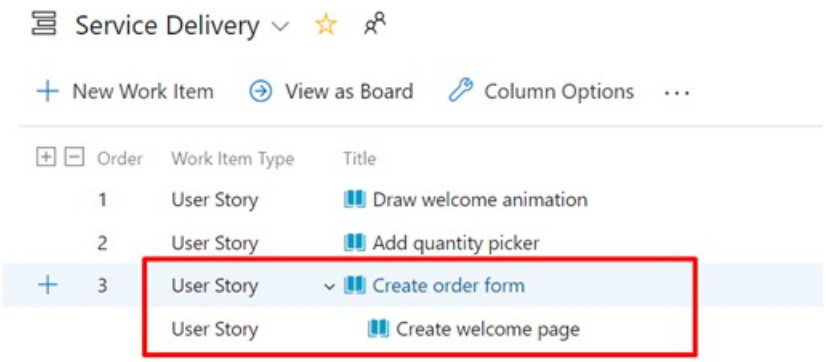
When you link work items of the same type with parent-child links—such as bug-bug or user story-user story—you create same-category links. Also, when you create parent-child links between work items that belong to the same category—such as the Requirements category or Task category—you create same-category links. The category a work item belongs to is determined by your process backlog levels and your team's selected bug behavior. To understand more about same-category hierarchy, see the section Recommended configuration.

Resolve the message that you can't reorder work items

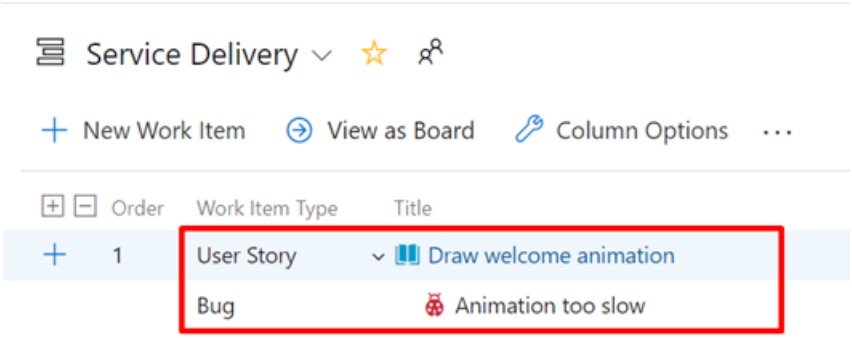
You may see a message such as `You cannot reorder work items and some work items may not be shown`. No work item IDs are listed.

To address this message, take the following actions:

1. Open your Backlog.
2. Review the list of items to determine which items of the same type are nested.
   For example, the following shows that a user story is a child of another user story.



   As another example, the following shows that a bug is a child of a user story. Because the team has configured their backlog to display user stories and bugs at the same level (Requirements category), this configuration results in a nested item that disables the ordering feature.



3. Remove all parent-child links that exist among nested items of the same work item type or same category. Or, change the link to 'Related'.
4. Refresh your Backlog.

The issue is now resolved and the message no longer displays.

Resolve the message that specifies work item IDs

You may see a message similar to

```
You cannot reorder work items and some work items may not be shown. See work item(s) 7 to either remove the parent to child link or change the link type to 'Related'." or "Work item 3 can't be reordered because its parent is on the same category"
```

.

To address this message, take the following action:

1. Open the work item listed in the error message.
2. Look for a parent or child link. Make sure this link goes to a work item within the same category as the work item you opened. This link goes to another work item that appears on the same backlog level as the work item you opened. Depending on your team's bug behavior setting, bugs may appear with requirements or tasks.

3. Remove the problem parent-child link. If you would like to keep these items associated, use 'Related' link type instead.

The issue is resolved and the message no longer displays.

Resolve the issue where work items that are in progress may disappear on a refresh

The message—

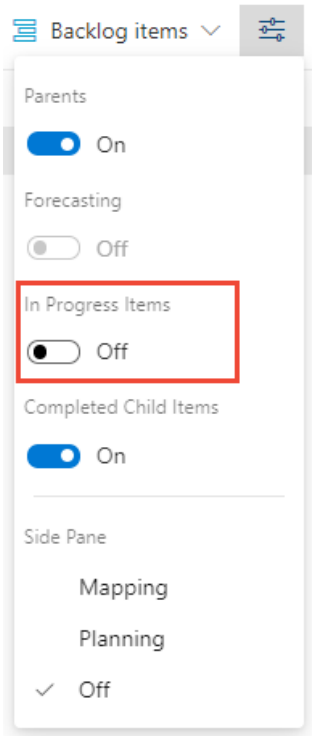
```
Items added to the backlog may disappear on a refresh because your team project marks them as "in progress".
Those items will appear when you change the "In progress" filter to Show.
```

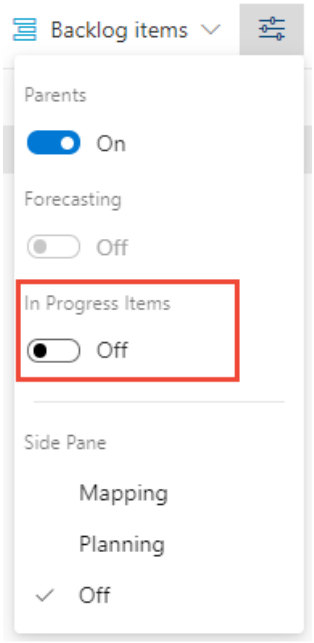
—indicates that the **In Progress** filter for the backlog has been turned off.

Upon refresh of your browser, the backlog displays those work items based on your selected filters.

To reset the filters, complete the following steps.

From the **View options** selector, you can choose to show or hide **In Progress items**. If you turn the **In Progress** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or states that map to the In Progress category state won't appear in the backlog.





Choose **In progress items** show or hide **In Progress** backlog items. If you turn the **In Progress items** control off, then items that are in the *Active*, *Committed*, or *Resolved* states or states that map to the In Progress category state won't appear in the backlog.

You usually choose to hide **In Progress items** when you want to forecast work. To learn more, see Forecast your product backlog.

Recommended configuration

While you can create a hierarchy of backlog items, tasks, and bugs—we recommend that you don't create same-category hierarchies. That is, don't create parent-child links among work items of the same type, such as story-story, bug-bug, task-task, or issue-issue. The reason is that the Backlog, Board, and Sprints experiences don't support reordering for same-category hierarchy. Since ordering is executed by hierarchy level, same-category hierarchy introduces confusion by ordering a work item that doesn't belong on that level.

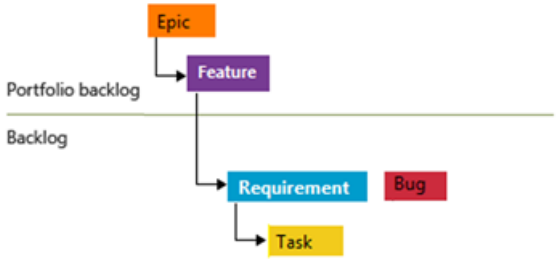
Instead of nesting requirements, bugs, and tasks, we recommend that you maintain a flat list. In other words, only create parent-child links one level deep between items that belong to a different category.

Use the Feature work item type when you want to group user stories (Agile), issues (Basic), product backlog items (Scrum), or requirements (CMMI). You can quickly map product backlog items to features, which creates parent-child links in the background.

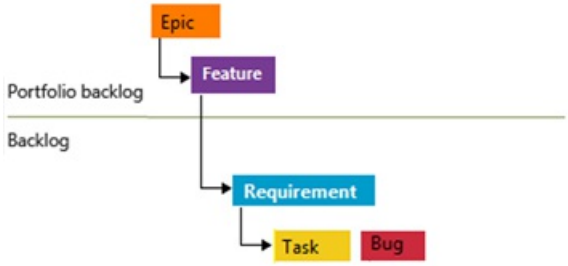
Track bugs as requirements or tasks

As mentioned previously, each team can choose how they want to track bugs to behave like requirements, or tasks, or as neither.

If you choose to track bugs as requirements, bugs should only be nested under the Feature level.



If you choose to track bugs as tasks, bugs should only be nested under the requirements level.



How backlogs and boards display nested items

For TFS 2018 and earlier versions, the Kanban board only shows the last node with nested items of a same-category hierarchy. For all versions, sprint backlogs and taskboards only show the last node in a same-category hierarchy, called the leaf node.

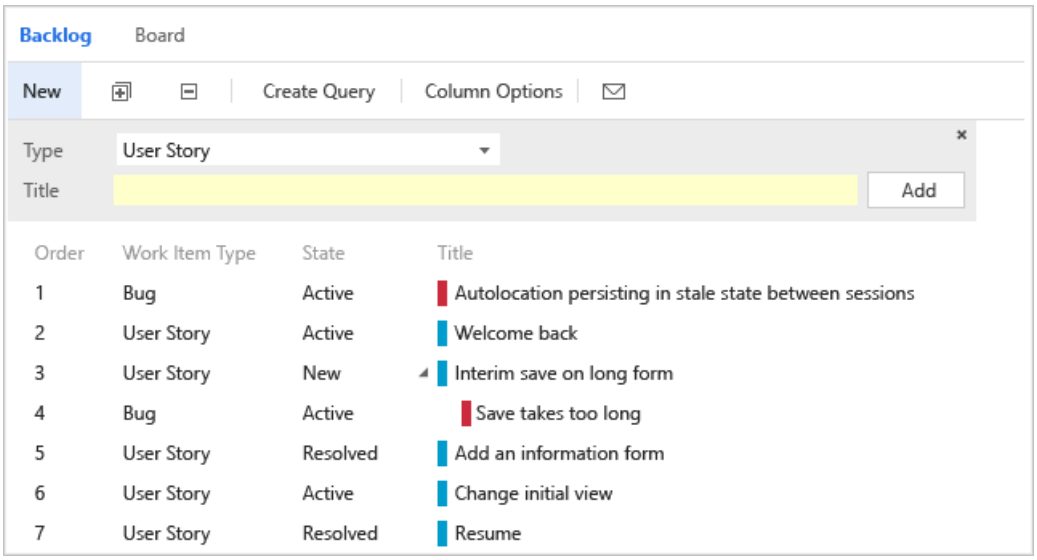
> **NOTE**
> For TFS 2018.2 and later versions, Kanban boards display all work items of nested same-category work items.

Product backlog and Kanban boards

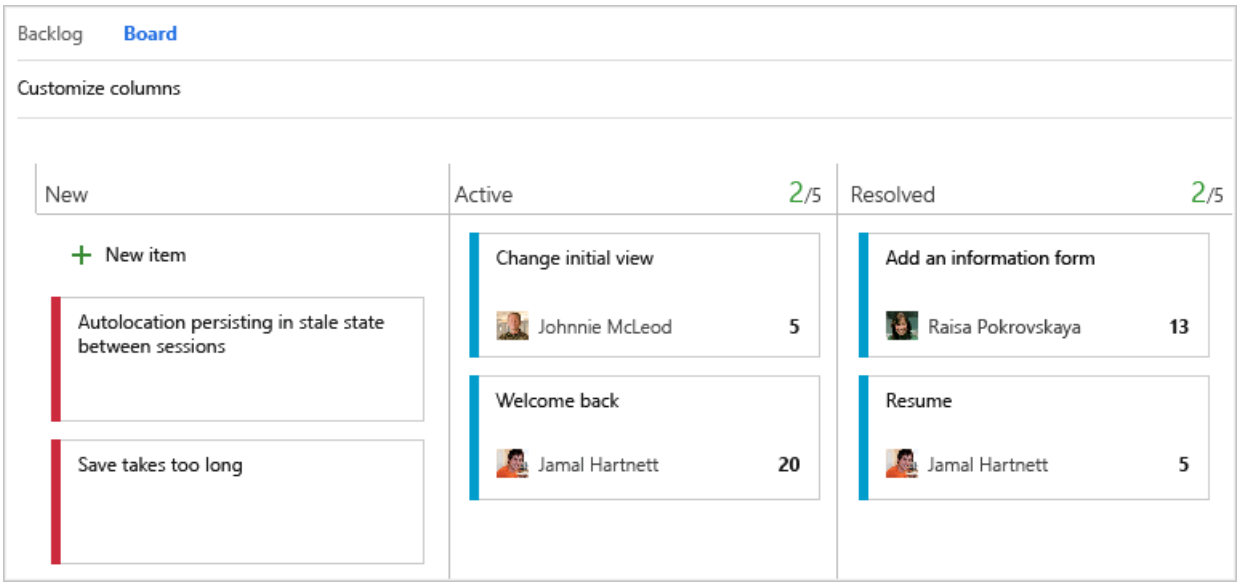
For example, if you link items within a same-category hierarchy that is four levels deep, only the items at the fourth level appear on the Kanban board, sprint backlog, and taskboard.

As shown in the following images, the third user story, *Interim save on long form*, has a child bug, *Save takes too long*. The child bug, *Save takes too long*, appears on the Kanban board, but not the parent user story.

All bugs and requirements appear on the backlog



Only leaf nodes appear on the Kanban board

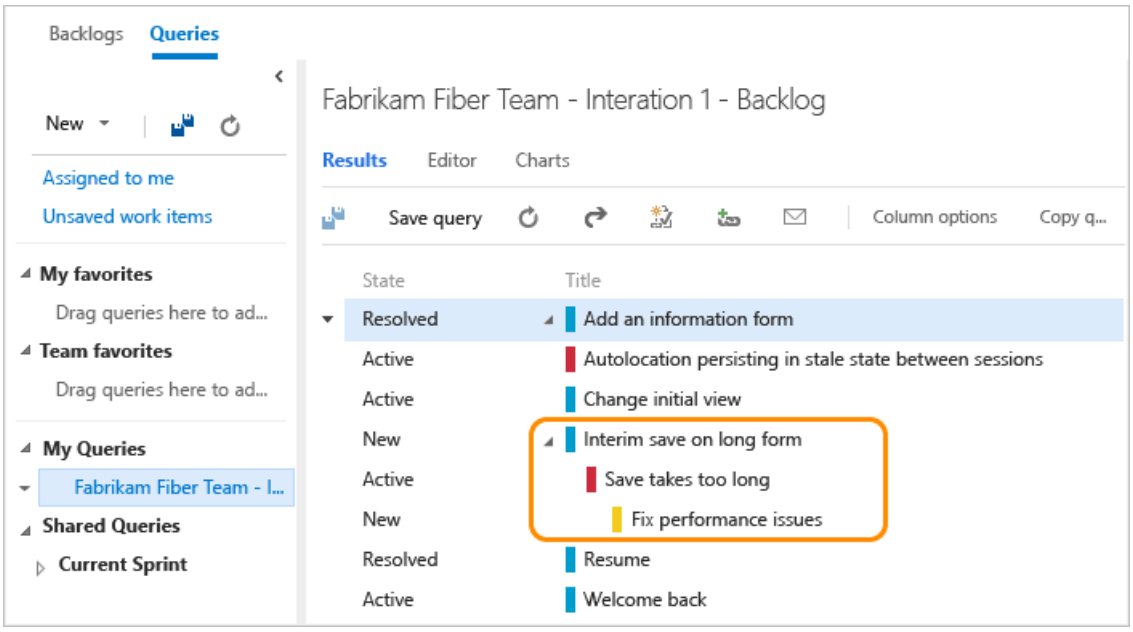


Sprint backlogs and taskboards

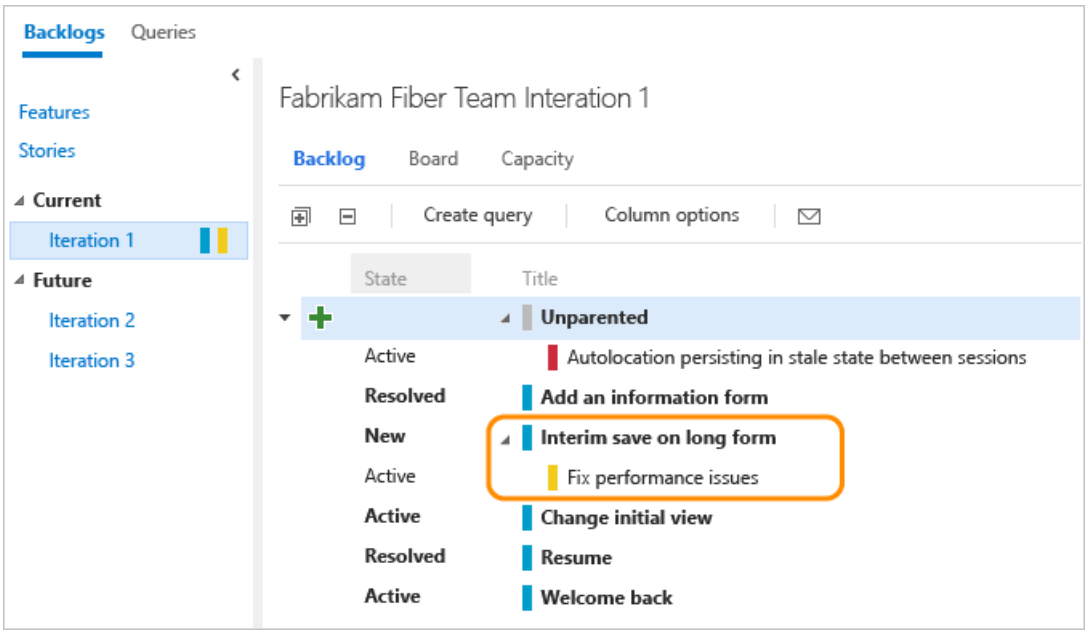
When bugs appear in the backlog with tasks, linking tasks and bugs to their parent requirements groups them correctly on the sprint backlog and taskboard.

But, if you create parent-child links between a requirement and a bug, and the bug and a task, as shown here, the task appears on the sprint backlog and taskboard, but not the bug.

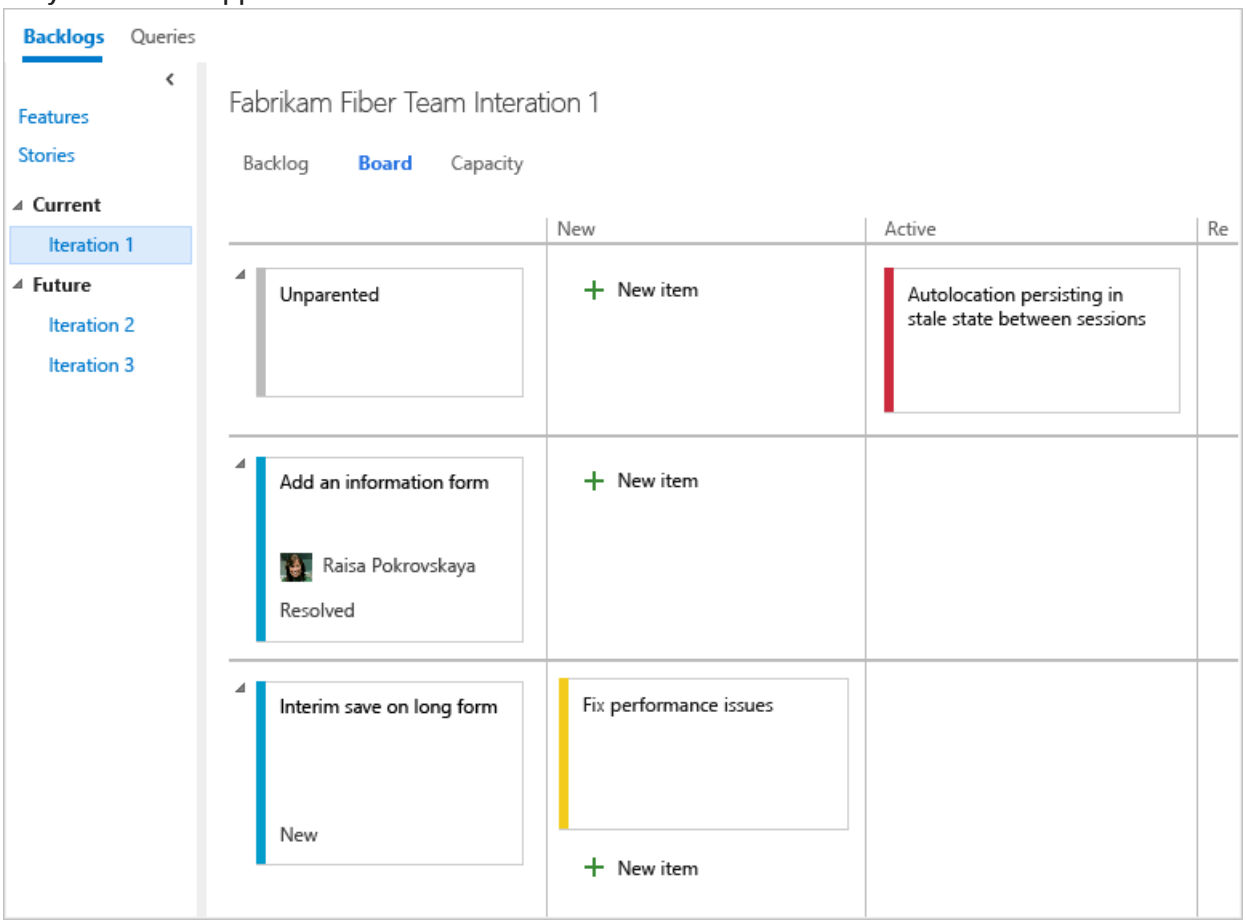
Hierarchy of items assigned to a sprint backlog



Only leaf nodes appear on sprint backlogs



Only leaf nodes appear on taskboards



Is there a workaround to display intermediate nodes within a hierarchy? Not at this time. You can always check the entire list of items assigned to a sprint by using the **Create Query** link.

Related articles

- Set up your Backlogs and Boards
- About Boards and Kanban, Limitations of multi-team Kanban board views
- Tasks supported by Backlogs, Boards, Taskboards, and Plans

Troubleshoot work item form caching issues

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To quickly render work items in your client or browser, several data elements are cached by the IndexDB process. A known issue exists in the cache rules that can cause a delay in server-side rule re-evaluations.

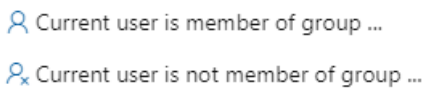
This issue may cause rules added to a work item type to evaluate improperly. Specifically, this issue occurs when modifying a work item in a web browser and a change to a user's group membership causes a change in how a conditional rule should be evaluated. For example, a user is added to a group that lifts a restricted state change, but the user's browser doesn't immediately accept the change in the user's status.

If you come across this issue and take no action, the issue will self-resolve. Each user's cache automatically updates every three days with a full update of membership information. Otherwise, you can resolve the issue by following the instructions provided in Clear the IndexDB cache later in this article.

Background information on issue cause

Each work item uses IndexedDB to cache work item type information and other meta data. Included in the information are rules added to the work item type. User and group-based rule restrictions are evaluated on the server-side. Permissions that prohibit or allow certain actions are applied to the user based on the evaluation.

For the **Inherited process model**, conditional rules include rules with the following conditions:



For the **Hosted XML** and **On-premises XML** process models, conditional rules include rules with the **for** or **not** attributes, such as:

```
<FIELD name="Triage Description">
    <READONLY not="[Project]\Triage Committee" />
</FIELD>
```

Rules are always evaluated from the web cache. Rules that are impacted by user or group membership changes aren't reevaluated in real time. While a potential solution is to invalidate the cache when updates to memberships are made, this solution runs up against performance constraints.

How to avoid the issue

To avoid triggering this issue in the first place, consider adding yourself to all relevant teams before you create a work item. This action ensures that the cache stores the correct information.

Clear the IndexDB cache

If your cache stores outdated rules, you can wait for the client cache to expire within three days or you can clear the cache. To clear the cache, run the following command in the browser command window and then refresh the browser:

```
window.indexedDB.deleteDatabase("wit")
```

Related resources

- Set permissions and access for work tracking
- Apply rules to workflow states (Inheritance process)
- Add a rule to a work item type (Inheritance process)
- Apply or ignore rules based on user or group (On-premises XML process)

Field descriptions for default and work item fields used in process templates

10/4/2022 • 4 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use this index to look up a description of each field used to track work items. This reference includes all fields defined within the core system processes/process templates: Basic, Agile, Scrum, and CMMI. The fields and work item types (WITs) available to you depend on the process you chose when you created your project.

To support other tracking needs, you can define your own custom work item fields.

To support other tracking needs, you can define your own custom work item fields using the Inheritance process model, or if your project collection is configured to use the On-premises XML process model, then see Modify or add a custom field.

To support other tracking needs, you can modify or add a custom field.

> **NOTE**
> The Analytics Service doesn't support reporting on plain text and HTML fields at this time.

Alphabetical index

Values in parenthesis indicate the following criteria:

- **System**: Core system field assigned to all work item types for all processes
- **Agile**: Used only by the Agile process
- **CMMI**: Used only by the CMMI process
- **Scrum**: Used only by the Scrum process
- **TCM**: Used to support Test case management

A

- Acceptance Criteria (Scrum)
- Accepted By
- Accepted Date
- Activated By
- Activated Date
- Activity
- Actual Attendee 1-8 (CMMI)
- Analysis (CMMI)
- Application Launch Instructions
- Application Start Information
- Application Type
- Area ID (System)
- Area Path (System)
- Assigned To
- Associated Context

- Associated Context Code
- Associated Context Owner
- Associated Context Type
- Attached File Count
- Authorized As (Not used)
- Automated Test Id (TCM)
- Automated Test Name (TCM)
- Automated Test Storage (TCM)
- Automated Test Type (TCM)
- AutomatedTestId (TCM)
- AutomatedTestName (TCM)
- Automation Status (TCM)

B

- Backlog Priority (Scrum)
- Blocked
- Board Column[1]
- Board Column Done[1]
- Board Lane[1]
- Business Value

C

- Called By (CMMI)
- Called Date (CMMI)
- Changed By (System)
- Changed Date (System)
- Closed By (System)
- Closed Date (System)
- Closed Status
- Closed Status Code
- Closing Comment
- Comment Count[2]
- Comments (CMMI)
- Committed (CMMI)
- Completed Work
- Contingency Plan (CMMI)
- Corrective Action Actual Resolution (CMMI)
- Corrective Action Plan (CMMI)
- Created By (System)
- Created Date (System)

D-E-F

- Discipline (CMMI)
- Description (System)
- Due Date
- Effort
- Escalate (CMMI)
- External Link Count

- Finish Date
- Found In Build (TCM)
- Found In Environment (CMMI)

H

- History (System)
- How Found (CMMI)
- Hyperlink Count

I

- ID (System)
- Impact Assessment (CMMI)
- Impact on Architecture (CMMI)
- Impact on Development (CMMI)
- Impact on Technical Publications (CMMI)
- Impact on Test (CMMI)
- Impact on User Experience (CMMI)
- Integrated in Build (TCM)
- Issue (TCM)
- Iteration ID (System)
- Iteration Path (System)

J-L-M-N

- Justification (CMMI)
- Link Comment (System)
- Link Description (System)
- Local Data Source (TCM)
- Meeting Type (CMMI)
- Minutes (CMMI)
- Mitigation Plan (CMMI)
- Mitigation Triggers (CMMI)
- Node Name (System)

O-P-Q

- Optional Attendee 1-8 (CMMI)
- Original Estimate
- Parameters (TCM)
- Parent[3]
- Priority
- Probability (CMMI)
- Proposed Fix (CMMI)
- Purpose (CMMI)
- Query Text (TCM)

R

- Rating
- Reason (System)
- Related Link Count (System)
- Remaining Work

- Remote Link Count[4] (System)
- Repro Steps
- Required Attendee 1-8 (CMMI)
- Requirement Type (CMMI)
- Requires Review (CMMI)
- Requires Test (CMMI)
- Resolution (Scrum)
- Resolved By
- Resolved Date
- Resolved Reason
- Rev (System)
- Reviewed By
- Reviewed Date
- Revised Date (System, TCM)
- Risk (Agile)
- Root Cause (CMMI)

S

- Severity
- Size (CMMI)
- Stack Rank
- Start Date
- State (System)
- State Change Date
- State Code
- Steps (TCM)
- Steps to Reproduce (TCM)
- Story Points (Agile)
- Subject Matter Expert (CMMI)
- Symptom (CMMI)
- System Info (TCM)

T

- Tags
- Target Date
- Target Resolve Date (CMMI)
- Task Type (CMMI)
- Team Project (System)
- Test Suite Audit (TCM)
- Test Suite Type (TCM)
- Test Suite Type ID (TCM)
- Time Criticality
- Title (System)
- Triage (CMMI)

U-V-W

- User Acceptance Test (CMMI)
- Value Area

- Watermark (System)
- Work Item Type (System)

> **NOTE**
>
> 1. These fields are available from Azure DevOps Services and TFS 2015.1 or later versions.
> 2. The Comment Count field is available from Azure DevOps Services and TFS 2017 or later versions.
> 3. These fields are available from Azure DevOps Services and Azure DevOps Server 2020.
> 4. These fields are available from Azure DevOps Services only.

By using the system fields or other fields you've added to your project collection, you can enable meaningful cross-project reports and queries. Also, any non-system field that is referenced in the workflow or forms section of the work item type definition must have a **FIELD** element that defines it in the **FIELDS** section of the work item type definition XML file. Also, you must specify any non-system field that you might want to use to generate a query or report in the **FIELDS** section.

Field reference articles

The following articles describe fields that are used in common by several WITs, or those fields that are functionally specific to just one or a few WITs.

Fields common to many work types

- Titles, IDs, and descriptive fields
- History and revision changes
- Areas and iterations
- Assignments and account-specific fields
- Planning, ranking, and priorities
- Work estimates, activity, and other numeric fields
- Build and test integration fields
- Links and attachment related fields

Fields used by specific work item types

- Code Review Request
- Code Review Response
- Feedback Request
- Feedback Response
- Shared Steps
- Test Case

Fields used to track CMMI work items

- Requirements
- Bugs
- Change Requests
- Issues
- Review Meetings
- Risks

Related articles

- About work item fields

- About managed queries
- Define a query
- Choose a process
- Reportable fields reference (on-premises Azure DevOps only)

Work item fields and attributes in Azure Boards

10/4/2022 • 16 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Work item fields are used to track information. Fields are defined for an organization and shared across all projects defined for that organization. You can use one of two tools to review the fields defined for the organization. These tools are available for both Inherited and Hosted XML process models.

- Process>Fields web page
- Work Item Field Explorer

Work item fields are used to track information. Fields are defined for a collection and shared across all projects defined for that collection. You can use one of two tools to review the fields defined for the Collection.

- Process>Fields web page: Available for Inherited process model
- Work Item Field Explorer: Available for Inherited and On-premises XML process models.

Work item fields are used to track information. Fields are defined for a collection and shared across all projects defined for that collection. To view all fields defined for a collection, you can use the Work Item Field Explorer tool, a plug-in to Visual Studio.

For a description of each field defined with a system process, see Work item field index.

Prerequisites

- To view the fields defined for an organization or collection, you must be a member of the **Project Collection Valid Users** application group or have the **View instance-level information** permission set to **Allow** for the organization or collection.

List or review fields

To list or review fields, you can use one of the following tools, depending on the process model—Inheritance, Hosted XML, or On-premises XML—you use. For an index of fields defined within the default processes, see Work item field index.

TOOL	INHERITANCE	HOSTED XML	ON-PREMISES XML
Web portal: List inherited and custom-defined fields	✔	✔1	
Work item field explorer	✔	✔	✔
witadmin listfields command line tool	✔	✔	✔

NOTE

1. Only supported for default processes (Agile, CMMI, Scrum).

Field data types and names

Each work item type specifies the fields defined for the work items that reference that type. Each field is associated with a number of attributes, many of which are set by the system and cannot be changed.

Each field is defined by the following three attributes.

- **Data type**: Specifies the type of data that can be entered into the field, such as Boolean, Double, Integer, HTML, and String. For descriptions of each data type, see Query fields, operators, and macros.
- **Friendly name**: Specifies the name assigned to the field and that you select for a **Field** in a query clause. This name may differ from the name displayed on the work item form.
- **Reference name**: Specifies the name that you use when creating WIQL query or an improvised work item template, using REST API commands, or defining XML work item type definitions. Once defined, the reference name cannot be changed.

For a description of each field attribute and how you can list them, see Field attributes and List field attributes later in this article. For an overview of WITs and work items, see Track work with user stories, issues, bugs, features, and epics.

What is a field? How are field names used?

Each work item type is associated with 31 system fields and several more type-specific fields. You use work items to plan and track your project.

Each field supports tracking a piece of information about the work to perform. Values you assign to a field are stored in the work tracking data store which you can create queries to determine status and trends.

For descriptions and usage of each field defined for the core system processes—Agile, Basic, Scrum, and CMMI processes—see Work item field index.

Field names

A work item field name uniquely identifies each work item field. Make sure your field names fall within these guidelines:

- Field names must be unique within the account/project collection
- Field names must be 128 or fewer Unicode characters
- Field names can't contain any leading or trailing spaces, nor two or more consecutive spaces
- Field names must contain at least one alphabetic character
- Field names can't contain the following characters: `.,;'`:~\/*|?"&%$!+=()[]{}<>`.

Because custom fields are defined for an organization or collection, you can't add a custom field to a process with the same field name that you add to another process.

For more information, see Naming restrictions and conventions.

System and predefined fields

All system defined fields have reference names that begin with *System*, for example, System.AreaPath, System.AssignedTo, and continue in that pattern.

Predefined fields defined by the default process begin with Microsoft.VSTS and then further differ based on their usage. Examples of predefined fields that are used in common, for scheduling purposes and integration with Office Project, for integration with Team Foundation Build, and integration with test case management (TCM) are as follows:

- Microsoft.VSTS.Common.Priority
- Microsoft.VSTS.Scheduling.DueDate
- Microsoft.VSTS.Build.FoundIn
- Microsoft.VSTS.TCM.Steps

For an overview of all system and predefined fields that are defined for the default processes/process templates, see Work item field index. For more information about specifying field names, see Naming restrictions.

Custom fields

Because custom fields are defined for an organization or project collection, you can't add a custom field to a process with the same field name that you add to another process.

When adding custom fields, note the following limits:

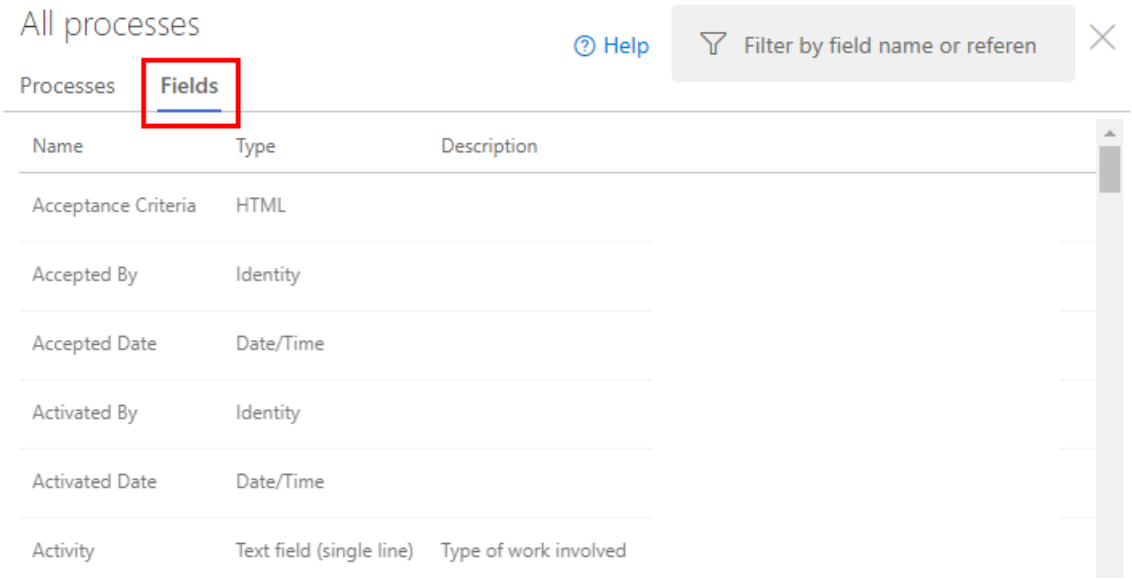
- A maximum of 64 fields can be defined for each WIT
- A maximum of 512 fields can be defined per process

The field data type determines the kind and size of data that you can store in the field. A field can have only one type defined within a project collection. This restriction encourages organizations to use common fields across projects and work item types.

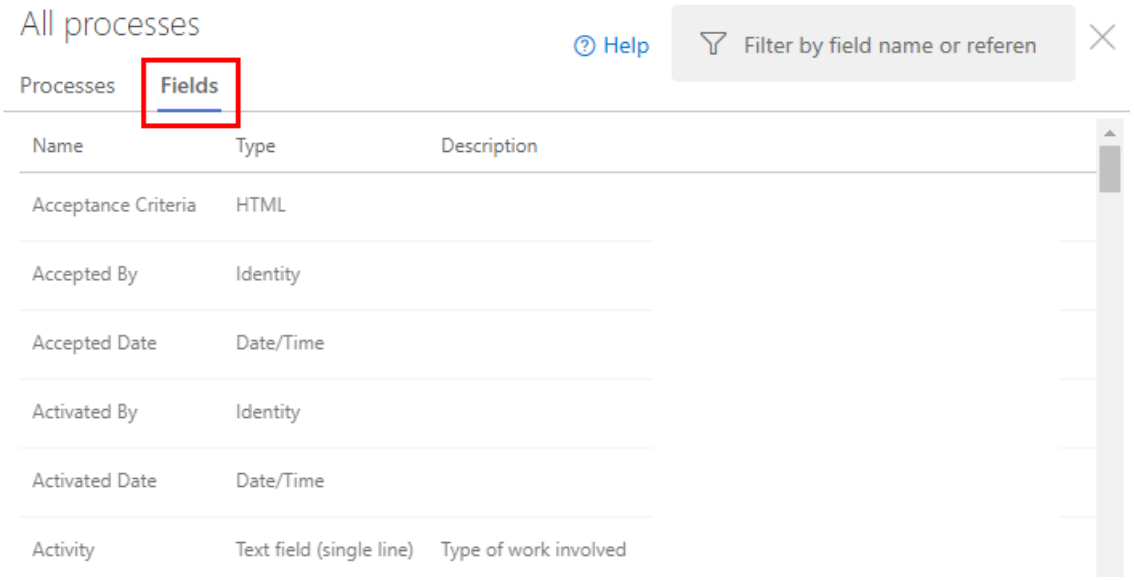
When you add a custom field to an inherited process, Azure DevOps assigns a reference name prefixed with *Custom* and then the name of the field with spaces removed. For example, you add a field named *DevOps Triage*, the reference name is **Custom.DevOpsTriage**. No spaces are allowed within the reference name.

How can I determine the field data type?

You can view the data type of fields defined for your organization by opening the Process>Fields page.



When your project collection uses the Inheritance process model to customize work tracking, you can view the data type of fields by opening the Process>Fields page.



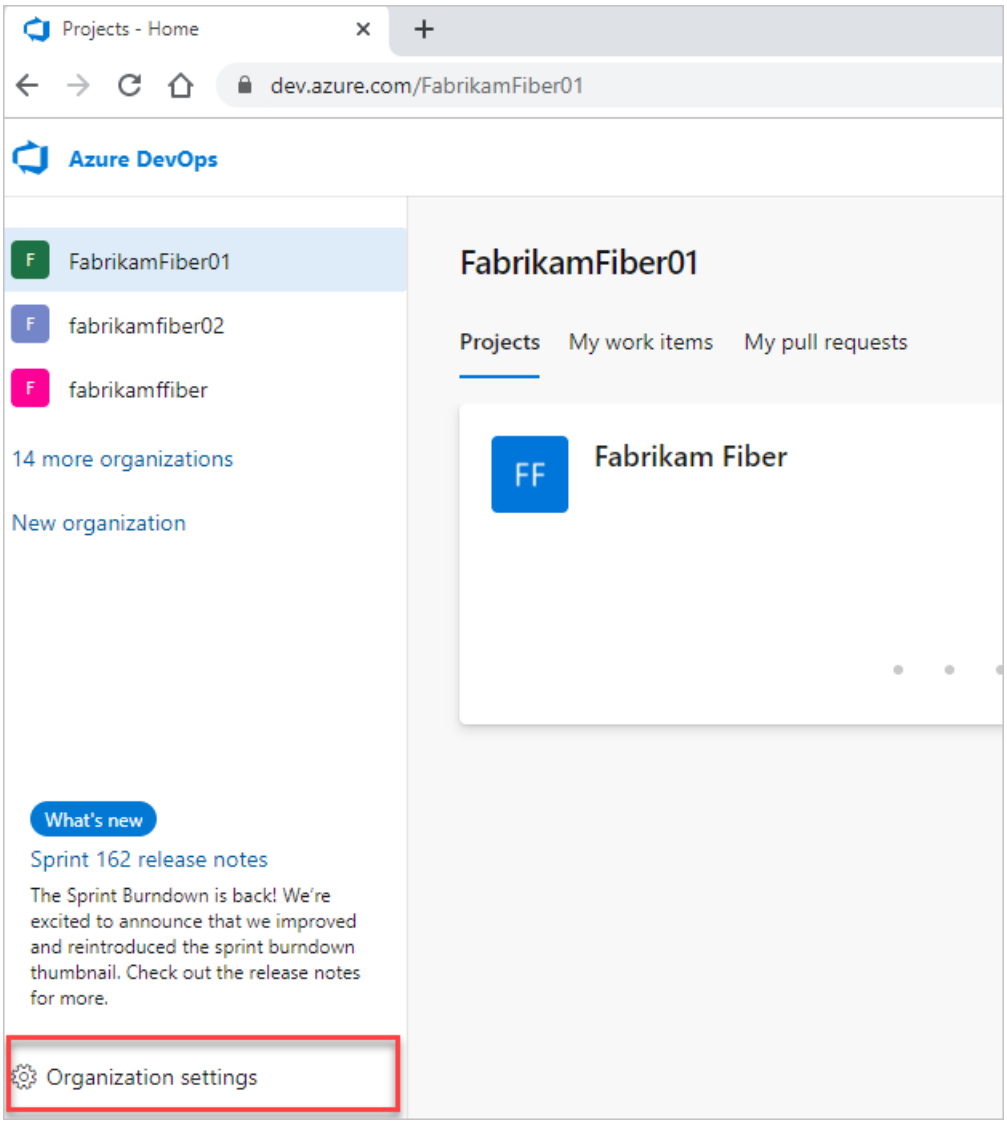
If the On-premises XML process model is used, you can look up the data type through the Work item field index. Or, you can open the Work Item Field Explorer to review the fields defined and their attribute assignments, or use the **witadmin listfields** command to list the field attributes. For details, see Work Item Field Explorer and List field attributes later in this article.

You can look up the data type through the Work item field index. Or, you can open the Work Item Field Explorer to review the fields defined and their attribute assignments, or use the **witadmin listfields** command to list the field attributes. For details, see Work Item Field Explorer and List field attributes later in this article.

Process>Fields web page

To review the list of fields defined for an organization or collection, open **Organization settings>Process>Fields**.

1. Choose the Azure DevOps logo to open **Projects**. Then choose **Organization settings**.



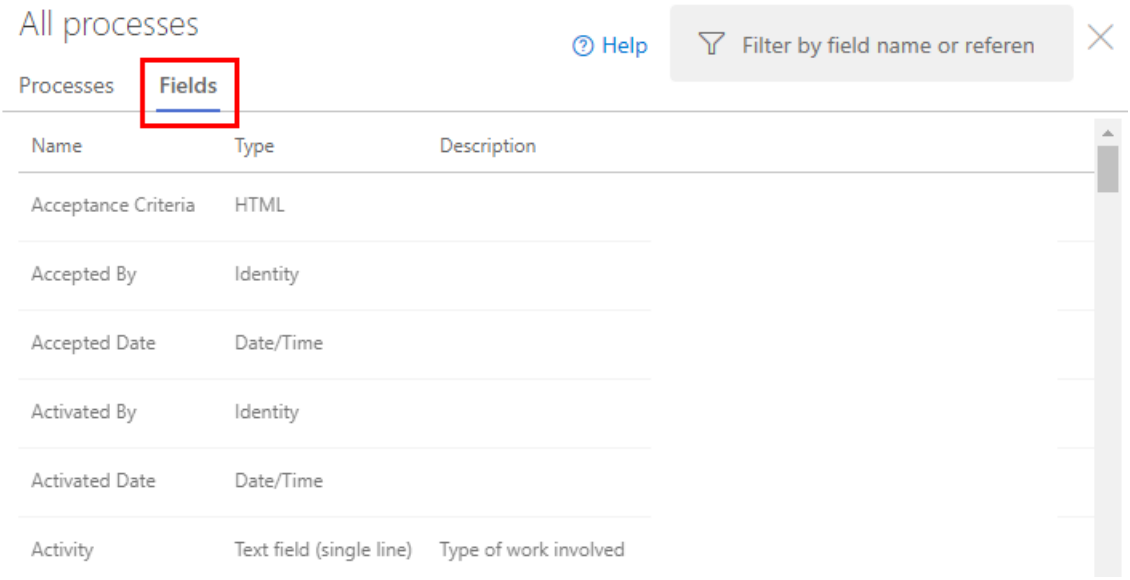
2. Then, choose **Process**.



> **NOTE**
> If you don't see **Process**, then you're working from TFS-2018 or earlier version. The **Process** page isn't supported. You must use the features supported for the On-premises XML process model.

3. Then, choose **Fields**.

   Fields listed correspond to all fields defined for the organization or collection. This includes all custom fields and those defined for system processes.

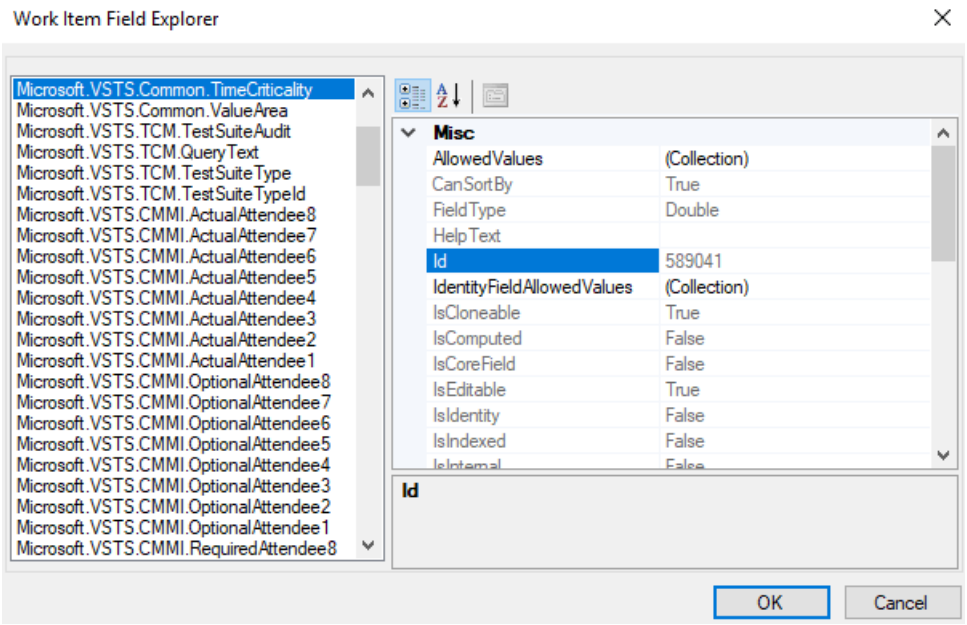


> **NOTE**
>
> If you don't see **Fields**, then you're collection uses the On-premises XML process. The Fields page isn't supported for that process.

For descriptions and usage of each field, as well as the Reference name for each field, you can look it up from the Work item field index. You can also get the Reference name of fields from the Work Item Types Field - List REST API.

Work Item Field Explorer

You can look up the assignments of field attributes using the Work Item Field Explorer tool.



To access the Work Item Field Explorer, you must install the Process Editor Tool. Based on the version of Visual Studio you have installed, get the Process Editor Tool from one of the following extensions.

- **Visual Studio 2019**: Process Template Editor.
- **Visual Studio 2017**: TFS Process Template Editor. You can also use this version of the Process Editor to modify the old-style work item forms. You can't use it to edit forms associated with the new web forms.

- **Visual Studio 2015**: TFS Power Tools.

Field attributes

There are many non-changeable and hidden attributes for each work item field. The following table describes each attribute. Attributes have different names based on if you get them through the **Fields - Get REST** API or view through the Work Item Field Explorer (**WIFE**) tool, and the FieldDefinition Properties.

Attributes assigned to a field depend on the platform and version you use. For example, some attributes aren't support with the Inheritance process. To look up the reference name for a field, see Work item field index.

Attribute

Attribute type

Description

REST:
WIFE: **AllowedValues**

collection

Gets the collection of valid values for a field that contains picklist values. You can change this by specifying a picklist or global list (on-premises).
Can change?=Yes

REST: **canSortBy**
WIFE: **CanSortBy**

boolean

Indicates whether you can sort query results with this field.
Can change?=No

REST: **description**
WIFE: **HelpText**

string

Specifies a description for the field, which also defines the help text that appears when you hover over the field within the work item form.
Can change?=Yes

REST:
WIFE: **ID**

Integer

Specifies the internal ID of the field.
Can change?=No

REST:
WIFE: **IsCloneable**

boolean

Indicates whether the value defined for the field is copied when a user chooses to copy a work item. For example, **Title**, **Tags**, and **Description** fields are copied, but the **ID** and **History** fields aren't copied.
Can change?=No

REST:
WIFE: **IsComputed**

boolean

Indicates if the value set by this field is computed by the system (True) or not (False). Examples of computed fields are ones that are set by the system, such as the **ID**, **Revised Date**, **Changed Date**, and **External Link Count**.
Can change?=No

REST:
WIFE: **IsCoreField**

boolean

Indicates whether this field is specified for all work item types.
Can change?=No

REST:
WIFE: **IsEditable**

boolean

Indicates if users can modify this field (True) or not (False). Examples of non-editable fields are ones that are set by the system, such as the **ID**, **Revision**, **Created By**, and **Changed By** fields
Can change?=No

REST: **isIdentity**
WIFE: **IsIdentity**

boolean

Indicates whether this field is an *Identity* field. Identity fields are string fields used to store user identities.
Can change?=No

REST:
WIFE: **IsIndexed**[1]

boolean

Indicates whether this field is indexed to support search.
Can change?=No

REST:
WIFE: **IsLongText**

boolean

Indicates that the field can contain more than 255 characters, such as fields assigned a data type of *PlainText*, *HTML*, or *History*.
Can change?=No

REST: **isPicklist**[2] WIFE:

boolean

Indicates whether the field is associated with a picklist. The value is set to *True* when a custom field is defined for Azure DevOps and Picklist (String) or Picklist (Integer) type is selected. The value is set to *False* for inherited fields that define picklists.
Can change?=No

REST: **isPicklistSuggested**[2] WIFE:

boolean

Indicates whether the field allows users to enter their own values for a picklist. The value is set to *True* when a custom field is defined for Azure DevOps, Picklist (String), or Picklist (Integer) type is selected, and the checkbox for **Allow users to set their own values** is checked.
Can change?=Yes

REST: **isQueryable**
WIFE: **IsQueryable**

boolean

Indicates if the field shows up within the set of fields you can add to filter a work item query (True), or not (False). Most fields are queryable.
Can change?=No

REST:
WIFE: **IsReportable** [3]

boolean

Indicates if the reportable attribute is defined or set to anything other than **None**. This attribute can be changed for on-premises environments.
Can change?=Yes

REST:
WIFE: **IsUsedInGlobalWorkflow**

boolean

Indicates if the field is defined within a global workflow.
Can change?=No

REST:
WIFE: **IsUserNameField**

boolean

Indicates if the field is used to display an Identity field.
Can change?=No

REST: **name**
WIFE: **Name**

string

Friendly name assigned to the field. The friendly name can't be changed for Azure DevOps, but can be changed for on-premises using the **witadmin changefield** command.
Can change?=On-prem only

REST: **picklistId**
WIFE: **HelpText**

GUID

If the field is a picklist, the identifier of the associated picklist, otherwise null. A unique GUID value is assigned when a custom field is defined for Azure DevOps and Picklist (String) or Picklist (Integer) type is selected.
Can change?=No

REST:
WIFE: **ProhibitedValues**

collection

Gets the collection of prohibited values for a field that specifies such values. You can only define prohibited values for on-premises deployments.
Can change?=On-prem only

REST: **readOnly**
WIFE:

boolean

Indicates whether the field is set to read only. For Azure DevOps Services, only custom fields can be changed to be read-only. System fields cannot be modified.
Can change?=Yes

REST: **referenceName**
WIFE: **ReferenceName**

string

Specifies the reference name of a field.
Can change?=No

REST:
WIFE: **ReportingAttributes**[3]

Specifies **Detail**, **Dimension**, or **Measure**, depending on whether and how you want the field to be included in reports. Data from fields that have a value other than **None** for this attribute are exported to the data warehouse and can be included in SQL reports.
Can change?=On-prem only

REST:
WIFE: **ReportingName**[3]

string

Specifies the label for a field when data appears in SQL reports. If you don't specify a value, the field's friendly name is used.
Can change?=On-premises only

REST:
WIFE: **ReportingReferenceName**[3]

string

Specifies a different reference name to a field that is used when data is exported to the relational data warehouse. If you don't specify a value, the fields reference name is used.
Can change?=On-premises only

REST: **supportedOperations**
WIFE:

set

The set of query operators that are valid for use when referencing this field. For a quick reference of supported operations based on data type, see Query quick reference, Operators, and macros supported for each data type.
Can change?=No

REST:
WIFE: **SupportsTextQuery**

boolean

Indicates whether the field supports text queries such as **Contains Words**, **Does Not Contains Words**.
Can change?=No

REST:
WIFE: **SystemType**

string

Specifies the data type of the field, referencing the system name such as System.DateTime or System.String.
Can change?=No

REST: **type**
WIFE: **FieldType**

string

Specifies the data type of the field, such as *Boolean*, *DateTime*, *Integer*, *String*, and so on. For a complete list and descriptions, see Query fields, operators, and macros.
Can change?=No

REST: **usage**
WIFE: **Usage**

string

Specifies whether the field is intended for use with work items (*WorkItem*) or work item link (*WorkItemLink*) objects. The usage for most fields is WorkItem. For a complete list of usage values, see Get Fields, FieldUsage.
Can change?=No

NOTE

1. For on-premises deployments, you can enable indexing for a field to improve query response times when filtering on the field. For more information, see Indexed fields later in this article.
2. The **isPicklist** and **isPicklistSuggested** attributes are only assigned to custom fields defined for an inherited process. The Inherited process model is supported for Azure DevOps Server 2019 and later versions. To learn more, see Inherited process model.
3. All reporting attributes are valid only for on-premises deployments whose projects have been configured to support SQL Server Reporting and SQL Server Analysis Services.

Reportable attributes

All reporting attributes are valid only for on-premises deployments whose projects have been configured to support SQL Server Reporting and SQL Server Analysis Services. For details, see Add reports to a project.

For a description of each reportable attribute, see Add or modify work item fields to support reporting.

For a list of fields that have reportable attributes defined by default, see Reportable fields reference.

Indexed fields

You can enable or disable indexing for a work item field by using the **witadmin indexfield** command. When you enable indexing for a field, you may increase the performance of finding work items whose queries specify

that field. By default, the following fields are indexed: Assigned To, Created Date, Changed By, State, Reason, Area ID, Iteration ID, and Work Item Type.

If you add a custom field that you use in many of your work item queries, you may want to enable indexing for that field. For more information, see Manage work item fields (witadmin).

List field attributes

You can list the attributes assigned to a field by using the **Fields - Get** REST API. Enter your organization name for *OrganizationName*.

```
https://dev.azure.com/OrganizationName/_apis/wit/fields/FieldReferenceName
```

For example, here we list the attributes for the Iteration Path, specifying the reference name, `System.IterationPath`, for the fabrikam organization.

```
https://dev.azure.com/fabrikam/_apis/wit/fields/System.IterationPath
```

Returned data:


```
{
"name": "Iteration Path",
"referenceName": "System.IterationPath",
"description": "The iteration within which this bug will be fixed",
"type": "treePath",
"usage": "workItem",
"readOnly": false,
"canSortBy": true,
"isQueryable": true,
"supportedOperations": [
{
"referenceName": "SupportedOperations.Under",
"name": "Under"
},
{
"referenceName": "SupportedOperations.NotUnder",
"name": "Not Under"
},
{
"referenceName": "SupportedOperations.Equals",
"name": "="
},
{
"referenceName": "SupportedOperations.NotEquals",
"name": "<>"
},
{
"referenceName": "SupportedOperations.In",
"name": "In"
},
{
"name": "Not In"
}
],
"isIdentity": false,
"isPicklist": false,
"isPicklistSuggested": false,
"url": "https://dev.azure.com/mseng/_apis/wit/fields/System.IterationPath"
}
```

You can list the attributes assigned to a field by using the **Fields - Get** REST API. Enter your organization name for *OrganizationName*. To get started using REST, see Azure DevOps Services REST API Reference

```
https://{ServerName:Port}/tfs/{Collection}/_apis/wit/fields/FieldReferenceName?api-version={version}
```

For example, here we list the attributes for the Iteration Path, specifying the reference name, `System.IterationPath`, for the fabrikam server.

```
https://fabrikam:8080/tfs/DefaultCollection/_apis/wit/fields/System.IterationPath?api-version=4.1
```

Returned data:


```
{
"name": "Iteration Path",
"referenceName": "System.IterationPath",
"description": "The iteration within which this bug will be fixed",
"type": "treePath",
"usage": "workItem",
"readOnly": false,
"canSortBy": true,
"isQueryable": true,
"supportedOperations": [
{
"referenceName": "SupportedOperations.Under",
"name": "Under"
},
{
"referenceName": "SupportedOperations.NotUnder",
"name": "Not Under"
},
{
"referenceName": "SupportedOperations.Equals",
"name": "="
},
{
"referenceName": "SupportedOperations.NotEquals",
"name": "<>"
},
{
"referenceName": "SupportedOperations.In",
"name": "In"
},
{
"name": "Not In"
}
],
"isIdentity": false,
"isPicklist": false,
"isPicklistSuggested": false,
"url": "https://fabrikam:8080/tfs/DefaultCollection/_apis/wit/fields/System.IterationPath?api-version=4.1"
}
```


List attributes using `witadmin` command-line tool

You can list select field attributes—such as the data type, reportable attributes, and indexing—using the **witadmin listfields** command.

For example, you can enter the following command to list the attributes defined for a specified field, such as Microsoft.VSTS.Common.Issue.

```
witadmin listfields /collection:http://fabrikam:8080/tfs/DefaultCollection /n:Microsoft.VSTS.Common.Issue
```

Field and attribute information appears for the named field, as shown in this example.

```
Field: Microsoft.VSTS.Common.Issue
Name: Issue
Type: String
Reportable As: dimension
Use: Adventure Works (Shared Steps), AW Future (Shared Steps), AW Current (Shared Steps)
Indexed: False
```

The **Use** parameter indicates the name of each project and the work item type where the field is used.

Project integration and project field mapping

You can change how work tracking fields map to fields in Project, and you can change how specific fields are published. See [The Microsoft Project Field Mapping File](/previous-versions/azure/devops/reference/xml/customize-project-field-mapping-file.

Add and modify fields

To add fields to a process, you add them to one or more work item types. To learn more, see Customize an inheritance process.

You can add or modify the fields contained within a WIT or add a custom WIT. To learn more, see:

- For project collections that use the Inheritance process model: Customize an inheritance process.
- For project collections that use the On-premises XML process model: Customize the On-premises XML process model.

You can change the field name, the index, and the report attributes for any field except system fields by using the **witadmin** command-line tool. For more information, see Manage work item fields-witadmin.

To add fields to a project, you add them to one or more work item types. See Add or modify a field to track work.

You can change the field name, the index, and the report attributes for any field except system fields by using the **witadmin** command-line tool. For more information, see Manage work item fields-witadmin.

Related articles

- Query quick reference
- Work item field index
- Add and manage fields for an inherited process

- Query quick reference
- Work item field index
- Choose the process model for your project collection
- Add or modify a field to track work
- Manage work item fields-witadmin

- Query quick reference
- Work item field index
- Add or modify a field to track work
- Manage work item fields-witadmin

Code review and feedback field reference in Azure Boards and Azure DevOps

10/4/2022 • 3 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can use the code review and feedback fields to create queries and reports that track the status of these processes. The fields appear in the following work item types, which are included with the default processes for Azure Boards and TFS: Code Review Request, Code Review Response, Feedback Request, and Feedback Response.

> **NOTE**
> If your Azure DevOps or TFS server has been upgraded from an earlier version you might need to update your project to get access to these work item types. See Configure features after an upgrade.

Fields used to track code reviews

The following fields are used to track code review requests and responses. You can only make a Code Review request against code maintained in a Team Foundation version control (TFVC) repository. A code review response is created for each person who provides review comments. See Day in the life of a Developer: Suspend work, fix a bug, and conduct a code review.

> **NOTE**
> Retrieving code review comments programmatically isn't a supported feature.

FIELD NAME	DESCRIPTION	DATA TYPE
Accepted By	The name of the code reviewer. Reference name=Microsoft.VSTS.CodeReview.AcceptedBy	String
Accepted Date	The date and time when the code-reviewer responded. Reference name=Microsoft.VSTS.CodeReview.AcceptedDate	DateTime
Associated Context	The name assigned to the code work requested for review. Reference name=Microsoft.VSTS.CodeReview.Context	String

FIELD NAME	DESCRIPTION	DATA TYPE
Associated Context Code	An integer value that captures whether the code review is for **1** (shelveset) or **2** (changeset). Reference name=Microsoft.VSTS.CodeReview.ContextCode	Integer
Associated Context Owner	The GUID assigned to the shelveset owner who requested the code review. Reference name=Microsoft.VSTS.CodeReview.ContextOwner	String
Associated Context Type	The type of code work requested for review: **Shelveset** or **Changeset**. Reference name=Microsoft.VSTS.CodeReview.ContextType	String
Closed Status	The status selected by the reviewer when closing the code review request. The number is stored in the system and written to the data warehouse as follows: - **0 — Not Reviewed** - **1 - Looks Good** - **2 - With Comments** - **3- Needs Work** - **4 - Declined** - **5 - Removed** Reference name=Microsoft.VSTS.CodeReview.ClosedStatus	String
Closed Status Code	A value ranging from 0 to 5 that corresponds to the status selected by the reviewer when closing the code review request. Reference name=Microsoft.VSTS.CodeReview.ClosedStatusCode	Integer
Closing Comments	The comment entered by the reviewer when closing the review request. Reference name=Microsoft.VSTS.CodeReview.ClosingComment	String
Reviewed By	The name of the team member who reviewed the code. The State transitions to Reviewed when the code reviewer responds. (Code Review Response) Reference name=Microsoft.VSTS.Common.ReviewedBy	String

FIELD NAME	DESCRIPTION	DATA TYPE
Reviewed Date	The date-time stamp when the reviewer closed the request. (Code Review Response) Reference name=Microsoft.VSTS.Common.ReviewedDate	Date-Time
State Code	Mirror field used to track the current state in code. Reference name=Microsoft.VSTS.Common.StateCode	Integer

Fields used to track feedback

The following fields track feedback requests and responses. You complete the first three fields in the feedback request form. A feedback response is created for each person and for each item for which feedback is provided. See Get feedback.

FIELD NAME	DESCRIPTION	DATA TYPE
Application Launch Instructions	Instructions to stakeholders on how to start the application. Reference name=Microsoft.VSTS.Feedback.ApplicationLaunchInstructions	HTML
Application Start Information	Guidelines to direct stakeholder feedback. Reference name=Microsoft.VSTS.Feedback.ApplicationStartInformation	PlainText
Application Type	The type of application that stakeholders will provide feedback on. The valid types are specified in the process configuration file, ProcessConfiguration. The default values are **Web Application**, **Remote Machine**, and **Client Application**. Reference name=Microsoft.VSTS.Feedback.ApplicationType	String

FIELD NAME	DESCRIPTION	DATA TYPE
Rating	The number of stars that an item receives from a reviewer in a star-based ranking system. (Feedback Response) The number is stored in the system and written to the data warehouse as follows: - **0 — Not Rated** - **1 - Poor** - **2 - Fair** - **3- Good** - **4- Very Good** - **5 - Excellent** Reference name=Microsoft.VSTS.Common.Rating	String

Related articles

- Index of work item fields
- Get feedback
- Day in the life of a Developer: Suspend work, fix a bug, and conduct a code review

Bugs, issues, and risks in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The following fields track information about bugs, issues, and risks. These work item types are defined within the process template for the CMMI process.

Bug tracking fields

These fields are neither reported nor indexed.

FIELD NAME	DESCRIPTION	DATA TYPE
Symptom	The unexpected behavior. Reference name=Microsoft.VSTS.CMMI.Symptom	HTML
Proposed Fix	The proposed change to fix the reported problem. Reference name=Microsoft.VSTS.CMMI.ProposedFix	HTML
Found in Environment	The software setup and configuration where the bug was found. Reference name=Microsoft.VSTS.CMMI.FoundInEnvironment	String
Root Cause	The cause of the error. You can specify one of the following values: - **Coding Error** - **Design Error** - **Specification Error** - **Communication Error** - **Unknown** To change the menu selection, see Customize a picklist. Reference name=Microsoft.VSTS.CMMI.RootCause	String
How Found	How the bug was found. For example, a bug might have been found during a customer review or through ad hoc testing. Reference name=Microsoft.VSTS.CMMI.HowFound	String

Issue tracking fields

These fields are neither reported nor indexed.

FIELD NAME	DESCRIPTION	DATA TYPE
Analysis	The root cause of the issue and one or more solutions that might resolve it. Reference name=Microsoft.VSTS.CMMI.Analysis	HTML
Corrective Action Actual Resolution	What the team actually did to correct the issue. Reference name=Microsoft.VSTS.CMMI.CorrectiveActionActualResolution	HTML
Corrective Action Plan	The proposed corrective action on which the team has agreed. Reference name=Microsoft.VSTS.CMMI.CorrectiveActionPlan	HTML
Target Resolve Date	The date when the issue becomes critical and starts to affect the critical path of the project plan. Reference name=Microsoft.VSTS.CMMI.TargetResolveDate	DateTime

Risk tracking fields

These fields are neither reported nor indexed.

FIELD NAME	DESCRIPTION	DATA TYPE
Contingency Plan	The actions to take if the risk occurs. You can create and link tasks to the Risk work item to track the work that the team must complete to implement the contingency plan. Also, you can create an Issue work item to track one or more issues on which the risk has an impact. Reference name=Microsoft.VSTS.CMMI.ContingencyPlan	HTML
Mitigation Plan	The actions to take to reduce the probability or impact of the risk. You can create and link tasks to the Risk work item to track the work that the team must complete to implement the mitigation plan. Reference name=Microsoft.VSTS.CMMI.MitigationPlan	HTML

FIELD NAME	DESCRIPTION	DATA TYPE
Mitigation Triggers	The conditions or events that determine how the team might mitigate a risk. For example, the triage team might authorize and obtain a reserve generator if the weather forecast is predicting an ice storm or hurricane to hit within 50 miles of the office within the next four days. Reference name=Microsoft.VSTS.CMMI.MitigationTriggers	HTML
Probability	A number that indicates the chance that the risk will occur. A valid probability number is between 1 and 99, where 99 indicates that the risk is almost certain to occur. Reference name=Microsoft.VSTS.CMMI.Probability	Integer

Related articles

Index of work item fields

Change request field reference for Capability Maturity Model Integration (CMMI) work items

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can track change requests for CMMI work items by using these six fields: Justification, Impact on Architecture, Impact on User Experience, Impact on Test, Impact on Development, and Impact on Technical Publications. A description and a reference name for each of the change request fields are provided in the following table. When you open a work item form for a change request, the **Justification** field appears on the **Justification** tab, and all other fields appear on the **Analysis** tab.

The Change Request work item type is provided only with the CMMI process.

None of these fields are reportable or indexed. They all have a data type of HTML.

FIELD NAME	DESCRIPTION	REFERENCE NAME
Justification	Why the change has been proposed and what value it would bring to the product and the customer.	Microsoft.VSTS.CMMI.Justification
Impact on Architecture	The impact that the change would have on architecture. You can use this field to describe in detail which sections of the architecture would be affected and how much the change would cost to implement.	Microsoft.VSTS.CMMI.ImpactOnArchitecture
Impact on User Experience	The impact that the change would have on the user experience. You can use this field to describe in detail which sections of the user interface would be affected and how much the change would cost to implement.	Microsoft.VSTS.CMMI.ImpactOnUserExperience
Impact on Test	The impact that the change would have on testing. You can use this field to describe in detail which tests would be affected and how much the change would cost to implement.	Microsoft.VSTS.CMMI.ImpactOnTest
Impact on Development	The impact that the change would have on development and product designs. You can use this field to describe in detail which development areas and designs would be affected and how much the change would cost to implement.	Microsoft.VSTS.CMMI.ImpactOnDevelopment

FIELD NAME	DESCRIPTION	REFERENCE NAME
Impact on Technical Publications	The impact that the change would have on product documentation. You can use this field to describe in detail which sections of documentation would be affected and how much the change would cost to implement.	Microsoft.VSTS.CMMI.ImpactOnTechnicalPublications

Related articles

- Index of work item fields

Requirements field reference for the CMMI process in Azure Boards

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you create a project using the CMMI process, you can define fields to track requirements to be developed and their importance to the overall product.

None of these fields are indexed. For more information about data types and field attributes, see Work item fields and attributes.

FIELD NAME	DESCRIPTION	DATA TYPE	REPORTABLE TYPE
Requirement Type	The kind of requirement to implement. You can specify one of the following values (see note 1): - **Business Objective** - **Feature** (default) - **Functional** - **Interface** - **Operational** - **Quality of Service** - **Safety** - **Scenario** - **Security** Reference Name=Microsoft.VSTS.CMMI.RequirementType	String	Dimension
Impact Assessment	The customer impact of not implementing this requirement. You might include details from the Kano model about whether this requirement is in the surprise, required, or obvious categories. Reference Name=Microsoft.VSTS.CMMI.ImpactAssessmentHtml	HTML	None

FIELD NAME	DESCRIPTION	DATA TYPE	REPORTABLE TYPE
User Acceptance Test	The status of the user acceptance test for a requirement. You can specify one of the following values (1): - **Pass** (default) - **Fail** - **Not Ready** - **Ready** - **Skipped** - **Info Received** You specify **Not Ready** when the requirement is in the **Active** state, and you specify **Ready** when the requirement is in the **Resolved** state. Reference Name=Microsoft.VSTS.CMMI.UserAcceptanceTest	String	None
Subject matter experts	The names of one to three team members who are familiar with the customer area that this requirement represents. You can specify the names of valid team members only. Reference Name=Microsoft.VSTS.CMMI.SubjectMatterExpert1 ... Microsoft.VSTS.CMMI.SubjectMatterExpert3	String	None

Notes:

1. To change the menu selection, see Customize a picklist.

Related articles

- Index of work item fields

Review meeting field reference

10/4/2022 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The following fields track information and changes for review meetings. Your team can specify this kind of information by using the Review type of work item that is provided with the CMMI process.

None of these fields are reportable or indexed. For more information about data types, see Work item fields and attributes.

FIELD NAME	DESCRIPTION	DATA TYPE
Purpose	The purpose and focus of the meeting. Reference name=Microsoft.VSTS.CMMI.Purpose	HTML
Comments	Additional information that you want to record. Reference name=Microsoft.VSTS.CMMI.Comments	HTML
Minutes	The details of what the team discussed and agreed upon during the meeting. You can use this field to record what the team reviewed, what criteria the team applied, and what problems the team identified. Reference name=Microsoft.VSTS.CMMI.Minutes	HTML
Meeting Type	The meeting venue. You can specify one of the following values: - Meeting - Offline Reference name=Microsoft.VSTS.CMMI.MeetingType	String
Called Date	The date and time when the meeting is scheduled. Reference name=Microsoft.VSTS.CMMI.CalledDate	DateTime

FIELD NAME	DESCRIPTION	DATA TYPE
Called By	The name of the team member who scheduled the meeting. Reference name=Microsoft.VSTS.CMMI.CalledBy	String
Required Attendee 1 ... **Required Attendee 8**	The names of each team member who is required to attend the meeting. Reference name=Microsoft.VSTS.CMMI.RequiredAttendee1 ... Microsoft.VSTS.CMMI.RequiredAttendee8	String
Optional Attendee 1 ... **Optional Attendee 8**	The name of each team member who is invited but not required to attend the meeting. Reference name=Microsoft.VSTS.CMMI.OptionalAttendee1 ... Microsoft.VSTS.CMMI.OptionalAttendee8	String
Actual Attendee 1 ... **Actual Attendee 8**	The name of each team member who attended the meeting. Reference name=Microsoft.VSTS.CMMI.ActualAttendee1 ... Microsoft.VSTS.CMMI.ActualAttendee8	String

Related articles

Index of work item fields

Add, rename, and delete dashboards in Azure DevOps

10/4/2022 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Share progress and status with your team using configurable team or project dashboards. Dashboards provide easy-to-read, easy access, real-time information. At a glance, you can make informed decisions without having to drill down into other parts of your project.

Share progress and status with your team using configurable team dashboards. Dashboards provide easy-to-read, easy access, real-time information. At a glance, you can make informed decisions without having to drill down into other parts of your project.

When a project is first created, a default team and default team dashboard is created labeled *Overview*. You can customize this dashboard by adding widgets. Each widget provides access to one or more features or functions. To learn more about each widget, see Widget catalog.

Project and team dashboards

When you add a dashboard, you can choose to make it a project dashboard or one specific to a team. Use project dashboards to display information or status about the project or when you want to control who can edit the dashboard. Use team dashboards to focus information specific to a team.

> **NOTE**
> Project dashboards are owned by the person that created the dashboard. The owner can set permissions as to who can edit the dashboard. Team dashboards are owned by team administrators and can be edited by any member of the team. All dashboards can be viewed by members of the project. All widgets available to team dashboards are available for project dashboards. For team-specific widgets, if you aren't able to select a team through the widget, then the team defaults to the default project team.

Prerequisites

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Anyone with access to a project, including stakeholders, can view dashboards.
- To add, edit, or manage a team dashboard, you must have **Basic** access or greater and be a team admin, a project admin, or have dashboard permissions. In general, you need to be a team member for the currently selected team to edit dashboards.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Install or enable the Analytics Marketplace extension. Analytics widgets aren't available if Analytics isn't installed, enabled, or running.
- Anyone with access to a project, including stakeholders, can view dashboards.
- To add, edit, or manage a team dashboard, you must have **Basic** access or greater and be a team admin, a project admin, or have dashboard permissions. In general, you need to be a team admin for the currently

selected team to edit dashboards. Request your current team or project admin to add you as a team admin.

- You must be a member of a project. If you don't have a project yet, create one.
- If you haven't been added as a project member, get added now.
- Anyone with access to a project, including stakeholders, can view dashboards.
- To add, edit, or manage a team dashboard, you must have **Basic** access or greater and be a team admin, a project admin, or have dashboard permissions. In general, you need to be a team admin for the currently selected team to edit dashboards. Request your current team or project admin to add you as a team admin.

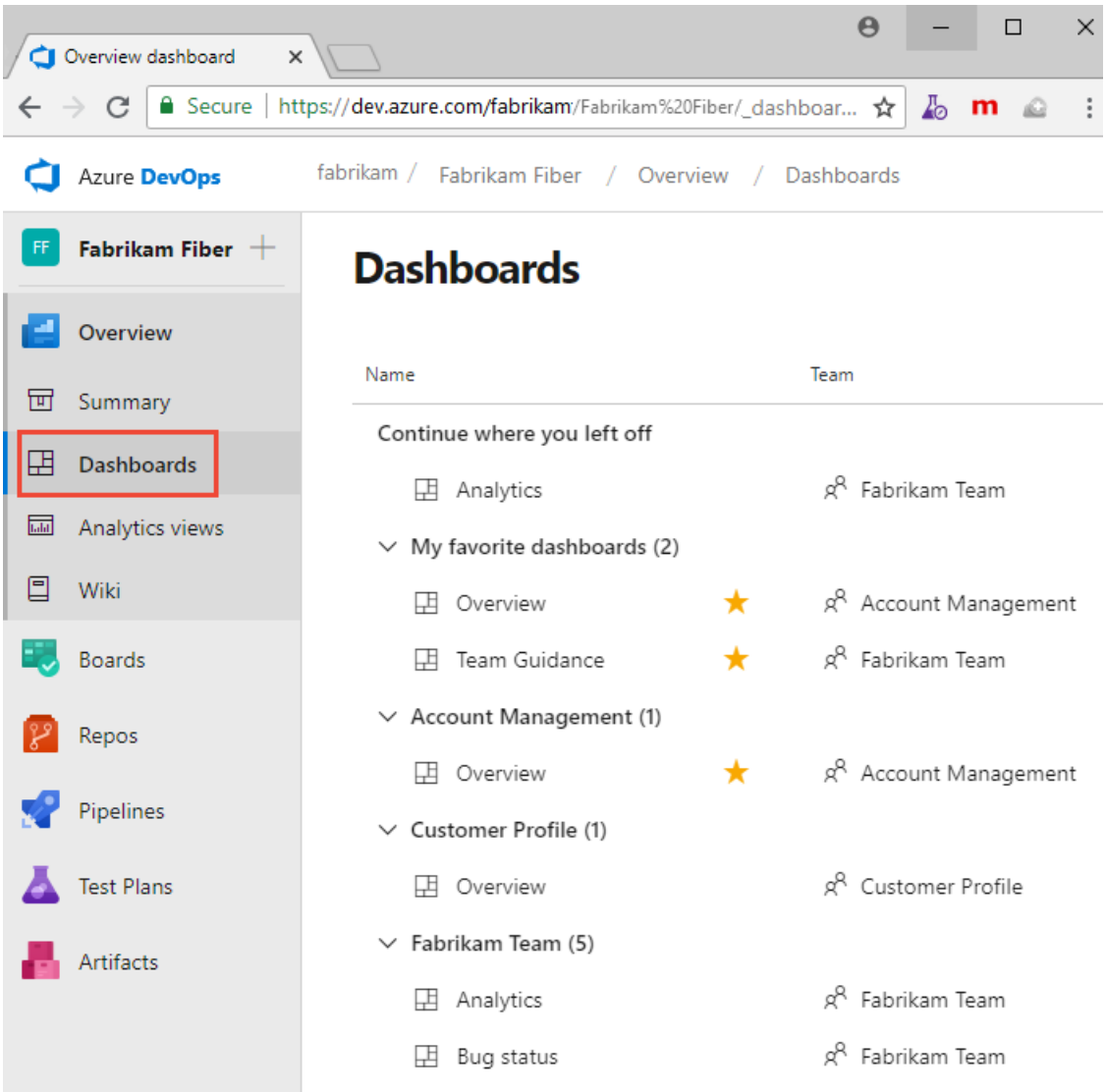
> **NOTE**
> Data displayed within a chart or widget is subject to permissions granted to the signed in user. To learn more, see FAQs on Azure DevOps dashboards, charts, and reports, Access and permissions.

Open Dashboards

All dashboards are associated with either a team or a project. From the **Overview > Dashboards** page, you can browse all dashboards and see which team they belong to, or if they are project dashboard.

All dashboards are associated with a team. From the **Overview > Dashboards** page, you can browse all dashboards and see which team they belong to.

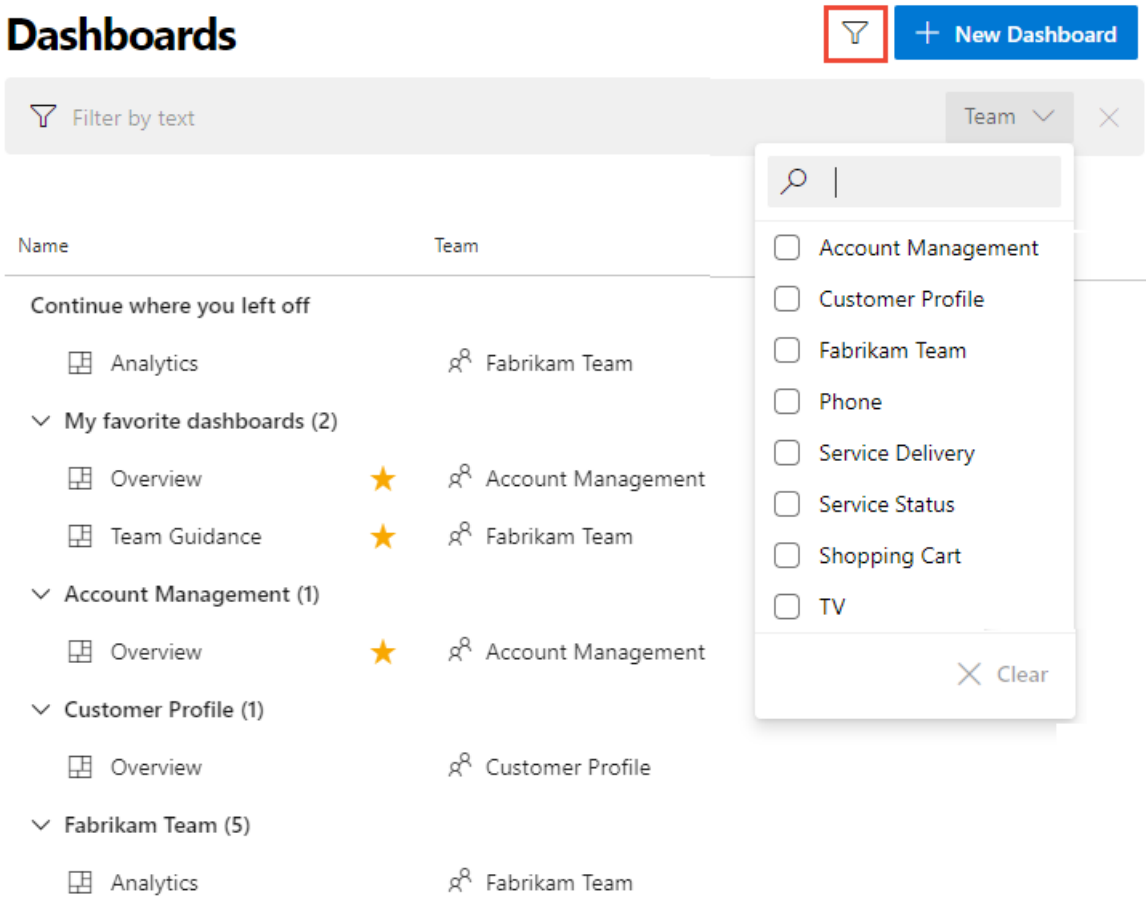
Open a web browser, connect to your project, and select **Overview > Dashboards**. The dashboard directory page opens.



It lists dashboards in the following order:

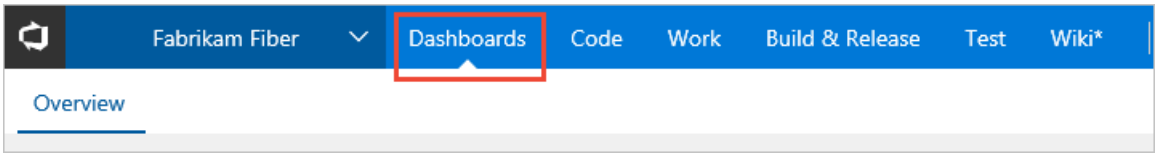
- Your last visited dashboard
- Your favorited dashboards
- All dashboards of teams that you belong to
- All dashboards defined for the project in alphabetical order.

Select the filter icon to filter the list by keyword or team. Keywords apply to dashboard titles, descriptions, and team names.



If you need to switch to a different project, select the Azure DevOps logo to browse all projects.

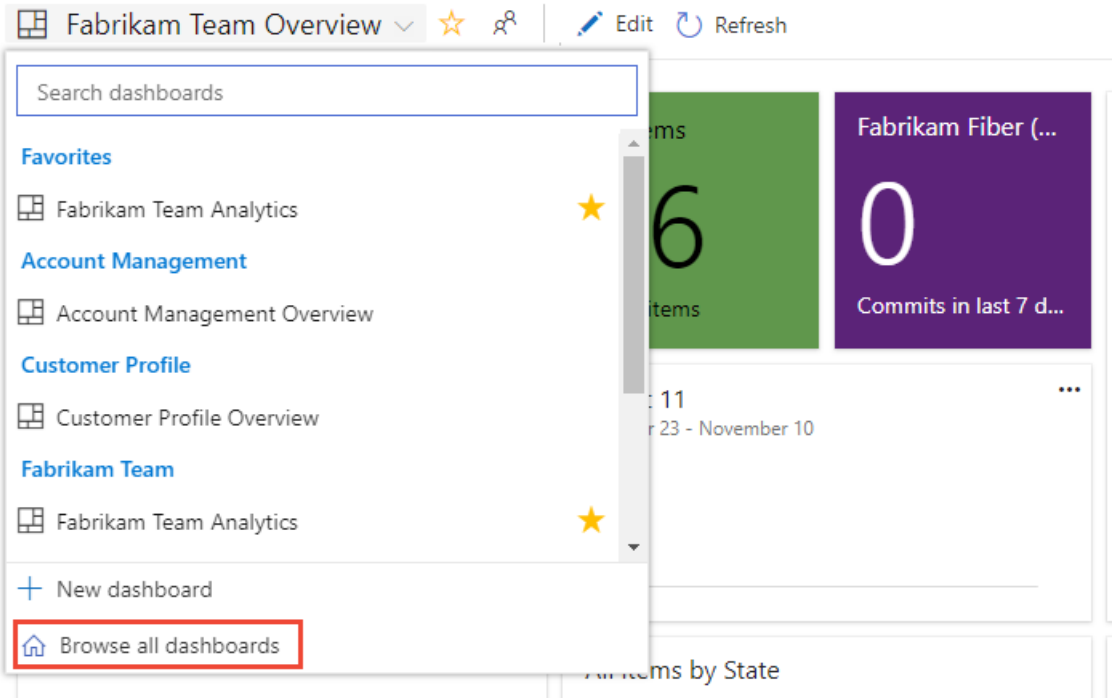
Open a web browser, connect to your project, and select **Dashboards**.



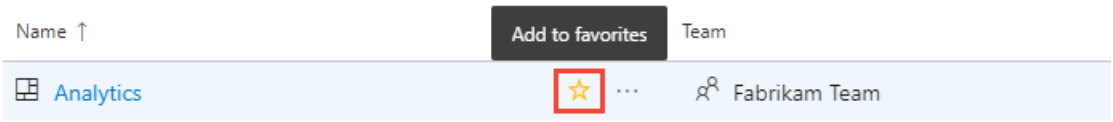
If you need to switch to a different project, select the Azure DevOps logo to browse all projects.

Select a dashboard

1. Select a dashboard from the directory list, or from the selector. To return to the dashboard directory, select the **Browse all dashboards** option.



2. To favorite a dashboard, hover over the dashboard and select the ☆.

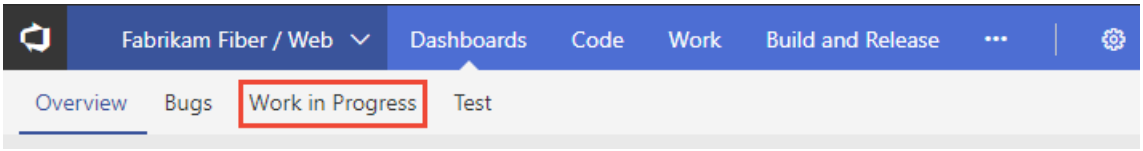


Favoriting a dashboard will cause it to appear under **My Favorites dashboards** list on the dashboards directory. Also, it will appear towards the top in the **Dashboards** selector and in your personal Favorites list.

1. Select the team whose dashboards you want to view. To switch your team focus, see Switch project or team.

2. From **Dashboards**, select the name of the dashboard to view it.

   For example, here we choose to view the Work in Progress dashboard.

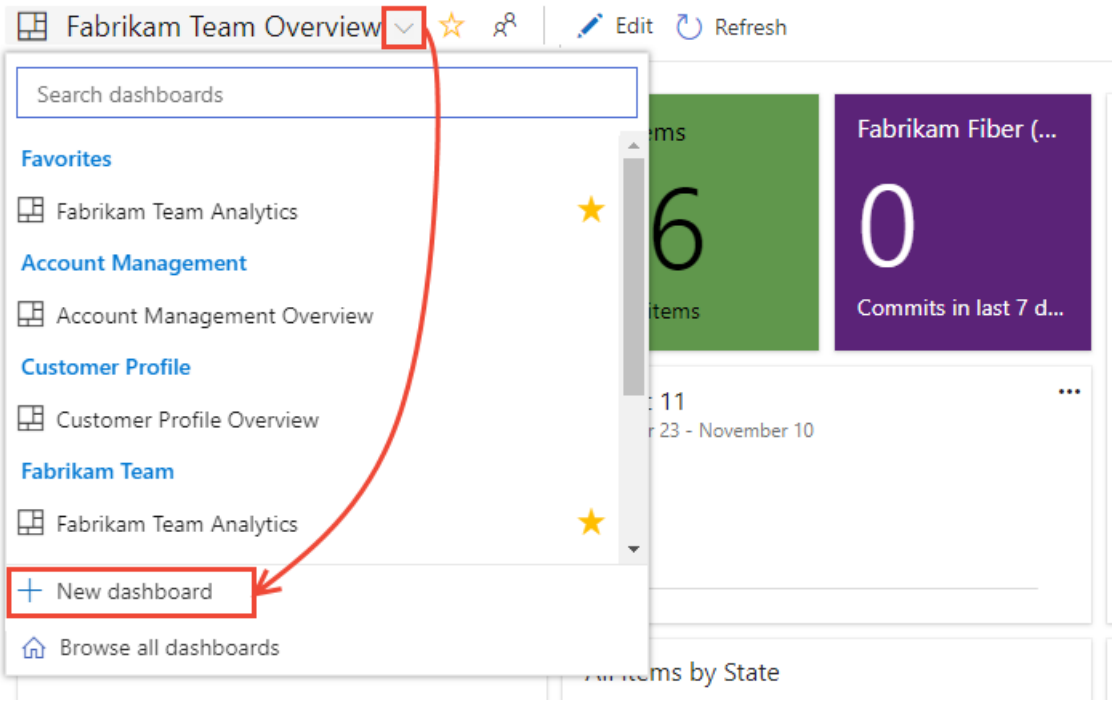


Add a dashboard

Add a new dashboard as needed to support your team's needs. You can also edit and rename any existing dashboards associated with your team.

> **NOTE**
> There is a limit of 500 dashboards per project. You'll receive an error message if you try to create a dashboard beyond that limit. Delete unused dashboards to resolve the error.

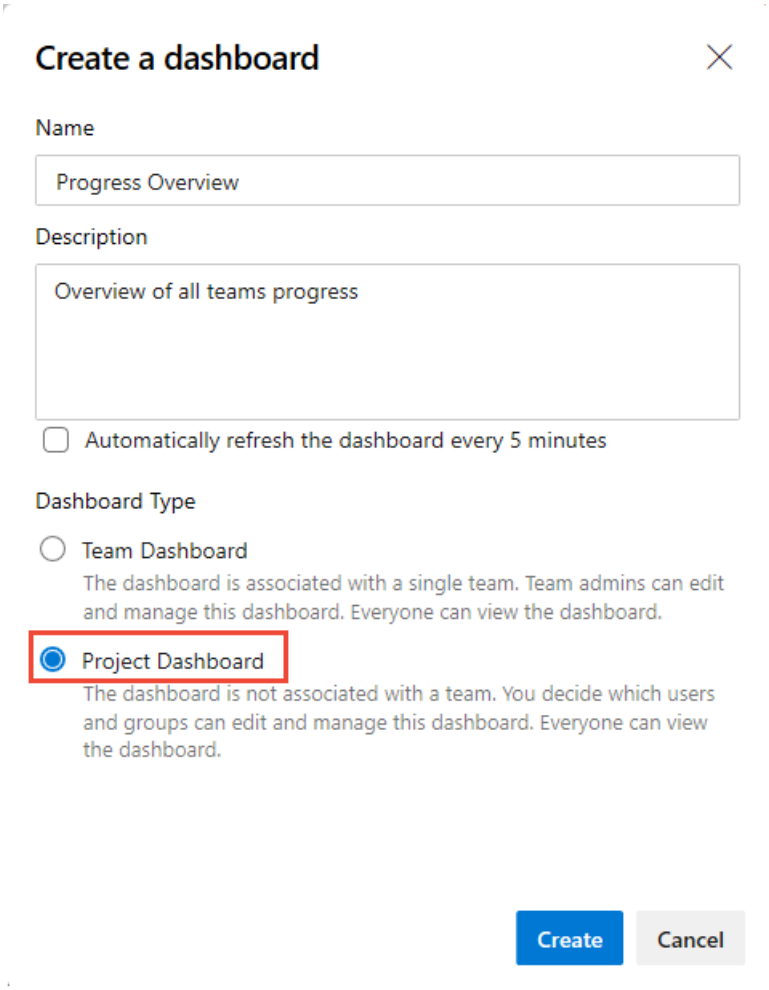
1. From the Dashboards directory, select **New Dashboard**. Or, when viewing a dashboard, open the selector and select the + **New Dashboard** option.



If you don't see the + **New Dashboard** option, then you're not a team admin for the currently selected team, or you don't have permissions to add and edit dashboards. Either switch the context to your team, or request you be added as a team admin.

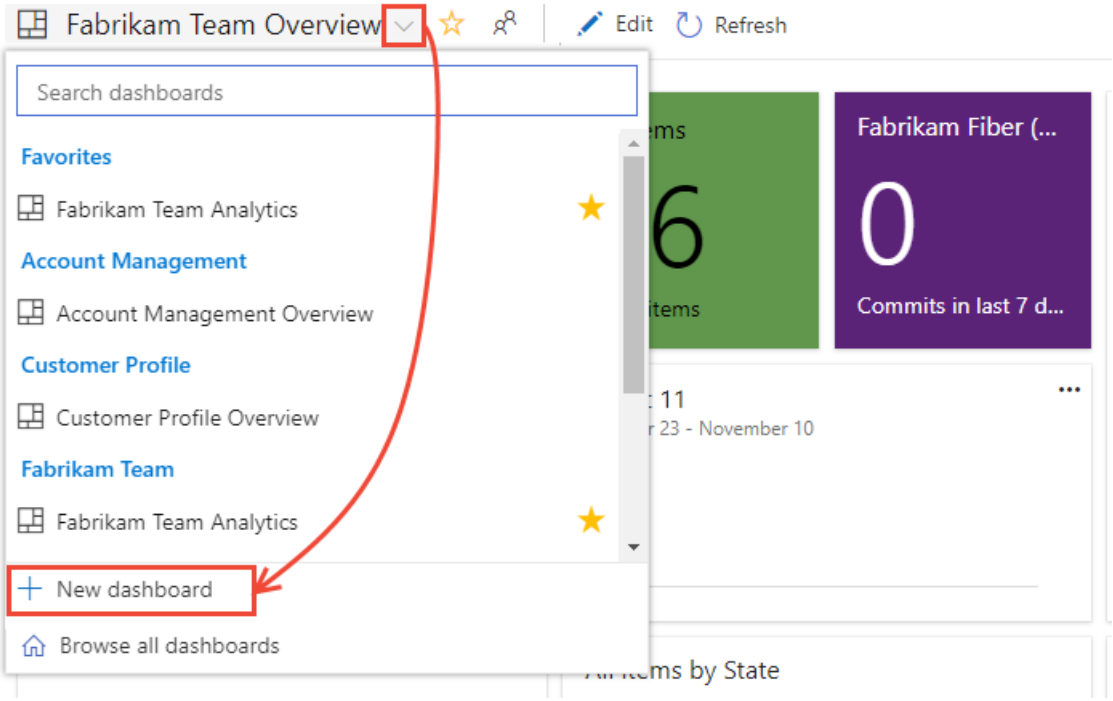
2. Enter the name of the dashboard and other information you want to capture.

   Here we choose to create a Project dashboard. To create a team dashboard, select **Team Dashboard** and then select a team. To add a team, see Add a team.



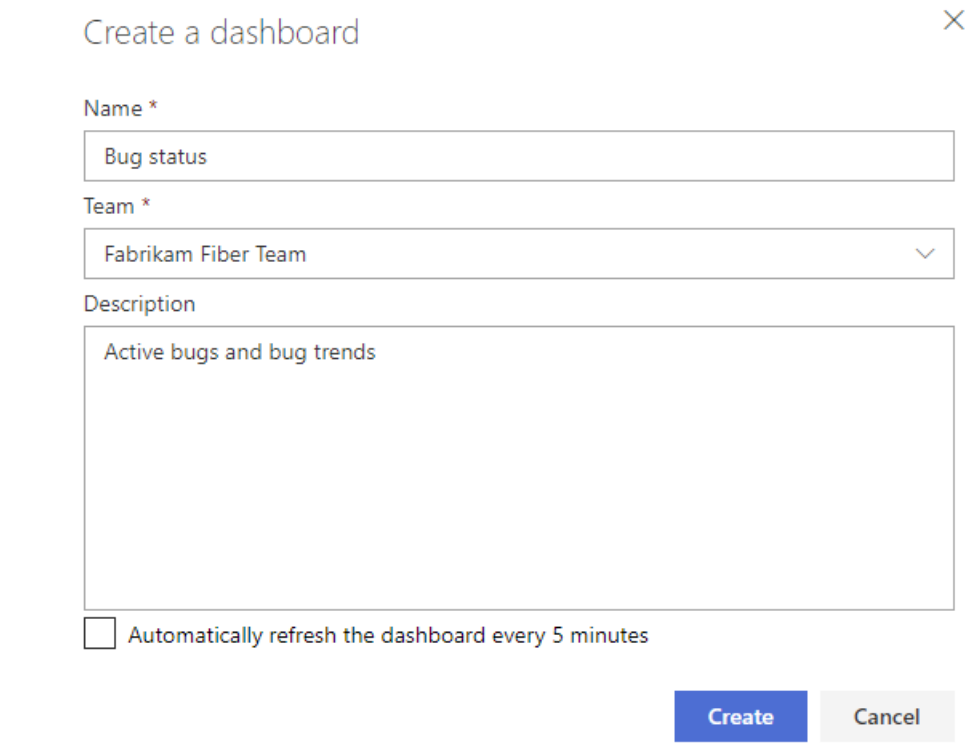
3. Select **Save**.
4. The widget catalog opens. You can add one or more widgets to the dashboard. You can then configure and resize each widget as needed.
5. You can move the widgets around the dashboard to place them where you want them.
6. When you're done making changes, select **Done Editing**.

1. From the Dashboards directory, select **New Dashboard**. Or, when viewing a dashboard, open the selector and select the + **New Dashboard** option.



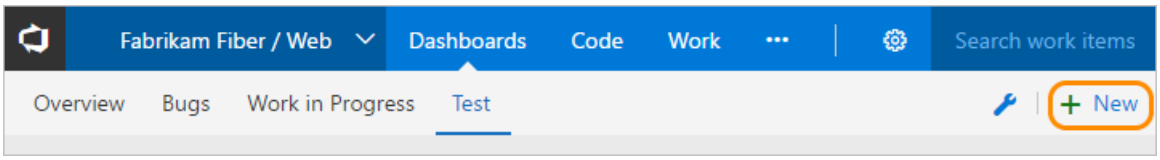
If you don't see the + **New Dashboard** option, then you're not a team admin for the currently selected team, or you don't have permissions to add and edit dashboards. Either switch the context to your team, or request you be added as a team admin.

2. Enter the name of the dashboard and other information you want to capture.



3. Select **Save**.
4. The widget catalog opens. You can add one or more widgets to the dashboard. You can then configure and resize each widget as needed.
5. You can move the widgets around the dashboard to place them where you want them.
6. When you're done making changes, select **Done Editing**.

From **Dashboards**, select **+ New Dashboard** and enter a dashboard name.



If you don't see **+ Edit** icon, then you're not a team admin for the currently selected team, or you don't have permissions to add and edit dashboards. Either switch the context to your team, or request you be added as a team admin.

With the dashboard selected, you can add widgets and charts to the dashboard. Or, you can add charts to a team dashboard from the Work, Build, or Test pages.

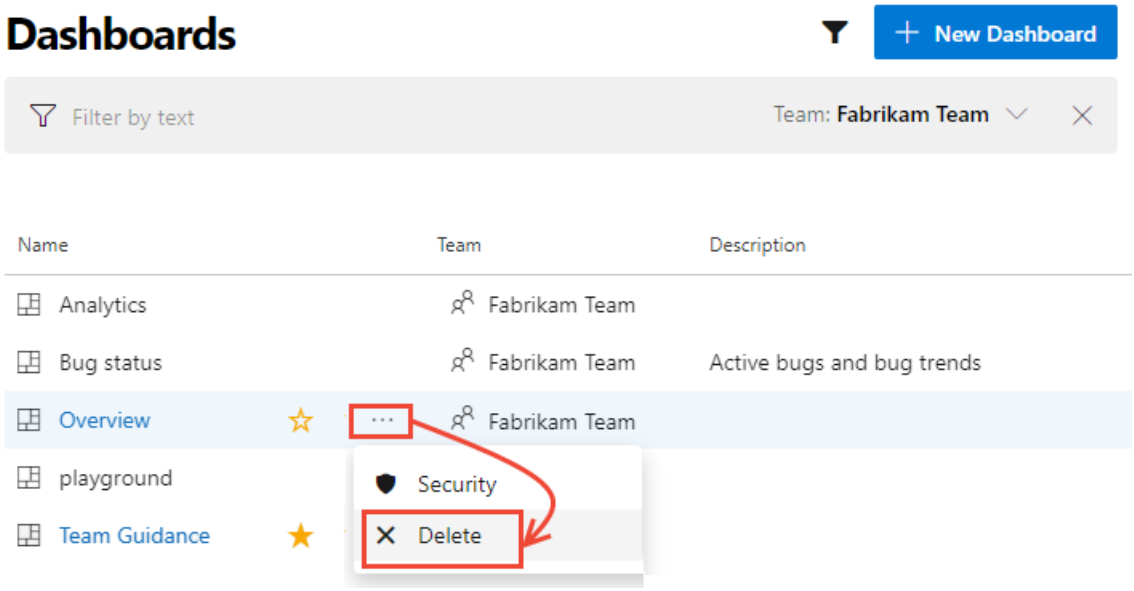
Rename, delete, and enable auto-refresh

You can rename or delete a dashboard. When you enable auto-refresh, the dashboard automatically updates every 5 minutes.

> **NOTE**
> To delete a project dashboard, you must be a member of the**Project Collection Administrators** group.

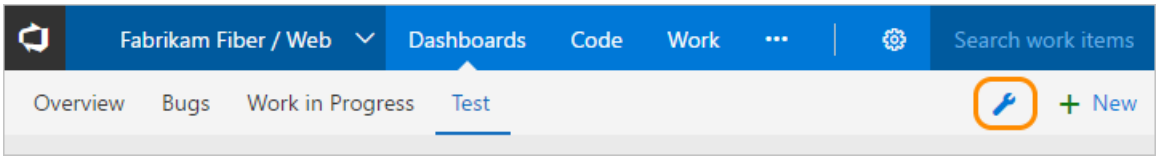
- To rename a dashboard, modify its description, or change its automatic refresh setting, open the dashboard, select the gear icon, and change the field options shown. Save your changes.

- To delete a dashboard, open the Dashboards directory, select ··· **More Actions** for the dashboard, and select **Delete**.

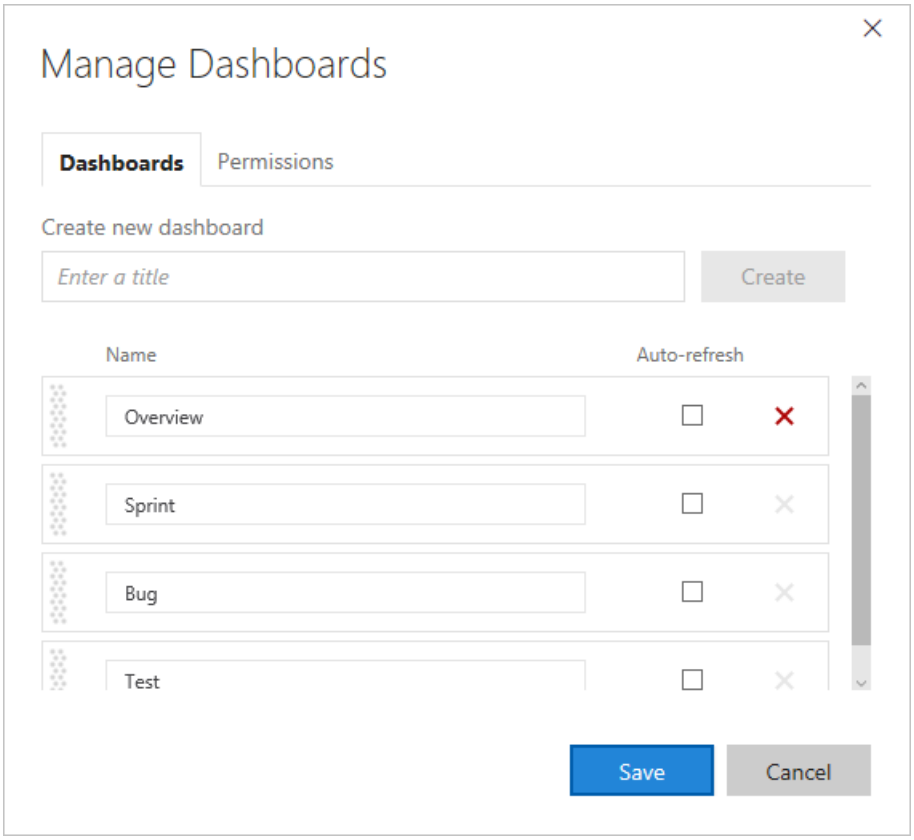


- To set permissions for a dashboard, select the **Security** option. For details, see Set dashboard permissions.

1. To manage dashboards, select the wrench icon.



2. Drag and drop the dashboards into the sequence you want them to appear.



3. (Optional) Select the Auto-refresh checkbox when you want the dashboard to refresh every five minutes.

4. To delete a dashboard, select the delete icon.

5. Select **Save** to save your changes.

You can also manage dashboard permissions.

Move or delete a widget

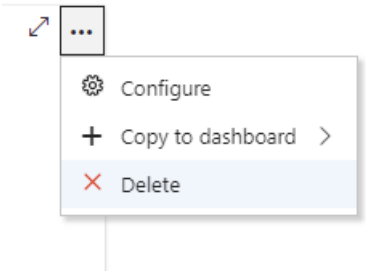
Just as you have to be a team admin, a project admin, or have the necessary permissions to add items to a dashboard, you must have the necessary permissions to remove items.

> **TIP**
> When you're in dashboard edit mode, you can remove, rearrange, and configure widgets, as well as add new widgets. Once you leave edit mode, the widget tiles remain locked, reducing the chances of accidentally moving a widget.

Select **Edit** to modify your dashboard.

You can now add widgets or drag tiles to reorder their sequence on the dashboard.

To remove a widget, select **More actions** and select **Delete** .



When you're finished with your changes, select **Done Editing** to exit dashboard edit mode.

Select  to modify your dashboard. You can then drag tiles to reorder their sequence on the dashboard.

You can now add widgets or drag tiles to reorder their sequence on the dashboard.

To remove a widget, select the widget's or delete icons.

When you're finished with your changes, select  to exit dashboard editing.

Extensibility

Using the REST API service, you can create a dashboard widget. To learn more about the REST APIs for dashboards and widgets, see Dashboards (API).

Try this next

As you can see, you can use team dashboards to provide guidance and keep your team in sync, providing visibility across your org about status, trends, and progress.

Add a widget to a dashboard

Related articles

- FAQs on Azure DevOps dashboards, charts, and reports
- Add a team
- Widget catalog
- Marketplace widgets

Tools and clients that connect to Azure DevOps

10/4/2022 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Our platform of software development tools began more than 20 years ago. We released Visual Basic and Visual Studio as an integrated development environment (IDE). Visual Studio supports many plug-ins that extend its functionality. In particular, the Team Explorer plug-in allows the Visual Studio client to connect to Azure DevOps to support source control, work tracking, build, and test operations.

Desktop client developer tools

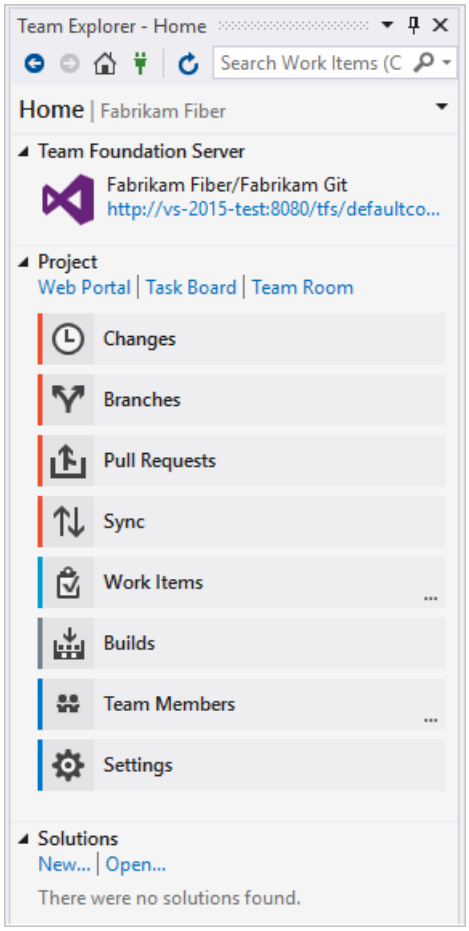
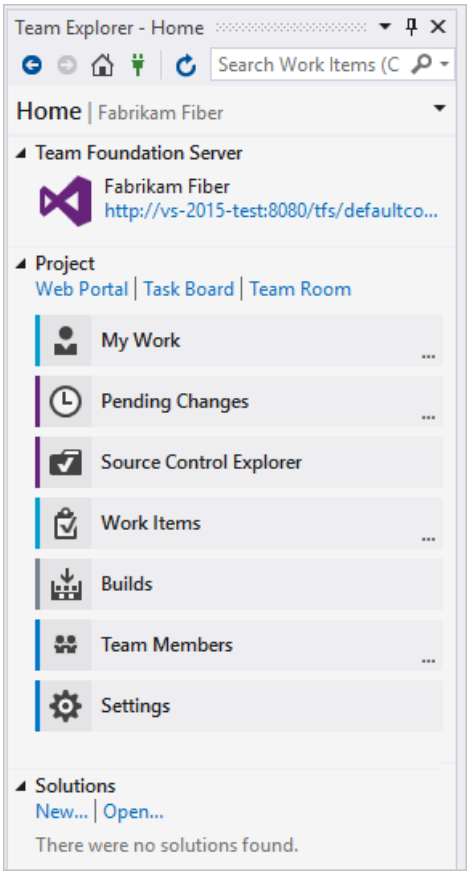
Developers have access to many tools through these versions of Visual Studio and plug-ins. To download any version of Visual Studio, go to the Visual Studio Downloads page. To understand what features you get with the Visual Studio versions, see Compare Visual Studio offerings.

- **Visual Studio Community**: A fully featured and extensible IDE for creating modern applications for Android, iOS, and Windows, including web applications and cloud services. (Replaces Visual Studio Express.)
- **Visual Studio Professional**: Development tools and services to support individual developers or small teams.
- **Visual Studio Enterprise**: Integrated, end-to-end development tools and solutions for teams of any size, and with a need to scale. It supports designing, building, and managing complex enterprise applications.
- **Visual Studio Test Professional**: Provides access to Microsoft Test and development tools to support quality and collaboration throughout the development process.
- **Visual Studio Team Explorer**: Free solution for non-developers to interact with Azure DevOps.
- **Eclipse/Team Explorer Everywhere**: Free plug in to support teams running Eclipse on Linux, macOS, or Windows that connects to Azure DevOps.
- **Android Studio with the Azure DevOps Services Plug-in for Android Studio**: Free plug in to support Android developers and connect to Git repositories on Azure DevOps.
- **IntelliJ with the Azure DevOps Services Plugin for IntelliJ**: Free plug in to support developers who use IntelliJ IDEA or Android Studio to connect to Git repositories on Azure DevOps.
- **Visual Studio Code**: Free, open-source code editor with a free extension to support connecting to Git repositories on Azure DevOps.

To get started with client libraries, see Client library samples.

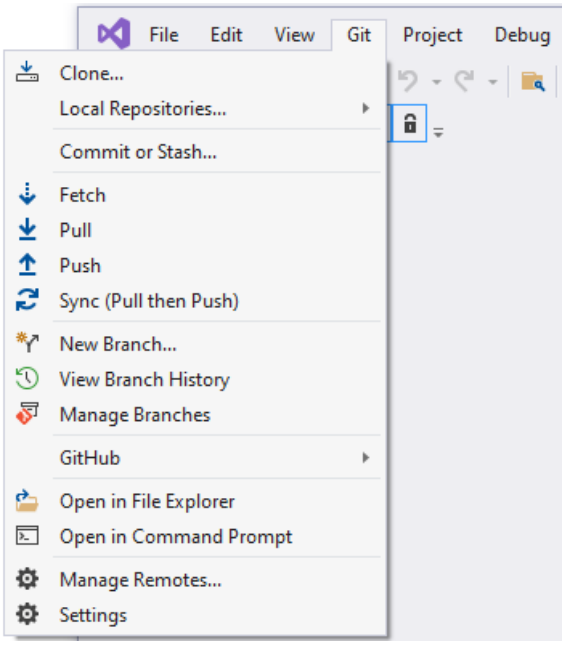
Team Explorer plug-in

Team Explorer, a plug-in to all Visual Studio versions, connects Visual Studio to projects defined in Azure DevOps. You can manage source code, work items, and builds. To learn more, see Work in Team Explorer.

HOME PAGE WITH GIT	HOME PAGE WITH TFVC
	

Visual Studio Git experience

Visual Studio 2019 and later versions provide a new Git experience through the **Git** menu as shown below. To learn more, see Git experience in Visual Studio and Side-by-side comparison of Git and Team Explorer.



Office integration tools

You can integrate the following Microsoft Office tools with Azure DevOps.

- Excel: Use Excel to add and bulk modify work items.

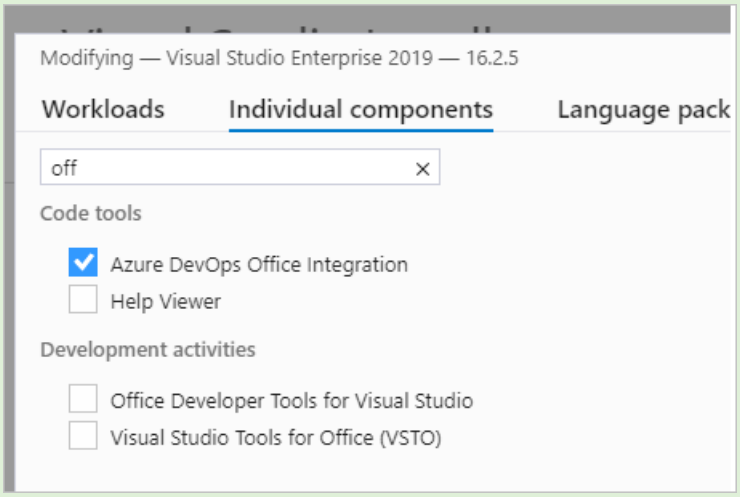
IMPORTANT

Starting with Visual Studio 2019, the Team Foundation plug-in for Office is deprecating support for Microsoft Project. Project integration and the TFSFieldMapping command is not supported for Azure DevOps Server 2019 nor for Azure DevOps Services. However, you can continue to use Microsoft Excel.

- Excel: Use Excel to add and bulk modify work items.

TIP

Check to make sure the Azure DevOps Office Integration component is selected in the Visual Studio Installer, per the following example.



When you install any edition of Visual Studio or Team Foundation Server Standalone Office Integration 2015 (free), the Team Foundation plug-in integrates work item tracking with select Office clients. The Team Foundation plug-in installs to your existing Office client. The plug-in supports Office 2007, Office 2010, or Office 2013 versions.

- Excel: Use Excel to add and bulk modify work items.
- Project: By using Project, you can plan projects, schedule tasks, assign resources, and track changes. You have access to features that TFS doesn't support, such as a project calendar, Gantt charts, and resource views.
- PowerPoint Storyboarding: Illustrate user stories and requirements by using PowerPoint. The Team Foundation plug-in installs to your existing PowerPoint client.

Task-specific clients

The following clients support specific tasks, such as managing testing efforts, providing feedback, or modifying work items:

- Azure Test Plans: Manage your test efforts, create and run manual tests, and create and track bugs that are found during test efforts.
- Test & Feedback extension (previously called the Exploratory Testing extension): This extension provides a lightweight plug-in to a web browser. Stakeholders can respond to feedback requests for user stories and features created in Azure DevOps. This extension is free to Stakeholders.
- Microsoft Feedback Client: Your Stakeholders can use this client to record feedback for your application as video, audio, or type-written comments. This client is installed with all versions of Visual Studio, or it can be installed from the free download. All feedback is stored in the work item data store and requires Stakeholders to have permissions.

Browser-based web tools

Web portal

The collaboration tools supported through the web portal are summarized under Essential services. New features are deployed every three weeks for Azure DevOps Services, and quarterly for Azure DevOps Server. For release notes, see Azure DevOps Services Features Timeline.

You can use the following browsers to access the web portal:

VERSION	EDGE	INTERNET EXPLORER	SAFARI (MAC)	FIREFOX	CHROME
Azure DevOps Services Azure DevOps Server 2020.1	Most recent	Not supported	14.1 and later	Most recent	Most recent
Azure DevOps Server 2020 Azure DevOps Server 2019 TFS 2018 TFS 2017	Most recent	11 and later	14.1 and later	Most recent	Most recent
TFS 2015	Most recent	9 and later	5 and later	Most recent	Most recent
TFS 2013		9 and later	5 and later	Most recent	Most recent

Microsoft Edge, Firefox, and Chrome automatically update themselves, so Azure DevOps supports the most recent version.

For more information, see Web portal navigation.

Browser-based extensions

Several extensions are built and maintained by the Azure DevOps Services product team:

- Code search: Increase cross-team collaboration and code sharing. Enables developers to quickly locate relevant information within the code base of all projects that are hosted within an organization or collection. You can discover implementation examples, browsing definitions, and error text.
- Work item search: To quickly find relevant work items, search across all work item fields over all projects in an organization. Do full-text searches across all fields to efficiently locate relevant work items. Use inline search filters, on any work item field, to quickly narrow down a list of work items.

Find more extensions in Azure DevOps **Organization settings** > **Extensions** > **Browse marketplace**. See also, Overview of extensions for Azure Boards.

Command-line tools

You can do many code development and administrative tasks by using the following command-line tools:

- az devops commands
- Git commands
- TFVC commands
- TCM commands
- Manage permissions with command line tool (az devops security)
- witadmin (work item tracking)

- Git commands
- TFVC commands

- TCM commands
- witadmin (work item tracking)
- TFSConfig
- TFSDeleteProject
- TFSSecurity
- TFSServiceControl

Integrated tool support for third-party applications

The following tools provide support for monitoring and interacting with Azure DevOps from a third-party application.

- **Azure Boards**:
  - Use the Azure Boards app with Slack to manage work items
  - Use the Azure Boards app in Microsoft Teams
- **Azure Repos**:
  - Azure Repos with Slack
  - Azure Repos with Microsoft Teams
- **Azure Pipelines**:
  - Use Azure Pipelines with Microsoft Teams
  - Azure Pipelines with Slack
  - Integrate with ServiceNow change management
  - Continuously deploy from a Jenkins build

Marketplace extensions

Visual Studio and Azure DevOps provide a wealth of features and functionality. They also provide a means to extend and share that functionality.

Extensions are simple add-ons that you can use to customize and extend your DevOps and work tracking experiences. They're written with standard technologies—HTML, JavaScript, and CSS. You can develop your own extensions by using your preferred dev tools.

You build extensions by using our RESTful API library. Publish your extensions to the Azure DevOps Marketplace. You can privately maintain or share them with millions of developers who use Visual Studio and Azure DevOps.

To learn more, visit the Azure DevOps Marketplace and see Overview of extensions.

REST APIs

The Azure DevOps APIs are based on REST, OAuth, JSON, and service hooks—all standard web technologies broadly supported in the industry.

REST APIs are provided to support building extensions to Azure DevOps. To learn more, see REST API overview.

Related articles

- A tour of services
- Software development roles
- Pricing
- Azure DevOps data protection overview