Contents

- Work items
- Overview
  - About work items
  - What is Azure Boards?
  - Best tool for the job
  - Start using Azure Boards
- Quickstarts
  - View & add work items
  - Add & update a work item
  - Drive Git development
  - Define area paths
  - Define iteration paths
- Tutorials
  - Follow work
  - Manage bugs
  - Manage issues
  - Move, change, or delete
  - Link work items
  - Bulk modify (web)
  - Copy/clone
  - Work as a Stakeholder
- Concepts
  - Choose a process
  - Agile process guidance
    - Agile workflow
  - Scrum process guidance
    - Scrum workflow
  - CMMI process guidance
    - CMMI workflow

- CMMI background
- Project management
  - Project activities
  - Project inception
  - Plan a project
  - Manage change
  - Manage risk
  - Iteration activities
  - Plan an iteration
  - Manage issues
- Engineering
  - Develop requirements
  - Arrange requirements into a product plan
  - Create a solution architecture
  - Implement development tasks
  - Build a product
  - Track bugs
- Work item fields
- Agile glossary
- Linking & traceability
- Workflow states & categories
- New work item experience
- Samples
- Query by title or ID
- Query by assignment or workflow
- Query by area or iteration
- Query history
- How-to Guides
- Use @mentions in discussions
- Use #ID to link to work items
- Add tags
- Share information in work items

- Bulk add or modify (Excel)
- Use work item templates
- Go mobile
- Email/print work items
- Delete test artifacts
- Change column options
- Switch project or team focus
- Set the Work Items experience (Visual Studio)
- Manage permissions (Security)

Reference

- Work item form controls
- Keyboard shortcuts
- Link types
- Define a work item URL
- Work item field index
  - Code review & feedback fields
  - Bugs, issues, & risks fields
  - Change request fields
  - Requirements fields
  - Review meeting fields
- Default permissions (Security)
- Changes made to process templates
- Client support for work tracking
- REST API, Work Items

Resources

- Boards (Kanban)
- Backlogs
- Sprints (Scrum)
- Queries
- Customization
- What is Agile?

Work items

1/31/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Track the features and requirements you're developing, code defects or bugs, and other particulars using work items.

5-Minute Quickstarts

- View and add work items
- Add work items
- Drive Git development

- Add work items
- Drive Git development

- Add work items

Step-by-Step Tutorials

- Follow work
- Manage bugs
- Manage issues
- Move, change, or delete items
- Link work items
- Bulk modify (web)

- Manage bugs
- Manage issues
- Remove or delete items
- Link work items
- Bulk modify (web)

Concepts

- Choose a process
- Agile process guidance
- CMMI process guidance
- Scrum process guidance
- Agile glossary

How-to Guides

- Use @mentions to further discussion
- Use #ID to link to work items
- Add tags to work items
- Use work item templates

- Add tags to work items
- Use work item templates

Reference

- Permissions and access for work tracking
- Work item form controls
- Keyboard shortcuts for work item forms & the Work Items page
- Work item field index

- Permissions and access for work tracking
- Work item field index

Resources

- Backlogs
- Kanban
- Scrum
- Queries
- Customization

Track work with user stories, issues, bugs, features, and epics

2/7/2019 • 13 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 an identifier which is unique within an organization or project collection. The work item types available to you are based on the process used when your project was created (Basic, Agile, Scrum, or CMMI).

In a nutshell:

- You use different work item types (WITs) to track different types of information. Specific tools
  - Add backlog items, such as Issues (Basic process), User Stories (Agile), Product Backlog Items (Scrum), Requirements (CMMI)
  - Define Features and Epics
  - Define Features and Epics
  - Define, triage, and manage Bugs
  - Add Tasks to backlog items
- You update the work item form to add information, update status, reassign to another project member or sprint, and to link work items, attach files, and add comments
- You can add and modify work items using the web portal and other supported clients
- You can assign a work item to one and only one project member
- You assign work items to a sprint via the iteration path
- You can use ad hoc search or queries to find or list work items
- You can use work item templates to quickly fill in work item fields
- Teams can choose how they track bugs, either as requirements or as tasks
- You or a project administrator can add and customize work item types
- To add or modify work items, you need to be a project member with Contributor permissions and Basic or Stakeholder access.

- You use different work item types (WITs) to track different types of information, such as user stories, bugs, and tasks
- You update the work item form to add information, update status, reassign to another project member or sprint, and to link work items and attach files
- You can add and modify work items using the web portal and other supported clients
- You can assign a work item to one and only one project member
- You assign work items to a sprint via the iteration path
- Teams can choose how they track bugs, either as requirements or as tasks
- You can use ad hoc search or queries to find or list work items
- You can use work item templates to quickly fill in work item fields
- You can add and customize work item types and your backlog
- To add or modify work items, you need to be a project member with Contributor permissions and Basic or Stakeholder access.

Work item types (WITs)

To track different types of work, different WITs are defined. The WITs available to you differ depending on the process or process template used to create your project.

For example, the following WITs are available to you when you choose the Agile process.

> **NOTE**
>
> The WITs available to you depend on the process chosen to create your project—Basic, Agile, Scrum, or CMMI. The items in your backlog may be called user stories, product backlog items (PBIs), or requirements. All three are similar: they describe the customer value to be delivered and the work to be performed.

To learn more about processes and process templates, see Choose a process.

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. To learn about the different tabs and sections of the form, see Work item form controls.

Each work item supports tracking data contained in work item fields. Also, it captures changes as updates are made within the **History** field. To learn more about each field, see Work item field index.

Each form contains a number of controls as shown below and described in Work item form controls.

The new form and its corresponding features are available from the web portal. The new form is automatically available when you add projects to a new collection. For existing projects, an admin is required to enable the new form. To learn about the different tabs and sections of the new form, see Work item form controls.

New web form

The new web form provides a number of experiences not provided with the old web form. To learn more, see New work item experience.

Track features, requirements, code defects, tasks, and issues using form-specific work item types
Unique identifier assigned by the system
Assign work to team members
Track the status of work as it progresses from unassigned, to in progress, to done
Additional tasks available through the Actions menu
Attach files to the work item.
Link work item to other work items, code changes, pull requests, and other objects
History maintains an audit trail of all changes
USER STORY 643*
View/add to Discussion
643 Cancel order form
Add tags
Jamal Hartnett
2
Add Tag
Save & Close
Follow
State
Active
Area
Fabrikam Fiber
Updated by Raisa Pokrovskaya 11/3/2015
Reason
Implementation...
Iteration
Fabrikam Fiber
Details
Description
Provide a cancellation order from similar to the screen shown
See the attached storyboard for details.
Order details here
Cancel
Planning
Story Points
Priority
2
Risk
Development
Add link
Development hasn't started on this item.
Related Work
Add link
There are no links in this group.
Classification
Value area
Business
Use to estimate work, build velocity charts, and forecast
Rich-text format toolbar appears once you click within the box
Acceptance Criteria
Discussion
Add a comment
Add and review comments, use @mention to pull someone into the discussion

Old web form

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 Services and TFS.

Web portal

Use the web portal to accomplish the following tasks.

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

- **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 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.

Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

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 an 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 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, then you should consider creating additional work items that you'll assign to each user responsible for the work to be completed
- 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. To learn more, see Access with Azure Active Directory (Azure AD).

Integration with Active Directory

When TFS is configured with Active Directory (AD), then TFS 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 AD or by adding accounts to existing or custom groups defined from the collection setting **Security** pages. To learn more, see Set up groups for use in TFS 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 Apply a field rule.

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.

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 therefore 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 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.

Find or list work items

You can use the search box to perform an ad hoc search to find specific work items based on select field criteria. Or, you can create a query to perform a managed search which will list work items based on your query criteria. With managed searches you can perform a number of other tasks, such as to triage work items, create a trend or status chart and add to the dashboard, and more.

To learn more, see these topics:

- Create managed queries to list, update, or chart work items
- View, run, or email a query
- Ad hoc vs 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 which have pre-populated values for your team's commonly used fields. For example, you can create a task template that will set 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 WIT

You can add or modify the fields contained within a WIT or add a custom WIT. 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 add or modify the fields contained within a WIT or add a custom WIT. To learn more, see Customize the On-premises XML process model.

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 Work as a Stakeholder.

To learn more about permissions and access, see Permissions and access for work tracking and About access levels.

To add users to a project, see Add users to a project or team.

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 for work item forms and the Work Items page
- Agile, Scrum, and CMMI processes
- Work item field index

What is Azure Boards?

1/31/2019 • 10 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

With Azure Boards, you can quickly and easily start tracking user stories, backlog items, task, features, and bugs associated with your project. You do this by adding one of three work item types—user stories, bugs, features, epics, and tasks—that the Agile process provides. As works progresses from not started to completed, you update the State workflow field from **New**, **Active**, **Resolved**, and **Closed**.

WORK ITEM TYPES	WORKFLOW

When you add a user story, bug, task, or feature, you create a work item. Each work item represents an object stored in the work item data store. Each work item is assigned a unique identifier (ID) within your projects.

> **NOTE**
>
> Your first project corresponds to the Agile process unless Scrum or CMMI were selected. If you want a project that uses the Scrum or CMMI process, then you can add another project and specify the process through advanced setting options. See Choose a process for a comparison of processes.

Add epics to track significant business initiatives, add features to track specific applications or set of work. Define user stories to track work that you'll assign to specific team members, and bugs to track code defects. Lastly, use tasks to track even smaller amounts of work for which you want to track time either in hours or days.

> **IMPORTANT**
>
> **Select the version that meets your location and process**: We are experimenting with a new acquisition model which is currently available for users located in the United States and that sign up through azure.com/boards. This model supports a new Basic process.
> For International users and others who sign up through another method, the Agile process is used. Select your version of this article based on your location and process used.
>
> - **Basic process**
> - **Agile process**

Track work, implement Kanban

- Quickly add and update the status of work you track from **New**, **Active**, **Resolved**, and **Closed** using the Kanban board. Add tasks as child items to user stories. Learn more: Track user stories, features, and tasks.
- Assign work to team members, tag with labels to support queries and filtering, or share information through descriptions, attachments, or links to network shared content.
- Prioritize work through drag-and-drop on your team backlog. Learn more: Create your backlog.

Update the status of issues

Prioritize your backlog of issues

Collaborate

- Collaborate with others through the **Discussion** section of the work item form, using **@mention**s and **#ID** controls to quickly include others in the conversation or link to other work items. Learn more: Discussion section.
- Create dashboards that track status and trends of work being accomplished. Learn more: Add and manage dashboards.
- Set notifications to get alerted when an issue is created or changed. Learn more: Set personal notifications.
- Elect to follow specific issues to get alerted when they are updated. Learn more: Follow a work item.

Get updated when a work item is updated

Work in sprints, implement Scrum

- Plan sprints by assigning work to current or future sprints. Learn more: Assign work to sprints.
- Forecast work that can get completed based on effort estimates: Learn more: Forecast work.
- Determine how much work can be done within a sprint. Learn more: Set team capacity.
- Bulk assign issues and tasks to team members and sprints. Learn more: Bulk modify work items.

Assign backlog items to a sprint

Work effectively

- Organize work into a hierarchy by grouping issues under epics, and tasks under issues. Learn more: Organize your backlog.
- Create queries and quickly triage issues and tasks. Learn more: View and run queries and Triage work.
- Create work item templates to help contributors quickly add and define open meaningful issues and tasks. Learn more: Use work item templates.
- 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. Learn more: View and add work items.

Group items to create a hierarchy

Connect with GitHub

If you use Azure Boards connected with GitHub, you can also do the following:

- From GitHub, use #AB to link GitHub commits and pull requests to your issues and tasks
- From Azure Boards issues and tasks, link to GitHub commits and pull requests.

Learn more: GitHub & Azure Boards.

Best tool for the job

Azure Boards provides the following main interactive lists and signboards. Each tool provides a filtered set of work items. All tools support viewing and defining work items. To learn more about effective use of these tools, see Best tool to add, update, and link work items.

- **Boards**: Boards present work items as cards and support 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.
- **Backlogs**: Backlogs present 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. Use to plan, prioritize, and organize work.
- **Sprints**: Sprint backlogs and taskboards provide a filtered view of work items a team assigned to a specific iteration path, or sprint. From your backlog, you can assign work to an iteration path by using drag-and-drop. You can then view that work in a separate *sprint backlog*. Use to implement Scrum practices.
- **Queries**: Queries are filtered lists of work items based on criteria that you define by using a query editor. You

use queries to support the following tasks:

  - Find groups of work items with something in common
  - List work items for the purposes of sharing with others or performing bulk updatesTriage a set of items to prioritize or assign them
  - Create status and trend charts that you then can add to dashboards.
- **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.

Teams and Backlogs, Boards, and Sprints

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 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.

Install and manage extensions

An extension is an installable unit that adds new capabilities to your projects. You can find extensions in the Visual Studio Marketplace, Azure DevOps tab to support planning and tracking of work items, sprints, scrums, and more and collaboration among team members.

Best tool to add, update, and link work items

2/7/2019 • 6 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Azure Boards provides you several tools—many designed to perform 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 do that 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.

Additional 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 inception to completion
- Quickly adding product backlog and portfolio backlog items
- Updating the status of backlog items
- Adding linked child items (task checklists, user stories or features)
- Filtering work items to focus on different views
- 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

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

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

- Bulk modifying work items
- Assigning work items, change the iteration
- Adding or removing tags
- Restoring work items from the recycle bin
- Monitoring team velocity

- Bulk modifying work items
- Assigning work items, change the iteration
- Adding or removing tags
- Monitoring team velocity

Sprints

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 perform 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.

OFFICE APPLICATION	BEST TOOL FOR:
Excel	• Adding or updating many work items and their fields • Adding or changing hierarchical links between work items
Project	• Importing a project plan to a project • Adding tasks to Project and publishing them as work items • Creating or updating parent-child links or predecessor-successor links

Storyboarding with PowerPoint	• Storyboarding features and user interface changes • Linking your storyboard to a work item • Adding and sharing storyboard shapes

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 • Perform 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

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

TFS Power Tools

Provides you access to these additional tools through the Team Explorer plug-in for Visual Studio. Additional requirements may apply.

- Process Template Editor
- Additional check-in policies for Team Foundation Version Control
- Team Explorer enhancements including Team Members
- Team Foundation Power Tool Command Line
- Test Attachment Cleaner
- Work Item Templates

Related articles

- Navigate in the web portal
- Navigate in Team Explorer
- Configure a Burndown or Burnup widget

Start using Azure Boards (Basic process)

1/31/2019 • 2 minutes to read • Edit Online

Azure Boards

Use this guide to sign up and start using Azure Boards.

> **IMPORTANT**
>
> **Select the version that meets your location and process**: We are experimenting with a new acquisition model which is currently available for users located in the United States and that sign up through azure.com/boards. This model supports a new Basic process.
> For International users and others who sign up through another method, the Agile process is used. Select your version of this article based on your location and process used.
>
> - **Basic process**
> - **Agile process**

Start with Sign up and invite some teammates.

Then, read Track issues and tasks to start adding and tracking issues 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 based on the Basic process, and start tracking issues and tasks. 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.

If you use GitHub and want to track your issues in Azure Boards, see GitHub & Azure Boards.

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

Reference

- Basic fields reference
- Key concepts
- Best tool for the job
- Default permissions & access (Security)
- Why use Azure Boards?
- Connect a project to GitHub

Additional resources

- Web portal navigation
- Work items
- Sprints (Scrum)
- Process customization

View and add work items using the Work Items page

2/14/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | Visual Studio 2019

View work items that you created or are assigned to you. The **Work Items** page provides several personalized pivots and 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 currently available from Azure DevOps Services, Azure DevOps Server 2019 RC1, 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 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 About access levels.
- 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 create 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.

- 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 About access levels.
- 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 create 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.

Open Work Items

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

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance as Azure DevOps Server 2019 only supports the **New Navigation** user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation
- Visual Studio 2019

(1) Check that you have selected the right project, then (2) choose **Boards>Work Items**.

> **NOTE**
>
> The new Work Items experience is available when you connect to a GitHub repository. If you connect to a TFVC repository, you'll continue to see the legacy query-focused experience.

> **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.

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.

- New navigation
- Previous navigation
- Visual Studio 2019

- **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. Simply choose the work item type from the **New Work Item** drop down menu.

For example, here we choose User Story.

- New navigation
- Previous navigation
- Visual Studio 2019

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.

- New navigation
- Previous navigation
- Visual Studio 2019

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 in the work item form to add and review comments about the work under development.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name. Or, bring a group into the discussion by typing **@** and the group name, such as a team or security group. 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.

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.

Use the icons—@ at-mention, # #-work-item-ID, and pull-request ID —to facilitate bringing others into the discussion or linking to work items or pull requests. Choose one of these icons and a menu displays with the most recent options that you've previously selected.

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. Simply choose

Open in Queries or the **Open selected items in Queries** option from the context menu.

Queries provide additional 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.

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

- Best tool to add, update, and link work items
- Move, change, or delete work items (Recycle Bin)
- Enable preview features
- Use work item form controls
- Keyboard shortcuts for work item forms and the Work Items page
- Work across projects

> **NOTE**
> You can create and manage work items from the command line or scripts using the Azure DevOps CLI.

1/31/2019 • 9 minutes to read • Edit Online

Add, update, and follow a work item

Add and update a work item

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You add work items to plan and manage your project. You use different types of work items to track different types of work—such as user stories or product backlog items, tasks, bugs, or issues. You can describe the work to be done, assign work, track status, and coordinate efforts within your team.

For additional clients that you can use, 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 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 About access levels.
- 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 create 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.

- 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 About access levels.
- 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 create 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.

Add a work item

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

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation
- Visual Studio 2019

Choose a **Boards** page—such as **Work Items**, **Boards**, or **Backlogs**. Then choose the + plus icon and select from the menu of options.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

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.

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.

Note that the location of the State field may differ depending on the work item type you are updating.

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 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 when you click your cursor within the each text box that can be formatted.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** or use the at-

mention icon. and select or type their name. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

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

Use the pull-request id icon to link to a pull request.

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.

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

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.

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.

You'll only receive notifications when other members of your team modifies 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 additional clients that you can use to add work items, see Clients that support tracking work items.

Drive Git development from a work item

1/31/2019 • 7 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

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.

> **IMPORTANT**
>
> This article addresses creating new branches and adding links to commits and 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.

The Development section records all Git development processes that support completion of the work item. This section 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.

> **NOTE**
> The Development section within the work item form is not supported in TFS 2015 and earlier versions. Consider upgrading to a later TFS version.

From it, you can quickly access branches, pull requests, and commits which are linked to the work item. Also, you can initiate a pull request for a branch you've created or linked to from the 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 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 About access levels.
- 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 create 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.

- 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 About access levels.
- 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 create 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.

Workflow process to create a branch and pull request

Consider creating a new branch when there are no linked code artifacts. If there is a branch 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 is always named `master`. 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 just 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 just 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 details, 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 master 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 (click the icon), and you'll see the links that have been added to support the operations you just 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 initiated from the Development section are also logged under the History tab.

To link a work item to an existing object, click 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 click the ✕ delete icon.

Or, you can select it from the Links tab and click the remove link icon.

Related articles

Learn more about tracking work with work items and developing with Git from these resources:

- Add work items
- Git overview
- Link GitHub commits and pull requests to work items
- TFVC overview
- Create your backlog

- Add work items
- Git overview
- TFVC overview
- Create your backlog

Keep in mind that the Development section 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 will be missing several of the features that the web portal makes available.

Associated work items in build

With Git commits, any work items that have been linked to a commit will be listed under the Associated work items in the build summary page.

Link types showing in the Development section

Links shown in this section appear as a result 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.

Define area paths and assign to a team

2/7/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can add area paths to support teams and to group work items based on product, feature, or business areas. Then, define area paths at the project level and assign them to a team under the team configuration. You can also create a hierarchy of area paths to support subcategories within categories.

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.

To understand how the system uses area paths, see About area and iteration paths.

> **IMPORTANT**
>
> Make sure that you select the correct version of this article for Azure DevOps Services or Azure DevOps Server, renamed from Team Foundation Server (TFS). The version selector is located above the table of contents.

Prerequisites

- Add Area Paths to a project. 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 Set permissions at the project- or collection-level.
- 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 set. To learn more, 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 sequence

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, you'll want to add one Area Path for each team that you'll define. For guidance, review About areas and iterations.

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**
> 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

Open Project Settings

From the web portal, open **Project Settings**.

> **NOTE**
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

You define both areas and iterations for a project from the **Project Settings>Work>Project configuration**.

1. Choose (1) **Project Settings**, expand **Work** if needed, and choose (2) **Project configuration** and then (3) **Areas**.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Add an area path

If you haven't added any areas or teams, you'll see that only one area is defined. You add area paths under the root area path for the project, or as a child to another area path.

- To add a child node, highlight the area path and then choose **New child**. Optionally, you can open the ···

context menu 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.

- To add a child node, highlight the area path and then choose **New child**. Optionally, you can open the ··· context menu 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.

1. Open **Areas**.

From the areas page, you can set the default area path used to filter the backlog. The default area path is also used when new work items a user creates new work items.

2. Add a new child node to the area you've selected.

Open team settings

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.

From a web browser, open the web portal administrative context for your team.

- New navigation
- Previous navigation

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 the team profile icon. Then choose **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.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Set team default area path(s)

All work items assigned to the area paths selected for a team appear on the backlogs and boards for that team. You can select one or more area paths and optionally include their sub-area paths. Choose to include sub-area paths when you want to support rollup views of work performed across several teams or areas.

All work items assigned to the area paths selected for a team appear on the backlogs and boards for that team. You can select a single area path, and optionally include their sub-area paths. Choose to include sub-area paths when you want to support rollup views of work performed across several teams or areas.

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 then 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.

1. Open **Areas** for the team context.

   Here, we show the Areas for the Web team.

2. Choose the area path(s) to be active for each team.

In this instance, we choose to activate all three sub-area 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.

1. Open the Areas admin page for the team context.

   Here, we navigate to the Web team. The checked box indicates the area paths selected for the team. To exclude sub-areas, select the option from the area path context menu.

2. Refresh the product backlog page for the team, and you'll see only those work items assigned to the Fabrikam Fiber\Web area path.

Rename, move, or delete an area path

When you rename an area or an iteration, or move the node within the tree hierarchy, the system automatically updates the work items and queries that reference the existing path or paths.

1. To rename an area or iteration path, choose the ··· actions icon for the node, and 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.

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 are assigned to the root area path and show up in the default team that's defined for the project.

Next steps

Set iteration paths or sprints

Related articles

As you can see, areas play a major role in supporting Agile tools 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

Define iteration paths (aka sprints) and configure team iterations

1/31/2019 • 13 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You add iteration paths to support teams perform sprint planning and to group work items based on a time-box interval or sprint. You define iteration paths at the project level and then each team selects the one to be active for them under the team configuration. You can create a flat iteration path structure or a hierarchy of it paths to support releases, sub-releases, and sprints.

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). Several tools reference the team's default and active 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.

Newly created projects contain a single, root area that corresponds to the project name. You add iteration paths under this root. Each project typically specifies a predefined set of iterations to help you get started tracking your work. All you need to do is specify the dates.

> **IMPORTANT**
>
> Make sure that you select the correct version of this article for Azure DevOps Services or Azure DevOps Server, renamed from Team Foundation Server (TFS). The version selector is located above the table of contents.

Prerequisites

- You add Iteration Paths to a project. If you don't have a project yet, create one now.
- To add an Iteration Path under the root node or edit or delete any child node, you must be a member of the **Project Administrators** group. To acquire these permissions, see Set permissions at the project- or collection-level.
- Or, 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 you want to manage: **Create child nodes**, **Delete this node**, and **Edit this node**, and **View permissions for this node**. By default, the user who created the project has these permissions set. To learn more, see Set permissions and access for work tracking.
- To set team Iteration Paths, you must be added as the team administrator or be a member of the **Project Administrators** group.

For naming restrictions on Iteration Paths, see About areas and iterations, Naming restrictions.

Get started sequence

If you are new to managing projects and teams, the most straight forward sequence for configuring iterations for your project and teams is as follows:

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. For guidance, review About areas and iterations.
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 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).
6. Each team should assign the default Iteration Path they selected to their work items. This is needed in order 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 perform 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

Backlog iteration versus default iteration

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.

For TFS 2015 and earlier versions, the default iteration is the same as the backlog iteration. The one value selected both filters items that appear on the team's backlogs and boards, and is assigned to work items created from the team context.

Open Project Settings

From the web portal, open **Project Settings**.

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

You define both areas and iterations for a project from the **Project Settings>Work>Project configuration**.

1. Choose (1) **Project Settings**, expand **Boards** if needed, and choose (2) **Project configuration** and then (3) **Iterations**.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Add iterations and set iteration dates

From **Iterations**, you can add iterations that teams can then select for their use. You add iterations in the same way you add areas. For more information about working within a sprint cadence, see Scrum and sprint planning tools.

You add and modify area paths from the **Work, Iterations** page from the project admin or settings context.

For Scrum-based projects, you'll see the following set of sprints.

1. To schedule the start and end dates for each sprint your teams will use, Highlight the sprint and choose **Set dates**. Or, you can open the ··· context menu for the iteration path and choose **Edit**.

   Choose the calendar icon to choose new dates.

2. When you're finished, you'll have a set of sprints scheduled - like this:

Your next step is to choose the sprints each team will use.

1. Open the **Iterations** tab for the project context.

   For Scrum-based projects, you'll see these set of sprints.

   You can change the name, location within the tree hierarchy, or set dates for any sprint. Simply open it (double-click or press Enter key) and specify the info you want.

2. Schedule the start and end dates for those sprints you plan to use.

After you set the start and end dates for one iteration, the calendar tool automatically attempts to set the next set of dates, based on the same iteration length you specified for the first. For example, if you set a three week sprint for Sprint 1, then when you select the start date for Sprint 2, the calendar tool automatically determines the start and end dates based on the next three weeks. You can accept or change these dates.

3. To add another sprint, select **New child** and name it what you want. Here, we call it Sprint 7.

Your next step is to select the sprints each team will use.

Open team settings

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.

From a web browser, open the web portal administrative context for your team.

- New navigation
- Previous navigation

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 the team profile icon. Then choose **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.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Select team sprints and 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.

- New navigation
- Previous navigation

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 used when a work item is created from the team dashboard (new work item widget) and queries page. You can use an explicit value or use **@CurrentIteration** to assign new work items to the team's current iteration. This is the same macro used in queries to list work items assigned to the currently active iteration assigned to the team.

   For example, you might want all new work items to be 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 you need, or the dates aren't set, you can add or edit iterations for the project, provided you have the required permissions. To learn more, see Define iteration paths (aka sprints).

5. To see the newly activated sprint backlogs, refresh your team's product backlog page.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Rename, move, or delete an iteration

When you rename an iteration, or move the node within the tree hierarchy, the system will automatically update the work items and queries that reference the existing path or paths.

1. To rename an iteration path, choose the ··· actions icon for the node, and 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.

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 will be 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.

Prior to archiving the iterations, consider if you have captured all the reports that you may want.

Chart progress by iteration

You can quickly generate queries to view the progress for those areas. As an example, you can visualize progress of work items 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 won't be able to gain much use of sprint planning tools.

Related articles

As you can see, iterations play a major role in supporting Agile tools and managing work items. You can learn more about working with these fields from these 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
- Query by date or current iteration
- Query by area or iteration path
- Set permissions and access for work tracking

Follow a user story, bug, issue, or other work item or pull request

2/7/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure Repos | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

To get notified of changes made to a work item or a pull request, you can elect to follow them.

This article shows you how to:

- Follow a work item
- Follow a pull request
- Manage work items that you're following

You must configure an SMTP server in order for team members to receive notifications.

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.

> **NOTE**
>
> The **Follow a work item** feature is available from TFS 2017 and later versions. The **Follow a pull request** feature is available from TFS 2017.1 and later versions. To update your on-premises TFS, visit the Visual Studio downloads page for Team Foundation Server.

You'll only receive notifications when other members of your team modifies 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 modifies 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.

> **NOTE**
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

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 perform 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**.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

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: 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.

Define, triage, and manage bugs

2/7/2019 • 15 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

How do you track and manage defects in your code? How do you make sure software problems and customer feedback get addressed in a timely manner to support high-quality software deployments? And, how do you do this while making good progress on new features?

At a minimum, you need a way to capture your software issues, prioritize them, assign them, and track progress. Moreover, you'll want to manage your bugs in ways that align with your Agile practices.

In a nutshell, you manage bugs through the following tasks:

- Capture information using the bug work item type
- Triage bugs by assigning a priority
- Update bug status throughout the bug lifecycle
- Monitor bug assignments and trends

> **IMPORTANT**
>
> Depending on the process chosen to create your project—Basic, Agile, Scrum, or CMMI— the items in your backlog may be called issues, user stories, product backlog items (PBIs), or requirements. All these work item types are similar: they describe the customer value to be delivered and the work to be performed.
>
> By default, user stories on Agile backlogs, product backlog items (PBIs) and bugs appear on Scrum backlogs, and requirements on CMMI backlogs. Each team can choose how bugs show up on their backlogs and boards. The Bug work item type isn't defined for the Basic process. Instead, you should track bugs and code defects using the Issue work item type.

Prerequisites

- 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 About access levels.
- 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 create 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.

- 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 About access levels.

- 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 create 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.

Define and list bugs

When bugs are managed along with requirements, you can add them through the product backlog or Kanban board. When bugs are managed along with tasks, you can add them to a sprint backlog or taskboard.

Or, you can define a bug from the web portal, Visual Studio/Team Explorer, a work item template, or using test tools. Using the bug work item form, you capture the code defect in the Title, Steps to Reproduce, and other fields.

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). For other bug related tasks, see the following articles:

- View, run, or email a work item query
- View and add work items using the Work Items page
- Triage work items
- Query by assignment or workflow changes

- View, run, or email a work item query
- Triage work items
- Query by assignment or workflow changes

Bug work item form

The bug work item form tracks similar information to the one shown for the Scrum process.

> **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.

- New web form
- Old web form

The new web form provides a number of experiences not provided with the old web form. To learn more, see New work item experience.

TIP

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

The new web form is only available from TFS 2017 and later versions.

Fields specific to bugs

When defining a bug, use these fields to capture both the initial issue and ongoing discoveries made when triaging, investigating, fixing, and closing the bug.

FIELD/TAB	USAGE

Steps to Reproduce (friendly name=Repro Steps)	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. Describe the criteria that the team should use to verify whether the code defect is fixed.
System Info	Information about the software and system configuration that is relevant to the test.
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. 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.
Found In Build Integrated in Build	When Test Manager creates bugs, it automatically populates **System Info** and **Found in Build** with information about the software environment and build where the bug occurred. To learn more about defining the software environments, see Test different configurations. When you resolve the bug, use **Integrated in Build** to indicate the name of the build that incorporates the code that fixes the bug. For on-premises TFS, 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 names, see build number format options.
Priority [1]	A subjective rating of the bug as it relates to the business or customer requirements. Priority indicates the order in which code defects should be fixed. 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.

Severity [1]	A subjective rating of the impact of a bug on the project or software system. For example: If clicking a remote link (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.

[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.

For information about fields specific to the CMMI process, see Bugs, issues, and risks field reference. For information about all other fields, see Work item field index.

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 when you click your cursor within the each text box that can be formatted.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** or use the at-mention icon. and select or type their name. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

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

Use the pull-request id icon to link to a pull request.

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.

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

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.

Capture bugs using test tools

You can create bugs during test sessions using one of the following tools:

- **Test & Feedback extension**: see Exploratory testing with the Test & Feedback extension
- **Test Runner**: see Update an existing bug while using Test Runner.

Triage bugs

Once you've started coding and testing, you'll want to hold periodic triage meetings to review and prioritize your bugs. How frequently you meet and for how long depends on your situation. Typically, the project owner runs the bug triage meetings, and team leads, business analysts and other stakeholders who can speak about specific project risks attend them.

The project owner can create or open a shared query for new and reopened bugs to generate a list of bugs to be triaged.

Bug queries

Open a shared query or use the query editor to create useful bug queries, such as the following:

- 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 3 weeks (`Created Date > @Today-21`)

Once you have the queries of interest to your team, you can create status or trend charts that you can also add the chart to a dashboard.

Triage mode in query results

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.

Assign bugs to a sprint

Once bugs have been triaged, it's time to assign them to a sprint to get fixed. By addressing a set of bugs to get fixed every sprint, your team can keep the total number of bugs to a reasonable size.

When bugs appear on the product backlog, you can assign bugs to sprints in the same way you do PBIs and user stories during your sprint planning sessions.

When bugs are treated as tasks, they're often automatically linked to a PBI or user story. So, assigning their parent PBI or user story to a sprint will assign the linked bugs to the same sprint as the parent PBI or user story during your sprint planning sessions.

Your team should consider fixing all bugs found during a sprint when testing a feature in development.

Fix, resolve and close bugs (update status)

Bug fixes that involve more than a single section of code may require significant regression testing and may involve other team members. Record any conversations that relate to assessing the risk of bug fixes in the bug work item history.

Bug workflow lifecycle

Once you fix a bug, you should update its workflow State. State choices vary depending on the process you use

—Scrum, Agile, or CMMI. The following images illustrate the workflow lifecycle defined for the default bug workflow for the Agile, Scrum, and CMMI processes.

AGILE	SCRUM	CMMI

For Scrum bugs, you simply change the State from Committed (similar to Active) to Done. For Agile and CMMI, you first resolve the bug, indicating that the bug has been 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, it can be reactivated by setting the State to Committed or Active.

Verify a fix

To verify a fix, a developer or tester should attempt to reproduce the bug and look for additional unexpected behavior. If necessary, they should 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, 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.

Verify a bug, re-run tests defined for web apps

Choose the **Verify** option to re-run tests which identified the bug. You can invoke the **Verify** option from the bug work item form context menu to launch the relevant test case in the web runner. Perform your validation using the web runner and update the bug work item directly within the web runner.

Choose the **Verify** option to re-run tests which identified the bug. (Requires TFS 2017.1 or later version.) You can invoke the **Verify** option from the bug work item form context menu to launch the relevant test case in the web runner. Perform your validation using the web runner and update the bug work item directly within the web runner.

To learn more about running test from the web portal, see Run tests for web apps.

Close a bug

You close a bug once it's verified as fixed. However, you may also close a bug for one of these reasons:

- Deferred - deferring a fix until the next product release
- Duplicate - bug has already been reported, 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 PBI or user story has been opened to track the bug

It's always a good idea to describe any additional details for closing a bug in the Discussion field (new web form) or the History field (old web form) to avoid future confusion as to why the bug was closed.

Monitor bug status, assignments, and trends

You can track the bug status, assignments, and trends using queries which you can then chart and add to a dashboard.

For example, here are two examples showing active bugs by priority trend and a snapshot of bugs by priority.

To learn more about queries, charts, and dashboards; see Create managed queries and Charts, and Dashboards.

Customize the bug form

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 the On-premises XML process model.

Try this next

Triage work items

Related articles

To track your bugs and integrate with other resources available to you, see these topics:

- Scrum and working with sprints best practices
- Follow a work item or pull request
- Move, change type, or delete work items
- Pre-populate fields using a template
- Copy or clone a work item

Integrate & Test resources

- UserVoice
- Zendesk
- Developer testing tools scenarios and capabilities
- Run tests for desktop apps

Use the Analytics service to create bug reports

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

If you work from an on-premises Azure DevOps Server or TFS and you have SQL Server Analysis Services and SQL Server Reporting Services configured for your project, you have access to the following reports (Agile and CMMI processes only).

- Bug Status
- Bug Trends
- Reactivations

To learn how to add SQL Server reports for a project, see Add reports to a project.

Use SonarQube to help manage technical debt

SonarQube provides a way of automatically measuring some technical debt. SonarQube finds important violations of best coding practices. You implement Sonar to ensure that developers follow important code metrics like appropriate class and method size or low cyclomatic complexity (a quantitative measure of the number of linearly independent paths through a program's source code).

By integrating your on-premises TFS with a SonarQube server, you can get the following data:

- Results of .NET and JavaScript code analysis
- Code clone analysis
- Code coverage data from tests
- Metrics for .NET and JavaScript

See Technical Debt Management: Announcing SonarQube integration with MSBuild and Team Build for details.

Manage issues or impediments

1/31/2019 • 5 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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. Using the impediment work item type helps 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 address feature ambiguity, personnel or resource issues, problems with environments, or other risks that impact 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. If you want to add these work item types to a backlog, see Customize your backlogs or boards.
>
> 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.

In this topic you'll learn:

- 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 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 About access levels.
- 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 create 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.

- 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 About access levels.
- 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 create 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.

> **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.

Add an issue or impediment

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

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.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Customize issues 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 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 Customize your backlogs or boards.

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.

Related articles

- Add work items
- Work item form controls
- Manage bugs or code defects
- Create your backlog

1/31/2019 • 20 minutes to read • Edit Online

Move, change, or delete work items

Azure Boards

Often times 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.

Delete or restore work items

Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can remove work items added to your backlog or taskboard that aren't relevant anymore. Simply change the State to Remove, or delete the work item. You can perform operations on individual work items or bulk modify several work items.

> **TIP**
> 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.

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
- How to remove work items from the backlog by changing the State to Removed
- How to delete work items
- How to restore or permanently delete work items (web portal)
- What permissions are required to delete work items

- How to remove work items from the backlog by changing the State to Removed
- How to delete work items
- How to restore or permanently delete work items (web portal)
- How to permanently delete work items (command-line tool)
- What permissions are required to delete work items

- How to remove work items from the backlog by changing the State to Removed
- How to permanently delete work items (command-line tool)
- What permissions are required to delete work items

You only have access to those actions that are supported on your platform and for which you have permissions. If you are a member of the Contributors group (anyone who has been added as a team member) or Project Administrators groups, you have access to the following features. For a simplified view of permissions assigned to built-in groups, see Permissions and access.

You can access the following actions for which you have permissions. If you are a member of the Contributors

group (anyone who has been added as a team member) or Project Administrators groups, you have access to the following features. For a simplified view of permissions assigned to built-in groups, see Permissions and access.

CONTRIBUTORS & STAKEHOLDERS	PROJECT ADMINISTRATORS
• Change work item type • Remove work items (change State) • Delete work items • Restore work items	• Move a work item to another project • Permanently delete work items • Permanently delete test artifacts

CONTRIBUTORS & STAKEHOLDERS	PROJECT ADMINISTRATORS
• Change work item type [1] • Remove work items (change State) • Delete work items • Restore work items	• Move a work item to another project [1] • Permanently delete work items • Permanently delete test artifacts

Notes:

1. 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.

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.

CONTRIBUTORS & STAKEHOLDERS	PROJECT ADMINISTRATORS
• Remove work items (change State) • Delete work items • Restore work items	• Permanently delete work items (web portal • Permanently delete work items (command-line) • Permanently delete test artifacts

CONTRIBUTORS & STAKEHOLDERS	PROJECT ADMINISTRATORS
• Remove work items (change State) • Delete work items • Restore work items	• Permanently delete work items (web portal • Permanently delete work items (command-line)

- **Contributors & Stakeholders**: Remove work items (change State)
- **Project Administrators**: Permanently delete work items (command-line)

> **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 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 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, delete, or remove work items, you must be granted **Stakeholder** access or higher For details, see About access levels.
- 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.
- To delete work items, you must be a member of the **Project Administrators** group or have the **Delete work items in this project** permission set to Allow. By default, the Contributors group has **Delete and restore work items** set to **Allow**.

> **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 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 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, delete, or remove work items, you must be granted **Stakeholder** access or higher. For details, see About access levels.
- 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.
- To delete work items, you must be a member of the **Project Administrators** group or have the **Delete work items in this project** permission set to **Allow**. By default, the Contributors group has **Delete and restore work items** set to **Allow**.

- 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 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 remove or delete work items, you must be granted **Stakeholder** access or higher. For details, see About access levels.
- To delete work items, you must be a member of the **Project Administrators** group or have the **Delete work items in this project** permission set to Allow. The **Contributors** group has **Delete and restore work items** at the project-level set to **Allow** by default.

- 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 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 delete or remove work items, you must be granted **Stakeholder** access or higher. For details, see About access levels.
- To delete work items, you must be a member of the **Project Administrators **group or have the **Delete work items in this project** permission set to **Allow**. By default, for TFS 2015.1 and earlier versions, the Contributors group has **Delete work items in this project** set to **Not set**. This setting causes the Contributors group to inherit the value from the closest parent that has it explicitly set.

To learn more, see Set permissions and access for work tracking or Set permissions at the project-level or project collection-level.

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 of work items associated with test management. Both Contributors and users assigned Stakeholder access can change the work item type.

You can change a single work item or several multi-selected work items to a new type.

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

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.

3. Save the work item to complete the change.

> **NOTE**
> The system automatically resets the State and Reason fields to the default initial values of the specified type.

From the Query results page, you must save all work items that you bulk-modified. When you bulk modify items from the backlog, they are automatically saved. Work items shown in bold text indicate that local changes have not 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.

You can only move work items from one project to another project within the organization or collection. 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.

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 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.

   Click ··· to open the context menu of one of the selected work items, and choose the **Move...** option.

2. Select the destination project 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.

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).

To cause removed items to not show up in queries, you must add a clause that filters on the **State** field.

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 test suite, or other test-related work item types, see Delete test artifacts.

- New navigation
- Previous navigation

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.
2. Confirm you want to actually delete the item(s).

3. You can also delete work items from your Kanban or taskboard.

Delete work items

> **NOTE**
>
> The **Delete and Recycle bin** features are available from TFS 2015.2 and later versions.

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 test suite, or other test-related work item types, see Delete test artifacts.

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.

2. Confirm you want to actually delete the item(s).

> **NOTE**
>
> The Delete work items confirmation dialog for on-premises TFS 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.

3. Using multi-select from a backlog or query results list, you can delete several work items at once.

4. You can also delete work items from your Kanban or taskboard.

Or, you can drag them to the

(Recycle bin). You can only access the (Recycle bin) from **Work**.

Restore or permanently delete work items

You restore deleted work items from the web portal Recycle bin.

- New navigation
- Previous navigation

1. Choose **Boards>Work Items** and then choose the **Recycle bin**.

A new browser tab opens with the query which lists work items added to the Recycle bin.

2. Select the items you want to restore and then choose **Restore**.

> **NOTE**
> You can't open work items that have been moved to the Recycle bin.

Optionally, you can choose to permanently delete the items.

> **NOTE**
> You'll only see the Permanently delete option if your Permanently delete work items permission is set to Allow.

3. Confirm your selection.

Restore or permanently delete work items

You restore deleted work items from the web portal Recycle bin.

> **NOTE**
>
> **Feature availability**: The Delete and Recycle bin features require TFS 2015.2 or later version.

1. Choose **Work>Work Items** and then choose the **Recycle bin**.

Or, from a backlog or query, open the Recycle bin.

A new browser tab opens with the query which lists work items added to the Recycle bin.

2. Select the items you want to restore and then choose **Restore**.

Optionally, you can choose to permanently delete the items.

> **NOTE**
> You'll only see the Permanently delete option if your Permanently delete work items permission is set to Allow.

3. Confirm your selection.

Permanently delete work items (command line)

To permanently delete work items from the web portal, you must be a member of the Project Administrators group or be granted explicit permissions to delete or restore work items.

Deleting work items from the command line is deprecated for TFS 2018.2 and later versions, and not supported for Azure Boards.

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 cannot be restored nor reactivated. You must be a member of the Project Administrators group of have your **Edit project-level information** permission set to Allow.

1. 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.

2. 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, simply enter the ID as shown:

   ```
   witadmin destroywi /collection:http://TFSServerName:8080/tfs/DefaultCollection /id:2003
   ```

1. 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 2017. (For other versions, see Remove work items permanently (witadmin destroywi)).

   ```
   %programfiles(x86)%\Microsoft Visual
   Studio\2017\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.

2. 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, simply enter the ID as shown:

   ```
   witadmin destroywi /collection:http://TFSServerName:8080/tfs/DefaultCollection /id:2003
   ```

1. 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 2015. (For other versions, see Remove work items permanently (witadmin destroywi)).

   ```
   cd %programfiles(x86)%\Microsoft Visual Studio 14.0\Common7\IDE
   ```

   > **NOTE**
   >
   > **Required permissions:** For TFS 2015.2 or later versions, you must have Permanently delete work items permission set to Allow. For TFS 2015.1 or earlier versions, you must be a member of the Project Administrators group of have Edit project-level information permissions set to Allow.

   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.

2. 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, simply enter the ID as shown:

   ```
   witadmin destroywi /collection:http://TFSServerName:8080/tfs/DefaultCollection /id:2003
   ```

1. 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 2013. (For other versions, see Remove work items permanently (witadmin destroywi)).

   ```
   cd %programfiles(x86)%\Microsoft Visual Studio 12.0\Common7\IDE
   ```

   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.

2. 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, simply enter the ID as shown:

   ```
   witadmin destroywi /collection:http://TFSServerName:8080/tfs/DefaultCollection /id:2003
   ```

Related articles

- Best tool to add, update, and link work items
- View and add work items using the Work Items page
- Remove work items permanently (witadmin destroywi)
- Delete test artifacts
- Create a test plan
- Control how long to keep test results

- Best tool to add, update, and link work items
- Delete test artifacts

- Add, update, and follow a work item
- Remove work items permanently (witadmin destroywi)

- Best tool to add, update, and link work items
- Add and update a work item
- Remove work items permanently (witadmin destroywi)

Delete and restore actions performed under the hood

Delete work items

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 work item to become non-queryable and therefore can't appear in any work tracking experience, query result, or report
- Updates charts accordingly, CFD, velocity, burndown and lightweight charts are updated to remove deleted work items
- Removes WIT 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.

Restore work items

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 2 new revisions, one for deletion, and one for restore
- Reattaches WIT extensions
- Updates charts accordingly, 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 similar
- Sets the area or iteration path fields to the root node if the previous area path or iteration paths were deleted.

Link user stories, issues, bugs, and other work items

1/31/2019 • 10 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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
- 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
- Find work items that you want to link to
- Bulk modify link relationships

For a list of all link types and supported link relationships, see Linking, traceability, and managing dependencies.

Prerequisites

- 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 About access levels.
- 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 with **Stakeholder** access for a public project have full access to backlog and board features 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 add or modify work items, you must be granted **Stakeholder** access or higher. For details, see About access levels.
- 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.

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 ··· actions icon 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**....

1. From the web portal, open a 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, and then choose **Link to a new item**... or **Link to an existing item**....

   Here we multi-select from the Queries page and choose **Link to a new item...**.

1. From the web portal, open a 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, and then choose **Link selected item(s) to a new work item**... or **Link to an existing item**....

   Here we multi-select from the Queries page and choose **Link selected item(s) to a new work 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. Simply 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 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 based on IDs, a query, or title keyword.

To view the work items selected for linking, you can choose the ⓘ.

If you are 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 have 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 are 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.

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 are 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 have 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, click 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. Click 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.

Additional bulk-modify link options

Additional 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 topic 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 or Project. See Bulk add or modify work items with Excel and Create your backlog and tasks using Project.

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 parent-child links
- Use Project to edit parent-child and predecessor-successor links
- Linking, traceability, and managing dependencies

- Map backlog items to portfolio backlog items
- Link work items to Git development objects
- Use Excel to edit parent-child links
- Use Project to edit parent-child and predecessor-successor links
- Linking, traceability, and managing dependencies

Bulk modify work items

1/31/2019 • 11 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Use bulk modify when you need to quickly make the same change to a number of 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 perform a bulk add of work items through the web portal.

With bulk modify, you can edit fields, add or remove tags, 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

- 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

Prerequisites

- 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 About access levels.
- 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 with **Stakeholder** access for a public project have full access to backlog and board features 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 add or modify work items, you must be granted **Stakeholder** access or higher. For details, see About

access levels.

- 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.

Supported tasks

All of the following actions can be performed by team members that belong to the Contributors group. Members provided with Stakeholder access can perform 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]

Notes:

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.

> **NOTE**
>
> To exercise the **Change work item type** or **Move work items to another project**, you must have disabled the data warehouse.

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)

Notes:

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 TFS, you must have an SMTP server configured for your deployment.

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
Bulk edit/update/delete	• Edit field(s) • Assign to • Move to iteration • Change position • Change parent • 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)

Notes:

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 TFS, you must have an SMTP server configured for your deployment.

Bulk edit multi-selected work items

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.

(Requires TFS 2015.1 or later versions)

You can use bulk modify by selecting work items from the backlog page or query results list. From the backlog page context menu, you can change the backlog priority of several items (Change position or Move to position), 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

Backlog multi-select menu

Query results multi-select menu

Backlog multi-select menu

Query results multi-select menu

To multi-select 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, click (···) 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.

Stories
Backlog
Board
New
Create query
Column options
Order
Story Points
State
Title
Iteration Path
Resume
Hello World Web Site
Show response on welcome page
Change the initial view
Add an information form
Secure sign on
Welcome back page
Check issues with permissions
Log on
Welcome back page
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

Order
Work Item Type
Title
State
Effort
Product Backlog Item
Hello World Web Site
New
Bug
Slow response on form
Committed
8
Product Backlog Item
Change initial view
New
Bug
Secure sign-in
New
Product Backlog Item
Welcome back page
New
Bug
Canadian addresses don't display properly
New
Interim save on long forms
New
Edit selected work item(s)...
Move to top
Move to position ...
Assign to
Move to iteration
Indent (change parent)
Copy as HTML
Email selected work item...
Backlog
Current (Sprint 1 - 7/27/2015 - 8/14/2015)
FUTURE ITERATIONS
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6

> **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 this, 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.

1. From the Query results page, you must save all work items that you bulk-modified. When you bulk modify items from the backlog, they are automatically saved. Work items shown in bold text indicate that local changes have not yet been saved to the data store.

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 are automatically saved. Work items shown in bold text indicate that local changes have not yet been saved to the data store.

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. Save all work items that you bulk-modified. Work items shown in bold text indicate that local changes have not yet been saved to the data store.

Move work items to a sprint

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

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.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Bulk modify tags

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.

Here we choose to add the *Beta* tag to the selected work items.

Bulk update of work items to add or remove tags from the web portal requires TFS 2015.2 or later version. To bulk edit work items when connecting to TFS 2015.1 or earlier versions, use Excel.

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.

Copy or clone work items

1/31/2019 • 7 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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. In addition, you can choose to change the project or work item type when copying/cloning a work item.

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.

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

Prerequisites

- 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 About access levels.
- 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 create 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.

- 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 About access levels.
- 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 create 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.

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. No history or attachments are copied.

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.

1. From the web portal, open the work item you want to copy or clone, open the ... context menu, and click **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 additional 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.

1. From the web portal, open the work item you want to copy or clone, open the ... context menu, and click **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 additional 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.

1. From the web portal, open the work item you want to copy or clone, and click the copy/clone icon. The copied work item is automatically linked to the original work item through a Related link type.

2. Choose the project (if copying to another project) and work item type if different from the copied work item. Optionally change the Title and provide additional details. The copied work item is automatically linked to the original work item through a Related link type.

1. From the web portal, open the work item you want to copy or clone, and click the copy/clone icon. The copied work item is automatically linked to the original work item through a Related link type.

2. Choose the project (if copying to another project) and work item type if different from the copied work item. Optionally change the Title and provide additional details. The copied work item is automatically linked to the original work item through a Related link type.

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 Change work item type isn't available to you, you can export a set of work items using Excel, copy them to a new Excel list, and re-import them by specifying a different work item type. See Bulk add or modify work items with Excel.

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 TFS, 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**.

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 as HTML**.

   Here we multi-select from the backlog page.

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 as HTML**.

   Here we multi-select from the backlog page.

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 click the copy-to-clipboard icon.

Right click the link ID to open the browser copy link option.

Related topics

- Bulk modify work items
- Move, change, or delete work items
- Pre-populate fields using work item templates

Get started as a Stakeholder

2/5/2019 • 11 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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. Stakeholder access is one of several supported access levels. To understand the full set of features Stakeholders have access to, see About access levels.

> **NOTE**
> For public projects, Stakeholder access gives users greater access to work tracking features. To learn more, see Default roles and access for public projects.

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. To understand the full set of features Stakeholders have access to, see About access levels.

With Stakeholder access, you can add and modify work items. 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. To understand the full set of features Stakeholders have access to, see About access levels.

Use this topic to learn:

- How to sign-in to a project
- How to add a work item
- How to view the product backlog and add new work to it
- How to view work in progress on the Kanban board
- Find work assigned to you, or query for other work items

For information about working with pipelines, see these articles:

- Build your GitHub repository
- Build OSS repositories

First time signing in

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.

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

   `http://ServerName:8080/tfs/DefaultCollection/ProjectName`

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

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

View and add work items

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

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

1. (1) Check that you have selected the right project, then (2) choose **Boards>Work Items**.

2. Using the drop-down menu, you can focus on relevant items inside a project using one of the seven pivots as described next.

- **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've elected to follow.
- **Mentioned**: lists work items in which you've been mentioned in the last 30 days.
- **My activity**: lists work items that you have 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.

3. To add a work item, simply choose the work item type from the **New Work Item** drop down menu.

   For example, here we choose User Story.

4. Enter a title and then save the work item. Before you can change the State from its initial default, you must save it.

> **NOTE**
> A caution icon on a tab indicates values that violate validation rules. You must correct information on that tab in order to save the work item.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

For descriptions of each field, see Work item field index.

> **NOTE**
> Depending on the process chosen when the project was created, 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.

Add tags to a work item

Tags are useful for filtering backlogs, boards, and queries

All users granted Stakeholder access for a private project can only add existing tags to work items, not add new tags. Even if the **Create tag definition** permission for the user is set to Allow. This is part of the Stakeholder access settings.

Azure DevOps Services users granted Stakeholder access for a public project can add new and existing tags to work items.

Check the backlog, add work items to the backlog

Work appears in the backlog in priority order. Work item types may include bugs depending on the settings made for the team.

- New navigation
- Previous navigation

1. (1) Check that you have 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 sprints** 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.

4. To view or edit a work item, select it and choose **Enter**.
5. To add a work item, choose the + **New Work Item**, enter a title and then press the Enter key or choose **Add to top**.

Repeat this step to capture all your ideas as work items.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Check work in progress

To view the team's progress, open the Kanban board. To view or edit a work item, choose a title and press **Enter**, or double-click the title.

- New navigation
- Previous navigation

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

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

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

- New navigation

- Previous navigation

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.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Related articles

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

- Add work items
- Create your backlog
- Kanban quickstart
- Provide Stakeholders access to edit build and release pipelines

If you want to provide a group of users access to provide feedback, then you don't need to give them Stakeholder access. Instead, simply give reviewers permissions to provide feedback.

Choose a process

2/7/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Anytime you create a project, you must choose a process or process template based on the process model you use.

- 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 as well as other sub-systems you access through Azure Boards or an on-premises Azure DevOps Server or Team Foundation Server (TFS). It supports Hosted XML and On-premises XML process models which support customization of projects through the modification and import of XML definition files.

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

> **TIP**
>
> With Azure DevOps Server, you can choose between using the Inherited process model or the On-premesis 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 here: %programfiles%/Azure DevOps Server 2019/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 are summarized below. (Currently, the Basic process is in Preview and available from the Azure Boards cloud service.) For simplicity, they are referred to as a "process."

Basic, Agile, Scrum, and CMMI

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 light-weight. 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.

Agile, Scrum, and CMMI

The default processes differ mainly in the work item types (WITs) they provide for planning and tracking work. Scrum is the next most light-weight. 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.

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

> **NOTE**
>
> Epics are supported on Azure Boards and TFS 2015 and later versions. Each team can choose the backlog levels that are active as described in Select backlog navigation levels for your team.

Basic Choose Basic when your team wants the simplest model that uses Issues, Tasks, and Epics to track work. Note: Basic is currently in a selective Preview for new users of Azure Boards only. 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 PBIs and bugs down 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 three 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

Notes:

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.

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	Scrum	CMMI
Epic 	**Epic** 	**Epic**

Feature

Feature

Feature

User story

Product backlog item

Requirement

Bug

Bug

Bug

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 additional 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, items in the closed and done state will 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.

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 Manger.
- 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 are not meant to be created manually and therefore are added to the Hidden Types category. Work item types that are added to the Hidden Types category don't appear in the menus used to create new work items.

> **NOTE**
>
> **Feature availability**: If you upgraded your project from TFS 2013 or an earlier version to a later version of TFS, you might have to add WITs that didn't exist in the earlier versions. For more information, see Configure features after a TFS upgrade.
>
> The following WITs were added with the indicated TFS version:
>
> - Shared Parameters added with TFS 2013.2
> - Test Plan and Test Suite added with TFS 2013.3

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 which test suites are defined for a test plan. However, these objects aren't connected to each other through link types. You can 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 that project. 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 a TFS upgrade

If you have additional questions, see Azure DevOps support page.

Agile process

2/7/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 are based on Agile principles and values.

In addition to the WITs, teams have access to a set of shared work item queries to track information, analyze progress, and make decisions.

> **NOTE**
> You can customize the work tracking system for your project based on the Agile process by creating and customizing an inherited process and applying that process to your project. To learn more, see Inheritance process model.

> **NOTE**
> The latest version of the Agile process uploads automatically when you install or upgrade to the latest version of TFS. You can customize projects and use the Process Template Manager to upload and download process templates.
>
> The following WITs are available as follows: Epic, TFS 2015 and later versions; Shared Parameters, TFS 2013.2 and later versions; and Test Plan and Test Suite, TFS 2013.3 and later versions.
>
> Additional artifacts, such as SQL Server reports and SharePoint dashboards, are only available when you connect to a project from an on-premises TFS. 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 on 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 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.

Descriptions of predefined queries are listed later in this article.

You can open these queries from the **Work/Queries** page in the web portal or from the **Work Items** page in Team Explorer. You can modify a query using the query editor to apply different filter criteria. Also, you can add queries to team dashboards.

From Team Explorer, you can open any work item query in Excel to perform bulk edits.

Quick tips on shared queries

If you are new to Azure Boards and TFS 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. 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 or Project, 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.

Monitor progress

All processes—Agile, Scrum, and CMMI—support building status and trend charts and dashboards. In addition, 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 open a shared 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 a number of Agile reports. For these reports to be useful, teams must perform 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.

SharePoint portal dashboards

You can access Agile process dashboards displayed through SharePoint. These dashboards display project data, support investigation tasks, and help teams to perform common tasks quickly. These dashboards support the display of web access parts for listing work items and reports that were built in the Analysis Services cube.

To use SharePoint dashboards your project must have a project portal configured and the project portal must point to a SharePoint site.

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
- Plan a sprint - to start working in Scrum
- Kanban - to start working in Kanban
- Excel or Project - to create a work breakdown structure

For more information on Agile tools:

- Team assets

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 TFS server. For Azure Boards, the latest version is always used. Starting with TFS 2012, the `version` element was added to the process template to support versioning of the templates. This element specifies a major and minor version. Prior to this change, the version was specified within the process template name.

TFS VERSION	AGILE PROCESS NAME	MAJOR VERSION
TFS 2018	Agile	16
TFS 2017	Agile	15
TFS 2015	Agile	7
TFS 2013	MSF for Agile Software Development	7
TFS 2012	MSF for Agile Software Development 6.0	6
TFS 2008	MSF for Agile Software Development - v4.*n*	

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 additional iterations, you can modify these queries to specify a different iteration and then save them to additional 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 are not 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 are not 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 are not 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 are not closed and sorts them by priority and then ID,
User Stories without Test Cases	Lists all user stories that do not 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 should not be assigned any hours. For more information, see Address inaccuracies published for summary values.

Workbooks

You can use the following Excel workbooks to review open issues and to rank and assign untriaged work items. Workbooks are only available when your project has been configured with a SharePoint portal. Each workbook references a shared query.

- The Issues workbook uses the Open Issues shared query.
- The Triage workbook uses the Bug Triage shared query.

Because these queries support workbooks, if you change these queries, it will affect those workbooks that use them.

Agile process work item types and workflow

2/7/2019 • 13 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Teams use the work item types (WITs) provided with the Agile process to plan and track progress of software projects. Teams define user stories to manage the backlog of work and then, using the Kanban board, 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 can map user stories to features. When teams work in sprints, they define tasks which 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**, teams can use the forecast feature and velocity charts to estimate future sprints or work efforts. By prioritizing the user stories on the backlog page (which is 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. 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.
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 additional guidance, 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 cannot ship without the feature. • **2**: Product cannot 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

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 when you click your cursor within the each text box that can be formatted.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** or use the at-mention icon. and select or type their name. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

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

Use the pull-request id icon to link to a pull request.

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.

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

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.

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 additional swim lanes or columns. For additional 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 do this 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, 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 Agile processes, teams forecast work and define tasks at the start of each sprint, and each team member performs a subset of those tasks. Tasks can include development, testing, and other kinds of work. For example, a developer can define tasks to implement user stories, 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

Original Estimate	The amount of estimated work required to complete a task. Typically, this field doesn't change after it is 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, you can update these fields using Project.

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 a number of 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 web portal, Visual Studio, or when testing with Test Manager.

Definitions for common WIT 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 WITs 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 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
- Plan a sprint - to start working in Scrum
- Kanban - to start working in Kanban
- Excel or Project - to create a work breakdown structure

For more information on Agile tools:

- Team assets

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 have 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	User story Bug Feedback Request Task Test Case	• No restrictions.
Implementation	User story Task	• Allows only **Parent** and **Child** links between user stories and tasks. • Excludes links to work items in other projects.
Links	Issue Shared steps	• 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 to user stories. • Excludes links to work items in other projects.
Storyboards	User Story	• Allows only **Storyboard** links.
Test Cases	User story Bug	• Allows only **Tested By** links. • Allows links only to test cases. • Excludes links to work items in other projects.
Tested User Stories	Test case	• Allows only **Tests** links. • Allows links only to user stories. • Excludes links to work items in other projects.

Scrum process

2/7/2019 • 7 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 are based on Scrum principles and values.

In addition to the WITs, teams have access to a set of shared work item queries to track information, analyze progress, and make decisions. If you work from an on-premises TFS, you also have access to additional SQL Server reports and SharePoint dashboards.

> **NOTE**
>
> You can customize the work tracking system for your project based on the Scrum process by creating and customizing an inherited process and applying that process to your project. To learn more, see Inheritance process model.

> **NOTE**
>
> The latest version of the Scrum process uploads automatically when you install or upgrade to the latest version of TFS. You can customize projects and use the Process Template Manager to upload and download process templates.
>
> The following WITs are available as follows: Epic, TFS 2015 and later versions; Shared Parameters, TFS 2013.2 and later versions; and Test Plan and Test Suite, TFS 2013.3 and later versions.
>
> Additional artifacts, such as SQL Server reports and SharePoint dashboards, are only available when you connect to a project from an on-premises TFS. Other resource requirements apply.

In addition to WITs, reports, and dashboards, teams have access to a set of shared work item queries to track information, analyze progress, and make decisions.

Plan and track work with Scrum

You build your project plan by creating a backlog of work items that represent the features, requirements, user stories, or other work to perform. 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 on 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

To list work items for a current sprint or the product backlog, use the shared queries that the Scrum process provides.

Descriptions of predefined queries are listed later in this article.

You can open a query from the work items page and then use the query editor to apply different filter criteria. Also, you can add queries to team dashboards. From Team Explorer, you can open any work item query in Excel to perform bulk edits.

> **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.

Quick tips on shared queries

If you are new to Azure Boards and TFS 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. 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 or Project, 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.

Monitor progress

All processes—Agile, Scrum, and CMMI—support building status and trend charts and dashboards. In addition, several charts are automatically built based on the Agile tools you use. These charts display within the web portal.

Scrum process and light-weight charts

To get started, you can open a shared 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 a number of Scrum reports. For these reports to be useful, teams must perform 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.

SharePoint portal dashboards

You can access the Release dashboard displayed through the SharePoint project portal. This dashboard displays project data, support investigation tasks, and help teams to perform common tasks quickly.

To access this dashboard, your project must have a project portal configured and the project portal must point to a SharePoint site.

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
- Plan a sprint - to start working in Scrum
- Kanban - to start working in Kanban
- Excel or Project - to create a work breakdown structure

For more information on Agile tools:

- Team assets

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 TFS server. For Azure Boards, the latest version is always used. Starting with TFS 2012, the `version` element was added to the process template to support versioning of the templates. This element specifies a major and minor version. Prior to this change, the version was specified within the process template name.

TFS VERSION	SCRUM PROCESS NAME	MAJOR VERSION
TFS 2018	Scrum	16
TFS 2017	Scrum	15
TFS 2015	Scrum	3
TFS 2013	Microsoft Visual Studio Scrum	7
TFS 2012	Microsoft Visual Studio Scrum 2.0	2

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 additional sprints, you can modify these queries to specify the latest sprint and save them to additional 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.

Scrum process work item types and workflow

2/7/2019 • 12 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 which 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 additional guidance, 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 when you click your cursor within the each text box that can be formatted.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** or use the at-mention icon. and select or type their name. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

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

Use the pull-request id icon to link to a pull request.

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.

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

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.

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 is 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 additional swim lanes or columns. For additional 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 performs 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 a number of 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 WIT 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 WITs 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 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
- Plan a sprint - to start working in Scrum
- Kanban - to start working in Kanban
- Excel or Project - to create a work breakdown structure

For more information on Agile tools:

- Team assets

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 have 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.

Feedback and support

We welcome your feedback.

For product related inquiries, review our User Voice page or report a problem at Developer Community. Also, you can get advice on Stack Overflow, or get support via our Support page.

Follow us on **Twitter @AzureDevOps**.

CMMI process

2/7/2019 • 11 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 are based on the Capability Maturity Model Integration (CMMI) process.

In addition to the WITs, teams have access to a set of shared work item queries to track information, analyze progress, and make decisions. If you work from an on-premises TFS, you also have access to additional .

> **NOTE**
>
> You can customize the work tracking system for your project based on the CMMI process by creating and customizing an inherited process and applying that process to your project. To learn more, see Inheritance process model.

> **NOTE**
>
> The latest version of the CMMI process uploads automatically when you install or upgrade to the latest version of TFS. You can customize projects and use the Process Template Manager to upload and download process templates.
>
> The following WITs are available as follows: Epic, TFS 2015 and later versions; Shared Parameters, TFS 2013.2 and later versions; and Test Plan and Test Suite, TFS 2013.3 and later versions.
>
> Additional artifacts, such as SQL Server reports and SharePoint dashboards, are only available when you connect to a project from TFS. 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 on 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

You can use shared work item queries to list work items based on their type, such as change requests, bugs, tasks, and requirements.

Descriptions of predefined queries are listed later in this article.

You can open a query from the work items page and then use the query editor to apply different filter criteria. Also, you can add queries to team dashboards.

> **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.

From Team Explorer, you can open any work item query in Excel to perform bulk edits.

Quick tips on shared queries

If you are new to Azure Boards and TFS 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. 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 or Project, 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.

Monitor progress

All processes—Agile, Scrum, and CMMI—support building status and trend charts and dashboards. In addition, 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 open a shared 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 a number of CMMI reports. For these reports to be useful, teams must perform

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.

SharePoint portal dashboards

You can access Agile process dashboards displayed through SharePoint. These dashboards display project data, support investigation tasks, and help teams to perform common tasks quickly. These dashboards support the display of web access parts for listing work items and reports that were built in the Analysis Services cube.

To use SharePoint dashboards your project must have a project portal configured and the project portal must point to a SharePoint site.

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
- Plan a sprint - to start working in Scrum
- Kanban - to start working in Kanban
- Excel or Project - to create a work breakdown structure

For more information on Agile tools:

- Team assets

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 TFS server. For Azure Boards, the latest version is always used. Starting with TFS 2012, the `version` element was added to the process template to support versioning of the templates. This element specifies a major and minor version. Prior to this change, the version was specified within the process template name.

TFS VERSION	CMMI NAME	MAJOR VERSION
TFS 2018	CMMI	16
TFS 2017	CMMI	15
TFS 2015	CMMI	7
TFS 2013	MSF for CMMI Process Improvement 2013	7
TFS 2012	MSF for CMMI Process Improvement 6.0	6
TFS 2008	MSF for CMMI Process Improvement - v4.*n*	

For a summary of updates made to process templates, see Changes made to process templates.

Additional 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 does not 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. You should adapt this guidance to your own situation, which will depend on the type and history of the product that you are 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.
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.

SHARED QUERY	DESCRIPTION
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.
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.

Workbooks

You can use the following Excel workbooks to review open issues and to rank and assign untriaged work items. Workbooks are only available when your project has been configured with a SharePoint portal. Each workbook references a shared query.

- The Issues workbook uses the Open Issues shared query
- The Triage workbook uses the Untriaged Work Items shared query

Because these queries support workbooks, if you change these queries, it will affect those workbooks that use them.

CMMI process work item types and workflow

2/7/2019 • 14 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 additional CMMI processes, teams can track change requests, risks, issues, and notes captured in review meetings. If you are 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. Alternatively, you can bulk add requirements using Excel or Project.

Later, you can open each requirement to provide more details and estimate its size.

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 additional guidance, 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 which 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 additional guidance, 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 is 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. These fields are filled in by Microsoft Project when you use it for scheduling. You can specify work in hours or in days. There are no inherent time units associated with this field.

Priority (Required)	A subjective rating of the requirement as it relates to the business. Allowed values are: • **1**: Product cannot ship without the item. • **2**: (default) Product cannot 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 when you click your cursor within the each text box that can be formatted.

Use the **@mention** control to notify another team member about the discussion. Simply type **@** or use the at-mention icon. and select or type their name. You can bring a group into the discussion by typing **@** and the group name, such as a team or security group.

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

Use the pull-request id icon to link to a pull request.

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.

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

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.

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.

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 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 additional swim lanes or columns. For additional customization options, see Customize your work tracking experience.

Map requirements to features

hen 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 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 additional 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 performs 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 additional guidance, 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 is assigned to `type="Activity"` in the ProcessConfiguration file. (2) For additional guidance, see Implement development tasks.
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 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 a number of 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

In addition to 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 WIT 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 WITs 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 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

- Plan a sprint - to start working in Scrum
- Kanban - to start working in Kanban
- Excel or Project - to create a work breakdown structure

For more information on Agile tools:

- Team assets

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 have 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.

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	Requirement Bug Change Request Feedback Request Issue Review Risk Shared Steps Task Test Case	• No restrictions.
Implementation	Task	• Allows only **Parent** and **Child** links between requirements and tasks. • Excludes links to work items in other projects.
Links	Code Review Request	• Allows only **Parent** and **Child** links to Code Review Response work items. • Excludes links to work items in other projects.

Requirements	Change Request	• Allows only **Affects** link type to link change requests to requirements. • Excludes links to work items in other projects.
Stories	Feedback Response	• Allows only **Related** links to requirements. • Excludes links to work items in other projects.
Storyboards	Requirement	• Allows only **Storyboard** links.
Test Cases	Requirement Bug	• Allows only **Tested By** links. • Allows links only to test cases. • Excludes links to work items in other projects.
Tested Requirements	Test case	• Allows only **Tests** links. • Allows links only to requirements. • Excludes links to work items in other projects.

Background to CMMI

1/25/2019 • 8 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 current CMMI product suite at the time of this writing. This model is extremely stable and should continue to be current well beyond 2010. You may also find "CMMI Distilled: A Practical Introduction to Integrated Process Improvement" to be a useful and accessible book about the topic.

The CMMI started life in 1987 as the Capability Maturity Model (CMM), a project at the Software Engineering Institute, which is a research center at Carnegie-Mellon University. This center was established and funded by the United States Department of Defense. The CMM for Software was first published in 1991 and is based on a checklist of critical success factors in software development projects during the late 70s and early 80s. The model is also informed by research at International Business Machines (IBM) Corporation and 20th-century quality assurance leaders Philip Crosby and W. Edwards Deming. Both the name, Capability Maturity Model, and the five levels in the Staged Representation (as discussed later in this article) were inspired by Crosby's Manufacturing Maturity Model. Applied mainly to defense programs, CMM has achieved considerable adoption and undergone several revisions and iterations. Its success led to the development of CMMs for a variety of subjects beyond software. The proliferation of new models was confusing, so the government funded a two-year project that involved more than 200 industry and academic experts to create a single, extensible framework that integrated systems engineering, software engineering, and product development. The result was CMMI.

The most important thing to understand about the CMMI-DEV is that it is a model. It is not a process or a prescription to be followed. It is a set of organizational behaviors that have proven to be of merit in software development and systems engineering. Why use such a model? What is its purpose? And how best should it be used? These are critical questions and are perhaps the most misunderstood issues with CMMI.

Why use a model?

Without a model of how our organizations work, which functions they need, and how those functions interact, it is difficult to lead efforts to improve. A model gives us an understanding of discrete elements in our organizations and helps us formulate language and discussion of what needs to be improved and how such improvement might be achieved. A model offers the following benefits:

- provides a common framework and language to help communicate
- leverages years of experience
- helps users keep the big picture in mind while focusing specifically on improvement
- is often supported by trainers and consultants
- can provide a standard to help solve disagreements

What is the purpose of the CMMI model?

The textbook will tell you that the purpose of the model is to assess the maturity of an organization's processes and to provide guidance on improving processes that will lead to improved products. When talking directly with people from the Software Engineering Institute, you might hear them say that the CMMI is a model for risk management and indicates an organization's ability to manage risk. This indication is evidence for the likelihood that the organization can deliver on its promises or deliver products of high quality that are attractive to the

market. Another way to think of this is that the model provides a good indicator of how an organization will perform under stress. A high maturity, high capability organization will take unexpected, stressful events in its stride, react, change, and proceed forward. A low maturity and lower capability organization will tend to panic under stress, blindly follow obviated procedures, or throw out all process altogether and retrench back to chaos.

The CMMI has not proven a good indicator of the economic performance of an organization. Although higher maturity organizations may manage risk better and be more predictable, there is evidence of risk aversion among higher maturity firms. This 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, therefore, 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 exhibit the quantitative management and optimizing behaviors, and lower maturity organizations, which are merely managed or following defined processes. Higher maturity organizations exhibit 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 does not get in the way. Where low maturity organizations tend to exhibit 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 second, the Continuous Representation, models process capability within each of the 22 process areas individually, allowing 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. Of course, 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 because implementers may forget that the CMMI is not a process or workflow model but provides goals for process and workflow to achieve. Meeting those 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 seems to have some greater success as a guide to process improvement, and 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 does not have artificial goals that are determined by maturity levels. The Continuous model also more naturally lends itself to applying process improvement in the 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 CMMI model is divided into 22 process areas, which are listed in the following table:

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
OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training
PI	Product Integration

ACRONYM	PROCESS AREA
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 sub-practices of generic and specific practices and typical work products.

It is very important that we understand that only generic and specific goals are required. Everything else is provided as a guide. The examples of the expected and informative components that are given in the CMMI literature are very often pulled from large space and defense-systems integration projects. These projects are run by companies that sponsor and support the Software Engineering Institute at Carnegie-Mellon University. These projects may 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

Project management (CMMI)

1/25/2019 • 4 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can use the Project Management section of the MSF for 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, objective 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 expended 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.

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 is needed to fulfill the feature requirements that are scheduled for that iteration. The iteration plan can be viewed through the sprint backlog page.

  Iterative work does not 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 all of the sales part of a shopping Web site for the first third of the project, all of the warehouse system in the second third, and all of 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 is 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 has not 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 are not 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 perform 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 this is not always necessary. If the teams do not 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

Project activities

1/25/2019 • 4 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 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 inception

Inception includes defining the project vision, which states what users will be able to 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 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 approximately estimated. The difference between general requirements and product requirements is an important one, and this is a significant activity. 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 subsequent 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 performed 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 objective 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. Just 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 are not 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 performing manual tests.

Related articles

- Background to CMMI
- CMMI process

Project inception

1/25/2019 • 6 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You arrange the basic resources of the project in an initial stage that is named Project Inception.

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. You should 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.

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 will be able to perform 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.

In addition, 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 substantial 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 make accurate estimates of 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 is one reason why a good product plan schedules some work on every principal component at an early stage.

Is this a scope-driven or date-driven project?

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. On the other hand, 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 this. Planning overhead is increased, and schedule buffering as a contingency against poor estimation will push the delivery date out more, which increases costs. Therefore, before you decide that you have a scope-driven project, be absolutely sure. It is 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 is 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 objective 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 is required. A communication plan is a useful tool for all staff, particularly 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. In addition to 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 is still a

high-level plan that defers many detailed scheduling decisions. This 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.

Additional 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 (CMMI)

1/25/2019 • 7 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The desired 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. Therefore, the plan provides an overall shape and is subject to review and refinement before the start of each iteration.

Gathering and modeling the requirements

This activity is about discussing what the system should do, with business stakeholders, prospective users, and subject matter experts. It is important to understand the business context. If you have 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.

Creating incremental product requirements

The requirements as you have gathered them from your customers are not directly appropriate for the purpose of 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; warehouse schedules delivery; and so on. These steps, or an equivalent activity diagram, are not incremental requirements.

Instead, the first increment of your system might offer only one item for sale, deliver to only one address, and perform only a test transaction with the payment service. The second increment might provide a catalog that consists of a simple list. Later increments might 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.

If you work with an existing product, the principle is the same, but you start from the existing functionality. If you are unfamiliar with its internal design, the cost of updates can be difficult to estimate. It is worth being liberal with your estimates for the earlier changes.

For more information, see Develop requirements.

Entering and editing 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.

Estimating 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.

Assigning product requirements to iterations

Representatives of the business stakeholders and the development team should work together to assign product requirements to iterations. Typically, you do this 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 do not 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 a variety of information, as the following illustrations show:

Some guidelines on prioritization

Many detailed schemes exist for prioritization. We will examine some of these 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 a simple 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, do not 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} is not 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 are truly optional add-ons.

2. Prioritize technical risk.

   If a scenario includes a technically risky element, develop it early in the schedule. Take a "fail early" approach to risk. If something cannot be accomplished, you want to know this 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 will not be sure about some requirements. It is difficult to predict what product behavior will work best in the business context. Prioritize work that is likely to reduce the uncertainties. This 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 to determine the requirements that can potentially bring more value to the customers earlier. Prioritize these requirements into earlier iterations. This 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 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 should be able to articulate 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 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, we want to prioritize the reduction of risk because of the possibility of failure. We want to prioritize common functionality because it is 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.

Planning 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 is 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.

Revising 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

1/25/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can use change request work items to track and control all changes to the product and supporting systems. All change requests are initiated 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 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 are 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 is important that the change request queue does not 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 that input to the change request work item. If the change requires changes to other planning documents, note that in the change request, and change those documents as appropriate. This will maintain the revision history and enable everyone to see the details. This 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 does not 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

1/25/2019 • 5 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Risk implies that actual outcomes may vary, sometimes significantly, from desired outcomes. Both the probability of this variance and the degree of variance between actual and desired 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 desired 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 implies that a list of risks is a list of potential future events. Each risk describes some event that may occur in the future. It may 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.

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 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 in the event of an occurrence.

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 have been sufficiently analyzed, it is 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 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.

You should 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 is 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 is 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 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.

Iteration activities

1/25/2019 • 5 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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.

At the start of 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 encounters 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 is 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 do not 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 will not be completed by the end of the iteration. In that case, the project manager or team leader initiates 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 are not considered during an iteration.

Triage

Some team members (usually 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 will know whether it was an improvement.
- When you will make that assessment.
- What you will do as a result.

Integration

If this project is part of a larger program, each team performs 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 perform 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 performing the merge, automatic and manual tests are run. This 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 perform 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 will have to perform another forward integration.

Preparing for the Next Iteration

Toward or at the end of an iteration, several project management activities are performed. These include reviewing risks and reviewing the plan with regard to change requests and changed development estimates.

For more information, see Project activities.

Plan an iteration (CMMI)

1/25/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Developing software in iterations means that 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 even the first iteration. Iterations let you receive feedback early so that 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 perform. 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

You should 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 allocate 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 should then be fed into 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 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 will want to pay extra attention to the Remaining Work, Unplanned Work, and Requirements Overview to make sure that the iteration is tracking against expectations.

Additional 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 issues (CMMI)

1/25/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can use the issue work item to help you track problems with the project plan and its activities and tasks. Issues are not 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. Resolving issues is not normal project work. Therefore, it must be tracked and given special attention. Tracking these project problems with issue work items and using 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 will 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 implies that the organization intends that the issue should not recur. This 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 performed 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 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 has not 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 expedite resolution and to avoid additional delay.

Additional 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

1/25/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The Engineering section of the MSF for CMMI Process Improvement guidance covers the value-added activities 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, and Validation. All of these are model level 2 or 3 process areas.

For more information about CMMI see Background to CMMI.

|**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|

Develop requirements

1/25/2019 • 16 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Requirements describe what the stakeholders expect from the product. You should 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 should neither discuss nor 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 does not attempt to replicate the very 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, should not be performed in strict order. Develop a domain model while you are 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.

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 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, therefore, 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 just 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, and 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.

- Do not 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, 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 of the request on dependent projects and users, and estimate the cost with regard to changes in your code. If a change request is accepted, update the requirement.
- Update the tests to conform to every requirements change.
- Designate 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 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 does not perform 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 will the users be able to do that they could not 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 will the product's users be able to do that they could not 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. For example, this would be a good title:

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 is recommended to use a sketch style that makes it clear that this is not 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.

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 her 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 in terms of "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.

Organizing 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 very 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 particularly 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, therefore, 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.

Modeling behaviors

Draw activity diagrams to summarize 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.

Modeling concepts

Draw domain class diagrams to describe the important entities and their relationships that are mentioned in the scenarios. For example, the DinnerNow model 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 do not 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 should not refer to the same thing as products. If the class diagram shows that a menu item belongs to exactly one menu, the scenarios should not 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."

In addition, it is 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, a customer focus group, or 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 are 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 perform, 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 is not valid, it is revised so that it is 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 does not 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 practicable, it should be tested in real conditions.

Inspecting and editing 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.

Additional 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

1/25/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 do not automatically tell you the plan.

this article outlines a method of obtaining a plan, starting from a set of requirements. This is just one method 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.

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 will 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 will be able to do with the product, that they could not do in previous iterations. There are no items, or very few items, on the plan that do not 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 is 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 is 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. Therefore, 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.

Scenario decomposition

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 lets you link test cases to the scenarios and thereby 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 does not 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. You must 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.

Quality of service features

Quality of service requirements are usually pervasive with regard to the software design. For example, security requirements are generally 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. For example, 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.

Product planning

Before the start of every iteration, hold a meeting to review the product plan. The first product planning meeting creates the plan, and subsequent 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 will not be stored in Team Foundation Server, but you can save the spreadsheet. When you open the spreadsheet again, click any cell in the work item table, and then click 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.

Iteration planning

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. The feature should not be marked complete until the linked test cases pass.

For more information, see Plan an iteration.

Create a solution architecture

1/25/2019 • 10 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 be used 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 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. In addition, 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 additional 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 additional 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 with regard 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 is important to address risk as early as possible in the project so that key strategic and architectural decisions can be made while it is 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, do not 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 objectives for each key scenario. These include objectives such as response time, throughput, and resource use. Identify the performance-related resources that have been budgeted to meet the performance objectives. 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 allocate 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 objectives. Consider tradeoffs that help meet performance objectives such as design and deployment alternatives. Reevaluate quality of service requirements if necessary.

Identify the scenarios that do not meet budget allocations. Measure the performance of the scenarios. Use prototyping in early iterations if code is not 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

1/25/2019 • 8 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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

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 is 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 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.

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 is also a good idea to document the rationale for key decisions that were made. For example, if there is 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.

Design Review

A design review is used to ensure that the new or revised design is technically accurate, 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 additional 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 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 Quality

Ensure that the design is testable. Does it build code that cannot be verified or validated in a reasonable manner? If so, you cannot 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 magnitude 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 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 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 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 this is the case, 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 Analyzing Application Quality by Using Code Analysis Tools and Enhancing Code Quality with Team Project 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 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.

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 do not 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.
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 will refactor the code.

Apply the refactoring incrementally, one change at a time. Change the code and all references to the modified area as necessary.

Perform unit tests so that the area remains semantically equivalent after the refactoring. Fix any unit tests that do not work. Perform code analysis, and fix any warnings. Perform unit tests again if code changes are made as a result 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 is 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 are not 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

1/25/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

To integrate development work into a component, a sub-system, 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 is not appropriate for your project, you should convene a meeting of technical stakeholders to discuss an appropriate plan for product integration. The meeting should 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

1/25/2019 • 5 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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.

Work item fields and attributes

2/7/2019 • 13 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Work item fields are used to track information. Each work item type (WIT) definition specifies the fields defined for that WIT. 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 that displayed on the work item form.
- **Reference name**: Specifies the name that you use when creating WIQL query or an ad hoc 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.

Field names

The field friendly name identifies each work item field. When adding a custom field, make sure the friendly name falls within these guidelines:

- Must be unique within the organization or project collection
- Must be 128 or fewer Unicode characters
- Can't contain any leading or trailing spaces, nor two or more consecutive spaces
- Must contain at least one alphabetic character
- Can't contain the following characters: `.,;'` :~\/*|?"&%$!+=()[]{}<>`.

For additional 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 256 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 based on the process name and the name of the field. For example, you add a field named Triage to the My Agile process, the reference name is **MyAgile.Triage**. 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.

Field attributes

There are a number of non-changeable and virtually hidden attributes for each work item field. The following table describes each attribute.

> **NOTE**
>
> The attribute listed in the first column of the table is supported through the **Fields - Get** REST API and **Work Item Types Field** - Get commands. The attribute listed in the second column is supported through the Work Item Field Explorer (WIFE) tool, and the FieldDefinition Properties. The attributes assigned to a field depend on the platform and version you use.

ATTRIBUTE (REST)	ATTRIBUTE (WIFE)	ATTRIBUTE TYPE	DESCRIPTION	CAN CHANGE?
	Allowed Values	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).	Yes
canSortBy	CanSortBy	boolean	Indicates whether you can sort query results with this field.	No

description	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.	Yes
	Id	boolean	Specifies the internal ID of the field.	No
	IsCloneable	boolean	Indicates whether the value defined for the field is copied when a user chooses to copy a work item. For example, the work item Title, Tags, and Description are copied, but the ID and History fields aren't copied.	No
	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.	No
	IsCoreField	boolean	Indicates whether this field is specified for all work item types.	No
	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	No
isIdentity	IsIdentity	boolean	Indicates whether this field is an Identity field. Identity fields are string fields used to store user identities.	No
	IsIndexed[1]	boolean	Indicates whether this field is indexed to support search.	No
	IsLongText	boolean	Indicates that the field can contain more than 255 characters, such as fields assigned a data type of PlainText, HTML, or History.	No
isPicklist[2]		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 Services and Picklist (String) or Picklist (Integer) type is selected. The value is set to False for inherited fields that define picklists.	No
isPicklistSuggested[2]			Indicates whether the field alliows users to enter their own values for a picklist. The value is set to True when a custom field is defined for Azure DevOps Services, Picklist (String) or Picklist (Integer) type is selected, and the checkbox for **Allow users to set their own values** is checked.	Yes
isQueryable	IsQueryable	boolean	Indicates if the field shows up within the set of fields you can add to filter a query (True) or not (False). Most fields are queryable.	No
	IsReportable [3]	boolean	Indicates if the reportable attribute is defined or set to anything other than **None**.	Yes

	IsUsedInGlobalWorkflow	boolean	Indicates if the field is defined within a global workflow.	No
	IsUserNameField	boolean	Indicates if the field is used to display an Identity field.	No
name	Name	string	Friendly name assigned to the field. The friendly name can't be changed for Azure DevOps Services, but can be changed for on-premises using the **witadmin changefield** command.	On-prem only
picklistId			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 Services and Picklist (String) or Picklist (Integer) type is selected.	No
	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.	On-prem only
readOnly		boolean	Indicates whether the field is set to read only. For Azure Cloud Services, only custom fields can be changed to be read only. System fields cannot be modified.	Yes
referenceName	ReferenceName	string	Specifies the reference name of the field definition.	No
	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 reports.	On-prem only
	ReportingName[3]	string	Specifies the label for a field when data appears in reports. If you do not specify a value, the field's friendly name is used.	On-prem only
	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 do not specify a value, the fields reference name is used.	On-prem only
supportedOperations		*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.	No
	SupportsTextQuery	boolean	Indicates whether the field supports text queries such as **Contains Words**, **Does Not Contains Words**.	No

	SystemType	string	Specifies the data type of the field, referencing the system name such as System.DateTime, System.String, and so on.	No
type	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	No
usage	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	No

Notes:

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 only supported for Azure DevOps Services and Azure DevOps Server 2019.
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.

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 (requires that you have installed a version of Visual Studio). Install the TFS Process Template editor from the Visual Studio Marketplace. You can 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.

To access the Work Item Field Explorer, you must install the Process Editor (requires that you have installed a version of Visual Studio) by installing the TFS Power Tools.

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.

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

Agile glossary

2/7/2019 • 11 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The Microsoft Agile glossary is a short dictionary of terms used in tracking work using Azure Boards and Team Foundation Server. See also:

- 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 is 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.

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

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 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.

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 TFS. To learn more, see On-premises process model.

Pick lists

Specifies an enumerated set of values that appear within a drop-down menu in a work item form and the **Value** column within the query editor. The method you use to customize a picklist varies depending on the field and the process model.

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 TFS. 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. 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.

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

Workflow is an integral aspect of a work item and is defined by it's corresponding work item type. The workflow determines the logical progression and regression of work items, tracking the status of work as it progresses from a New or Active state to Closed or Completed state. It also specifies the values that appear in the drop-down menus for the State and Reason fields. Learn more: Workflow states and state categories.

Related articles

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

Linking, traceability, and managing dependencies

1/31/2019 • 13 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

By linking work items and other objects, you can track related work, dependencies, and changes made over time. All links are defined with a specific link type. For example, you can use Parent/Child links to link work items to support a hierarchical tree structure. Whereas, the Commit and Branch link types support links between work items and commits and branches, respectively.

In a nutshell, you can:

- Link work items to other work items
- Link work items to test cases, and test cases to other test items and test results
- Link work items to code-related objects such as branches, commits, pull requests, and more
- Link select git code-related objects to builds
- Link work items to a web address or to a storyboard or a document on a network share
- Link work items to architecture diagrams (requires Visual Studio Enterprise edition).

Below, you can learn how to link objects and which link types to use. You can link objects from the web portal or Visual Studio Team Explorer.

> **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.

this article describes the link types available for your use. For details on linking work items, see Add link to work items.

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 additional 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.

- New web form
- Old web form
- Visual Studio
- Team Explorer Everywhere

From the new work item form, you can add a link using the **Related Work** section or from the **Links** tab.

Open a work item and choose 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 initiate 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 perform these actions:

- 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 perform 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.

The new web form is only available from TFS 2017 and later versions.

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 or Project to change the link structure. See Bulk add or modify work items with Excel and Create your backlog and tasks using Project.

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 builds

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, and Versioned Item.

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 section within the new work item form.

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 (git), changesets (TFVC), pull requests (git).

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 new work item form (available from Azure Boards and TFS 2017), you can drive your git development from the work item as shown in the following image.

Work items linked to GitHub commits and pull requests

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 used are **GitHub Commit** and **GitHub Pull Request**.

To learn more, see GitHub & Azure Boards.

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.

From the **Storyboards** tab, you can link storyboards that you created using PowerPoint Storyboarding or other application. The Storyboards tab and links control is available from the web and client work item form for TFS 2015.

For process templates associated with TFS 2015 and earlier versions, the **Storyboards** tab links control was added to those work item types used to defined requirements, user stories, or features. When you make changes to a linked storyboard, the work item continues to link to the file with the latest changes.

By using the Storyboard link type, you differentiate the link your 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. You can also link from a Power Point file to a work item using this link type as described in

Storyboard your ideas using PowerPoint.

Work items linked to architectural diagrams

You choose the **Model** link type when you want to link a work item to an architectural diagram created using Visual Studio Enterprise. This link type is only available when you work from Visual Studio Enterprise. With this link type, you can track tasks, test cases, bugs, requirements, issues, or other kinds of work that are associated with specific parts of your model.

To link a work item to a diagram, open the work item in Visual Studio, choose the **All Links** or **Other Links** tab and choose the **Model** link type. You can also open the linked diagram from the links tab.

Or, you can link a model element to a work item. See Link model elements and work items.

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 are 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 additional information, see these topics:

- Add link to multiple work items
- 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.

How workflow states and state categories are used in Backlogs and Boards

1/31/2019 • 5 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

All workflows consist of states, transitions, and reasons. Workflows are defined for a work item type (WIT). 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 upon its creation to closure. For example, the four main states defined for the User Story (Agile process) define a progression of four states, from New, Active, Resolved, to 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 of the user story, product backlog item, and requirement WITs are as shown.

Workflow states

USER STORY, AGILE PROCESS	PRODUCT BACKLOG ITEM, SCRUM PROCESS	REQUIREMENT, CMMI PROCESS

State categories

State categories, on the other hand, 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, and Complete.

Here's how the default, inherited states map to the state categories for all three system processes plus test case management WITs. The workflow states for Test Case, Test Design, and Test Suite are the same across all three

system processes.

CATEGORIES	AGILE	SCRUM	CMMI	TEST WITS
Proposed: Assign to states associated with newly added work items that should appear on the backlog. The first column on the Kanban or taskboard maps to a Proposed state.	New	New Approved To Do (Task)	Proposed	Design (Test Case)
In Progress: Assign to states that represent active work. Work items assigned to states mapped to this category will appear in the backlog (unless you choose to hide them) and make up the middle columns on the Kanban boards.	Active Resolved (Epic, Feature, User Story)	Committed Open (Impediment)	Active Resolved (Epic, Feature, Requirement, Task)	Active (Test Plan) In Planning (Test Suite) In Progress (Test Suite) Ready (Test Case)
Resolved: Assign to states that represent a solution has been implemented, but are not yet verified. Generally these states apply to bug WITs. Work items in a Resolved state appear on the backlog by default. The Agile tools treat the Resolved state category exactly the same as the In Progress state category.	Resolved (Bug)	n/a	Resolved (Bug, Issue, Review, Risk)	n/a
Completed: Assigned to states that represent work 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. Note that you can't modify states in this category nor can you add states to this category.	Closed	Done	Closed	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. > **Note:** You should avoid using the Removed state and Removed state category as they are in the process of being deprecated. Instead, you should Delete work items to remove them from the backlog.	Removed	Removed	n/a	n/a

When to add a State versus a Kanban column

Both States and Kanban columns are used 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 WITs that reference that process.

Also, by adding custom states to support those workflow states that several teams want to track, you avoid the confusion that can arise when team's create a query based on a Kanban column. 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 work around is to formalize the columns by adding custom states which 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 have the option to automatically complete those work items when you successfully complete the PR. As shown in the following image, all you have to do is 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 WIT:

- The work item, whose WIT 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 TFS 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.

Related articles

- Lead Time and Cycle Time control charts (widgets)
- Customize a workflow for a process

- Lead Time and Cycle Time control charts (widgets)

- Change the workflow for a work item type

- ProcessConfiguration XML element reference
- Customize your work tracking experience

- Change the workflow for a work item type
- ProcessConfiguration XML element reference
- Customize your work tracking experience

New work item tracking experience

2/7/2019 • 4 minutes to read • Edit Online

TFS 2018 | TFS 2017

> **IMPORTANT**
>
> The new form and its corresponding features are available from the web portal for TFS 2017 and later versions.
>
> - For TFS 2017 users, the new form is automatically available when you add projects to a new collection. For existing projects, an admin is required to enable the new form.
> - For TFS 2015 users, the new form isn't available. You must upgrade to TFS 2017 or later version to access the new form.

Work items have received a facelift. To provide a fresher, more modern experience to tracking work, we've replaced our "old and clunky" form. Along with the noticeable responsive form layout, with the nre form you now have access to these features:

- **Discussion control** with a discussion indicator in the banner that summarizes the number of comments in the discussion and from which you can quickly jump to the discussion and add your comments; this control also supports:
  - **@mention** which allows you to quickly add team members to receive an email notification with your comments and a link to the work item in which you mentioned them.
  - **#ID** which provides a quick way to create links to other work items within the Discussion control
- **Related Work control** that provides a quick method to link to and view linked work items
- **Development control** where you're able to engage with code changes, pull requests, and builds through interactive experiences as described in Drive Git development for a work item
- **History page** with improved readability and with indicators for important changes such as reassignment, change of state, discussion comment, and more
- **Attachment control** that supports drag-and-drop of attachments and attachment preview.
- **Custom control support** that enables adding custom field, group, and page extensions such as multi-value control and toggle control
- Easy copy to clipboard functionality.

Here's what you'll see when you preview the new form.

For guidance in using the new form features, see Add work items to plan and track your project.

New form features

Consistent look and feel

First off, you'll notice that the form reflects the color of your work item type—bug, user story, or task. And, we've done away with duplicate titles to cut down on clutter. Whether you're working from a backlog, board, or query—your experience remains the same.

Great consumption experience

Your data has more structure and organization. Plus, you can focus more easily on areas of interest by collapsing groups to hide details you don't care about. And as you resize the form, it quickly responds to provide you with a great viewing experience even within a limited screen width.

More viewing area to support core experiences

All the core tabs—Details, History, Links, and Attachments—now have more screen space. See History and auditing for details on the new History tab.

Discussion

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

Click the discussion icon, which indicates how many comments have been added, to move your focus to the discussion section. Click the full screen icon to expand the display of the discussion section within the form.

Within the discussion section, you can 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.

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

Access to other tasks

Looking for the toolbar task options? You access them now through the ··· actions icon.

Switch to the new experience

Depending on the opt-in mode your admin selected, you'll either automatically see the new web form when you open a work item, or you'll have the choice to switch to the new form.

When you switch, all forms displayed through the web portal for all work item types will now display with the new experiences.

As this switch is set on a per-user basis, other users will continue to see the old form until they also choose to switch.

Switch back to the old form

If your admin has enabled opt-in to support switch back. Simply choose the ··· actions icon to open the context menu to switch back to the old layouts.

If you don't see this option, then the new form has been enabled for all users, and switching back is managed by your admin.

When you switch back, all forms revert to the layout they had previously. If you do choose to switch back, we'd appreciate hearing what didn't work for you with the new form layout.

Related articles

The new form also supports customization through the user interface with the Inheritance process model. To learn more about process models and what's supported with each, see Customize your work tracking experience.

To manage the rollout of the new form or customize it, see:

- Manage new form rollout
- Customize the new form (On-premises XML process model)

For guidance in using the new form, see:

- Add work items to plan and track your project
- Drive Git development for a work item

For news of upcoming features, or extending the work tracking experience, see these resources:

- Azure DevOps Services Features Timeline
- REST API, Work Item Types

Will this impact Visual Studio work item forms?

Not at this time.

Query by titles, IDs, and rich-text fields

2/11/2019 • 6 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

When you want to find work items based on a keyword or phrase, you can do so by using single-line text (String), multi-line text (PlainText), and rich-text (HTML) fields.

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)	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

Notes:

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.

Keyword or phrase query

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.

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.

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

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. Also, 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 .

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 TFS, 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 indepth 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

Notes:

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
- Create 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
- Work item query language (WIQL) syntax
- Fetch work items with queries programmatically

SDK resources

To programmatically interact with queries, see Query for Bugs, Tasks, and Other Work Items.

Query by assignment or workflow changes

2/7/2019 • 14 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Workflow states support tracking the status of work as it moves from a new state to a closed or a 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 (WITs) 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 is 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 WITs. Jump to Workflow and Kanban query fields for the set of fields you can include in queries and which WITs 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 what the field data type is, see the 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)	= , <> , > , < , >= , <= , =[Field], <>[Field], >[Field], <[Field], >=[Field], <=[Field], Contains, Does Not Contain, In, Not In, In Group, Not In Group, Was Ever

Notes:

1. The **Boolean** data type field is supported for TFS 2017 and later versions.

Use the **In** and **Not In** operators when you want 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, team security group, or other security group. For more information, see Query fields, operators, and macros.

Identity based queries

You can 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 my (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 are not members of a group. Examples of groups are teams, built-in security groups, custom security groups, Active Directory groups, and work item categories.

Workflow change based queries

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`

Kanban board change queries

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.

> **NOTE**
> Kanban query fields are available with TFS 2015.1 or later versions.

For example, 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 that contains "Test"	`Board Lane _ Contains _ Test`

> **IMPORTANT**
> Work items that appear on more then 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.

Workflow and Kanban board fields

You can use the following fields to filter your queries or build reports. Some of these fields are populated with information as a work item progresses from one state to another, or you move an item in the Kanban board to a different column or swimlane. Several of these fields do not appear on the work item form, but they are tracked for those WITs listed in the following table.

For more information about field attributes, see Work item fields and attributes.

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE
Activated By[1, 2]	The name of the team member who created the work item or changed its status from closed, completed, or done state to a new or active state. Reference name=Microsoft.VSTS.Common.ActivatedBy	String (Identity)	All
Activated Date [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	DateTime	All
Assigned To[1, 2, 3]	The name of the team member who currently owns the work item. For additional information, see Note 1 on synchronization and person-name fields and Assign work items to a team member. Reference name=System.AssignedTo	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	String	Requirement Category[4]
Board Column Done	The current assignment of the work item to Doing (False) or Done (True) Kanban column. Only assigned when split-columns has been enabled for a Kanban board column. Reference name=System.BoardColumnDone	Boolean	Requirement Category[4]
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	String	Requirement Category[4]
Closed By[1, 2]	The name of the team member who set the state to closed, completed, or done. Reference name=System.ClosedBy	String	All
Closed Date	The date and time when a work item was closed. Reference name=Microsoft.VSTS.Common.ClosedDate	DateTime	All
Created By[1, 2]	The name of the team member who created the work item. Reference name=Microsoft.VSTS.Common.CreatedBy	String (Identity)	All

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE
Created Date	The date and time when a work item was created. Reference name=Microsoft.VSTS.Common.CreatedDate	DateTime	All
Reason [2, 3]	The reason why the work item is in the current state. Values are defined within the `WORKFLOW` section of the WIT definition using the `REASON` element. To modify the defined reasons, see Change the workflow for a work item type. Reference name=System.Reason	String	All (except Test Case and Shared Steps)
Resolved By [1, 2]	The date and time when the work item was moved into a resolved or done state. Reference name=Microsoft.VSTS.Common.ResolvedBy	String (Identity)	All
Resolved Date [2]	The date and time when the work item was moved into a resolved or done state. Reference name=Microsoft.VSTS.Common.ResolvedDate	DateTime	All
Resolved Reason [2]	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	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	String (Identity)	Code Review Response
State [2, 3]	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. Values are defined within the `WORKFLOW` section of the WIT definition using the `STATE` element. To add a custom state to Azure Boards, see Customize the workflow for a process. To add or modify States or the workflow for TFS, see Change the workflow for a work item type. Reference name=System.State	String	All

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE
State Changed Date	The date and time when the value of the State field changed. Reference name=Microsoft.VSTS.Common.StateChangeDate	DateTime	All

Notes

1. By default, the server synchronizes system-defined person-name fields with Active Directory or Azure Active Directory, if these are configured. 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.

2. Reportable field with attribute set to Dimension. Reportable data is exported to the data warehouse and can be included in Excel or SQL Server reports. For on-premises TFS, use the **witadmin changefield** command to change the reportable attribute for a field.
3. Indexed field. Enabling indexing for a field may increase the performance of finding work items whose queries specify that field. For on-premises TFS, use the **witadmin indexfield** command to change the index attribute for a field.
4. This field applies to all work item types that appear on the Kanban board. This includes all WITs added to the Requirement Category and may include those added to the Bug Category based on the team setting for Show bugs on boards and backlogs. If you want to modify a board-related field, such as Board Column or Board Lane, from the work item form, you must add it to the form. For more information, see Add and manage fields (Inheritance process model) or Add or modify a work item field (On-premises XML process model).

Workflow fields

You can use the following fields to filter your queries or build reports. Some of these fields are populated with information as a work item progresses from one state to another. Several of these fields do not appear on the work item form, but they are tracked for those WITs listed in the following table. For more information about field attributes, see Work item fields and attributes.

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE
Activated By [1, 2]	The name of the team member who created the work item or changed its status from closed, completed, or done state to a new or active state. Reference name=Microsoft.VSTS.Common.ActivatedBy	String (Identity)	All
Activated Date [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	DateTime	All

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE
Assigned To [1, 2, 3]	The name of the team member who currently owns the work item. For additional information, see Note 1 on synchronization and person-name fields and Assign work items to a team member. Reference name=System.AssignedTo	String (Identity)	All
Closed By [1, 2]	The name of the team member who set the state to closed, completed, or done. Reference name=System.ClosedBy	String	All
Closed Date	The date and time when a work item was closed. Reference name=Microsoft.VSTS.Common.ClosedDate	DateTime	All
Created By [1, 2]	The name of the team member who created the work item. Reference name=Microsoft.VSTS.Common.CreatedBy	String (Identity)	All
Created Date	The date and time when a work item was created. Reference name=Microsoft.VSTS.Common.CreatedDate	DateTime	All
Reason [2, 3]	The reason why the work item is in the current state. Values are defined within the `WORKFLOW` section of the WIT definition using the `REASON` element. To modify the defined reasons, see Change the workflow for a work item type. Reference name=System.Reason	String	All (except Test Case and Shared Steps)
Resolved By [1, 2]	The date and time when the work item was moved into a resolved or done state. Reference name=Microsoft.VSTS.Common.ResolvedBy	String (Identity)	All
Resolved Date [2]	The date and time when the work item was moved into a resolved or done state. Reference name=Microsoft.VSTS.Common.ResolvedDate	DateTime	All
Resolved Reason [2]	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	String	All (Agile, CMMI)

FIELD NAME	DESCRIPTION	DATA TYPE	WORK ITEM TYPE
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	String (Identity)	Code Review Response
State [2, 3]	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. Values are defined within the `WORKFLOW` section of the WIT definition using the `STATE` element. To add a custom state to Azure Boards, see Customize the workflow for a process. To add or modify States or the workflow for an on-premises server, see Change the workflow for a work item type. Reference name=System.State	String	All
State Changed Date	The date and time when the value of the State field changed. Reference name=Microsoft.VSTS.Common.StateChangeDate	DateTime	All

Notes

1. By default, the server synchronizes system-defined person-name fields with Active Directory or Azure Active Directory, if these are configured. 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.

   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.

2. Reportable field with attribute set to Dimension. Reportable data is exported to the data warehouse and can be included in Excel or SQL Server reports. For on-premises server, use the **witadmin changefield** command to change the reportable attribute for a field.

3. Indexed field. Enabling indexing for a field may increase the performance of finding work items whose queries specify that field. For on-premises server, use the **witadmin indexfield** command to change the index attribute for a field.

Related articles

- Query quick reference
- Work item fields and attributes

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

- Work item query language (WIQL) syntax
- Fetch work items with queries programmatically

SDK resources

To programmatically interact with queries, see Query for Bugs, Tasks, and Other Work Items.

Query by area or iteration path

2/7/2019 • 4 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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.

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.

In addition to 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.
@TeamAreas	Filter items based on area path(s) assigned to a specific team.

> **NOTE**
> The **@CurrentIteration** macro is supported for Azure Boards and TFS 2015 and later versions. The **@CurrentIteration +/- n** and **@TeamAreas** macros are supported for Azure DevOps Services and Azure DevOps Server 2019 and later versions.

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

Using the **Node Name** field, you can filter on work items assigned to area paths based on a keyword using the **Contains** operator. The **Node Name** specifies the leaf 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 leaf node contains the word "Azure".

Here's another example which 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 will not be able to specify it in Microsoft Project. To avoid this problem, define path names of no more than 10 characters, and do not nest nodes more than 14 levels deep.

You can't apply most field rules to the System.AreaPath and System.IterationPath fields. To learn more, see Apply a field rule.

The following fields do not 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
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 leaf 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 is not. 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 paths (aka sprints) and configure team iterations
- Set permissions and access for work tracking

Supported field rules

For the On-premises XML process model, you can apply rules to fields. However, for system fields (System.XXX), such as the Area Path and Iteration Path, you can specify only a small subset of rules, such as `HELPTEXT` and `READONLY` to fields.

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
- Work item query language (WIQL) syntax
- Fetch work items with queries programmatically

SDK resources

To programmatically interact with queries, see Query for Bugs, Tasks, and Other Work Items.

Query by history

2/7/2019 • 7 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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.

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

View the History of a work item

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 perform 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 use the query editor to add the **History** field to a query clause. Comments entered into the **History** field are queryable. Change history entries, such as which fields were changed, aren't queryable.

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 will not 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 will not 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 you user.

- 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.

> **NOTE**
>
> With TFS 2017 and later versions, the **History** field is no longer a rich-text field. To annotate the work item history, add to the **Description** or **Discussion** fields.

To view only the comments that were added to the log, choose the **Discussion Only** tab. To view all changes made to the item, choose the **All Changes** tab, and then choose the **show all changes** link for a specific date and time.

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 are 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. You can 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. Reference name=System.Rev, Data type=Integer	All
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 additional 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

Notes:

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

To learn more about creating and saving queries, see also:

- Query editor
- Query fields, operators, and macros
- Query by date or current iteration

Get history programmatically

See Work item tracking for ways to extract history information programmatically.

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
- Work item query language (WIQL) syntax
- Fetch work items with queries programmatically

SDK resources

To programmatically interact with queries, see Query for Bugs, Tasks, and Other Work Items.

Use @mentions in work items and pull requests

1/31/2019 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015

The **@mention** control allows you to quickly pull someone into a work item or pull request.

> **NOTE**
>
> The **@mention** control is available from TFS 2015 Update 2 and later versions.

For team members to receive notifications, you must configure an SMTP server.

When you're leaving a code comment in a pull request, you can enter **@** to trigger the **@mention** identity picker. From the identity selector, you see a list of those people that you've recently mentioned. Choose one of those names or enter the name of the person you are looking for to perform a directory search.

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 the search icon, and then select from the options listed.

To **@mention** a user you've never selected previously, just continue to enter the entire name to perform your search against the full directory.

Names of those that you mention appear in blue text. Choose the **@mention link name** to open the user's contact card, which can provide you additional context for why they were pulled into the conversation.

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. You can also use the **@mention** control in any text field on the work item form.

Related articles

- Work item form controls
- Pull requests

Use #ID to link to work items

1/25/2019 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015

Use the **#ID** control to quickly link objects to work items.

> **NOTE**
>
> The **#ID** special control feature is available from TFS 2015 Update 1 and later versions.

Link a pull request to a work item

When you are leaving a code comment in a pull request, you can enter **#** to trigger the **#ID** work item picker. The picker displays a list of 50 work items that you have recently modified or that are assigned to you.

You can narrow the list of suggested work items by entering keywords that match the work item type, ID, or title, or you can enter the exact work item ID.

To further filter the list, continue to enter keywords until you've found a match. You can enter up to five keywords.

Link to work items in pull requests, comments, and commits

You can also use the **#ID** control in pull request discussions, commit comments, changeset comments, and shelveset comments.

> **NOTE**
>
> Requires TFS 2015 Update 2 or a later version.

Link to work items from a Wiki page

Use the **#ID** control to link to a work item from within a Wiki page.

To learn more about the built-in wiki, see Add & edit wiki pages.

Related articles

- Link work items
- Save work with commits
- Pull requests
- Check in your work to the team code base

Add work item tags to categorize and filter lists and boards

1/31/2019 • 7 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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.

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 perform bulk modifications of tags.

> **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 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 About access levels.
- 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 create 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.

- 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 About access levels.
- 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 create 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.

> **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.

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.

From the web portal, open a work item and add a tag. Choose **Add** 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, simply 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 Add administrators, set permissions at the project-level or project collection-level.

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.

Use Excel to bulk modify tags.

> **NOTE**
>
> Bulk modify of tags from the Visual Studio client 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.

> **NOTE**
>
> Adding tags to a query is supported from TFS 2013.2 and later versions.

For example, here we query for all work items that are tagged either `Web` or `Service`.

> **TIP**
>
> You can use the **Contains** or **Does Not Contain** operators. To learn more about queries, see Create managed queries.

Show tags in your backlog or query results

Click **Column Options** to add the Tags field to the product backlog or a work item query. If the option doesn't appear, click the ··· to select it from the menu of options.

All tags that have been added to the listed work items appear.

1. Click **Column Options** to add the Tags field to the product backlog or a work item query. If the option doesn't appear, click the ··· actions icon to select it from the menu of options.

2. All tags that are assigned 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 clicking the filter icon.

Check the boxes of those tags that you want to filter on. Keep the OR selection to perform a logical OR for all the tags you selected. Or, click the AND option to perform a logical AND on all the selected tags.

1. Turn on filtering and choose a tag.

The list refreshes. Only those work items with the selected tag are displayed. Filtering the list disables add-a-backlog-item panel, stack ranking, and forecasting.

2. To apply a sub-filter, choose another tag. You filter successively by choosing from the set of tags that appear in the filter tag bar. To start your filter process over, choose All to show all tags.

3. To show all items, choose All or choose the Tag filter image on backlog and queries pages filter icon to turn filtering off.

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 3 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 3 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](

Requires TFS 2015.1 or later version.

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

Limits on number of tags

While no hard limit exists, creating more than 100K tags for a project collection can negatively impact performance. Also, the auto-complete 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.
```

Simply 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 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

1/31/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013 | Visual Studio | 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 fields. If you need to maintain the information in a different format, you can easily link to or attach a file.

Additional ways to share information include using dashboards, README files, and project Wikis.

Additional ways to share information include using dashboards, and SharePoint portals.

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 (as well as many others not listed here for the CMMI process).

- *Acceptance Criteria*
- *Description*
- *Steps*
- *System Info*

> **NOTE**
> The *History* field is no longer a rich-text field. To annotate the work item history, add your content to the Description or Discussion fields.

- *Acceptance Criteria*
- *Description*
- *History*
- *Steps*
- *System Info*

In addition, 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
- Team Explorer Everywhere

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 facilitate bringing others into the discussion or linking 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 for work item forms.

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.

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.

Link items

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.

Again, 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

You and your team can centralize reference materials by attaching files to your work items. That way they are 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.

Choose the or **Attachments** tab to attach a file with supplemental information.

- Browser
- Visual Studio
- Team Explorer Everywhere

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 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.

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.

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.

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.

> **TIP**
> 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. Also, 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.

You can edit, open, save, or delete an attachment by clicking an attachment and opening it's ··· actions menu.

Choose the **Attachments** tab icon to attach a file to the work item.

Choose the + plus icon to add an attachment. Choose an item and then choose the open icon to open the attachment or download icon to save a copy.

Attachment size

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 2GB.

Storyboard

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**
>
> 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.

- Browser
- Visual Studio
- Team Explorer Everywhere

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.

From the **Storyboards** tab, click **Start Storyboarding** to open Storyboarding with PowerPoint. Or, you can link to an existing storyboard.

To open PowerPoint with storyboarding, see Storyboard your ideas using PowerPoint.

Email a work item query list

A common way teams share information is through a list of work items. 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.

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.

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.

SharePoint project portal

You can use a SharePoint site to share and access documents from your project. Your team can use the SharePoint site, also referred to as the project portal, to share information in the following ways:

- Share data contained in reports or dashboards
- Share team progress using predefined or customized SharePoint dashboards
- Share documents, files, images
- Share team knowledge and processes using the SharePoint wiki.

To learn more, see Share information using the project portal.

Related articles

As you can see, there are many ways to share information using work items alone. See these additional tools and features to support planning, tracking, and sharing information with your team.

- Dashboards
- Add and edit a wiki

Marketplace extensions

You may find additional ways to share information and collaborate as a team by adding a Marketplace extension.

Bulk add or modify work items with Excel

2/13/2019 • 12 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

When you have a lot of work items to add or modify, using Microsoft Excel can save you time. 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 parent-child links.

You can also bulk add and modify work items using Microsoft Project.

In this article you'll learn:

- Connect to a project from Excel
- Add or edit work items and publish your changes to Azure Boards or TFS
- Refresh your Excel worksheet with the latest changes made to the work tracking data store
- Select user accounts
- Add hierarchically linked backlog items and tasks
- Work with different list types

Prerequisites

- 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 or TFS.

- Microsoft Excel 2007, Microsoft Excel 2010, or Microsoft Excel 2013
- Visual Studio 2013 or Visual Studio 2015 or Team Foundation Server Standalone Office Integration (free)
- Permissions to connect to the project.

To learn more about compatibility requirements, see Compatibility with Azure DevOps.

Add work items

1. If you don't have Microsoft Excel 2007 or a more recent version, install it. For Azure Boards and TFS 2017 and later versions, you'll need Excel 2010 or a later version.

2. If you haven't installed a version of Visual Studio (2010 or later) or the Team Foundation Server Standalone Office Integration (free), you'll need to install one of these versions to connect to an Azure DevOps or TFS project.

   > **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 supports connection to Azure Boards and TFS from Excel, Project, and the PowerPoint-based storyboarding tool.

3. 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 step 2 or TFS-Office integration issues.

   (1) Choose Team tab, (2) place your cursor in Cell A1, and then (3) choose **New List**.

> **TIP**
>
> If the **Team** ribbon no longer appears, you might need to re-enable it.

4. Connect to your project where you want to add work items. If you can't connect, get added as a team member.

If it is your first time connecting from Excel, you might have to add the URL to the list of recognized servers.

Azure Boards

(1) Choose **Servers...**, (2) choose **Add...**, (3) enter the URL of your Azure Boards organization, (4) check that the preview matches the URL that you entered, and then choose **OK**.

TFS

(1) Choose **Servers...**, (2) choose **Add...**, (3) enter the name of your TFS instance. As needed, change the Port number if your deployment uses a non-default port number. The Preview entry should display the correct URL for your deployment. (4) Choose **OK**.

Choose **Close** to close the Add/Remove servers dialog. From the Select a Team Foundation Server dialog, make sure the server you added is selected, and then choose **Connect**.

5. From the New List dialog, choose **Input list**.

6. Your worksheet is now bound to your project as a flat list. What this means is that you can add work items to the project from the worksheet or add work items to the worksheet from the project.

7. 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.

8. 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.

9. Also, note how you can open a work item in the web portal to add more information.

You can make changes to work items in Excel, Project, the web portal, or Visual Studio, or Team Explorer Everywhere.

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.

Select user accounts

NOTE

To access 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.

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.

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 in order to access the Select User feature.

2. Click a person-named field to activate the Select User feature in the team ribbon.

A person-named field is any field whose `syncnamechanges` attribute has been set to synchronize with Active Directory, Azure 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 those user accounts directly from the field.

TIP

Without the Select User feature active, you must enter user names exactly as they are in the database, or you'll receive data validation errors upon trying to publish.

Add linked backlog items and tasks

You can bulk add a nested list of work items, such as a work break down 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, as described in the following steps.

Here's how a three-level nested tree of items appears in Excel:

1. Follow steps 1 through 6 from the previous procedure.
2. Convert your flat list to a tree list by adding a tree level.

Notice how the list type is now labeled **Tree**, and an additional **Title 2** column has been inserted.

	A	B	C	D	E	F
1	Project: Fabrikam Fiber	Server: vs-2015-test\DefaultCollection		Query: [None]	List type: Tree	
2	ID	Title 1	Title 2	Work Item Type	State	Reason
3	100	Hello World web site		User Story	New	New
4	101	Welcome back		User Story	New	New
5	102	Resume		User Story	New	New
6	103	Change background color		User Story	New	New

3. Enter titles for backlog items under **Title 1** and for tasks, under **Title 2**. Also, select the corresponding work item type for each. Here we specify Task.

	A	B	C	D	E	F
1	Project: Fabrikam Fiber	Server: vs-2015-test\DefaultCollection		Query: [None]	List type: Tree	
2	ID	Title 1	Title 2	Work Item Type	State	Reason
3	100	Hello World web site		User Story	New	New
4			Change background color	Task	New	New
5			Change page layout	Task	New	New
6			Develop about page	Task	New	New
7	101	Welcome back		User Story	New	New
8			Add welcome back text	Task	New	New
9			Display user's name in header	Task	New	New

4. Just as before, publish your worksheet and notice how IDs are assigned to the new work items.

	A	B	C	D	E	F
1	Project: Fabrikam Fiber	Server: vs-2015-test\DefaultCollection		Query: [None]	List type: Tree	
2	ID	Title 1	Title 2	Work Item Type	State	Reason
3	100	Hello World web site		User Story	New	New
4	104		Change background color	Task	New	New
5	105		Change page layout	Task	New	New
6	106		Develop about page	Task	New	New
7	101	Welcome back		User Story	New	New
8	107		Add welcome back text	Task	New	New
9	108		Display user's name in header	Task	New	New
10	102	Resume		User Story	New	New

In the background, parent-child links are created for each task listed under a user story.

Useful tips when working with a tree list

- The plug-in 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 parent and child work items is blank.
  - 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 in an Excel tree list.
- You can use the Indent or Outdent 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** <number>, if it does not, add a tree level.
- 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.
- If you receive error TF208104, changes you made to the fields are published, but all changes you made to the link hierarchy are not published. At least one of the link relationships defined for the work item is locked by another process, such as Project Server integration. For more information, see Addressing Error TF208104: Hierarchical Link Relationship Is Locked.
- When you move a work item, make sure that you select the entire table row.

Work with different list types

Select your list structure based on these guidelines.

TASK	LIST STRUCTURE	LIST REFRESH
Create and publish many unrelated work items	Flat list	Input list
Perform bulk edits on many unrelated work items	Flat list	Query list or input list
Perform bulk edits on many work items and their dependent or related work items	Tree list	Query list
Perform top down planning and publish parent-child linked work items	Tree list	Input list
View and modify the hierarchy and parent-child link relationships of many work items	Tree list	Query list
Review reports based on a filtered set of work items that change over time	Flat or tree list	Query list

Enable the Tree commands

If the Tree group commands are not available, your worksheet is configured as a flat list. You can convert the flat list to a tree list as described in step 2 in Add backlog items and tasks and their parent-child links using a tree list.

Convert a tree to a flat list

First, publish whatever changes you have made. Then, on the **Team** ribbon, choose **Configure**, **List**, and then choose **Refresh work items only**. This will flatten the tree structure and change the query list to an input list.

Remove a tree level

Remove any content entered under the tree-level **Title** *number* column you want to remove—the highest numbered column—and, then **Refresh** your worksheet.

> **TIP**
> Always publish changes that you have 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 will lose any data you have not published.

Convert from an input list to a query

On the **Team** ribbon, choose **Configure**, **List**, and then select the query you want to use. The worksheet will refresh with only those work items returned by the query. Also, if you select a tree query, then the list becomes a tree list.

Add existing work items to a list

If you're working with a non-query input list, you can add work items by choosing Get Work Items from the Team ribbon. If you're working from a query, then you need to modify your query to contain the work items you want. Then refresh your list.

Bulk edit test cases

You can't use Excel to export and import test case steps or other test artifacts. Instead, use the grid view to bulk edit test cases supported via the web portal.

Related articles

While the examples shown here represent connecting to an on-premises TFS, you can connect to Azure Boards and bulk add and modify work items. Once you've connected to the cloud server, you use the same procedures to work in Excel.

- Bulk modify work items (web portal)
- Basic Excel tasks

To bulk edit links, you can use these clients:

- Use Excel to edit parent-child or hierarchical links.
- Use Project to edit parent-child and predecessor-successor link relationships.
- Use the web portal, to map backlog items to portfolio backlog items which creates parent-child links.
- Use either the web portal or Team Explorer, to modify parent-child links by dragging items within a hierarchical backlog page or within a tree query.

To resolve publishing errors, see one of these topics:

- Data conflicts
- Data validation errors
- Invalid links

Delete work items

You can't delete work items from Excel. The only way to delete work items is from the web portal or the **witadmin** command line tool. For details, see Move, change, or delete work items .

Direct link query converts to flat list

When you open a direct links query in Excel, the Team Foundation add-in converts the list to a flat list. While you can modify values for the fields and add work items, you can't view nor modify link relationships.

Multiple worksheets

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 a project collection.

To bulk add or modify work items in a different project, open a new Excel workbook.

Use Excel cut and paste functions

You can use many Excel features, such as cut, paste, automatic fill, format, sort (flat list only), filter, and add formulas. You can cut and paste rows to re-sequence items within a list and change link relationships among work items.

To drag a work item, select the work item or contiguous set of work items that you want to move, open the context menu and choose **Select**, **Table Row**, point to the border of the selection, and—when the pointer becomes a move pointer —drag the row to another location.

> **TIP**
> When you refresh the work item list, not all formats may be retained. For example, date formats are set by the server data store. Any changes you make to a date format field will be overwritten with the date format used by the server.

Web portal and Excel access

To open Excel from a web portal query, install the VSTS Open in Excel marketplace extension.

To use Excel, you must get the Team Foundation add-in, which installs when you install Visual Studio or Team Explorer. If you don't have one of these versions installed, install it now. You can install Visual Studio Community for free.

Once you've installed Visual Studio, open Excel and look for the Team ribbon.

Disable the Team Foundation Add-in

If you want to disable the add-in, see Add or remove an add-in.

Enable the Developer tab

See Show the Developer Tab on the Ribbon.

Excel for Mac

macOS is not supported. You need to use Excel on the same computer where you have installed Visual Studio or Team Explorer in order to get the Team Foundation add-in. These applications require Windows.

Use templates to add and update work items

2/5/2019 • 14 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013 | Visual Studio 2015

With work item templates you can quickly create work items which have pre-populated values for your team's commonly used fields. For example, you can create a task template that will set the area path, iteration path, and discipline or activity whenever you use it to create a task.

> **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.

Prerequisites

- To add, capture, or edit work item templates through the web portal, you must be a member of the team under which you add them.
- To apply a team template, you must be a Contributor of the project.

- To add, capture, or edit 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 or be granted Stakeholder access.

- To add, capture, or edit work item templates through Visual Studio Team Explorer, you must 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.

- Browser
- Visual Studio

1. From the web portal, open a work item which 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 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.

1. From the web portal, open a work item which you'll use as the basis for a template.

Within the web portal, work item templates are associated with a team. Only those templates defined for a team are accessible when working in a team context, either through a work item form or bulk update context menu.

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.

If you connect to an on-premises TFS and primarily create work items working in the web portal, you can create a hyperlink that captures the default values you specify for a template. Choose the hyperlink, and it opens the template in the web portal.

1. From the web portal Queries page, open a new work item.

2. Fill in the default values you want the template to specify. You can leave required fields empty, or place some text in them

   Here, we fill in several values, including tags, but leave the Title blank. When you're done, copy the URL for the template.

3. The URL defines each predefined field, for example (line breaks added for clarity):

```
http://fabrikamprime:8080/tfs/DefaultCollection/Fabrikam%20Fiber/_workItems/create/Task?%5B
System.AssignedTo%5D=Jamal+Hartnett&%5B
System.Description%5D=%3Cp%3EAlways+include+Remaining+Work+and+links+to+any+related+bugs+or+user+storie
s.%3C%2Fp%3E&%5B
System.Tags%5D=Web%3B+Phone%3B+Service&%5B
Microsoft.VSTS.Common.Activity%5D=Development&%5B
System.IterationPath%5D=Fabrikam+Fiber%5CIteration+1
```

> **TIP**
>
> The URL won't contain defaults defined for the work item type. To specify a default field value, see Add or modify a field. Also, there's a 2000 character limit recognized by browsers. There's no work around to that limit.

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.

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. You can then share these links through email, a network share, team portal, or a team dashboard or Welcome page.

Copy the link to a template

- Browser
- Visual Studio

1. Open settings for a team.

New navigation

Choose **Project Settings**.

Expand **Boards** and choose **Team configuration**. Then, choose **Templates**.

Previous navigation

Choose the ⚙ gear icon to open the settings for a team.

Choose the **Work>Templates** tab. Then, choose the ··· actions icon for the template you want to copy and select **Copy link**.

2. Copy the link to a shared network or send to your team via email. Also, consider adding a link the team dashboard.

3. 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 web page as a hyperlink.

- Browser
- Visual Studio

1. Open settings for a team.

   Choose **Project Settings**.

2. Expand **Boards** and choose **Team configuration**.

3. Choose **Templates** tab. Then, choose the ··· actions icon for the template you want to copy and select **Copy link**.

4. Copy the link to a shared network or send to your team via email. Also, consider adding a link the team dashboard.

5. 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 web page as a hyperlink.

See Capture a work item as a template for TFS web portal.

Add a template link to the 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.

> **NOTE**
> Multiple team dashboards and the Markdown widget are available for Azure Boards and TFS 2015.2 and later versions.

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.

Apply a template to existing work item(s)

You can apply a template to a single work item or perform a bulk update of several work items.

- Browser
- Visual Studio

Apply a template within a work item form

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 pre-defined 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 (bulk update)

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.

You can't apply a template to existing work items from the web browser.

Manage work item templates

- Browser
- Visual Studio

You can view the list of templates defined for each work item type, and also add, edit, copy, delete, and copy the link of a template. All templates are defined and managed for a team.

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.

   New navigation

   Choose **Project Settings**.

   Expand **Boards** and choose **Team configuration**. If you need to switch to a different team, use the team selector.

   Previous navigation

   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 click 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.

Define a template from scratch

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, click Copy link to capture the URL for the template that you can use to add work items using the template.

Edit, delete, or copy link to a template

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**.

Copy or duplicate 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.

Managing work item templates from the web portal is only supported from TFS 2017 and later versions. For TFS 2013 and TFS 2015, you can manage work item templates in Visual Studio 2013 or Visual Studio 2015 by downloading TFS Power Tools. With the Work Item Templates power tool add-in, you can:

- Create and edit work item templates
- Create and update work items using a template
- Create a template from an existing work item
- Bulk-edit work items by applying a template
- Use templates created in Visual Studio to create work items through the Visual Studio client.

> **NOTE**
> Visual Studio 2015 is the last version in which you can install TFS power tools. You can use the Work Item Templates tool whenever you connect to TFS 2015 version or an earlier version. Templates you define through the power tool are separate from those defined through the web portal for TFS.

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 even when you apply a work item template to an existing work item.

Define an ad hoc 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.

```
http://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*, and specifies values for the Assigned To, Description, Tags, Activity, and Iteration Path fields.

```
http://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 recognized by browsers.

You can save the URL as a text file or add the URL to a dashboard or web page as a hyperlink.

Q & A

Q: How do I set a default template for a team?

A: The feature to set a default template for a team isn't a supported at this time.

From the Work Item Templates tool, you can set a default for each work item type. However, that default is set only for you and available only when working through Team Explorer. Open the template context menu and select **Set as default**.

Q: How do I delete a work item template?

A: From the work item type page, choose ··· the actions icon for an existing template and select the **Delete** option.

How do I create a template with links to other work items?

A: You can't create templates that contain default links to work items. However, you can use the Marketplace

extension, Work item form one click actions, to create rules that support adding linked work items with a single click.

Or, you can use a Kanban board to add child tasks, backlog items, or features. To learn more, see:

- Add task checklists
- Add, run, and update inline tests
- Add features and epics

Otherwise, you can quickly copy tasks, bugs, and other items using Excel to apply parent-child links in a tree list.

You can't create templates that contain default links to work items. However, you can quickly copy tasks, bugs, and other items using Excel to apply parent-child links in a tree list.

Or, you can use a Kanban board to add child tasks, backlog items, or features. To learn more, see:

- Add task checklists
- Add, run, and update inline tests
- Add features and epics

View and update work items via the mobile browser

1/31/2019 • 3 minutes to read • Edit Online

Azure DevOps Services | 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 Azure DevOps Server 2019 and TFS 2018. 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:

- 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, click 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

Email or print user stories, bugs, and other work items

2/7/2019 • 5 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013 | 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.

Here's a list of the most common ways in which teams share information and plans using work item tracking.

> **NOTE**
> Some features are only available from the web portal or a Team Foundation client such as Visual Studio or the Eclipse plug-in, Team Explorer Everywhere (TEE).

TASK/FEATURE	WEB PORTAL	VISUAL STUDIO	TEE (ECLIPSE PLUG-IN)
Email summary list with links to work item(s)	✓	✓	✓
Print work item(s)		✓	
Email link to a work item query	✓	✓	
Email query results list	✓	✓	
Release summary	✓		

> **NOTE**
> For the email feature to work, your administrator for Azure DevOps Server or Team Foundation Server must configure an SMTP server.

In addition, 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.

Some of the most common ways information is shared within a team or across teams is by emailing lists or links to 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.

- Browser
- Visual Studio 2015
- Team Explorer Everywhere

From the web portal, open the work item, choose the ··· actions icon, and select the **Email work item** option.

From the web portal, open the work item, choose the ··· actions icon, and select the **Email work item** option.

From the web portal, open the work item and choose the ✉ mail icon.

> **NOTE**
> If you connect to an on-premises TFS, your TFS admin 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 do this 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.

- Browser
- Visual Studio 2015
- 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.

If you want to mail a list of all items in the backlog or query, simply choose the ✉ mail icon.

To email items from the web portal for TFS 2015: 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, simply choose the ✉ mail icon.

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 the details of a work item, 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.

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.

- Browser
- Visual Studio 2015
- Team Explorer Everywhere

From the web portal, simply copy the URL from the web browser address or hover over the title and then click the copy-to-clipboard icon.

From the web portal for an on-premises TFS, open the work item and then from the context menu for the browser, choose the copy link option.

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
- Configure an SMTP server

Delete test artifacts

1/25/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure Test Plans | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

While test artifacts such as test plans, test suites, test cases, and so on are 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

- You must be a member of the Project Administrators group or have the **Delete test artifacts** permission set to **Allow**.
- You must also have your access level set to 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.

> **NOTE**
>
> The permanently delete feature of test artifacts is available for TFS 2017.1 and later versions.

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

1. To delete a test case, open it from the web portal and choose the** Permanently delete** option from the actions menu. (Bulk deletion is not 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 test plans and test suites directly from **Test**.

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

Change column options

1/31/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

From each backlog page or query, you can add or remove columns. 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.

> **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.

Start by opening the **Column Options** dialog. Each user can set their own column options which persist for each product or portfolio backlog across the user's sessions.

> **NOTE**
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

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.

You can add most fields listed in the Work item field index. Note that all fields defined within the project collection or organization are available for selection, even those 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. Note that all fields defined within the project collection or organization are available for selection, even those 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. Note that all fields defined within the project collection or organization are available for selection, even those that aren't used for your particular project.

Find the field you want to add from the **Available columns** set and choose **>** (greater-than character) to move it into the **Selected columns** list. You can then change the order of the columns with the ↑/↓ up and down arrows. To remove a field, select it and then choose the **<** (less-than character).

You can add most fields listed in the Work item field index. Note that all fields defined within the project collection or organization are available for selection, even those that aren't used for your particular project.

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, click on the field and drag it to a new location
- To re-size a column, click 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 click 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 additional keyboard shortcuts, enter **Shift-?** to display available commands based on the page you're on.

Related articles

- Work item field index
- Backlogs, boards, and plans
- Create managed queries

Switch project, repository, team

1/31/2019 • 6 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Several features depend on the project, repository, or team that you have selected. For example, dashboards, backlogs, and board views will change depending on the project and team you select.

Also, when you add a work item, the system references the default area and iteration paths defined for the team context. Work items you add from the team dashboard (new work item widget) and queries page are assigned the team default iteration. Work items you add from a team backlog or board, are assigned the team default backlog iteration. To learn more, see About teams and Agile tools.

View and open a project

From the **Projects** page you can quickly navigate to a project that you have permissions to view.

> **NOTE**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface.

- New navigation
- Previous navigation

1. Choose the Azure DevOps logo to open **Projects**.

The projects you most recently viewed are displayed, followed by a list of all projects in alphabetic order.

2. Hover over the dots and you can open the service of interest for that project.

3. You can filter the project and team list using the *Filter projects* search box. Simply type a keyword contained within the name of a project or team. Here we type **Fabrikam** to find all projects or teams with *Fabrikam* in their name.

4. Choose **Create Project** to add a project. You must be an account administrator or a member of the Project Collection Administrators group to add a project.

View and open a repository

- New navigation
- Previous navigation

1. Choose **Repos>Files**.

2. Select the repository of interest from the repository selector.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Switch to a different team

- New navigation
- Previous navigation

From a user page, one under—**Boards**, **Repos**, **Pipelines**, or **Test Plans**—you can't switch to a different team, you can only select team artifacts.

From a **Project Settings>Work>Team configuration** page, you select a team from the team selector breadcrumb.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Related articles

- Work across projects
- Add teams

Set the Work Items experience in Visual Studio 2019

1/31/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | Visual Studio 2019

Visual Studio 2019 supports switching between the default view of the Work Items page and the legacy view. The default view is designed to match the Work Items page available in the web portal. The legacy view supports the Work Items page available with the previous versions of Visual Studio.

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 • Open query in Project • Email query results list using Outlook • Create reports from query in Excel (TFS only)

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).

Related articles

- View and add work items using the Work Items page

Set permissions and access for work tracking

2/5/2019 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You grant or restrict access to various work tracking features by granting users or groups specific permissions for an object, project, or collection. Or, when you assign a user as a team administrator, they have permissions to manage all assets for the specific team. Add users to the 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 About access levels, Stakeholder access.

ROLE OR PERMISSION LEVEL	FUNCTIONAL AREAS SET
Team administrator role	• Manage teams and configure team tools • Define and edit team dashboards • Add and manage team-level work item templates • Add team administrators To add a user to the team administrator role, see Add a team administrator.
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 • Delete and restore work items • Move work items out of a project • Permanently delete work items • Delete test artifacts • Edit shared work item queries • Add teams and team administrators • Create and manage area and iteration paths • Edit project-level permissions • Customize a project (On-premises XML or Hosted process models)
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 project-level.

Edit project-level or collection-level/instance-level information

The **Edit project-level information** and **Edit instance-level information** (also referred to as Edit collection-level information) provide permissions to several work tracking features as summarized below. To add users or set permissions at these levels, see Add administrators, set permissions at the project-level or project collection-level.

EDIT PROJECT-LEVEL INFORMATION	EDIT INSTANCE-LEVEL INFORMATION
• Add and administer teams and all team-related features • Create and modify areas and iterations • Edit shared work item queries • Edit project level permission ACLs • Manage process templates • Customize a project • Create and modify global lists • Edit event subscriptions (email or SOAP) on project level events.	• Add and administer teams and all team-related features • Create and modify areas and iterations • Edit check-in policies • Edit shared work item queries • Edit project level and collection level permission ACLs • Manage process templates • Customize a project or process • Create and modify global lists • Edit event subscriptions (email or SOAP) on project or collection level events.

Create child nodes, modify work items under an area 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**
>
> Your web portal uses either the **New navigation** or **Previous navigation** user interface. Choose the **New navigation** tab if the **New Navigation** feature is enabled. You'll see a vertical sidebar along with other navigational features when **New Navigation** has been enabled for the signed-in user or the organization. Choose **Previous navigation** when you see a top-level, blue-bar—indicating that **New navigation** isn't enabled. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **New navigation** tab for guidance. Azure DevOps Server 2019 supports the **New Navigation** user interface. For more information, see Web portal navigation.

> **NOTE**
>
> Choose the **Previous navigation** tab for guidance. TFS 2018 and earlier versions only support the previous navigation user interface. For more information, see Web portal navigation.

- New navigation
- Previous navigation

You define both areas and iterations for a project from the **Project Settings>Work>Project configuration**.

1. Choose (1) **Project Settings**, expand **Work** if needed, and choose (2) **Project configuration** and then (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. If you don't see the group you want, try adding it first.

   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 About permissions and groups.

Choose the **Previous navigation** tab for guidance. **New navigation** isn't supported for TFS 2018 and earlier versions.

Define and edit 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.

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 Set permissions at the project-level or project collection-level.)

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 artifacts

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 Advanced. 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.

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

> **NOTE**
> You can use one or more of the following options with the On-premises XML process models. To learn more about process models, see Customize work tracking experience.

You can restrict access to work tracking objects in one of two ways:

- By adding WITs to the Hidden Categories group, you can prevent the majority of project contributors from creating them. You can create a hyperlink to a template that opens the work item form and share that link with those team members who you do want to create them.
- Set a condition field rule, a condition-based field rule or a combination of the two that applies to a group. You can restrict changes from being made to a field by specifying a qualifying rule and making it apply for a specific group. Conditional rules can include **CANNOTLOSEVALUE**, **EMPTY**, **FROZEN**, **NOTSAMEAS**, **READONLY**, and **REQUIRED** elements.

For more information about how to customize WITs, see Modify or add a custom work item type (WIT).

Related articles

- Set permissions on queries and query folders
- Permissions and access for work tracking
- Permissions and groups reference

Work item form controls

2/7/2019 • 6 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Different types of work items track different data. Each work item form contains some standard fields—such as title, assigned to, and area and iteration path—as well as fields specific to the type. You can link work items to one another, as well as to changesets and source code files.

As the following image shows, each work item form comes with a number of controls, fields, and tabs.

> **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.

Work item form controls

CONTROL	FUNCTION
	Copy URL of work item to clipboard (appears on hover over work item title)
	Go to Discussions section
···	Open Actions menu for additional work item tasks
	Refresh work item with latest changes
	Revert changes to work item
	Open History tab
	Open Links tab
	Open Attachments tab
/	Enter or exit full display mode of a section within the form
/	Collapse or expand a section within the form
	Add new work item and link to existing work item (May appear under ··· Actions menu)
	Change work item type (Appears under ··· Actions menu)
	Move work item to a different project (Appears under ··· Actions menu)
	Copy work item and optionally change work item type (Appears under ··· Actions menu)
	Email work item (Appears under ··· Actions menu)
	Recycle work item (Appears under ··· Actions menu)
	Storyboard with PowerPoint (Appears under ··· Actions menu)

Update work status

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 Closed or Done. The following image shows the work flow states for the Agile process user story. If you want to discard a work item, change the state to Removed.

Typical workflow progression:

- Create a user story in the default state, New.
- Change the state from New to Active.
- Change the state from Active to Resolved.
- Change the state from Resolved to Closed.

Atypical transitions:

- Change the state from New to Removed.
- Change the state from Removed to New.
- Change the state from Active to Removed.
- Change the state from Resolved to Active.
- Change the state from Closed to Resolved.

Removed work items remain in the data store and can be reactivated by changing the State. If you want to permanently remove a work item, you can delete it.

With each update, the Reason field also updates and 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.

Link items to support traceability

By linking work items using Related or Dependent link types, you can track work that is dependent on other work. Each work item contains one or more tabs with link controls. These controls support linking the work item to one or more objects.

There are three links controls provided on most forms. The Development and Related Work scoped links controls appear on the Details tab. The 🔗 Links tab provides access to all links made to the work item.

Add links

From each links control, you can perform these actions:

- To open an associated item or object, click the linked item
- To delete a link, highlight it and click the ✕ delete icon
- To link to an existing item, or create and link a new work item, select one of the menu options.

To learn more, see Add links to work items.

Development scoped links control

The Development links control displays all of your development links, whether based on a git or Team Foundation version control (TFVC) repository. It displays links in a set order, and provides calls-to-action that support users to drive development from a work item.

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.

Git lets you link work items to commits by using the Commit link type. To learn how, see Manage and commit

your changes.

Related scoped links control

The Related Work links control displays links to other work items in a set order on the front page of the form. It supports these link types: Duplicate/Duplicate of, Parent/Child, Predecessor/Successor, Related, and Tests/Tested by. To learn more about different link types, see Linking, traceability, and managing dependencies.

Links control tab

In addition, the Links control tab provides access to all links made to the work item—both work items and external objects.

Add to the discussion

Add and review comments made about the work being performed by going to the discussion section. This way, you capture all comments within the work item rather than maintaining a long email thread.

Click the discussion icon, which indicates how many comments have been added, to move your focus to the discussion section. Click the full screen icon to expand the display of the discussion section within the form.

Within the discussion section, you can use the **@mention** control to notify another team member about the discussion. Simply type **@** and their name.

> **NOTE**
>
> You can also use group mentions. Simply start type the name of a team or a security group, click the search icon and then select from the options listed.

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.

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

Copy the URL

From the web portal, simply copy the URL from the web browser address or hover over the title and then click the copy-to-clipboard icon. For other copy options, see Copy or clone work items.

Start storyboarding

> **NOTE**
>
> The **Start storyboarding** menu option is only available from the new web form. However, from TFS, you can choose the **Start Storyboarding** link from the **Storyboard** tab from a backlog item, or simply open PowerPoint. See Storyboard your ideas using PowerPoint for requirements and usage.

You can storyboard your ideas using PowerPoint to bring your ideas to life with storyboard shapes, text, animation, and all the other features that PowerPoint Storyboarding provides. From any work item, you can open PowerPoint by choosing the Start storyboarding menu option.

Find or list work items

Type an ID in the Search work items box and then choose the search icon.

Use the ▾ context menu to add a filter based on assignment, status, a keyword, or a work item type.

Or, using the ▾ context menu, add filters to find items based on assignment, status, a keyword, or a work item type. For example, enter **A=@Me T=Task** to list all tasks assigned to you. Use the **=**, **:**, and **-** operators to specify the operations: Equals, Contains, and Not, respectively.

Once you have a results list, you can modify the filter criteria to refine your search results.

Related articles

Keep in mind that the work item tracking experience and forms that appear in Visual Studio won't show several of

the features that the web portal makes available. To add a field or customize a work item type, see Customize your work tracking experience.

See also:

- Keyboard shortcuts for work item forms and the Work Items page

2/7/2019 • 2 minutes to read • Edit Online

Keyboard shortcuts for work item forms and the Work Items page

Keyboard shortcuts for work item forms

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

With keyboard shortcuts, you can quickly navigate within a web page or the web work item form.

Boards>Work Items

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

> **NOTE**
> The following shortcuts are available from the web portal for Azure DevOps Services.

	Work **l** Open backlog **b** Open board **i** Open current iteration **t** Open task board **q** Open queries **z** Toggle full screen **Work items** **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 **Shift+?** from within the form.

> **NOTE**
> **Feature availability:** The following shortcuts are available from Azure DevOps Services.

	Work Item Form **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

::: moniker range="azure-devops"

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

::: moniker-end

::: moniker range=">= tfs-2017 <= azure-devops-2019"

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

::: moniker-end

Related articles

- Keyboard shortcuts for Azure DevOps and Team Explorer

Link type reference

2/7/2019 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Four link types are supported. A specific field maintains a count of links for each of these link types.

- **Work link types** support creating relationships among work items including select test case management work items
- **Remote work link types** support creating relationships among work items that are defined in different organizations
- **Hyperlink** supports linking a work item to any URL or network share
- **External link types** supports connecting a work item to an external object, such as a code object, build, or wiki page.

Three link types are supported. A specific field maintains a count of links for each of these link types.

- **Work link types** support creating relationships among work items including select test case management work items
- **Hyperlink** supports linking a work item to any URL or network share
- **External link types** supports connecting a work item to an external object, such as a code object, build, or storyboard.

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), 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.

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 task board. Typical uses include: • Create a work breakdown structure (WBS). See Schedule tasks and assign resources using Microsoft Project. • Map backlog items to portfolio backlog items. Mapping items automatically creates parent-child links between the items. To learn about mapping, see Organize your backlog. • 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 or Project 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 which are duplicates of one another. Restrictions and recommendations: • A work item can have only one Duplicate. • Only use Duplicate/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	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 Project, linked tasks are represented as predecessor-successor links in Azure Boards. Typically used to track work that must be completed prior to 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 into the Team Foundation client.
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 TFS, 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.

Notes:

1. Available from TFS 2017 and later versions.

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.

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)	Microsoft.VSTS.Common.ProducedFor.Forward Microsoft.VSTS.Common.ConsumesFrom.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.RemoteRelated 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.

Hyperlink type

There is 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. Note that **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. They support linking work items to other objects as shown in the

following image. A work item's *External Link Count* corresponds to the sum of all links defined with a external 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.

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.
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	Build	Build	Used to link a work item to a build.
Integrated in build	Build	Build	Used to link a work item to a build.

Pull Request	Git	PullRequestId	Used to link a work item to a pull request.
Model Link	ArchitectureTools	ModelLink	(Not supported within the web portal) Used to link a work item to a diagram—such as an Activity, Component, Layer, Use Case, or other diagram—stored within the system. You can link diagrams to work items only from the Visual Studio client.
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 on a network. See Storyboard your ideas using PowerPoint for details.
Storyboard	Requirements	Storyboard	Used to link a work item to a file on a network. See Storyboard your ideas using PowerPoint for details.
Tag	Git	Tag	Used to link a work item to a tag that's been defined for a git commit or git repository. 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. Note that **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.

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 topics:

- Link type element reference
- Manage link types (witadmin)

Related articles

- Link work items to track dependencies
- Add link to multiple work items
- Use mapping to link backlog items to features and epics
- Bulk modify links using Excel
- Link type topologies and restrictions
- Artifact Link Types API

Define the URL for a work item

2/7/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can define the URL for a work item using the syntax provided based on the version or platform you work from.

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.
- *TeamProjectName* specifies the project name
- *WorkItemNumber* specifies the ID of the bug, task, or other work item.

```
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
```

http://*ServerName:Port*/**tfs/***CollectionName/TeamProjectName*/**_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/TeamProjectName*/**_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
```

Work item field index

2/7/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 additional tracking needs, you can define your own custom work item fields.

To support additional tracking needs, you can modify or add a custom field.

Alphabetical index

Values in parenthesis indicate the following:

- **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)

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)

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

- 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)
- 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)
- Priority
- Probability (CMMI)
- Proposed Fix (CMMI)
- Purpose (CMMI)
- Query Text (TCM)
- 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)

By using the system fields or other fields you have added to your project collection, you can enable meaningful cross-project reports and queries. In addition, 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 topics

The following articles describe fields that are used in common by several WITs, or those 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
- Create managed queries
- Define a query
- Choose a process
- Reportable fields reference (on-premises TFS only)

Code review and feedback field reference

1/26/2019 • 3 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 TFS application 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's been requested to provide review comments. See Day in the life of a Developer: Suspend work, fix a bug, and conduct a code review.

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
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

FIELD NAME	DESCRIPTION	DATA TYPE
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
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 requested. 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
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 field reference (CMMI)

2/7/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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: <br/- **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.Mitigation Triggers	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 (CMMI)

1/26/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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
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 (CMMI)

2/7/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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 (CMMI)

2/7/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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
Called By	The name of the team member who scheduled the meeting. Reference name=Microsoft.VSTS.CMMI.CalledBy	String

FIELD NAME	DESCRIPTION	DATA TYPE
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

Permissions and access for work tracking

2/5/2019 • 10 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

As a member of an Azure DevOps project, you can use the majority of 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 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 the majority of 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 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 ☐ checkmark 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 Add administrators, set permissions at the project-level or project collection-level.

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.

General 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.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items	☐	☐	☐	☐
Add work items, add tags to work items *(Stakeholders can assign existing tags to work items, but can't add new tags)*	☐		☐	☐
Change work item type	☐		☐	☐
Move work item to another project			☐	☐
Email work items	☐		☐	☐

Apply a work item template				
Delete work items (able to restore from the Recycle bin)				
Permanently delete work items				
Provide feedback (through the Microsoft Feedback client)				
Request feedback				

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	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items				
Add work items, add tags to work items *(Stakeholders can assign existing tags to work items, but can't add new tags)*				
Email work items				
Apply a work item template				
Delete work items (able to restore from the Recycle bin)				
Permanently delete work items				
Provide feedback (through the Microsoft Feedback client)				
Request feedback				

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items				
Add work items, add tags to work items *(Stakeholders can assign existing tags to work items, but can't add new tags)*				
Email work items				

Delete work items (able to restore from the Recycle bin)			✓	✓
Permanently delete work items				✓
Provide feedback (through the Microsoft Feedback client)	✓	✓	✓	✓
Request feedback			✓	✓

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items	✓	✓	✓	✓
Add work items, add tags to work items *(Stakeholders can assign existing tags to work items, but can't add new tags)*	✓		✓	✓
Email work items	✓		✓	✓
Permanently delete work items				✓
Provide feedback (through the Microsoft Feedback client)	✓	✓	✓	✓
Request feedback			✓	✓

Boards feature access

You use **Boards** to implement Kanban methods. Boards present work items as cards and support quick status updates through drag-and-drop.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View boards and open work items	✓	✓	✓	✓
Add work items to a board; update status, reorder, or reparent child tasks through drag-and-drop; update a field on a card			✓	✓
Add child tasks to a checklist	✓		✓	✓
Assign to a sprint (from card menu)	✓		✓	✓
Customize a board, configure team settings *(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)*	✓			✓

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View boards and open work items	✓	✓	✓	✓

Add work items to a board; update status through drag-and-drop			✓	✓
Assign to a sprint	✓		✓	✓
Customize a board, configure team settings *(Stakeholders assigned as a team administrator or Project Administrator can configure team 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	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View backlogs and open work items	✓	✓	✓	✓
Add work items to a backlog *(Stakeholders can only add items to the bottom of the 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			✓	✓
Customize a backlog, configure team settings *(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)*	✓			✓

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View backlogs and open work items	✓	✓	✓	✓
Add work items to a backlog *(Stakeholders can only add items to the bottom of the backlog)*	✓		✓	✓
Use bulk edit features	✓		✓	✓
Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane			✓	✓
Customize a backlog, configure team settings *(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)*	✓			✓

Sprints feature access

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	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View sprint backlogs, taskboards, and open work items	✓	✓	✓	✓
Add work items to a sprint backlog *(Stakeholders can add backlog items to the bottom of a sprint backlog)*	✓		✓	✓
Add work items to a taskboard *(Stakeholders can add backlog items but not tasks)*			✓	✓
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 (work details)	✓	✓	✓	✓
Set team capacity			✓	✓
Use bulk edit features	✓		✓	✓
Define sprints, set sprint dates				✓
Customize a sprint backlog or taskboard, configure team settings *(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)*	✓			✓

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View sprint backlogs, taskboards, and open work items	✓	✓	✓	✓
Add work items to a sprint backlog *(Stakeholders can add backlog items to the bottom of a sprint backlog)*	✓		✓	✓
Add work items to a taskboard *(Stakeholders can add backlog items but not tasks)*			✓	✓
Prioritize/reorder a sprint backlog or taskboard; add child items to a backlog item; reassign items to another using drag-and-drop			✓	✓
View team capacity (work details)	✓	✓	✓	✓

Set team capacity			✓	✓
Use bulk edit features	✓		✓	✓
Define sprints, set sprint dates				✓
Customize a sprint backlog or taskboard, configure team settings *(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)*	✓			✓

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.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	PROJECT ADMINS
View and run managed queries	✓	✓	✓	✓
Create and save managed My queries	✓		✓	✓
Create and save managed Shared queries *(Stakeholders can't save Shared queries even if granted permissions)*				✓
View query charts		✓	✓	✓
Create query charts			✓	✓
Powerful semantic work-tracking search	✓	✓	✓	✓

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View and run managed queries	✓	✓	✓	✓
Create and save managed queries *(Stakeholders can't save shared queries)*	✓		✓	✓
View query charts		✓	✓	✓
Create query charts			✓	✓

Delivery plans feature access

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.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	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 feature access

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 admin context Security page.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	PROJECT ADMINS
Provide feedback using the Test & Feedback extension	✓	✓	✓	✓
Exploratory testing, view test runs		✓	✓	✓
Manage test plans and test suites Manage test configurations and test environments Exploratory testing, create and delete test runs Request feedback using the Test & Feedback extension			✓	✓
Test Manager (purchased separately)			✓	✓

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

Resources defined for the project

You set project-level information permissions from **Project Settings**>**Security**. You set permissions for area and iteration paths under **Project Settings**>**Boards**. These resources are defined for a project which all valid users of the project can view.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ACCOUNT OWNER/ PROJECT ADMINS
View project-level information	✓	✓	✓	✓	✓
Area node: Edit work items under the node			✓	✓	✓
Area nodes and Iteration nodes: Create, delete, edit child nodes					✓
Edit project-level information				✓	

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.

::: moniker range="<= tfs-2017"

If your on-premises TFS deployment includes reporting or SharePoint Products, add users to those resources. See Grant permissions to view or create SQL Server reports in TFS and Set SharePoint site permissions.

Related notes

- Set permissions and access for work tracking
- Get started as a Stakeholder
- Add another team
- Manage teams and configure team tools

Changes made to process templates

1/25/2019 • 9 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

> **NOTE**
>
> this article describes changes made to the core system processes—Agile, Scrum, or CMMI with updates made to the on-premises Team Foundation Server (TFS). These processes are available from both Azure Boards and TFS.
>
> Projects hosted on Azure Boards update automatically with each service upgrade. Whereas, updates to projects defined on an on-premises TFS may require running the Configure Feature Wizard after updating to a later version of TFS.

To support the addition of new features, changes are introduced periodically to the core system processes or process template. 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.

If you've customized your project and haven't upgraded TFS 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 update TFS for a description of each feature added with the updates.

TFS UPDATE	CHANGES INTRODUCED
TFS 2017	Added the **WebLayout** section within the **FORM** section of all work item type (WIT) 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 TFS to TFS 2017, the new web form is automatically available when you add projects to a collection. For existing projects, an admin 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 WIT 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. Time Criticality captures how the business value decreases 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 differentiates items based on work performed 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 necessitated adding fields to the bug WIT definition as well as adding a process

configuration behavior.

The following changes were made to the default process templates:

- WITs added: Epic
- Miscellaneous WIT changes:
  - Feature: Added `Time Criticality`, `Effort`, 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 WIT forms to reflect additions of new fields; added `ID` field to all forms
  - Added WIT `refname` attribute to all WIT 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 WIT 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 WIT 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 WIT definitions:

- Product backlog item: Added `Priority` and `Value Area` fields; added transition from `Committed` to `Removed`
- 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 WIT definitions:

- Scrum - Bug and Product backlog item: Removed the `Backlog Priority` field
- Agile - Bug: Added the `Story Points` field; User Story: Removed the `Stack Rank` field
- CMMI: Added the `Size` field to the Bug definition. Removed the `Stack Rank` field from the Requirement definition.

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:** • WITs 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:** • WITs 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:** • WITs added: Feature • Categories added: Feature Category • Process configuration changes: Replaced AgileConfiguration and CommonConfiguration with a single file, ProcessConfiguration, supports these additional features: portfolio backlogs, color assignment to WITs, 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. • Reporting Services updates: Backlog Overview (Scrum), Stories Overview and Stories Progress (Agile), and Requirements Overview and Requirements Progress (CMMI). These updates reflect changes required with the introduction of the Feature portfolio backlog work item type as described in this blog post: Update your Overview and Progress reports to support the Portfolio backlogs. 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 WIT definitions to support status updates via Kanban and taskboards** Each of the default process templates that TFS provides was updated to support additional 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 Workflow and 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:
	• WITs added: Code Review Request, Code Review Response, Feedback Request, and Feedback Response
	• WIT 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.
	• 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 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 Agile Stories Overview report.
	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. See Update the Workflow for Agile Team Projects.
	• 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 WIT.
	• 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.

Related upgrade notes

To update your existing projects hosted on an on-premises TFS to access new features, run the Configure Features wizard. In the event that 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
- Additional configuration options
- Upload reports to a project

Which tools and clients connect to Azure DevOps Services and Team Foundation Server (TFS)?

2/7/2019 • 8 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Learn about the tools and clients that can connect to Azure DevOps Services and TFS.

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 Services and TFS to support source control, work tracking, build, and test operations.

The available tools that communicate with Azure DevOps Services and TFS are included as follows:

- Desktop client developer tools
- Office integration tools
- Web-based tools
- Command-line tools
- Marketplace extensions
- REST APIs

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 those 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 Team Foundation Server and Visual Studio Team Services.
- **[Eclipse/Team Explorer Everywhere]**(../java/download-eclipse-plug-in.md): Free plug-in to support teams running Eclipse on Linux, macOS, or Windows that connect to Azure DevOps Services or TFS.
- **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 Services or TFS.
- **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 Services or TFS.
- **Visual Studio Code**: Free, open-source code editor with a free extension to support connecting to Git repositories on Azure DevOps Services or TFS.

Team Explorer plug-in

Team Explorer, a plug-in to all Visual Studio versions, connects Visual Studio to projects defined in Azure DevOps

Services or TFS. 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

Office integration tools

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.
- Project Professional: With Project Professional and the Team Foundation Server Extensions for Project Server, you can manage projects that synchronize data that exists in both TFS and Project Server. Project managers and software development teams can use the tools that they prefer, work at the level of precision that supports their needs, and easily share information.
- PowerPoint Storyboarding: Illustrate user stories and requirements by using PowerPoint. The Team Foundation plug-in installs to your existing PowerPoint client.

> **IMPORTANT**
>
> Support for integrating TFS with Project Server is deprecated for TFS 2017. However, synchronization support is provided by a Microsoft partner. See Synchronize TFS with Project Server for details.

Task-specific clients

The following clients support specific tasks, such as managing testing efforts, providing feedback, or modifying work items:

- Microsoft Test Manager: Manage your test efforts, create and run manual tests, and create and track bugs that are found during test efforts. Test Manager is installed with Visual Studio Test Professional and Visual Studio Enterprise.
- 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 Services or TFS. 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.

> **IMPORTANT**
>
> Test Manager is deprecated for TFS 2017.

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 TFS. For release notes, see Azure DevOps Services Features Timeline.

You can use these browsers to access the web portal (Azure DevOps Services and TFS):

VERSION	MICROSOFT EDGE	INTERNET EXPLORER	SAFARI (MAC)	FIREFOX	CHROME
Azure DevOps Services	most recent	11 and later	9.1 and later	most recent	most recent
TFS 15	most recent	11 and later	9.1 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 Services and TFS support the most recent version.

To learn more, see Web portal navigation.

Browser-based extensions

The following extensions are available from the Azure DevOps Marketplace and are built and maintained by the Azure DevOps Services product team:

- Test Manager: Run tests by using your browser with simple pass/fail of steps, add comments/attachments, take screenshots, and file bugs. You can accomplish all of this with automatic end-to-end traceability.
- Azure Artifacts: Build packages of reusable code components and share them. The Azure Artifacts extension enables continuous delivery workflows by supporting multiple packaging protocols such as NuGet and npm. It makes packages available to your team, your builds, and your releases.
- Code search: Increase cross-team collaboration and code sharing by enabling developers to quickly locate

relevant information within the code base of all projects hosted within an organization or collection. You can discover implementation examples, browsing definitions, and error text.

- Work item search: Quickly find relevant work items by searching across all work item fields over all projects in an organization. Perform 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 additional extensions from the Azure DevOps Marketplace.

Application monitoring tools

To monitor your applications, you can use Azure Application Insights for web apps or HockeyApp for mobile apps.

Monitor web applications with Application Insights

Application Insights is an extensible application performance management (APM) service for web developers. Use it to monitor your live web application. Application Insights automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. Application Insights is designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms—including .NET, Node.js, and Java EE—hosted on-premises or in the cloud.

With Application Insights, you can do the following:

- Gain actionable insights through application performance management and instant analytics
- Detect and diagnose exceptions and application performance issues
- Monitor Azure websites, including websites hosted in containers, plus websites on-premises and with other cloud providers
- Seamlessly integrate with your DevOps pipeline by using Azure DevOps Services, GitHub, and our webhooks
- Get started from within Visual Studio, or monitor existing apps without redeploying

To learn more, see Microsoft Azure - Application Insights.

Monitor mobile applications with HockeyApp

With HockeyApp, you can develop, distribute, and beta test your mobile apps. HockeyApp supports the following apps and functionality:

- Android, Cordova, iOS, macOS, Unity, Windows, and Xamarin apps
- Live, reliable crash reports
- Collection of in-app feedback from real users
- Open-source SDKs to let you know what code is running in your apps
- Integration with your existing build system and work item management solution

To learn more, see Microsoft Azure - HockeyApp.

Command-line tools

You can perform many code development and administrative tasks by using command-line tools, such as the following tools:

- Git commands
- TFVC commands
- TFSConfig
- TFSDeleteProject
- TFSSecurity
- TFSServiceControl
- witadmin (work item tracking)

Marketplace extensions

Visual Studio, Azure DevOps Services, and TFS 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, where you can privately maintain or share them with millions of developers who use Visual Studio, Azure DevOps Services, and TFS.

To learn more, visit the Azure DevOps Marketplace and see Overview of extensions.

REST APIs

The Azure DevOps Services and TFS 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 Services and TFS. To learn more, see REST API overview.

Related articles

- Key concepts
- Essential services
- Software development roles
- Pricing
- Azure DevOps data protection overview

Boards (Kanban)

2/13/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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. If you want to add another Kanban board, you do that by adding a team. For details, see About teams and Agile tools.

5-Minute Quickstarts

- Kanban board quickstart

Step-by-Step Tutorials

- Kanban board basics
- Task checklists
- Epics & features
- Add inline tests
- Cumulative flow
- Enable live updates

- Kanban basics
- Task checklists
- Epics & features
- Cumulative flow

- Kanban basics
- Features
- Cumulative flow

Concepts

- Kanban best practices
- Kanban key concepts
- About teams and Agile tools

How-to Guides

- Add columns
- Customize cards
- Set WIP limits
- Split columns
- Expedite work (swimlanes)
- Definition of done

- Add columns

- Customize cards
- Set WIP limits

Reference

- Kanban board controls
- Kanban board keyboard shortcuts
- Work item field index
- Permissions for work tracking

- Work item field index
- Permissions for work tracking

Resources

- Backlogs
- Sprints (Scrum)
- Queries
- Work item customization
- What is Agile?
- What is Agile development?

Backlogs

1/25/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Start planning your project by quickly adding user stories or requirements to your product backlog. Once you have your plan in place, you can start driving your code development efforts.

Start planning your project by quickly adding user stories or requirements to your product backlog.

If your 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. If you want to add another product backlog, you do that by adding a team. For details, see About teams and Agile tools.

5-Minute Quickstarts

- Create your backlog
- Drive Git development
- Define area paths
- Define iteration paths

5-Minute Quickstarts

- Create your backlog
- Define area paths
- Define iteration paths

Step-by-Step Tutorials

- Define features & epics
- Organize backlogs
- Bulk modify work items

Concepts

- About teams and Agile tools
- Refine your backlog
- About permissions and access
- Share information in work items and social tools

How-to Guides

- Filter backlogs & queries
- Change column options
- Email/print work items

Reference

- Permissions and access for work tracking

- Work item field index
- Backlog keyboard shortcuts

Resources

- Work Items
- Boards (Kanban)
- Sprints (Scrum)
- Queries
- Work item customization
- What is Agile?
- What is Agile development?
- Agile culture

Sprints (Scrum)

1/25/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

With Scrum, you can schedule and plan sprints, update your taskboard, and monitor your sprint burndown. For an overview of the Scrum lifecycle, see What is Scrum?.

5-Minute Quickstarts

Start tracking work by adding work items, creating your backlog, or scheduling sprints.

- Schedule sprints
- Define iteration paths

Step-by-Step Tutorials

Whether you use scrum, Kanban, or a combination of Agile methods, you can get started tracking bugs, driving your Git development, and more using the Agile tools available to you.

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. Share your sprint plan
6. Update the Taskboard

Concepts

- Scrum best practices
- Sprints and Scrum key concepts
- Scrum and sprint planning tools
- About areas and iterations
- About permissions and access
- Share information in work items and social tools

How-to Guides

- Filter backlogs & queries
- Copy list of work items
- Bulk add or modify (Web)

Reference

- Permissions and access for work tracking
- Work item field index
- Backlog keyboard shortcuts

Resources

- Backlogs
- Kanban
- Work item queries
- Work item customization
- What is Scrum?
- What is Agile development?

Queries

2/7/2019 • 2 minutes to read • Edit Online

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

List bugs, user stories, or other work items based on field criteria you specify. Then, you can review with your team, triage, or perform bulk updates.

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.

5-Minute Quickstarts

- View, run, or email a query
- Perform an ad hoc search (search box query)

Step-by-Step Tutorials

- Create or edit a managed query
- Triage work items
- Bulk modify work items

Samples

- Query by title, ID, or description
- Query by assignment or workflow changes
- Query by area or iteration path
- Query by date or current iteration
- Query history
- Query a numeric field
- Query by picklist value
- Query by build & test integration fields
- Query by links or attachments

How-to Guides

- Configure work item query-based charts
- Organize queries, add a query folder
- Change column options
- Email/print work items

Reference

- Query quick reference
- Query fields, operators & macros
- Query keyboard shortcuts
- Work item field index
- WIQL syntax

- Permissions and access for work tracking

Resources

- Backlogs
- Kanban
- Scrum
- Work item customization
- Wiql Editor (Marketplace extension)
- Enhanced Export (Marketplace extension)

Customization

1/31/2019 • 2 minutes to read • Edit Online

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You customize your work tracking experience to support your business and reporting needs. The most common customizations include adding a custom field, modifying a work item form, or adding a custom work item type.

Most customers use the Inheritance process model, which provides a convenient user interface to support customization of the work tracking experience.

A select few customers use the Hosted XML process model, which requires that they have opted into this method. This model relies on updating XML files and then importing the process template of these files. To learn more, see Customize your work tracking experience.

With Azure DevOps Server 2019, you have a choice of using the Inheritance process model or the On-premises XML process model to support customizations. The choice is made when you create a project collection and choose the process model for the projects that you'll create in the collection. For details, see On-premises XML process model.

Team Foundation Server uses the On-premises XML process model to support customizations. This model relies on updating and importing XML files using the **witadmin** command line tool. For details, see On-premises XML process model.

5-Minute Quickstarts

- Define area paths
- Define iteration paths or sprints
- Add a custom field
- Add a custom work item type

5-Minute Quickstarts

- Define area paths
- Define iteration paths or sprints

Tutorials

- Customize a project
- Create and manage a process

Concepts

- About areas and iterations
- Differences between process models
- Workflow states & state categories
- Inheritance process model
- Hosted XML process model
- On-premises XML process model

How-to Guides

Use the guidance provided in the following topics based on the process model or process template that you want to customize.

- **Inheritance process model**
  - Customize a project
  - Create and manage a process
- **Hosted XML process model**
  - Supported upgrade operations
  - Clone a Hosted XML process to Inheritance
  - Change a project from Hosted XML to Inheritance
  - Customize a Hosted XML process
  - Import a process
- **On-premises XML process model**
  - Add or modify a field
  - Add or modify a work item type
- **Process templates**
  - Upload or download a process template
  - Customize a process template

Reference

- Index to XML element reference
- All WITD XML elements reference
- All FIELD XML elements reference
- All WORKFLOW XML elements reference
- WebLayout and Control elements
- Process configuration XML element reference

Resources

- Azure Boards
- Scale & configure teams
- Marketplace extensions
- Extensibility and REST APIs