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

18/11/2018 • 5 minutos para ler • Colaboradores

Neste artigo

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

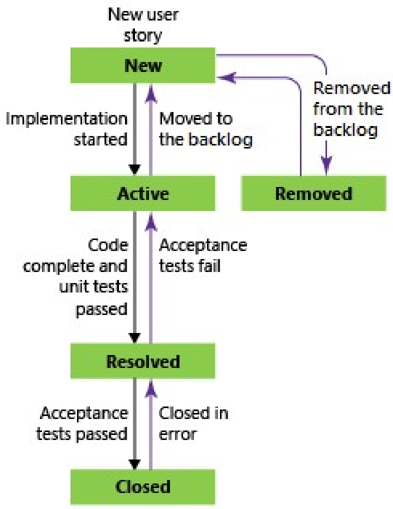
Workflow states define how a work item progresses upon its creation to closure. For example, the four main states defined for the User Story (Agile process) define a progression of four states, from New, Active, Resolved, to Closed. (The Removed state supports removing a work item from appearing on the 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.

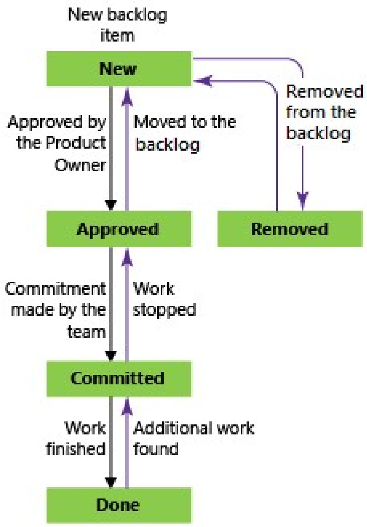
Workflow states

User Story, Agile process	Product backlog item, Scrum process	Requirement, CMMI process

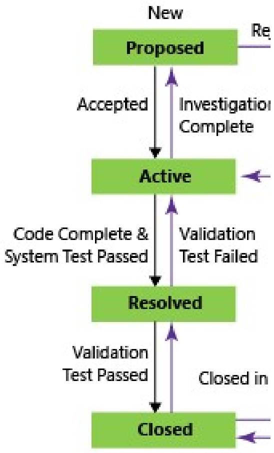
User Story, Agile process



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.	Removed	Removed	n/a	n/a

When to add a State versus a Kanban column

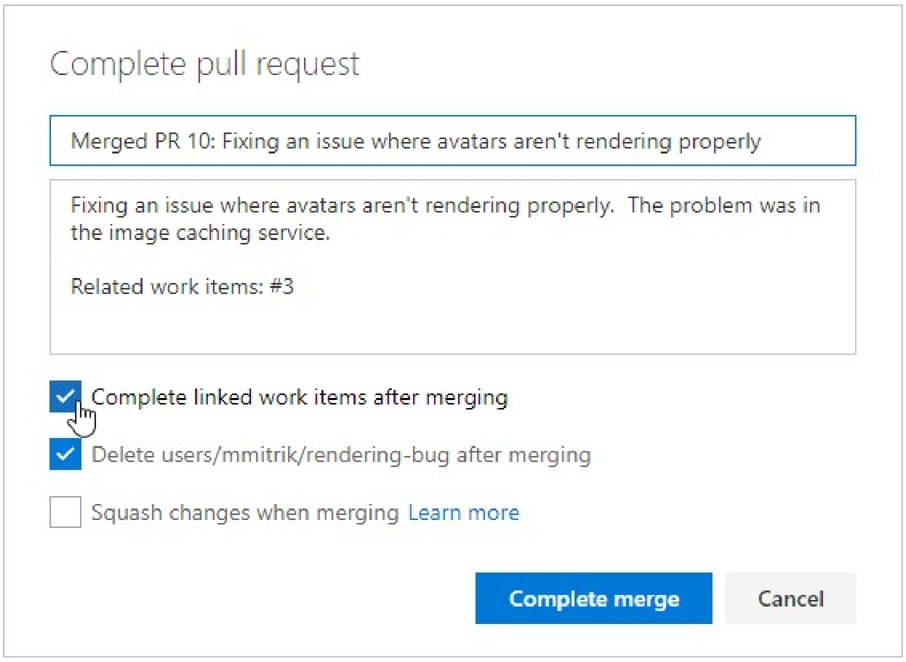
Both States and Kanban columns are used to track the status of work. Workflow states are shared across a project while Kanban columns are shared within a team. Only project collection admins can add custom states, while team admins can add Kanban columns.

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.



Complete pull request

Merged PR 10: Fixing an issue where avatars aren't rendering properly

Fixing an issue where avatars aren't rendering properly. The problem was in the image caching service.

Related work items: #3

☑ Complete linked work items after merging

☑ Delete users/mmitrik/rendering-bug after merging

□ Squash changes when merging Learn more

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