

# Best practices for Agile project management

Article 03/23/2023 11 contributors

[Feedback](#)

## In this article

[Configure your teams](#)[Configure your sprints](#)[Choose your work item types](#)[Create your product plan](#)[Use tags to support queries and filtering](#)[Forecast and milestone planning](#)[Manage dependencies](#)[Work in sprints](#)[Review progress and feature deliverables](#)[Process improvement](#)[Next steps](#)[Related articles](#)[Show less](#)

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

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

### Note

If your team is committed to practicing Kanban or Scrum methods, see instead [About Boards and Kanban](#) or the [tutorials for implementing Scrum](#).

Most of the guidance in this article is valid for both the cloud and on-premises versions. However, some of the features included in this article, such as Rollup, Analytics, and some portfolio planning tools, are only available for the cloud at this time.

## Configure your teams

Azure Boards provides each team a set of Agile tools to plan and track work. Each project defines a default team, which you can start using immediately. However, if you have several development or

feature teams, we recommend that you define a team in Azure DevOps for each feature team. This way, each team can work autonomously while collaborating with each other.

## Best practice tips:

- Configure teams along the value streams your organization wants to deliver
- Define a team for each development group of six to 12 developers
- Configure development teams to support rollup to project management feature teams

For more information about configuring teams, see the following articles:

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

## Configure your sprints

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

## Best practice tips:

- Define a sprint cadence for use by all teams within your product group
- Define at least six or more iterations that support planning for the next six to 12 months
- Determine how teams use iterations to manage backlog items
  - Unassigned sprint work is assigned to the default backlog, or
  - Unassigned sprint work is assigned to a designated future backlog sprint

For more information about configuring sprints, see the following articles:

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

## Choose your work item types

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

If your project is based on another process, such as Basic, Scrum, or CMMI, you have the following choices. Each team determines how they want to track bugs.

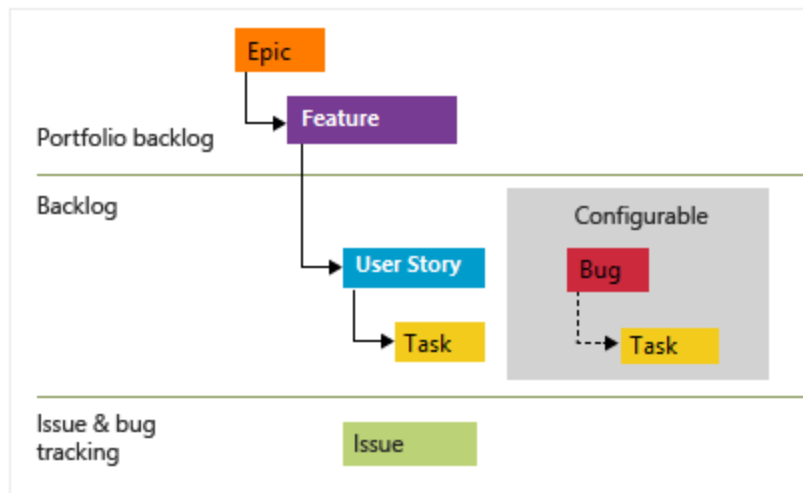
Agile process

Basic process

Scrum process

CMMI process

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



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

### Note

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

## Best practice tips:

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

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

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

For more information about work item types, see the following articles:

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

## Create your product plan

Create your product plan using the Features Backlog. The development team then creates their product plan using the Product Backlog. Periodically you should review and refine your product plans.

### Features backlog

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

Backlog

Analytics

Feature Timeline

Epic Roadmap

...

Features

Order

Work Item Type

Title

1

Feature

Customer Web - Phase 1

2

Feature

Customer Phone - Phase 1

3

Feature

Mobile shopping cart

4

Feature

Customer Service - Web

5

Feature

Mobile feedback

6

Feature

Customer Service - Phone

7

Feature

Text alerts

8

Feature

Refresh web look, feel, performance factors

### Product backlog

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

Backlog Analytics Feature Timeline Epic Roadmap

+ New Work Item

→ View as Board

+ -

Order

Work Item Type

Title

1

Feature

▼ 🏆 Customer Web - Phase 1

User Story

📄 About screen

User Story

> 📄 Hello World Web Site

User Story

> 📄 Change background color

User Story

📄 Slow response on information form

User Story

> 📄 Change initial view

+ 2

Feature

▼ 🏆 Customer Phone - Phase 1

User Story

> 📄 Phone sign in

User Story

> 📄 Request support

3

Feature

> 🏆 Mobile shopping cart

4

Feature

> 🏆 Customer Service - Web

5

Feature

> 🏆 Mobile feedback

6

Feature

> 🏆 Customer Service - Phone

7

Feature

🏆 Text alerts

8

Feature

> 🏆 Refresh web look, feel, performance factors

## Refine each backlog

Periodically review each backlog by doing the following tasks:

- Define work to be performed
- Reorder work items using drag-and-drop so that they appear in priority order
- Open work items and add details
- Assign work to team members or to sprints
- Capture technical debt and nonfeature work required to support a healthy ecosystem of delivery
- Map unparented work to the Feature they belong to
- Estimate size of requirements to help determine team velocity and support forecasting (optional)

### Tip

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

## Best practice tips:

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

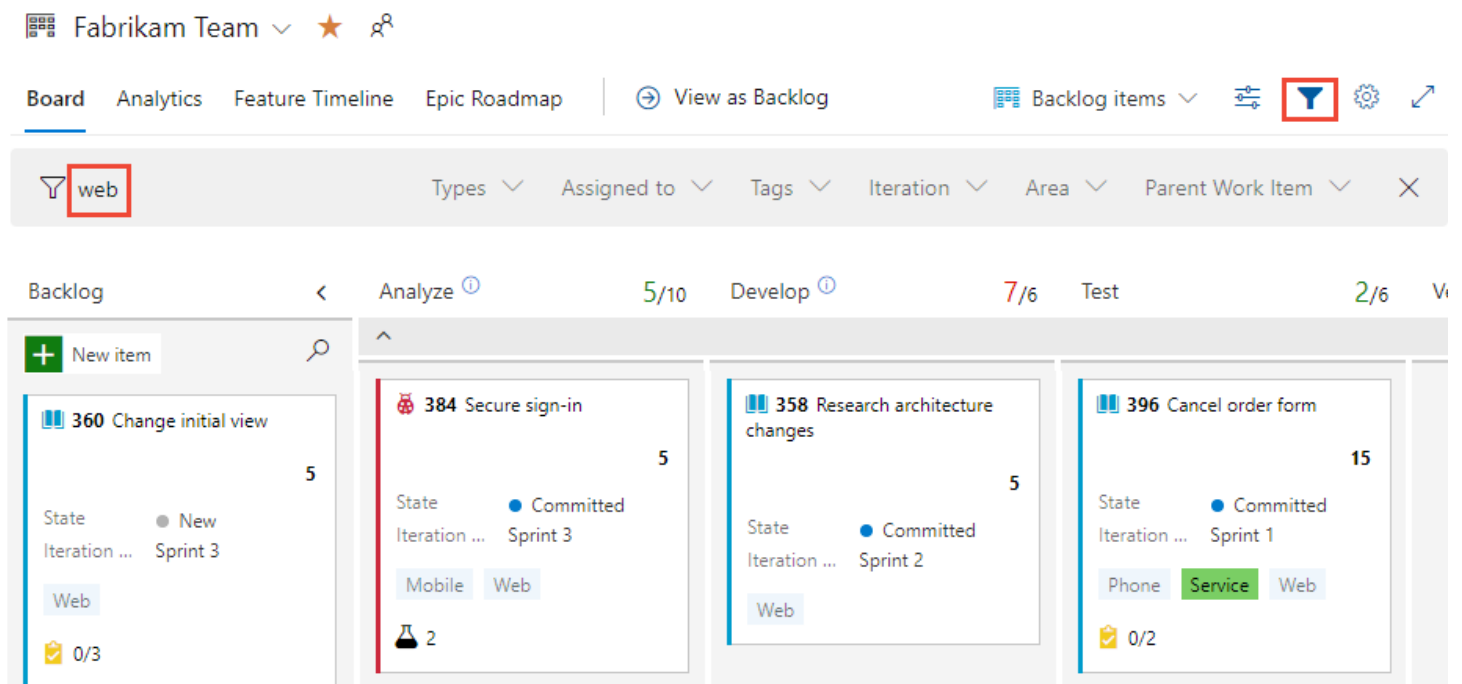
For more information, see the following articles:

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

## Use tags to support queries and filtering

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

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



## Best practice tips:

- Have a policy in place about how your teams use tags
- Indicate how you use tags to support queries, filtering, reporting

- Consider using tags to identify cross-team or cross-project dependencies

For more information, see the following articles:

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

## Forecast and milestone planning

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

## Order the features backlog in priority order

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

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

Backlog Analytics Feature Timeline Epic Roadmap					
+ New Work Item View as Board					
+ -	Order	Work Item Type	State	Title	
	1	Feature	New	> Customer Web - Phase 1	
	2	Feature	Active	> Customer Phone - Phase 1	
	3	Feature	Active	> Mobile feedback	
	4	Feature	Active	New home page design	
	5	Feature	New	> Refresh web look, feel, performance factors	
	6	Feature	Active	> Customer Service - Web	
	7	Feature	Active	> Customer Service - Phone	
	8	Feature	New	> Mobile shopping cart	
	9	Feature	Active	> Convert legacy Odata code	

## Order the requirements backlog based on parent features

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

Backlog

Analytics

Feature Timeline

Epic Roadmap

+ New Work Item

View as Board

Column Options

Order

State

Parent

Story Points

Title

1

New

Customer Web - Phase 1

3

About screen

2

New

Customer Web - Phase 1

3

> Change initial view

3

New

Customer Web - Phase 1

5

> Hello World Web Site

4

New

Customer Web - Phase 1

3

Slow response on information form

5

New

Customer Web - Phase 1

5

> Change background color

6

New

Customer Phone - Phase 1

2

> Phone sign in

7

New

Customer Phone - Phase 1

3

> Request support

8

New

Mobile feedback

8

Design feedback interface

9

New

Mobile feedback

8

Develop mobile interface

10

New

Refresh web look, feel, performance factors

3

> Add an information form

11

New

Refresh web look, feel, performance factors

5

Check performance

12

New

Refresh web look, feel, performance factors

3

Interim save on long form

13

New

Customer Service - Web

5

Technician dashboard improvements

14

New

Customer Service - Web

8

Scheduler

15

New

Customer Service - Web

3

Check issues with permissions

16

New

Customer Service - Phone

5

Switch context issues

17

New

Customer Service - Phone

5

> GSP locator interface

18

New

Customer Service - Phone

2

Check service status

## Forecast the requirements backlog

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

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



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

#### Forecasting based on velocity of 12

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

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

## Update your Features board

With a forecast of when a feature ships, you can update each feature's iteration path. Quickly assign values to a feature by adding those fields to the card on the Kanban board as shown in the following image.

Board Analytics Feature Timeline Epic Roadmap | [View as Backlog](#)

---

New < Active 0/5 Resolved

New item

**919** Customer Web - Phase 1  
 Iteration ... Sprint 3  
 0/5

**915** Customer Phone - Phase 1  
 Iteration ... Sprint 3  
 0/2

**932** Mobile feedback  
 Iteration ... Sprint 5  
 0/2

**935** Refresh web look, feel, performance factors  
 Iteration ... Sprint 6  
 0/3

## Milestone planning

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

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

## Manage dependencies

In Microsoft Project, you manage tasks that depend on the completion of other tasks by linking them. To manage dependencies in Azure Boards, you can add similar linking by adding

Predecessor/Successor link types to work items. Add these links from the **Add link** dialog for a work item.

## Add link dialog

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

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



### Add link

You are adding a link from:

  949 About screen  
Updated 8/12/2020, ● New

Link type

Successor

☒ Work items to link

Enter ID or search for work items



Comment

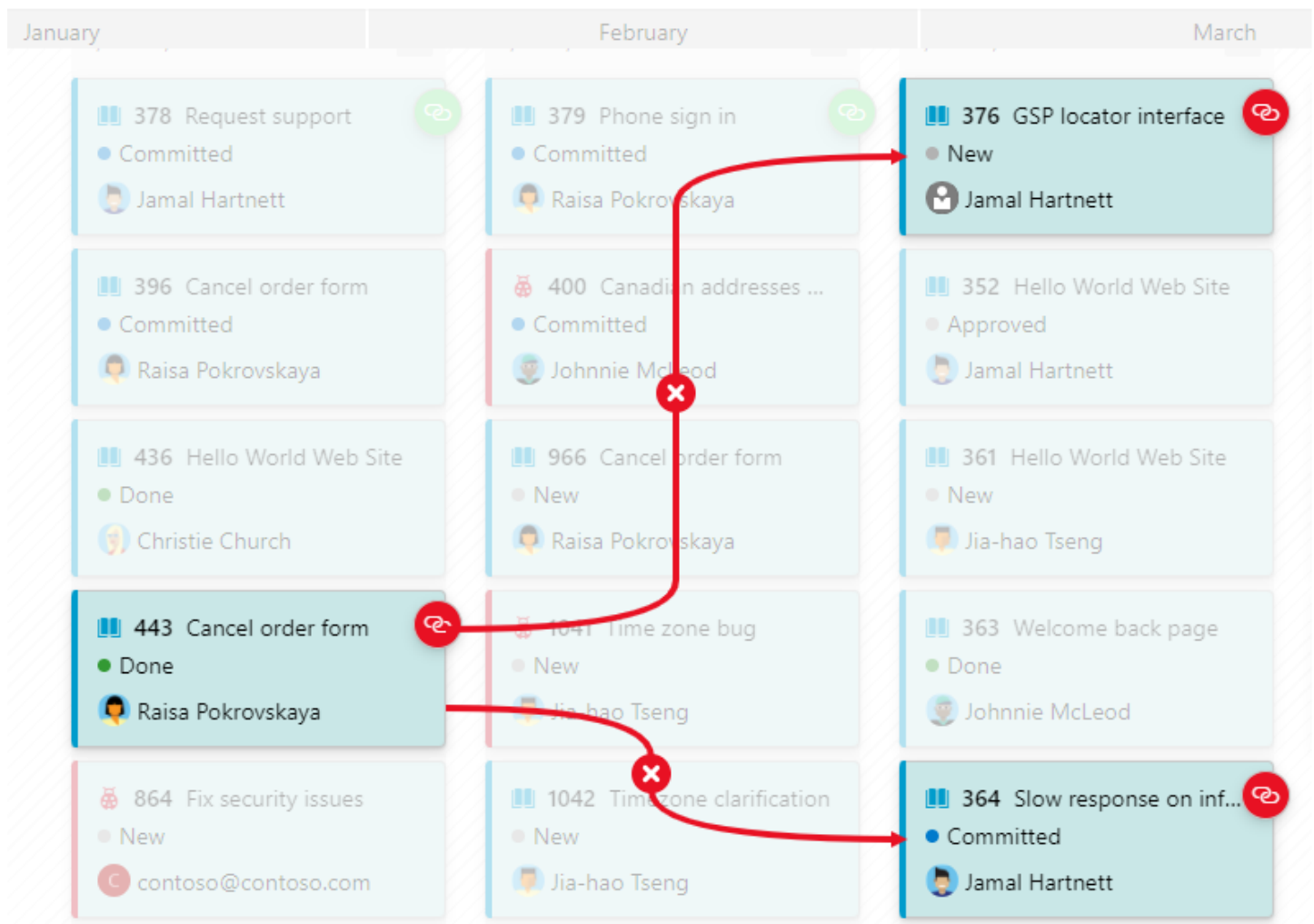
OK

Cancel



## Visualize work item relationships

You can view dependencies and identify dependencies that have issues with Delivery Plans. As shown in the following image, you can toggle the display of dependency lines between linked work items. For more information, see [Track dependencies using Delivery Plans](#).



## Minimum Viable Product versus Critical Path Management

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

### Best practice tips:

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

### Note

Marketplace extensions are not supported features of Azure Boards and therefore not supported by the product team. For questions, suggestions, or issues you have when using these

extensions, visit their corresponding extension page.

For more information, see the following articles:

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

## Work in sprints

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

### Sprint burndown chart

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

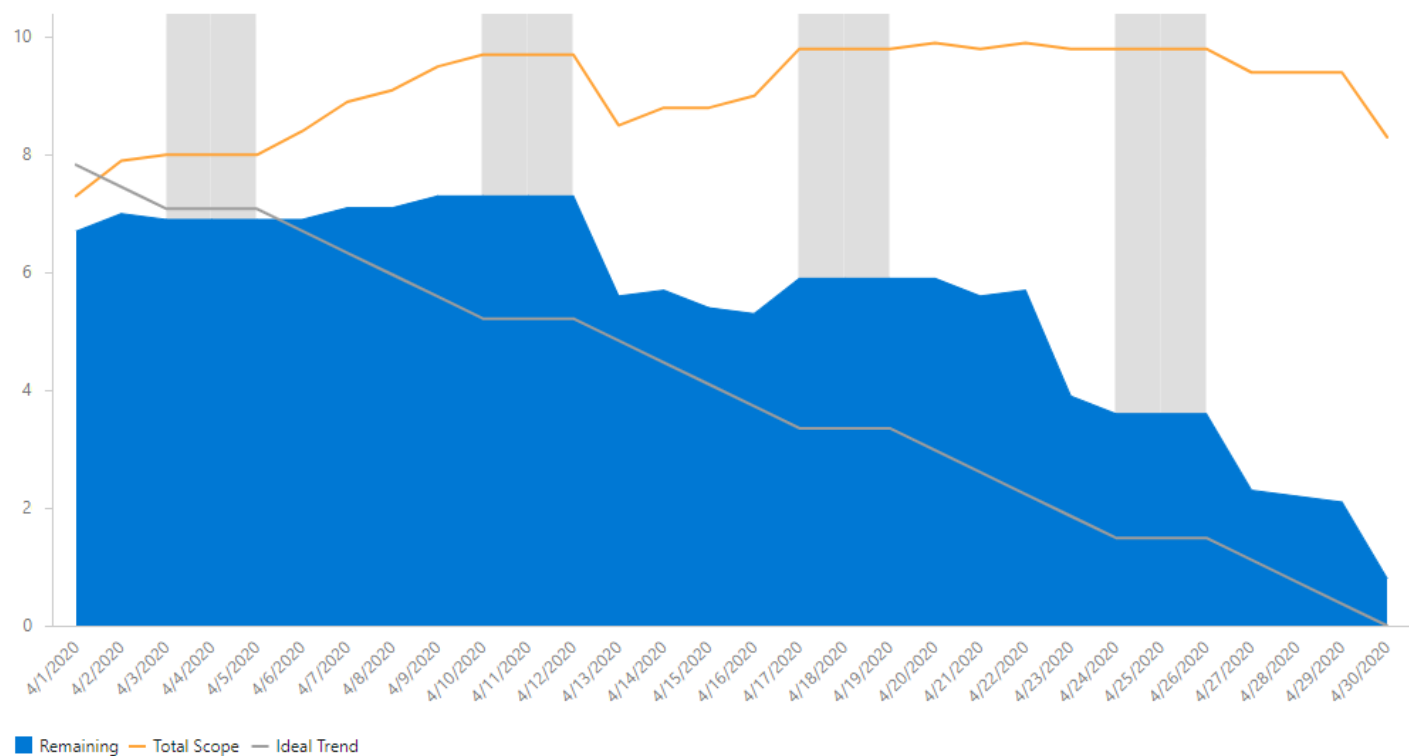
4/1/2020 - 4/30/2020

Completed 100%

Average burndown 2

Stories Remaining 0

Total Scope Increase 1



### Best practice tips

Each sprint, perform the following tasks:

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

For more information, see the following articles:

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

## Review progress and feature deliverables

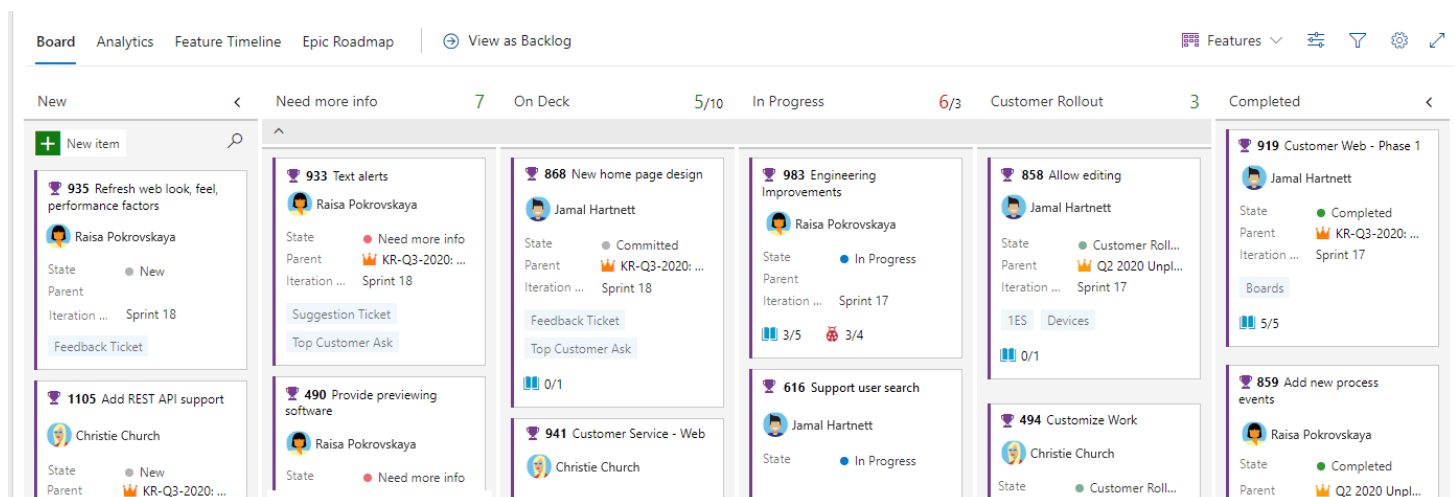
The three main tools you should use to review progress and deliverables are:

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

### Features Kanban board

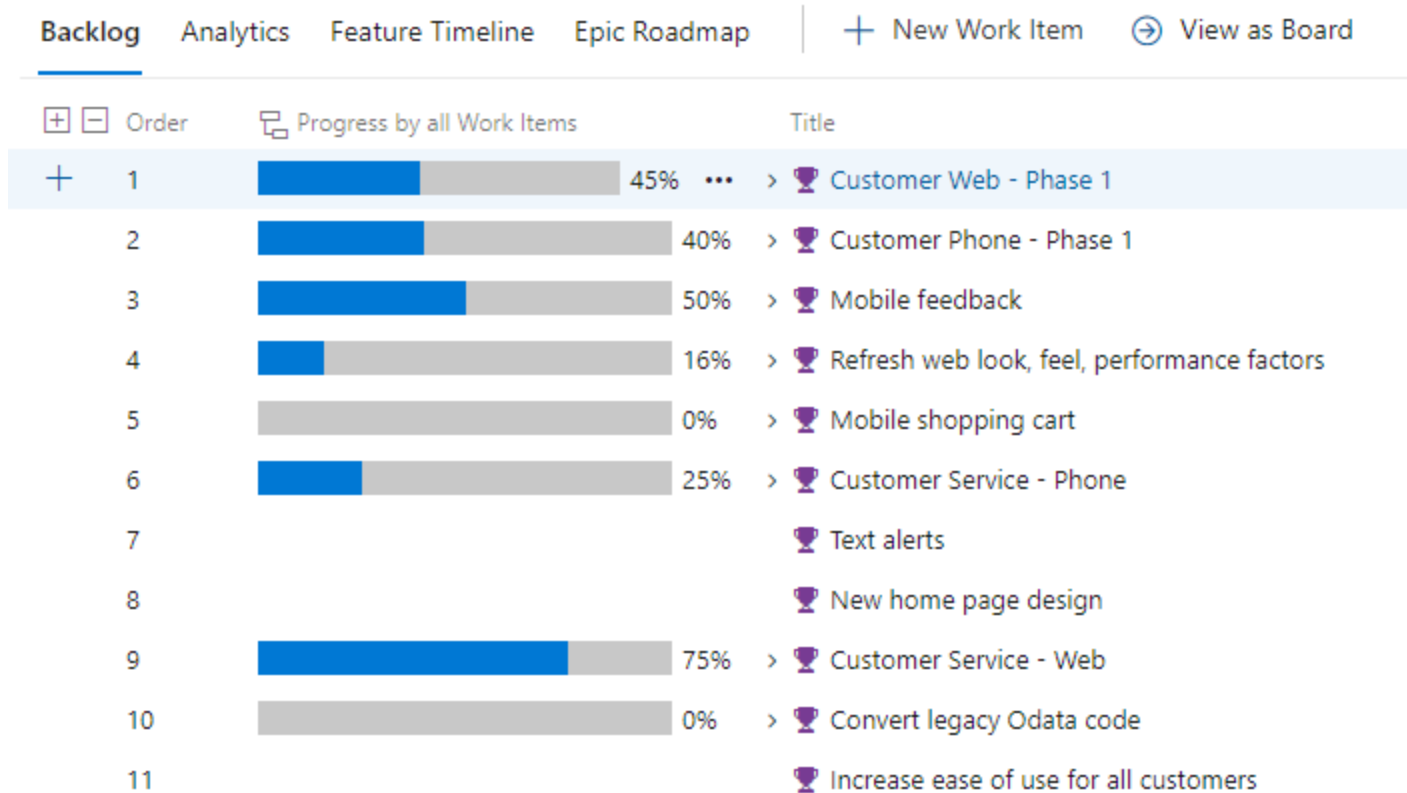
Your Features board is another place to review progress and ensure the continuous flow of deliverables. The following image illustrates a customized Features board, including in progress columns like *Need more info*, *Spec Complete*, *In Progress*, and *Customer Rollout*. These columns provide a more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

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



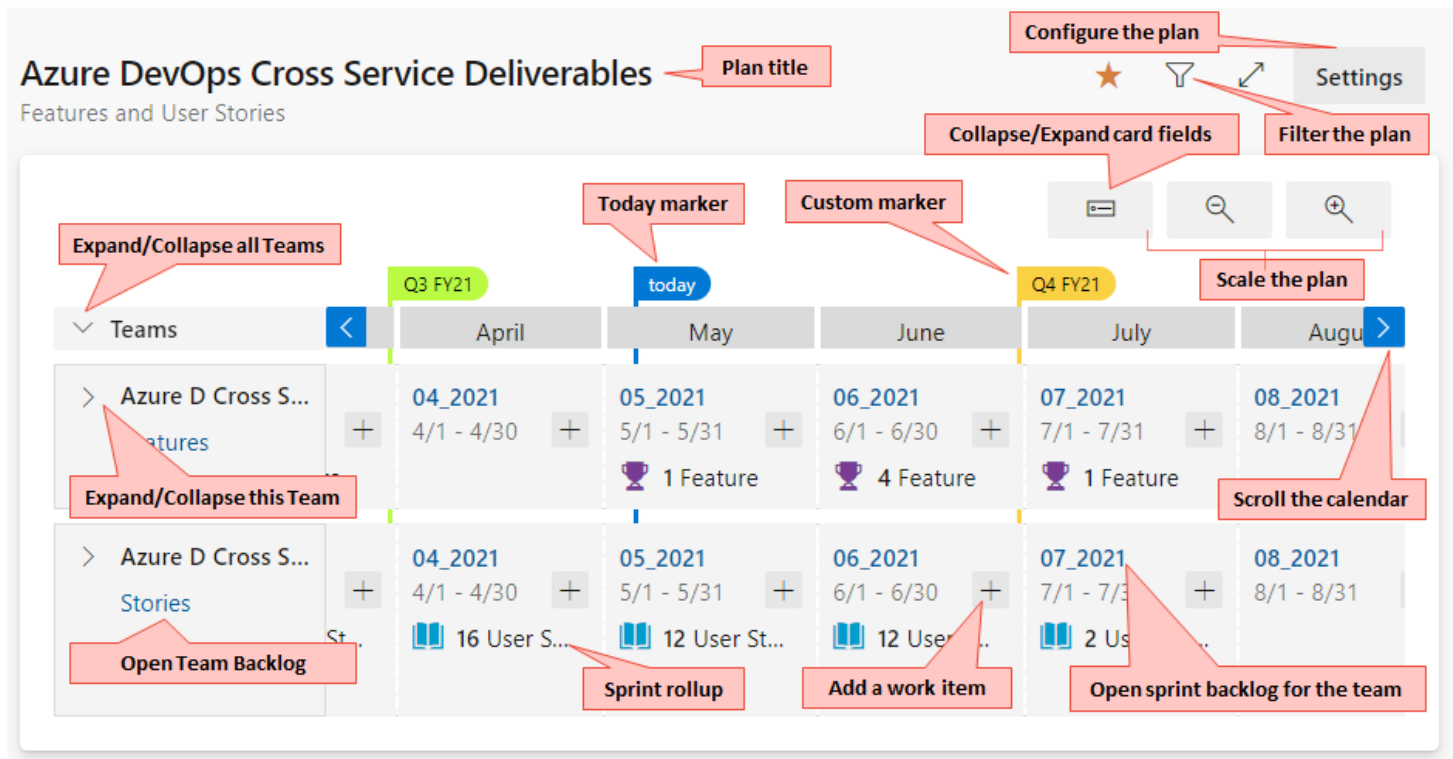
### Rollup

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



## Delivery plans and multiple team deliverables

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



## Best practice tips

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

For more information, see the following articles:

- Add columns to your Kanban board
- Customize cards
- Filter your Kanban board
- Display rollup progress or totals
- Review team Delivery Plans

## Process improvement

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

- Planning sprints
- Setting sprint goals



- Conducting regular retrospectives

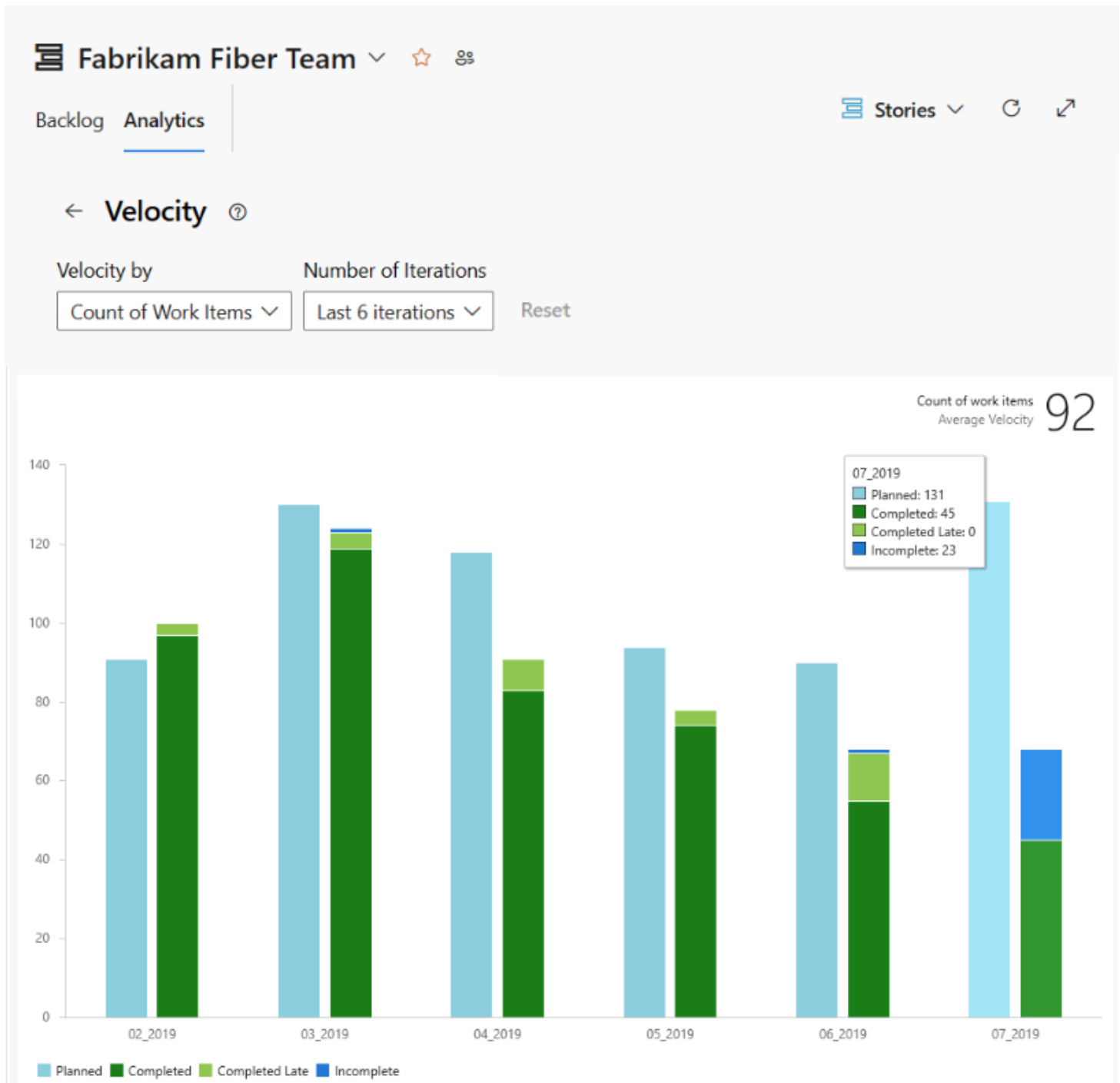
Consider the following questions when setting goals:

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

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

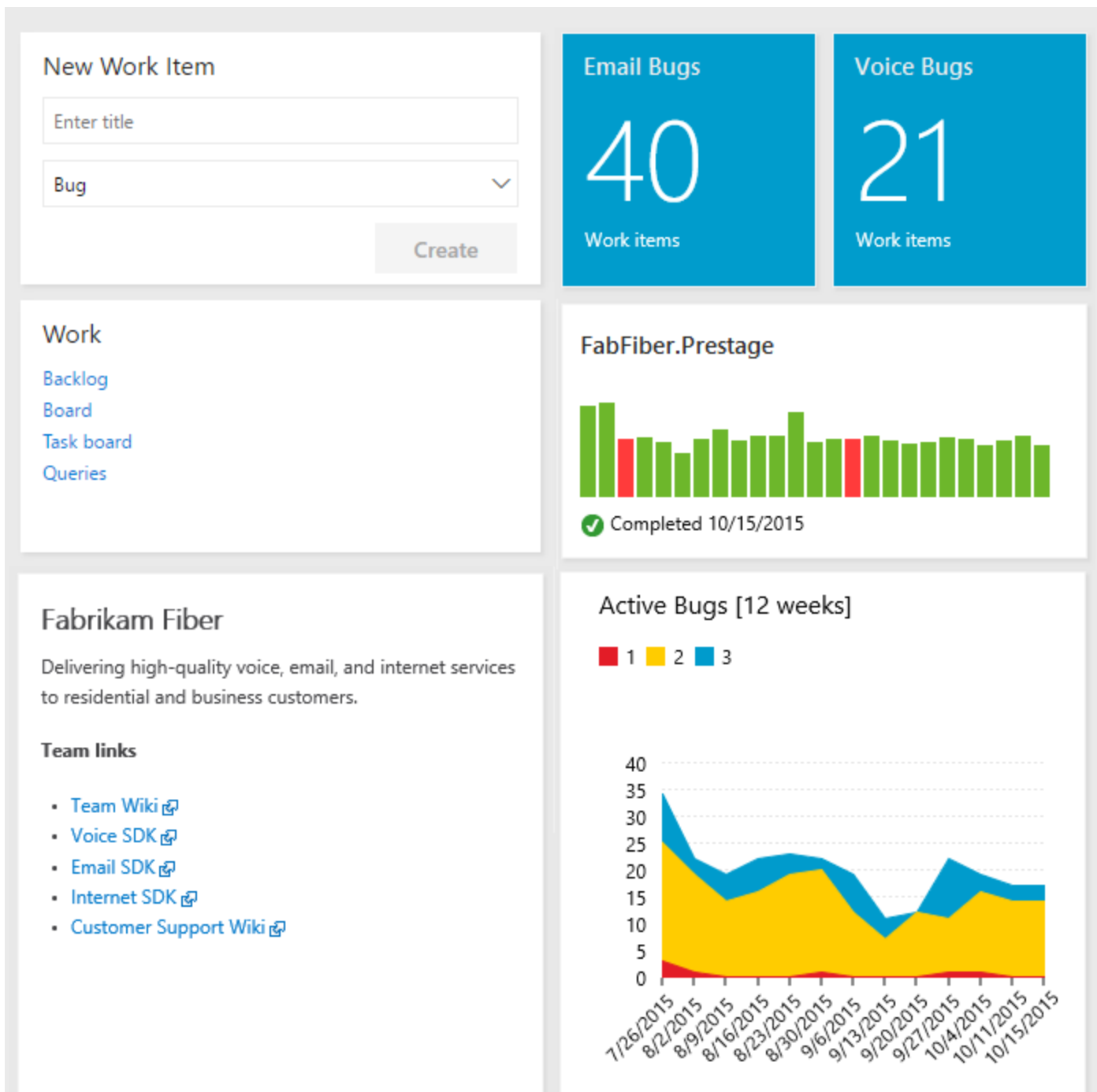
## Team velocity

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



## Team dashboards

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



## Best practice tips

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