

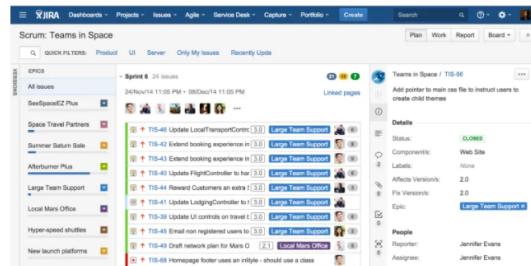
4 best practices for sprint planning meetings

This article is the second in a series on agile ceremonies. Learn how we do sprint retrospectives, too. So let's walk through four tenets of sprint planning we find most helpful.

Every new year brings a renewed sense of purpose and in the software world, promises to ship better software. At Atlassian, we rely heavily on the [agile ceremony](#) of sprint planning to refocus execution, minimize surprises, and guarantee overall higher quality code during the new year.

1. Check your roadmaps before you meet

Time passes faster than we think. So it's a good idea to review your project's roadmap during the first two weeks of the new year. The roadmap sets the context for two important agile concepts: **epics** and **versions**, which provide the backbone for agile program planning and help track delivery of longer-term work. Make sure the roadmap is current, visible to the whole team, and that epics and versions are correctly listed inside of Jira before your sprint planning meeting.



2. Hold a meeting before the meeting

Sprint planning involves two key tasks: grooming the [backlog](#) and deciding which work to complete in the upcoming sprint. At Atlassian, we've found that backlog grooming is best done in a separate meeting with the product owner and [scrum master](#) before the actual sprint planning meeting. We make this pre-meeting optional for the full development team.

What is backlog grooming?

[Backlog grooming](#) ensures that the backlog is healthy. What does that look like? Glad you asked. A healthy backlog:

- Prioritizes each work item, with the most important work listed at the top
- Includes fully-formed user stories the development team can begin to execute on
- Contains an up-to-date estimate for each work item

Teams at Atlassian hold short backlog grooming meetings a couple days before sprint planning. Plan on spending 30 minutes in this meeting to triage the issues you're most likely to take on in the next two sprints. Sometimes you'll find items that don't have enough detail to be executed on or need more contextual information from the product owner. That's the beauty of grooming the backlog in advance. Those gaps can be filled in between meetings so they aren't blockers (or time-wasters) during the actual sprint planning. Thus, grooming the backlog before hand gives the team more time during sprint planning to consider their options for the sprint and ear-mark items for the next sprint, as needed.

Team Activity: Some teams struggle with estimation. Story points provide a sound framework for estimating work. Engage the team in an activity called [silent estimation](#). To start the exercise, place the story point values (0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100) in columns on a white board. Then ask the team to place the user stories in the column they think is most accurate. Most of the stories will gravitate to one number, and if there is disagreement on the story point value it's time to open a discussion as to why.

PUBLISHED APRIL 7, 2019 IN
AGILE

 DAN RADIGAN

SUBSCRIBE TO WORK LIFE

Get stories about tech and teams in your inbox

[Subscribe](#)

Tip: Check out our [guide to project management](#), perfect for agile teams. Even non-project managers can deliver the goods without losing their minds.

3. Inform sprint planning with data

The focus of the sprint planning meeting is to set and agree upon the sprint goal—the amount of work the team believes it can complete during the sprint. The product owner, scrum master, and the full development team all need to be in attendance. At Atlassian, we recommend a minimum of one hour for each week of the sprint that you plan to cover in the meeting. For example, start with a two-hour sprint planning meeting to cover a two-week sprint. Ideally, schedule sprint planning early in the week. Then the team's context and flow is disrupted less by the weekend.



TIP

Learned through experience: Don't schedule the sprint retrospective and the sprint planning meeting together. Give the team enough space to digest the retrospective and effectively contribute to sprint planning—perhaps by inserting a team lunch in between them.

RELATED ARTICLE

[What's your virtual meeting IQ?](#)

BY SARAH GOFF-DUPONT IN QUIZZES

At the start of the meeting, the scrum master presents any relevant action items from the team's retrospective. Next, the product owner give product or market updates so everyone is on the same page and has the broader context fresh in their minds.

After the debriefs, it's the product owner's responsibility to start the actual planning conversation. To get started, the product owner uses the team's average velocity (the amount of work typically completed in a sprint) to compile a suggested set of stories for the sprint, called the "sprint forecast", that maximizes value to the customer. The product owner should also consider these three factors:

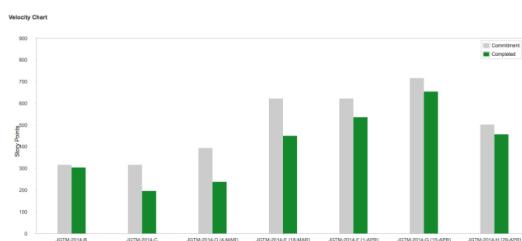
- Public holidays, personal vacation, and team-wide events
- Priority of stories in the backlog (ideally, they suggest the top-ranked items)
- How (and whether) that body of work will get the product closer to its final goal

TIP

Product owners can use the sprint marker to calculate velocity.

A screenshot of the Jira interface for the 'Scrum: Teams In Space' project. The left sidebar shows various filters like 'EPICs', 'All issues', and 'Sprint 6'. The main area lists 24 issues under 'Sprint 6'. One issue, T15-44, is highlighted with a red border and labeled 'Selected'. The details for T15-44 show it is a task for 'Local Team Support' assigned to Jennifer Evans, with a status of 'CLOSED' and a fix version of '2.0'. The 'Details' section also includes components like 'Web Site' and 'Affects Versions'.

If the team is new and doesn't have an established velocity, the product owner shouldn't suggest a sprint forecast. Instead, this should be an all-team exercise because it's important to have buy-in from each member. At first, the team will use its best judgment with regard to the forecast, and work through a few sprints of trial and error. Once the team's velocity—nicely visualized by the velocity chart in Jira—is established numerically, use that metric to guide the sprint forecast.



Once the product owner presents their ideas for the sprint forecast, the team can validate (and/or adjust) it and agree to a plan of action for the sprint.

TIP

The team should consider ways to reduce technical debt with each sprint. Creating a quick filter that highlights bugs in the team's backlog is an easy way to highlight important bugs to be pulled into the sprint as the team goes about working on user stories in various areas of the code base. The "tech debt" quick filter uses the JQL "type in (bug)" to limit the view to bugs. Just extend the JQL query to include additional issue types as needed.

The screenshot shows a Jira Scrum board titled 'Scrum: Teams in Space'. On the left, there's a sidebar with 'EPICS' and a tree view of projects like 'SeeSpaceEZ Plus', 'Space Travel Partners', 'Summer Satin Sale', 'Afterburner Plus', 'Large Team Support', 'Local Mars Office', 'Hyper-speed shuttle', 'New launch platforms', 'Delicious Space Nutrition', 'Speculation', and 'Monogram Delight'. The main board has two columns: 'Sprint 6' (4 of 24 issues visible) and 'Backlog' (7 of 44 issues visible). A ticket for 'TIS-87 Developer toolbox does not display by default' is selected in the 'Sprint 6' column. The ticket details show it's a 'CODE REVIEW' status, assigned to 'Jennifer Evans', created on 16Oct14 at 9:43 AM, and updated on 20Oct14 at 1:48 PM. The ticket description includes several bullet points about developer toolbars.

The next step in sprint planning is to walk through the stories and describe what work is required to complete each story. As the team plans, make sure someone is capturing the key points inside each Jira ticket. That way both the decision and the rationale are easy to see later. The team should consider questions like...

- Has the story's definition changed since it was written? Is there new contextual information the team needs to consider?
- Is the story's estimate still valid? Does the entire team agree on the estimate? If not, the scrum master should guide the team through re-estimation.
- What tasks are required to complete this story? Use sub-tasks to help parallelize work and optimize flow. If the team finds unique stories as they break down work, promote those tasks to fully independent stories.
- What are the testing implications for this story? How can we automate testing? (Remember that manual test scripts are essentially technical debt.)
- Are any specialist skill sets required? How can we optimize the specialist's time without blocking the rest of the team?
- How does this story affect the product's architecture? Are there specific people the team needs to involve in design and code review?
- Are there any dependencies between stories? Can we complete all of the work during the sprint given those dependencies?

There's a strong temptation to rush through this detailed exercise. But good planning pays numerous dividends once the sprint starts. The focus here is to understand how the work is going to get done, with the scrum master facilitating conversation amongst the team. It's important that everyone is heard so the team feels a sense of ownership once the plan is in place.

TIP

During sprint planning, it's easy to move stories in and out of the sprint as the team creates its sprint forecast. Just right-click on an issue to move it in or out of the sprint.

The screenshot shows the Jira backlog with a list of issues. A context menu is open over the ticket 'TIS-9 After 100,000 requests the SeeSpaceEZ Plus [2.0]'. The menu options include 'Send to', 'Sprint 6' (which is highlighted), 'Top of Backlog', 'Bottom of Backlog', 'Add flag', 'View in Issue Navigator', 'View in Excel', and 'Bulk Change'.

4. Ready, set, sprint!

At this point in the meeting, the team should be comfortable with the sprint forecast. At the end of sprint planning it's good practice to get verbal approval from everyone in the room about what the team is actually committing to shipping at the end of the sprint. Also, establish that each team member has at least one task to start on and nobody is duplicating work.

Team engagement and morale will naturally fluctuate from sprint to sprint. These variations often show up during sprint planning, but resist the temptation to dig into it right then and there. Instead, use the team's retrospective to understand any issues that are impacting morale. Teams that respond quickly to culture and development concerns are happier, more productive, and write better code.

Hungry for more? Check out The Agile Coach – a site with articles, webinars,

[www.theagilecoach.com](#)