

# Learn Scrum with JIRA Software

## Scrum tutorial

In this tutorial, we'll give you step-by-step instructions on how to drive a Scrum project, prioritize and organize your backlog into sprints, run the Scrum ceremonies and more, all within JIRA Software.

**What is scrum?** Scrum is an iterative software development framework. In this tutorial, you will have a backlog to prioritize your work into sprints, and a board with a simple workflow.

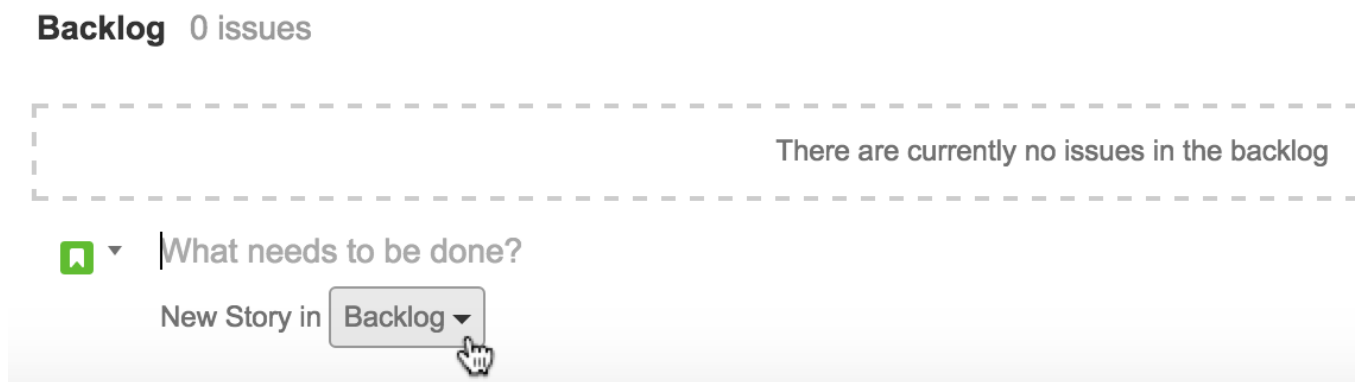
### Step 1: Create a Scrum project

Once you start an account of JIRA Software and log in (link under prerequisite), you will have the option to create a project. When you're prompted to select a project type, ensure you select **Scrum software development**. Otherwise, you can learn how to start a [Kanban project here](#).

A Scrum board is automatically created with a Scrum software development project. Once you've created your project, you will land on the empty backlog of your Scrum board. Now is your chance to start filling your backlog with user stories or tasks you have to complete your project, feature, or product.

### Step 2: Create user stories or tasks in the backlog

In JIRA Software, we call user stories, tasks, and bugs "**issues**". Quickly create a few user stories with the quick create option on the backlog. If you don't have a project or feature in mind, just create sample user stories to get started and see how it works.



**What are user stories?** In an agile framework, user stories are the smallest units of work. As a *{type of user}*, I want *{goal}* so that I *{receive benefit}*.

Let's use a website as a simple example to create a user story.

*As a customer, I want to be able to create an account so that I can see the purchases I made in the last year to help me budget for next year.*

User stories are sketched out by the product owner, and then the entire product team collectively determines detailed requirements. These are the granular pieces of work that help define the implementation items for the story and the upcoming sprint.

Once you've created 6 or more user stories, you can start prioritizing your stories in the backlog. In JIRA Software, you rank or prioritize your stories by dragging and dropping them in your backlog, in the order of importance that works for your team.

### Step 3: Estimate your stories

Before you begin your first sprint, it's important to estimate the stories in the backlog so you can determine how long chunks of work will take to complete.

**What is agile estimation?** Traditional software teams give estimates in a time format: days, weeks, months. Many agile teams, however, have

transitioned to story points. Story points rate the relative effort of work in a Fibonacci-like format: 0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100. It may sound counter-intuitive, but that abstraction is actually helpful because it pushes the team to make tougher decisions around the difficulty of work.

Estimates will also help you gauge how much work you should add to the next sprint based on the number of team members you have. After a few sprints, your team will get better at figuring out how much work they can do each sprint, which will help avoid over-committing.

In JIRA Software, you can estimate issues by going to the backlog and entering a number in the **Estimate** field for each issue.

## Step 4: Create a sprint

Create your first sprint in the backlog so you can start organizing issues into sprints.

**What is a sprint?** In Scrum, teams forecast to complete a set of user stories during a fixed time duration, known as a sprint. Generally speaking, sprints are one, two, or four weeks long. It's up to the team to determine the length of a sprint — we recommend starting with two weeks. That's long enough to get something accomplished, but not so long that the team isn't getting regular feedback. Once a sprint cadence is determined, the team perpetually operates on that cadence. Fixed length sprints reinforce estimation skills and predict the future velocity for the team as they work through the backlog.



The screenshot shows the JIRA Backlog with 5 issues. Each issue has a green icon, a priority indicator (up arrow), a key, a title, and an estimate in a grey circle. At the bottom, there is a '+ Create issue' button.

Issue Key	Title	Estimate
TIS-1	Add support for teams larger than 20 people	4
TIS-2	Plans for our Summer Saturn Sale	8
TIS-3	Make working with our space travel partners easier	16
TIS-4	500 Error when requesting a reservation	6
TIS-5	Requesting available flights is now taking > 5 seconds	4

[Create sprint](#)  
Create your first sprint for your board

## Step 5: Hold the sprint planning meeting

Before starting a sprint, you should hold the sprint planning meeting with the rest of your team. In this meeting, you will estimate any un-estimated stories from the prioritized backlog, get the commitment from the team on what they can complete, and ultimately build the first sprint.

You should also consider any upcoming time-off your team may have, like holidays, planned leaves — as well as other issues that may impact the completion of the sprint tasks.

Also, note that in step 3, your team should have already estimated the stories in story points, and in their order of priority. During sprint planning, you should just be estimating the stories that haven't been estimated as yet — perhaps due to lack of clarity at the time of story estimation.

### What is the sprint planning meeting?

**Attendees:** Required: development team, scrum master, product owner

**When:** At the beginning of a sprint.

**Duration:** Usually an hour per week of iteration — e.g. a two-week sprint kicks off with a two-hour planning meeting.

**Agile Framework:** Scrum. (Kanban teams also plan, of course, but they are not on a fixed iteration schedule with formal sprint planning)

**Purpose:** Sprint planning sets up the entire team for success throughout the sprint. Coming into the meeting, the product owner will have a prioritized product backlog. They discuss each item with the development team, and the group collectively estimates the effort involved. The development team will then make a sprint forecast that outlines how much work the team can complete from the product backlog. That body of work

then becomes the sprint backlog.

## **Step 6: Drag and drop issues into your sprint**

The backlog is where you can prioritize your most critical user stories. In the sprint planning meeting, you can decide what issues you want to work on in your first sprint. This is important, because you want to avoid scope creep. Have a think about how much work your team is willing to commit to, and what you want to achieve in the sprint.

When you're ready, drag the stories into the sprint you just created.

## **Step 7: Start your first sprint**

Name the sprint. Some teams name the sprint based on their goal. If there is a commonality between the issues in the sprint, name the sprint around that theme. Otherwise, you can name the sprint whatever you like. Add a duration of the sprint and start and end dates. The start and end dates should align to your team's schedule. For example, some teams start sprints on a Monday and then end on a Friday morning in the next week. Other teams decide to start and end their sprints mid-week. It's up to you! If you're unsure how long your sprints should be, we recommend trying two weeks.

Once you start your sprint, you will be taken to the Active sprints tab in the project.

**JIRA** Dashboards ▾ Projects ▾ Issues ▾ Tempo ▾ Capture ▾ Boards ▾ Portfolio ▾ [Create](#)

**Teams in Space**  
TIS board ▾

[Backlog](#)

[Active sprints](#)

[Releases](#)

[Reports](#)

[Issues](#)

[Components](#)

[Test sessions](#)

[+ Add shortcut](#)

[+ Invite your team](#)

[Project settings](#)

## TIS Sprint 1

**QUICK FILTERS:** [Only My Issues](#) [Recently Updated](#)

To Do

In Progress

- Add support for teams larger than 20 people  
TIS-1
- Plans for our Summer Saturn Sale  
TIS-2
- Make working with our space travel partners easier  
TIS-3
- 500 Error when requesting a reservation  
TIS-4
- Requesting available flights is now taking > 5 seconds  
TIS-5

This is where your team will work to pick up items from the to-do column and move them into in-progress and eventually, done!

## Step 8: Hold the daily standup meeting

After your sprint has started, have your team meet daily, typically in the morning, to review what everyone is working on. The purpose of this is to see if anyone on your team is experiencing any roadblocks towards the completion of sprint tasks.

## What is the daily standup meeting?

**Attendees Required:** development team, scrum master, product owner

**Optional:** team stakeholders

**When:** Once per day, typically in the morning

**Duration:** No more than 15 minutes. Don't book a conference room and conduct the standup sitting down. Standing up helps keep the meeting short!

**Agile Framework:** Scrum and Kanban.

**Purpose:** The daily standup is designed to inform everyone quickly of what's going on across the team. It's not a full status meeting. The tone should be light and fun, but informative. Have each team member answer the following questions:

- What did I complete yesterday?
- What will I work on today?
- Am I blocked by anything?

There's an implicit accountability in reporting what work you completed yesterday in front of your peers. No one wants to be the team member who is constantly doing the same thing and not making progress.

**ProTip:** Some teams use timers to keep everyone on track. Others toss a ball across the team to make sure everyone's paying attention. Many distributed teams use video-conferencing or group chat to close the distance gap. Your team is unique — your standup should be, too!

You can use the Active sprints of your Scrum board during the daily standup, so that each member can view the tasks they're working on.

## View the Burndown Chart

It's actually a good idea to check the Burndown Chart during a sprint. In JIRA Software, the Burndown Chart shows the actual and estimated amount of work to be done in a sprint. To view this chart, click Reports from the sidebar, and then select the Burndown Chart from the reports

dropdown.

## **What is a Burndown Chart and how do you read it**

A Burndown Chart shows the actual and estimated amount of work to be done in a sprint. The horizontal x-axis in a Burndown Chart indicates time, while the vertical y-axis indicates issues.

Use the Burndown Chart to track the total work remaining for a sprint, and to project the likelihood of achieving the sprint goal. By tracking the remaining work throughout the iteration, a team can manage its progress and respond accordingly.

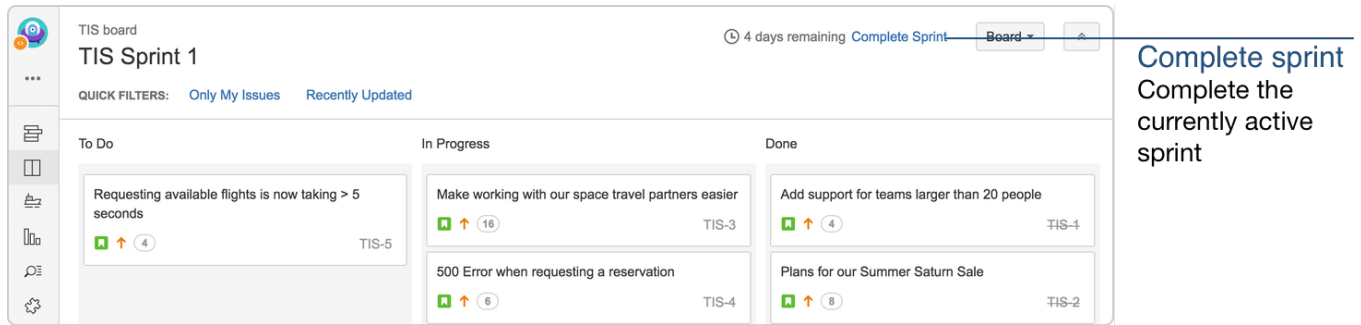
Scrum teams organize development into time-boxed sprints. At the outset of the sprint, the team forecasts how much work they can complete during a sprint. The Burndown Chart then tracks the completion of work throughout the sprint. The x-axis represents time, and the y-axis refers to the amount of work left to complete, measured in either story points or hours. Scrum teams then use the Burndown Chart to monitor how far off the team members are from completing the forecasted work by the end of the sprint.

A team that consistently meets its forecast is a compelling advertisement for agile in their organization. But don't let that tempt you to fudge the numbers by declaring an item complete before it is. It may look good in the short term, but in the long run, this only hampers learning and improvement. In this scenario, a team using the Burndown Chart has more opportunity to take the necessary actions to stay on track, and eventually hit the sprint goal.

## **Step 9: Complete the sprint**

At the end of the sprint, you must complete it.





Once you complete the sprint, you'll be able to review the Sprint Report.

## What is the Sprint Report and how to read it

The Sprint Report lists the work completed, work not completed, and any work added after the sprint started.

### Questions to ask:

- Did the team meet the sprint forecast?
- Was there work added or removed during the middle of the sprint?
- Did any work not get completed within the sprint?
- If so, why?

**Goal:** Understand how the sprint forecast compared to the actual delivery.

## Step 10: Hold the sprint retrospective meeting

After you complete the sprint, have your team do a retrospective.

Document your retrospective somewhere. May we suggest Confluence?

### What is a sprint retrospective meeting?

**Attendees Required:** development team, scrum master, product owner

**When:** At the end of an iteration.

**Duration:** 60 minutes.

**Agile Framework:** Scrum and Kanban. Scrum teams do sprint

retrospectives based on a fixed cadence. Kanban teams can benefit from occasional retrospectives, too.

**Purpose:** Agile is about getting rapid feedback to make the product and development culture better. Retrospectives help the team understand what worked well—and what didn't.

Retrospectives aren't just a time for complaints without action. Use retrospectives to find out what's working so the team can continue to focus on those areas. Also, find out what's not working and use the time to find creative solutions and develop an action plan. Continuous improvement is what sustains and drives development within an agile team, and retrospectives are a key part of that.

Questions to ask:

- What did we do well during the sprint?
- What could we have done better?
- What are we going to do better for next time?

**ProTip:** Even if things are going well across the team, don't stop doing retrospectives. Retrospectives provide ongoing guidance for the team to keep things going well.

## Step 11: Repeat from step 4

At this point, you've got the basics on building your backlog with user stories, organizing your user stories into sprints, starting your sprint, and holding Scrum ceremonies. You can decide if this is working for your team, or if you'd like to move forward into some more advanced topics.

After you and your team have mastered the steps above. Move on to the advanced article: [How to do advanced scrum practices with JIRA Software](https://www.atlassian.com/agile/how-to-do-scrum-with-jira-software).