

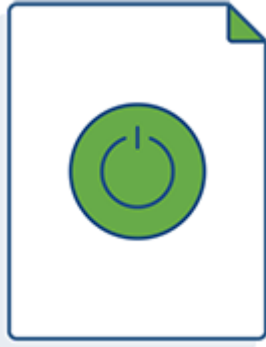
JIRA Software Lab Guide

Contents

JIRA Software Roles	3	
Admins		3
Getting started with JIRA Software	3	
JIRA Software workflow		4
Getting started as a JIRA Software manager		5
Lab 1 : Create your project	7	
Lab 2 : Add users	8	
Lab 3 : Configure estimation and tracking	9	
Lab 4 : Create issues	11	
Lab 5 : Estimate issues	13	
Lab 6 : Rank the backlog	13	
Lab 7 : Create a sprint	14	
Lab 8 : Start your sprint	15	
Lab 9 : Transition issues	15	
Lab 10 : View the Burndown Chart	16	
Lab 11 : End the sprint	17	
Lab 12 : View the Sprint Report	18	
Lab 13 : Velocity Chart	19	
Lab 14 : Find your issue	23	
Lab 15 : Work on an issue	24	
Lab 16 : Collaborate with your team	24	
Lab 17 : Create a branch	24	
Lab 18 : Start coding	25	
Lab 19 : Send your work for review	25	
Lab 20 : Finish your work	25	
Lab 21 : Learn to plan and estimate for scrum teams	26	
Lab 22 : Use epics for context	27	
Lab 23 : To create a version	28	
Lab 24 : Customize the team board	29	
Lab 25 : Estimating	30	
Lab 26 : Analyze team reports	32	
Lab 27 : Optimize future plans	35	

JIRA Software Roles

JIRA Software integrates development tools with agile features to help agile teams build great software.



Admins

- Getting started guide
- Installing JIRA Software
- JIRA Software overview
- Planning and estimating



team members

- Working on agile projects
- Searching for issues
- Working with issues

Scrum Masters/project

leads

- Leading agile projects
- Building backlogs
- Getting to work

Getting started with JIRA Software

JIRA Software unlocks the power of agile by giving your team the tools to easily create & estimate stories, build a sprint backlog, identify team commitments & velocity, visualize team activity, and report on your

team's progress.

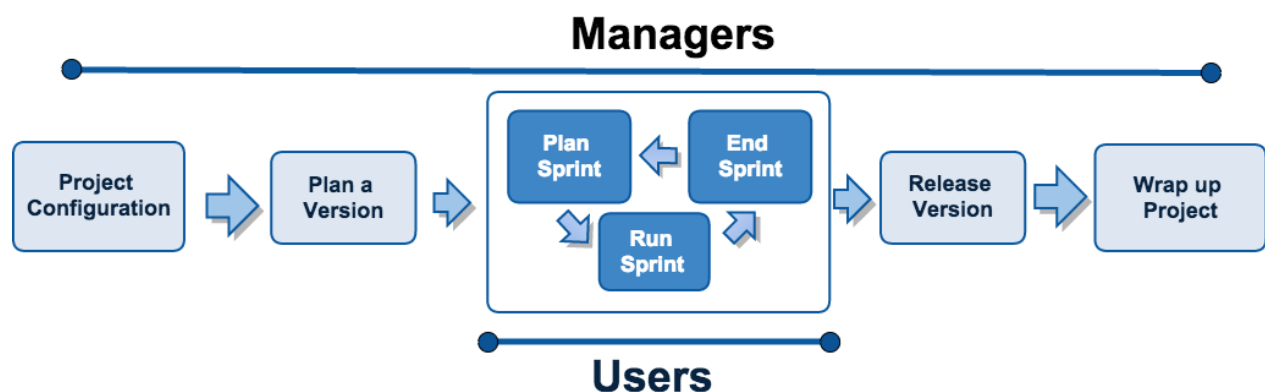
To give your team a tour through a complete project, the following guide contains two labs that show a simple agile workflow at a small software company. This guide touches on some of the most used features and follows the development team at Teams in Space as they work to improve their next generation space travel software.

In this simple workflow, a manager sets up and runs a project while individual users add content and work with issues within a sprint. The following pages are set up to follow this structure and are divided into two labs:

- [Getting Started as a JIRA Software Manager](#): A guide for managers who set up and run the agile project
- [Getting Started as a JIRA Software User](#): A guide for users who work on and resolve issues

JIRA Software workflow

The simplified workflow and setup in this lab is intended for teams using scrum, so some of the concepts presented may not apply to all teams. Here's how the manager and users work together to complete a sprint:



Roles

Typically, variations on the following roles can be found in an agile development environment. We recommend that managers complete both labs while users complete the user lab.

JIRA Software Managers

User with administrative rights for your instance of JIRA Software. These roles are usually filled by scrum masters or development managers.

Managers can...

- Access all features in JIRA Software
- Create and update projects
- Add and remove users
- Start and end sprints
- Perform other agile tasks

JIRA Software Users

User who works on and resolves issues. These roles are normally filled by software developers.

Users can...

- Create and manage issues
- Check code review status
- Create branches (with integrated development tools)

[Get started as a manager](#)

[Get started as a user](#)

Getting started as a JIRA Software manager

Welcome to JIRA Software! This lab will help you set up your workspace, and create and edit a project for your team. To get your project up and running, we'll also show you how to create, set up, and organize your scrum board.

By the end of this lab, you will have:

- Created a new software project
- Added users

- Prepared your backlog
- Started and completed a sprint
- Evaluated the results

But first, let's set up your JIRA Software workspace.

Audience: Scrum masters and software development managers that want to streamline their agile workflow!

Time: 1h 30m

Let's go!

Setting up your workspace

1. Setting up your workspace
2. Customizing your project
3. Creating your backlog
4. Grooming your backlog
5. Planning your sprint
6. Tracking your progress
7. Wrapping up your work
8. Doing more with your agile projects

When setting up your JIRA Software workspace, you'll need to do the following:

1. Sign up for a JIRA Software site.
2. Create your project.
3. View your Scrum board.

Sign up for a JIRA Software site (OPTIONAL – skip this step)

If you want to install JIRA Software Server instead of signing up for JIRA Software Cloud, see these instructions: [Installing JIRA applications](#), then skip to the 'Create a project' step below.

You will need your own JIRA Software site for this lab. Let's set you up with a JIRA Software Cloud site.

Cloud is our hosted offering and will allow you to set up your own site without installing a thing! If you already

have a site, you can skip this step.

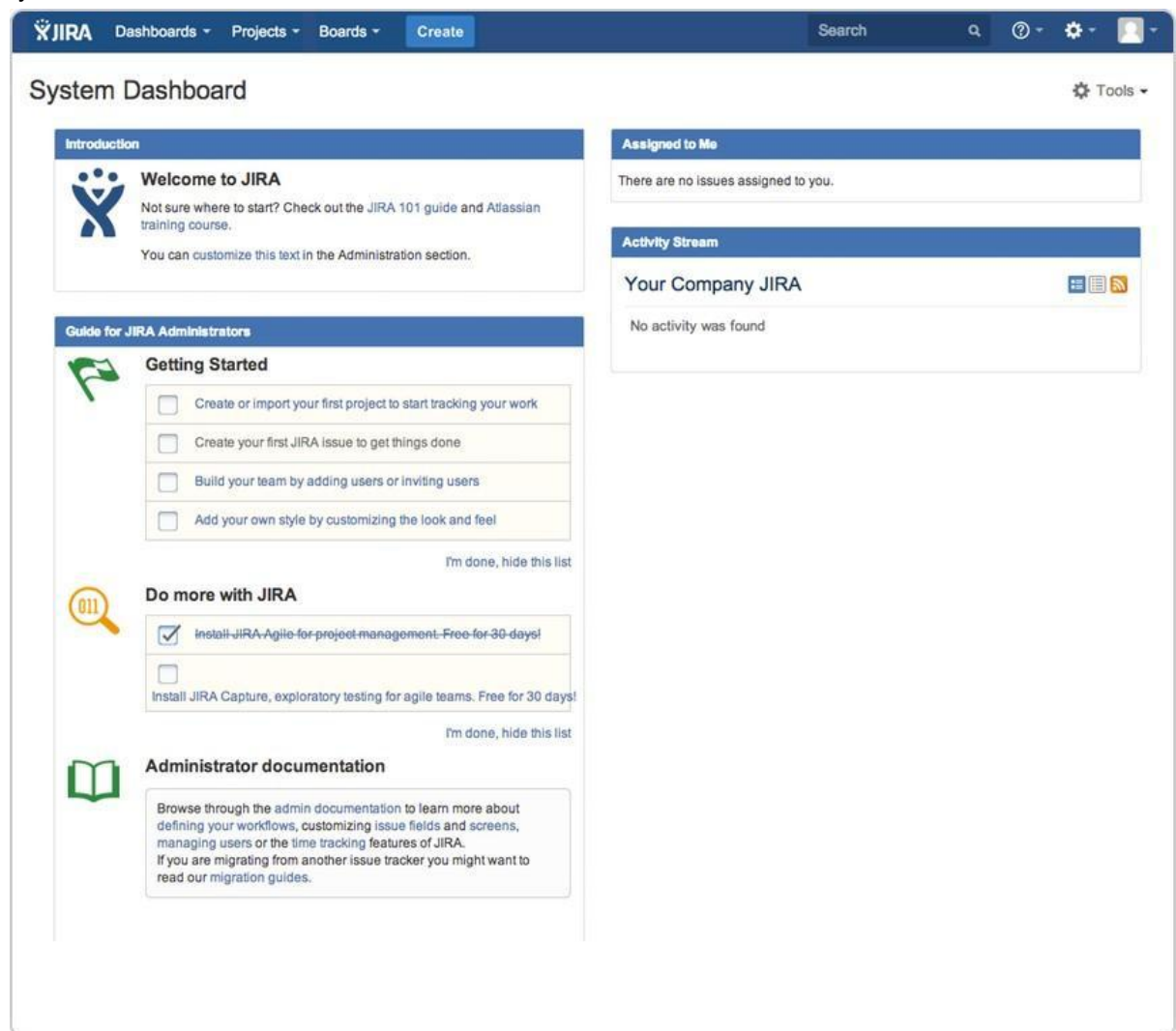
Signing up for JIRA Software will provide you with a fully functional JIRA Software site for one month.

1. Open [this link](#) in a new tab to view the Atlassian Cloud signup page.
2. Follow the signup form steps to enter your site URL and administrator username.
3. Once you have completed the signup process, grab a quick coffee (tea works as well!) — it takes about 10 minutes to create your JIRA Software Cloud site. You'll receive an email when your Cloud site is ready.

Lab 1 : Create your project

A JIRA Software project is a collection of issues and tools that allows your team to coordinate the development of a piece of software. Every project contains configurable boards and workflows that you can create and customize to fit your team's workflows.

1. Log in to your Cloud site using the link and credentials you set when you signed up. You'll see the System Dashboard, as shown below.



2. Choose **Projects > Create project**, and then select your project type. Typically, you would choose 'Scrum software development' for iteration planning, or 'Kanban software development' for constraint-based task management. For this lab, let's do Scrum since most software developers use scrum in agile projects.
3. Choose **Scrum software development > Next**.

Create project

SOFTWARE

- Scrum software development**
Agile development with a board, sprints and stories. Connects with source and build tools.
- Kanban software development**
Optimise development flow with a board. Connects with source and build tools.
- Basic software development**
Track development tasks and bugs. Connects with source and build tools.

BUSINESS

- Task management**
Quickly organize and assign simple tasks for you and your team.
- Project management**
Plan, track and report on all of your work within a project.
- Process management**
Track all the work activity as it transitions through a streamlined process.

[Import a project](#) | [Create with shared configuration](#) **Next** [Cancel](#)

4. In the Name field, enter **Teams in Space**.
5. Choose **Submit** to create your new project.

View your Scrum board

A Scrum board is automatically created with your new project. Use your Scrum board to view and work on issues, such as new features or bugs. You can rank, view, edit, and track issues on your scrum board via the **Backlog**, **Active sprints**, and **Reports**. Don't worry, we'll discuss these three along the way.

This is what your Scrum board will look like:

Teams in Space
Teams in Space

Backlog

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

FILL YOUR BACKLOG WITH ISSUES
This is your team backlog. Create and estimate new issues, and prioritize your backlog using drag and drop.

Backlog 0 issues

[Create Sprint](#)

[+ Create issue](#)

Congratulations, you now know the basics of the JIRA Software interface, and have just created your first project! Let's go ahead and customize it!

JIRA Software makes it easy to customize your board to fit your workflow. In this step, let's set up your software team, which will be the Teams in Space team. Teams in Space is an imaginary new company pioneering space travel for innovative travel providers.

Lab 2 : Add users

A software team without any members just won't cut it! Let's go ahead and configure Teams in Space's development team. The team consists of you (the manager) and two developers (Jennifer and Kevin).

1. Choose



> **User management.**

2. Select **Users**.

3. In the Create new users page, enter the following details for each user:

Full name	Email address
Jennifer Evans	jevans@veryrealemail.com
Kevin Campbell	kcampbell@veryrealemail.com

4. Click **Create users**.

Lab 3 : Configure estimation and tracking

One of the first things you need to do for a new Scrum project is to configure estimation and tracking. This is essential for understanding how much work your team has and how much it can do, as you build a backlog, run sprints, and review reports.

Scrum teams use different methods to calculate the amount of work involved in completing an issue, and in turn, a sprint. Many teams separate **estimation** (used for measuring the amount of work in the backlog and calculating velocity) from **tracking** (used for measuring burndown of hours used during the sprint), using different units for each. For example, some teams *estimate* tasks in *story points*, then *track* tasks using *hours*.

In this lab, Teams in Space uses story points for both estimation and tracking, as per the instructions below.

1. Navigate to the desired board, then click **Board > Configure**.
2. Click the **Estimation** tab.
3. In the **Estimation Statistic** field, select **Story Points**. Leave the **Time Tracking** field set to None.

Estimation

Issues can be estimated when in the Backlog to get an idea of how much work is being committed to in a sprint. [Read more about estimation and tracking.](#)

Estimation Statistic Story Points

Estimate issues in the Backlog by entering values for **Story Points**. Your velocity from sprint to sprint will be measured against these estimates.

Time Tracking ☒ None

Issues will burn down their **Story Points** value upon completion.

☐ Remaining Estimate and Time Spent

Track time against issues using JIRA's Remaining Estimate and Time Spent fields.

Great! You've successfully added users and configured your project to use story points for estimates. Now, let's work on your backlog!

Your backlog is a list of tasks that represents outstanding work in a project. Usually, a project would have issues in the backlog, and you can add these issues to a sprint so your team can work on them. Since Teams in Space is a new project, you won't have issues on your backlog. Let's create some work for your team.

How do I get to my board?

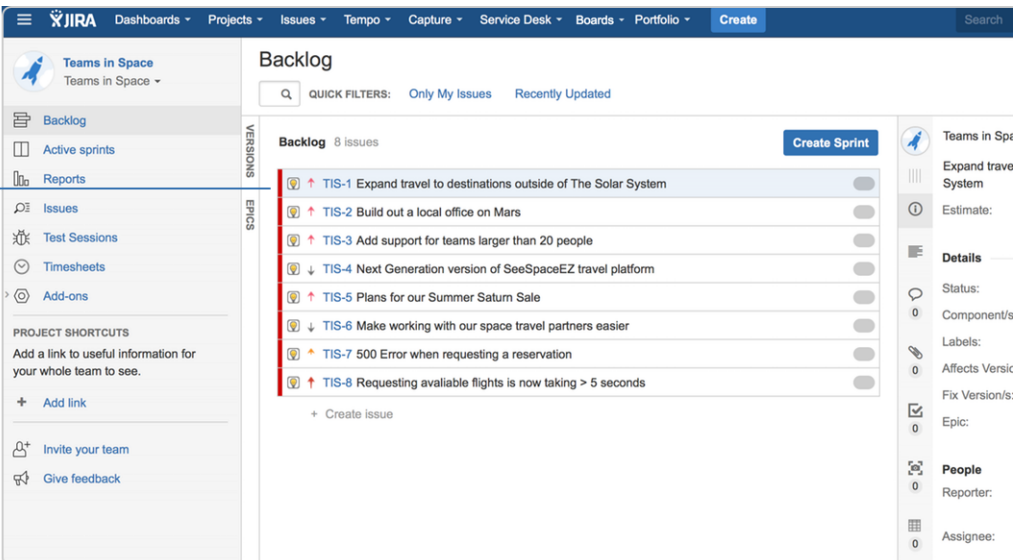
If you navigate away from your board, you can easily return to it by selecting Boards from the top navigation menu, and selecting your board "Teams in Space" from the list of recent boards.

Backlog

The Backlog gives you a place to organize your sprints. You can create new issues or sub-tasks, organize

your backlog, create versions, organize via epics, and start sprints.

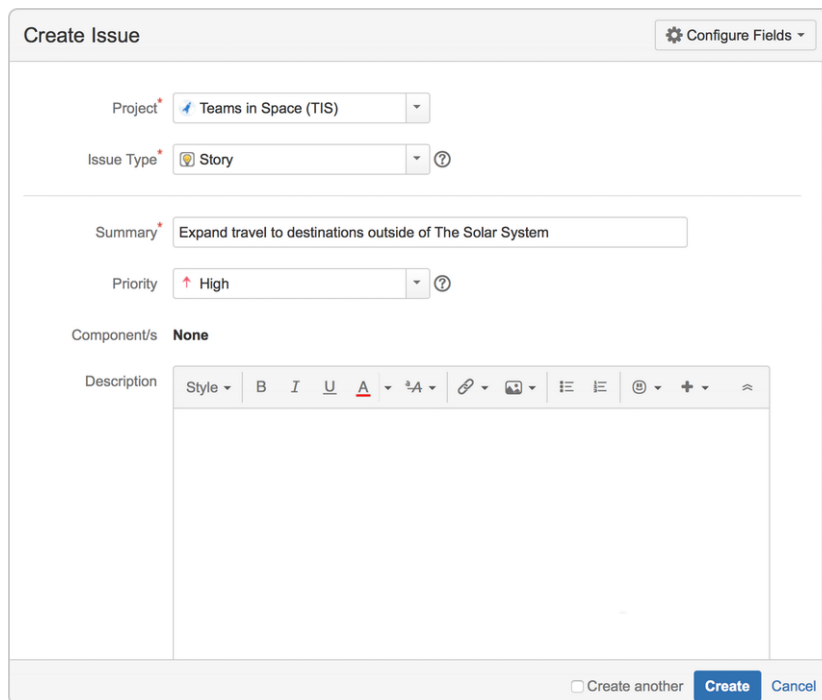
Backlog
A list of the current issues in the backlog



Lab 4 : Create issues

By default, any team member can create issues. In this lab, however, you will create all of the backlog issues.

1. On the Teams in Space board, choose **Create** at the top of the screen.



Tip! If you select the **Create** another checkbox, JIRA Software will create your issue and automatically pre-populate a new 'Create Issue' dialog box with your previous issue details.

Fill in the fields using the data shown below. Only the fields with * are mandatory.

- **Project:** Teams in Space
 - **Issue Type:** Story
 - **Summary:** Expand travel to destinations outside of The Solar System
 - **Priority:** High
 - Leave all other fields blank or at their default values.
2. Choose **Create** to make a new issue. An issue key (**TIS-1**) is created for this issue, which comes in handy when searching for issues later.

Add more issues

One issue isn't enough to get your team working! Let's add more issues so you can create and run a complete sprint. Create the following issues using the same steps as above.

Type	Summary	Priority
Story	Build out a local office on Mars	High
Story	Add support for teams larger than 20 people	High
Story	Next Generation version of SeeSpaceEZ travel platform	Low
Story	Plans for our Summer Saturn Sale	High
Story	Make working with our space travel partners easier	Low
Story	500 Error when requesting a reservation	Medium
Story	Requesting available flights is now taking > 5 seconds	Highest

Great! Your backlog's all set! Now, let's plan some work for your developers to do!

An essential part of agile is regularly "grooming" or reviewing the contents of your backlog, particularly before starting any new work.

Adjust your backlog

Before starting your sprint, you need to prepare your backlog. You can easily adjust your backlog by:

- Right-clicking on issues to view, estimate, or add details
- Ranking your issues by dragging and dropping
- Creating new issues

- Editing an issue using the issue detail view

Lab 5 : Estimate issues

Now, let's add some estimates to the issues in our backlog. This way, you can easily determine what you can accomplish, and your team can also have a way of measuring the success of the sprint.

1. On the Teams in Space board, select **Backlog**.
2. Select each issue on the left-hand side of the screen to display the issue details on the right-hand side of the screen.
3. Click the **Estimate** field on the right-hand side of the screen for each of the issues, and enter the following information for each issue:

Issue	Estimate
Expand travel to destinations outside of The Solar System	6
Build out a local office on Mars	2
Add support for teams larger than 20 people	4
Next Generation version of SeeSpaceEZ travel platform	3
Plans for our Summer Saturn Sale	8
Make working with our space travel partners easier	16
500 Error when requesting a reservation	6
Requesting available flights is now taking > 5 seconds	4

Lab 6 : Rank the backlog

By default, the issues in your backlog are ranked in the order in which you added them. You can change the rank of your issues according to their relative priority. This helps you organize the issues in your backlog more effectively.

Rankings let you determine whether an issue is more important or urgent than another issue. For example, you may have two separate issues that are both of 'High' priority. Using JIRA Software ranking, you can assign one of the issues a higher ranking than the other.

1. Find issue **TIS-8** in your backlog. This issue has the 'Highest' priority, and therefore should be at the top of your backlog.
2. Select **TIS-8** and drag it to the top.
3. Move issues **TIS-2** and **TIS-5** to positions two and three in the backlog. These issues have 'High' priority, but they're not as high a priority as TIS-8.

The screenshot shows the JIRA Backlog interface with 8 issues. The issues are ranked by priority, with TIS-8 at the top. The interface includes a 'Create Sprint' button and a 'Create issue' link at the bottom.

Issue	Estimate
TIS-8 Requesting available flights is now taking > 5 seconds	4
TIS-2 Build out a local office on Mars	2
TIS-5 Plans for our Summer Saturn Sale	8
TIS-1 Expand travel to destinations outside of The Solar System	6
TIS-3 Add support for teams larger than 20 people	4
TIS-4 Next Generation version of SeeSpaceEZ travel platform	3
TIS-6 Make working with our space travel partners easier	16
TIS-7 500 Error when requesting a reservation	6

Great! You've just groomed your backlog. Now, let's plan out the details of your sprint!

A sprint is a short period (ideally two to four weeks) during which a development team implements and delivers a discrete product increment, e.g. a working milestone version. In this lab, your team will be working in two-week long sprints. Let's go ahead and create a sprint for your team.

Sprint planning

Before creating and starting a sprint, your Scrum team would typically hold a sprint planning meeting. In this meeting, your team should:

- Review the estimates for selected issues
- Break down the selected issues into an initial list of sprint tasks
- Consider upcoming employee time-off, holidays, and other issues that may impact the completion of these sprint tasks
- Gauge the team's capacity to complete these sprint tasks

By the end of the meeting, your team should be confident enough to commit to completing the work in the sprint.

In this lab, we will assume that the Teams in Space team can handle 20 story points of work in a sprint, and that everyone is available for the full sprint. Typically, you would know how much work your team can complete in a sprint by reviewing information from past sprints, usually through velocity and sprint reports.

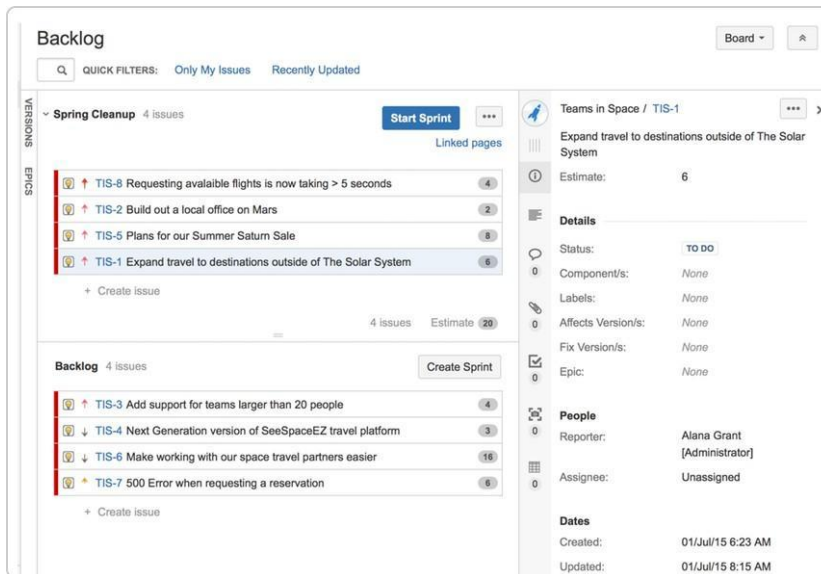
Need help scheduling?

By using JIRA Software in tandem with Confluence, you can embed your sprints using Team Calendars for Confluence. This helps you see the duration of your sprint and how your team's availability or other team events could impact the sprint.

Lab 7 : Create a sprint

1. On the Teams in Space board, click **Backlog**.
2. Click **Create Sprint** at the top of the Backlog.
3. Your new upcoming sprint will be added to your board, below any other future sprints. Select the **Sprint 1** text and edit the name of the sprint to 'Spring Cleanup'.

4. The top 4 issues in the Backlog are equal to 20 story points. This is what the team estimated that they could accomplish in the upcoming sprint. Drag and drop the top four issues from the Backlog into your new sprint.



Streamline your work

By connecting your JIRA Software instance to a Confluence instance, you can link Confluence pages to your sprints via JIRA Software to build stronger user stories and to better plan for sprints or releases. For example, you may want to write up the sprint meeting notes in Confluence and link them to the sprint.

Lab 8 : Start your sprint

Now that you have created a sprint, you can go ahead and start it.

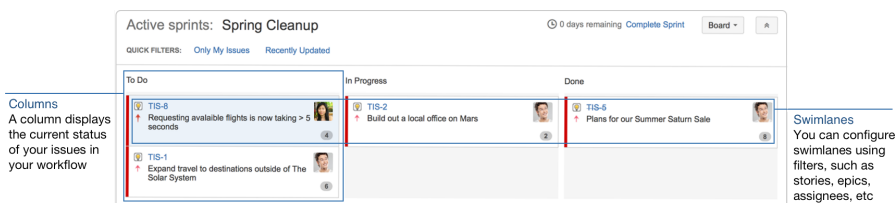
1. Click **Start Sprint**.
2. Today's date and current time become the start date and time for the sprint. For the purpose of this lab, enter an end date of 5 minutes from the start date and time.
3. Select **Start** to start the sprint and move the issues into Active sprints.

Congratulations! You've successfully got Teams in Space up and running! Now, let's look at Active sprints to track your team's progress.

During your sprint, you and your team will need to monitor your progress to make sure that everyone is on the same page. There are several tools that you can use to do this, which are described below.

Active sprints

The **Active sprints** page is where you monitor the progress of your team's work during a sprint. Here, your team can transition issues through a series of columns (statuses), allowing everyone to quickly visualize the progress being made in the sprint. You can also edit issues by adding information, such as descriptions, attachments, and comments.



More about Active sprints

Issues that you ranked higher in your backlog appear at the top of the column, to make it easier for users to select the most important issue to work on.

Lab 9 : Transition issues

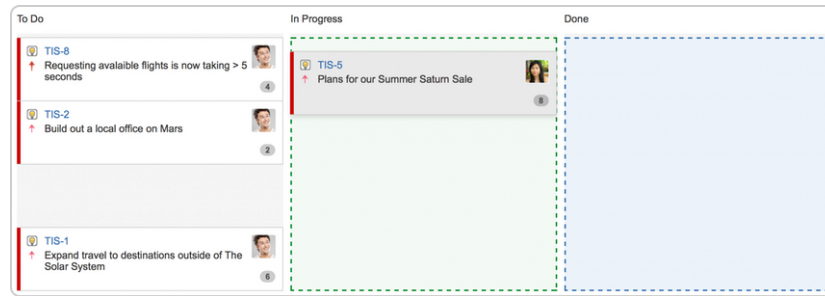
During the sprint, let's say your team was only able to finish three issues, TIS-1, TIS-2, and TIS-5. Issue TIS-8 remains in the 'To Do' column since no

Tip!

Use <Ctrl> or

team member was able to work on the issue. Let's show on the board what happened during the sprint by transitioning issues from one column to another.

1. On the Teams in Space board, click **Active sprints**.
2. Select **TIS-5** and move the issue to the 'In Progress' column.



3. Select **TIS -5**, **TIS-2**, and **TIS-1**, and then move the issues to the 'Done' column.

Lab 10 : View the Burndown Chart

You've just seen how your team is progressing, from the Active sprints of your board. The Burndown Chart is another useful tracking tool, which can help you visualize your team's progress, as well as determine whether your team is on target to achieve the sprint goal.

1. On the Teams in Space board, click **Reports**.
2. Select **Burndown Chart**.



The grey line in your Burndown Chart is a guide showing the rate of work required to complete the sprint. The red line, on the other hand, shows the actual work completed by your team.

If your Burndown Chart shows the red line above the grey line, your team may not achieve the sprint goal. You may want to consider removing some issues from the sprint. Any changes to scope (e.g. issues added to sprint, issues removed from sprint) are shown in the table below the graph.

Well done! You now know how to track the progress of your team's sprints! Next, let's wrap up your work!

After your team finishes the work in a sprint, you and your team can then perform a retrospective of the sprint. Sprint retrospectives help determine where the team succeeded, and where improvements can be made. The more sprints completed by your team, the more data you can use to find significant areas for improvement.

Lab 11 : End the sprint

Once your team reaches the end date of the sprint, you need to end the sprint — regardless if this means some issues in the sprint are not yet completed.

1. On the Teams in Space board, click **Active sprints**.
2. Select **Spring Cleanup** from the Active sprints drop-down.
3. Click **Complete Sprint**. The Complete Sprint dialog box will be displayed, showing the number of issues that are completed in the sprint, and the number of issues that were not completed.

Complete Sprint

Spring Cleanup

3 issues were Done.
1 incomplete issue will be returned to the top of the backlog.

Sub-tasks are not included in the total(s) above, and are always included in the same sprint as their parent issue.

Complete

Cancel

In this case, the incomplete issues are moved to the backlog. However, if you had more sprints planned, then they would be moved to the next planned sprint. Note, you can always add issues that are returned to the backlog to another sprint, if you wish.

Lab 12 : View the Sprint Report

After a sprint, your team can hold a sprint retrospective meeting, to determine your wins for the sprint, as well as point out the potential areas of improvement that your team can tackle in future sprints. You can use the Sprint Report during sprint retrospective meetings to do this.

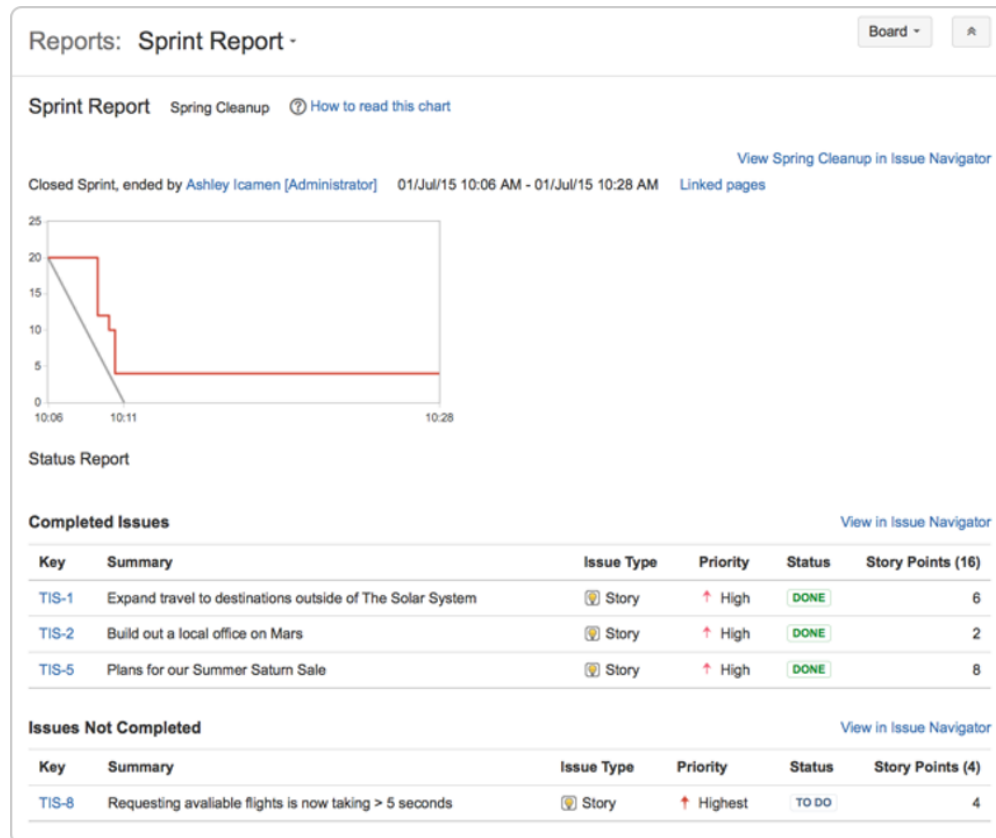
The Sprint Report shows a status list of issues in each sprint. It also provides a breakdown of the progress, status, and estimation information for each issue. You can also use the Sprint Report to perform progress checks in the middle of a sprint.

Looking for more reports?

There are many other reports available for your team to use in the Reports page.

You can also create your own reports, or use reports in the Atlassian Marketplace.

1. On the Teams in Space board, click **Reports**.
2. Select **Sprint Report** from the Reports drop-down.
3. Select **Sprint 1** from the Sprint Report drop-down.



Other tools you can use

Other useful tools can give your team a better visual of the improvement areas – and more importantly, these tools can help your team figure out action plans for these areas. For this lab, some of the tools you may want to use are:

- Velocity Chart
- Release Hub

Lab 13 : Velocity Chart

You can use the Velocity Chart to track the amount of work your team completes from sprint to sprint. Using the Velocity Chart lets you predict a more realistic amount of work that your team can commit in future sprints.

1. On the Teams in Space board, click **Reports**.
2. Select **Velocity Chart** from the Reports drop-down.



Release Hub

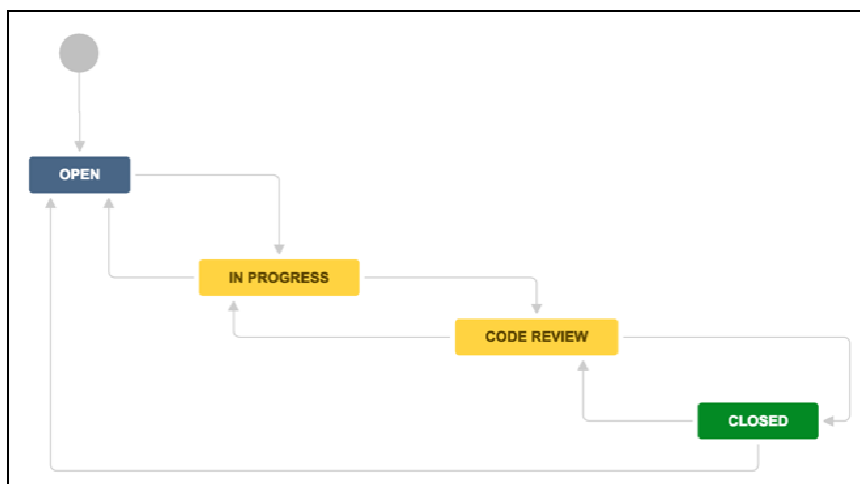
Aside from using reports (like the Sprint Report and Velocity Chart), you can also [monitor the progress of a version](#) after you complete a sprint. Monitoring a version's progress helps you see problems early, as well as determine the likelihood of releasing a version on time.

One more step to go! Let's look at some tips and tricks to help you get the most out of JIRA Software.

Want to become a JIRA Software ninja? Take a look at these advanced topics!


Workflows

A workflow is a set of *statuses* and *transitions* that an issue goes through during its lifecycle. JIRA Software makes it easy to create and edit workflows that fit your team's needs. Click [here](#) for more information on creating and editing workflows.



Development panel

JIRA Software can be connected to a range of development tools, such as repository managers, code review applications, and build and deployment managers. These tools can help you keep your project tracking in sync with your development work.

 Teams in Space / TIS-47

UI should allow users to book on large or personal space craft

EditCommentAssignMoreStop ProgressReady for Testing

Details

Type:Story

Priority:Major

Affects Version/s:1.5

Component/s:Web Site

Labels:None

Status:IN PROGRESS (View Workflow)

Resolution:Unresolved

Fix Version/s:3.0

People

Assignee:Ke

Reporter:Jer

Votes:0

Watchers:0 Sta

Description

Currently LocalTransportController makes an assumption that all the participants in the group are on the same itinerary. Many of our local travel providers limit reservations to 4 people.

The UI should walk the user through booking a large travel vendor for the group and allow certain users to opt out for a personal space craft.

Activity

AllCommentsWork LogHistoryActivity

There are no comments yet on this issue.

Comment

Dates

Created:25/May

Updated:Yesterd

Time Tracking

Estimated:

Remaining:

Logged:

Development

4 commits

1 pull requestMERGED

1 build

Deployed to QA

Notifications

Email other JIRA users a link to an issue either by sharing the issue with them, or by mentioning them in an issue's **Description** or **Comment** field.

The screenshot shows the 'Link to Issue' dialog box in JIRA. It has a 'Link to Issue' field with the URL 'http://localhost:8080/browse/TIS-47'. Below it is a 'User name or email' field with 'jennifer' entered. A dropdown menu shows 'Showing 1 of 1 matching users' with 'Jennifer Evans - jevans@veryr...' selected. There are 'Share' and 'Cancel' buttons at the bottom. The background shows the 'People' section of an issue with fields for Assignee, Reporter, Votes, Watchers, Dates (Created, Updated), and Time Tracking (Estimated, Remaining, Logged).

Custom cards

Customize issue cards for your boards to bring the right information to your team's attention at a glance. You can change the card colors to help people quickly identify cards on your board as being of a particular issue type, priority, assignee, or any JQL that you want. Click [here](#) for more information on custom cards.

The screenshot shows a JIRA Kanban board with three columns: 'To Do', 'In Progress', and 'Done'. The 'To Do' column has a card for 'TIS-8' with a red border, titled 'Requesting available flights is now taking > 5 seconds', dated '01/Jul/15 10:03 AM', and assigned to 'Kevin Campbell'. The 'In Progress' column has a card for 'TIS-3' with a blue border, titled 'Add support for teams larger than 20 people', dated '01/Jul/15 10:02 AM', and assigned to 'Jennifer Evans'. The 'Done' column is empty.

Confluence integration

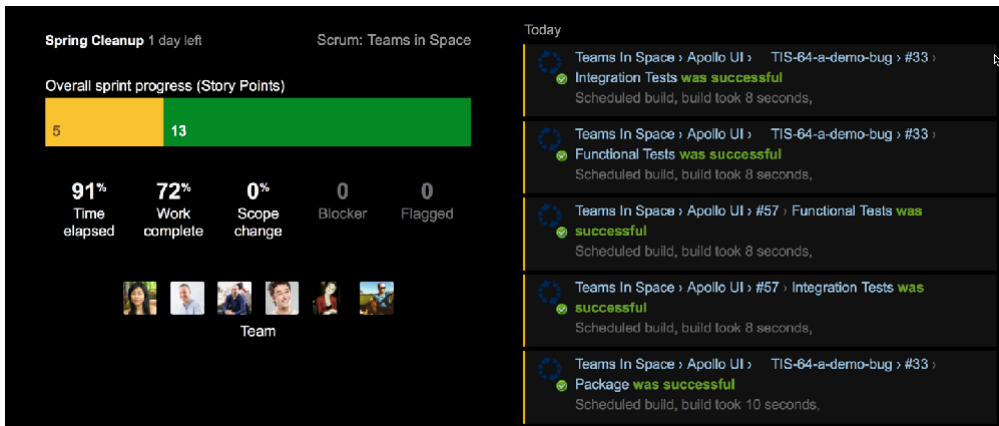
You can link your JIRA Software instance to a Confluence instance to create and link Confluence pages to your [sprints](#) and [epics](#). For example, you can link your sprint meeting notes in Confluence directly to the relevant sprint. This makes it easy for your team to quickly share information about the sprint. Click on [sprints](#) or [epics](#) for more information on linking your project to Confluence.

The screenshot shows a JIRA board with a 'Linked pages' dialog box open. The dialog box has a title 'Linked pages' and a list of pages with 'Teams in Space Home' selected. There are 'Link page' and 'Create page' buttons at the bottom. The background shows a board with a 'Sprint 2' section containing two issues: 'TIS-8' and 'TIS-3'.

Wallboards

A wallboard runs on a wall display and is used to monitor vital data about a project's progress. You can use wallboards to share information with your team during stand-ups. You can also add gadgets to a wallboard –

make sure to add gadgets that are useful to your team, such as a gadget that displays the status of a sprint, including issue status, time remaining, and build status.



What next?

So that's it — we hope this guide has helped you get a feel for JIRA Software. You can continue your training by completing the [Getting started as a JIRA Software user](#) lab, checking out [our documentation](#), or visiting our community on [Atlassian Answers](#) for more information.

[Previous](#)

Getting started as a JIRA Software user

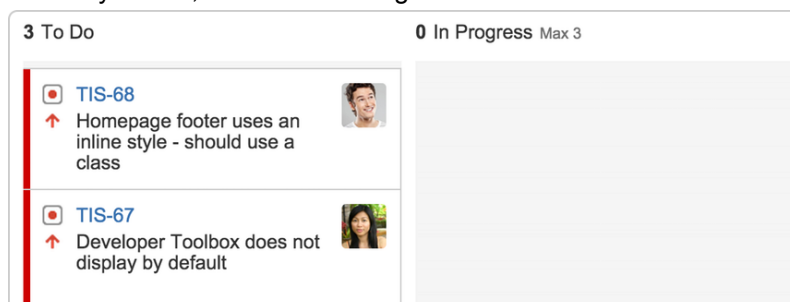
Welcome to your JIRA Software project! In this lab, we'll introduce you to your project and walk you through a simple workflow – from finding an issue assigned to you, to completing an issue. By the end of this lab, you will have completed the following steps as a JIRA Software user going through a simple workflow:

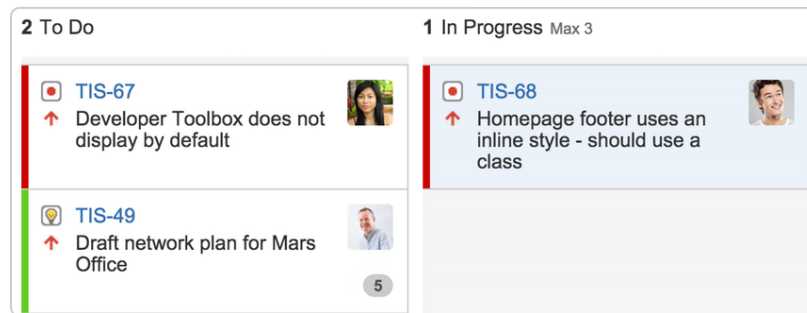
[Find your issue](#) | [Work on an issue](#) | [Collaborate with your team](#) | [Create a branch](#) | [Start coding](#) | [Send your work for review](#) | [Finish your work](#)

Lab 14 : Find your issue

Typically, your Development Manager will have already created a Scrum or Kanban project, and populated the backlog with issues. There are a number of ways to navigate around your project, but the easiest way to view information is by using boards. A board displays issues from one or more projects – it gives you a flexible way of viewing, managing, and reviewing your work. In this step, we describe how you would typically find an issue to work on, in the active sprint of the board of a Scrum project.

1. Select **Boards** from the top menu, and then select a board from the drop-down list.
2. Select **Active sprints** to see the issues in the sprint of the selected board.
3. Issues are represented by issue cards in the Active sprints of your board. The cards show quick reference information, like issue keys, assignees, and descriptions. Locate the top card in the **To Do** column and drag it to the **In Progress** column. By doing this, you are actually moving to assign the issue to yourself, and start working on it.





Lab 15 : Work on an issue

By moving an issue to In Progress, you are indicating that work has started on it. At this stage, you will need to assign it to yourself, so that everyone knows who is working on it. You may also want to add some technical details about the issue – it's helpful for your teammates to know as much information as possible about the issues you're working on.

1. In the **Active sprints** of your board, select the issue in the **In Progress** column. The issue is displayed in the Issue Detail view.
2. In the Issue Detail view, select **Edit** from the 'cog' drop-down. The 'Edit Issue' dialog will be displayed.
3. If the issue is not assigned to you yet, type your name in the **Assignee** field.
4. Add some technical details about the issue as necessary.
Note, you may want to specify a component or a version (if components and versions have been configured for your project), or add attachments as needed, etc.
5. When you're done adding technical details, click the **Update** button.

Lab 16 : Collaborate with your team

While you're working on an issue, you will probably need to share some information, clarify some requirements, or discuss some details about the issue with your team. This is easy to do with issue comments.

1. Select the issue to add a comment.
2. In the Issue Detail view, locate and click the **Comment** button.
3. In the **Comment** text box, type your comment.
4. To email other users about your comment, simply mention these users in the **Comment** text box (by typing @User's Name). An email will be sent to the users' email addresses that are registered with their JIRA accounts.
5. Click the **Add** button to save the comment.

Lab 17 : Create a branch

After assigning an issue to yourself and entering technical details about the issue, you're ready to do some coding. It is recommended that you create a branch when you start working on an issue. This way, you'll have your own stream of work that won't interfere with the rest of the team's work. It also ensures that your changes get reviewed before being merged back into the master branch.

JIRA Software can be linked to a code hosting or repository management source, like [Bitbucket Cloud](#) or [Bitbucket Server](#). This lets you open, collaborate, and manage source code directly from within JIRA. Integrating an application lets you and your team create a branch directly from an issue, giving you a faster workflow from selecting an issue to coding.

Before creating a branch, you should already know how to use branches in the source repository that your team is using ([Bitbucket Cloud instructions](#), [Bitbucket Server instructions](#)).

1. Select the issue for which you want to create a branch.
2. In the Issue Detail view, locate the **Development** panel, and click **Create branch**. The Create branch dialog will be displayed.
3. Include the issue key in the branch name. If you have workflow triggers configured, the status of the issue may automatically transition to 'In Progress'. See [Configuring workflow triggers](#) for more information.
4. Enter other details for the branch as necessary.
5. Click the **Create branch** button. Your branch will be created in your source repository.

Lab 18 : Start coding

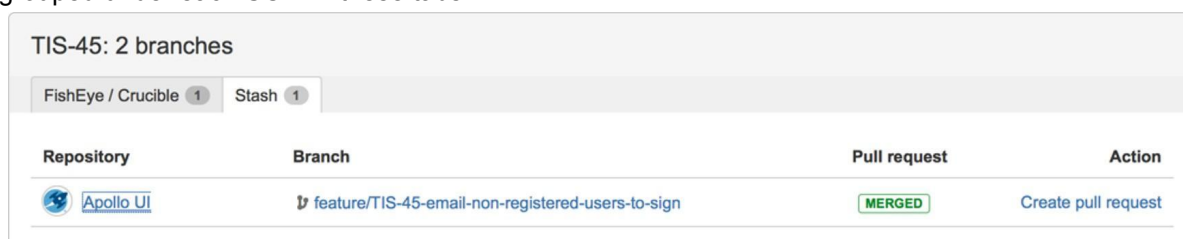
Now that you have created a new branch, you can start coding without worrying about your changes affecting the master branch. Ensure that you add the issue key that references the issue that you're working on. When you [reference your issue key\(s\) in your development work](#), your connected development tools will also have links back to the relevant issues.

1. Go to your source repository where your new branch is created, and access your new branch.
2. In your new branch, implement your code or fixes as necessary.

Lab 19 : Send your work for review

Use pull requests to tell your teammates about changes you've pushed to the repository. Once a pull request is sent, your team can review, discuss changes, or push follow-up commits.

1. Click **branches** in the Development panel to open a dialog in JIRA Software that shows linked branches in the Software Configuration Manager (SCM). If JIRA Software has been linked to more than one SCM, a tab will show for each SCM application (e.g. Bitbucket Server). The branches will be grouped under each SCM in these tabs.



2. Click a repository or branch to open the linked SCM at the relevant repository or branch.
3. Hover over a **Pull request** status (e.g. MERGED) to show a popup displaying a link to the pull request.
4. Click **Create pull request** to create a pull request for the branch (to merge it back to master).
5. Include the issue key in the title of the pull request. If you have workflow triggers configured, the status of the issue may automatically transition to 'In Review'. See [Configuring workflow triggers](#) for more information.

Lab 20 : Finish your work

After your work has been reviewed and merged with the master branch, the last step is to close or resolve the issue.

1. Locate your card in the **Active sprints** of your board.
2. Drag and drop the card to the **Done** column. If you have workflow triggers configured, the status of the issue may automatically transition to 'Closed'. You can skip the next steps. See [Configuring workflow triggers](#) for more information.
3. Select the **Resolution** for the issue from the drop-down menu.

Close Issue: TIS-20

① Closing an issue indicates that there is no more work to be done on it, and that it has been verified as complete.

Resolution * Fixed ⓘ

Fix Version/s 3.0 ×

Start typing to get a list of possible matches or press down to select.

Assignee Jennifer Evans

[Assign to me](#)

Time Spent (eg. 3w 4d 12h) ⓘ

Date Started 30/Jun/15 10:41 AM

Remaining Estimate

- ☒ Adjust automatically
- ☐ Leave estimate unset
- ☐ Set to (eg. 3w 4d 12h)
- ☐ Reduce by (eg. 3w 4d 12h)

Comment

Style B I U A A- Link Image Bulleted List Numbered List Quote Unquote

Close Issue Cancel

4. Provide details as necessary.
5. Select **Close Issue**.

Great job! You have successfully gone through a simple workflow using JIRA Software!

To do more awesome stuff, like managing your issues and customizing JIRA Software,

Lab 21 : Learn to plan and estimate for scrum teams

Estimating and planning for agile projects can be really challenging regardless of how experienced you are. There's always competing priorities, resourcing issues, business demands and time constraints.

This topic can't make any of those things go away, but if you're a newish to agile scrum or just new to JIRA Software, it will give you an overview of the JIRA Software features to help you plan and estimate.

Learning objectives

The aim is to give you a good introduction to:

- Planning concepts and functions
- Estimating methodologies
- Analysis tools and techniques
- Optimization approaches

Learning activities

There's optional activities outlined at the bottom of each topic. You don't have to do these, but it does help to put the theory into practice. You might want to create a dummy project to do this.

Pre-requisites

You need:

- Access to JIRA Software are. If you don't have a working copy, trial it [here](#).
- The ability to create projects and configure boards.

Learn more about agile

It helps if you're familiar with the agile language and concepts we'll use. So if this is a bit new to you or you feel a bit rusty, check out the [Agile Coach](#), which covers lots of the basics.

If you are working with an existing instance of JIRA software, you

need this access to do some activities. If you [create a dummy project](#), you will get this access by default.

They say the best laid plans always go awry, and sometimes they do, but you still have to make them. At least in JIRA Software, the plans you make are highly flexible, customizable, and configurable.

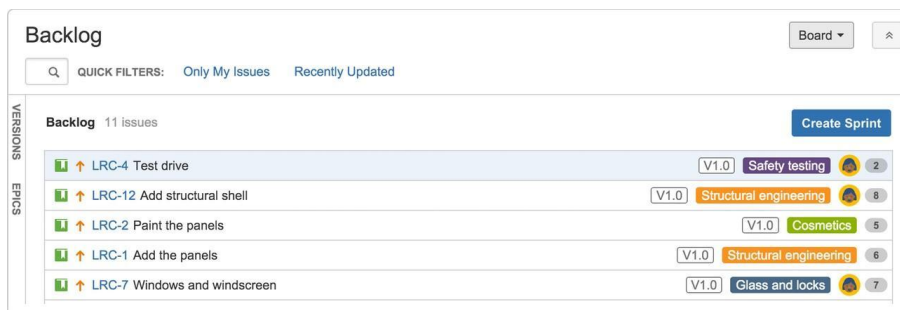
Start with a backlog

All agile projects start with a [backlog](#) of issues that represent the work that needs to be done: user stories, tasks, bug fixes, etc. Anything that's been determined is a discrete piece of work.

- Backlogs are all about priorities and doing the most important things first
- Use quick filters to just see the issues you want
- Create new issues and start sprints in the backlog

To get to your backlog, open your board and click **Backlog**. Control board details shown using the options in the Board menu.

You can add issues quickly by clicking **+ Create issue** below the backlog list.



"You're just setting yourself up for failure if you try to plan sprints too far ahead. You'll waste energy making decisions about things that will almost certainly change scope or priority in a few weeks. The backlog is always waiting to be re-prioritized"

~ Atlassian Team Lead

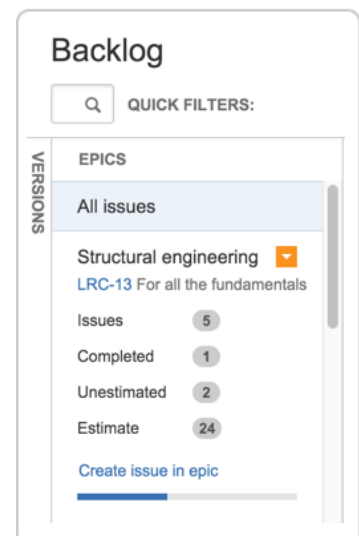
Lab 22 : Use epics for context

Epics are designed to tell the bigger story. They can be used to capture broader work themes, such as 'performance' or 'user interface', or to track larger items of work. There are advantages to using [epics](#).

- Epics can span multiple projects, which is especially useful if your board is set up to show issues from multiple projects.
- You can link an epic to a Confluence page to provide more information, such as design specs, business case, requirements, etc.

To create an epic

1. In your board backlog, click **EPICS** on the left side of the board, then click **Create epic**.
2. Complete the details of the epic in the window that appears and click **Create**.
3. Drag and drop backlog issues across to the epic they belong to.



"Versions work best for our long term planning because they align to our service-level agreements. Our customers expect weekly releases."

~ Release Manager

Versions keep the release in order

Lots of teams release **versions** of their product as updates or feature sets for customers. Allocating issues to versions can help the team plan sprint work in the lead up to a release.

Lab 23 : To create a version

1. Click **VERSIONS** on the left side of the board, then click **Create version**.
2. Complete the details and click **Create**.

You can then drag and drop backlog issues across to the version they belong to.

Create Version

Project

Little Red Car

Name

Prototype v1.0

Description

Pre-release engineering test

Start Date

Release Date

Create

Cancel

See [Configuring versions in a Scrum project](#) for more information.

Rank and prioritize

When your backlog issues are listed in order of priority, it's far easier to create sprints that are based on the work that needs to be done next.

Ranking is easily done by dragging and dropping issues in the backlog.

"Ranking helps us knock over sprint planning really quickly because we can all see what has to get done next."

~ Atlassian Dev Lead

Backlog

QUICK FILTERS: Only My Issues Recently Updated

Backlog 11 issues

Create Sprint

LRC-4 Test drive

V1.0 Safety testing 2

LRC-12 Add structural shell

V1.0 Structural engineering 8

LRC-2 Paint the panels

V1.0 Cosmetics 5

LRC-1 Add the panels

V1.0 Structural engineering 6

LRC-7 Windows and windscreen

V1.0 Glass and locks 7

LRC-8 Fit the Upholstery

V1.0 Cosmetics

Rinse and repeat

The best thing about scrum projects is that as you progress through sprints, the more clearly you can see what your team is capable of.

There's a heap of reports and charts that you can consult to help with understanding your team's productivity rates. We'll talk about how some of these can help you get better at planning a bit further along.

LEARNING ACTIVITY

Create a dummy project and choose **Scrum software development** as the project type.

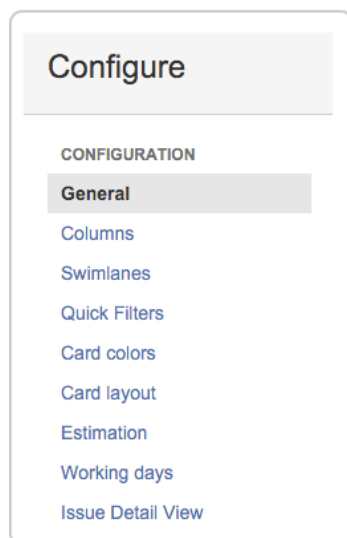
- Create a backlog of issues. You just need a title to add new ones.
- Re-prioritize the backlog rank by dragging and dropping
- Create an epic and add a few issues to it
- Create a version and add a few issues to it

Lab 24 : Customize the team board

1. [Plan for the team](#)
2. Customize the team board
3. Estimate in story points
4. Analyze team reports
5. Optimize future plans

The [board](#) is where all the action happens so you want to make sure that it's set up the way your team needs. You can change a heap of settings in the [board configuration](#) screens. Here's just a few.

⚠ You need to have special Board Admin rights to change board configurations. If you created the board, you already have them.



Board customization

Control everything from the board configuration page. In your board, go to **Board > Configure**.

Columns

Add statuses and change the workflow to suit your team structure. For instance if you want to show testing progress separate to dev progress, add a column. You can also switch on an indicator to help identify slow moving issues.

Card layout

Show up to three extra fields on the card layout. Say, if you want to show labels or components.

Card colors

If your team is 'visual' you can color code cards according to issue type, assignee, priorities, or queries

Quick filters

Set up [quick filters](#) so you can just click once to show specific issues. You can set up a filter for issue types, assignees, unassigned, etc. Pretty much anything you need.

Go to Board > Configure > Quick Filters.

Filters work based on JQL (JIRA Query language) which are really simple search statements. The result is something like:

✓ TIP: as you start typing in the JQL field, the syntax helper pops up to assist

See [Configuring a board](#) to learn about the other things you might need.

"We just couldn't work as a global team without quick filters. The guys in San Fran and Gdansk need to be able to see exactly the same set of issues as us when we're discussing work online."

~ Dev Manager

Quick Filters

Quick Filters can be used to further filter the issues in the board based on the additional JQL query.

Name	JQL	Description
Engineering	✓ component = Engineering	All issues related to vehicle engineering

Add

▼ **TIP: Create filters from saved searches**

If you have a giant backlog, you probably use the [issue search](#) pretty regularly to find the issues you want. And if you're repeating searches using similar criteria, then other people probably are too.

Next time you are searching for a set of issues you know you will need again, save your search as a filter. Once you save a filter, you can share it, favorite it, get emails of search results, use it with a dashboard, and heaps more.

1. Go to **Issues > Search for issues**.
2. Enter your search filters and click **Save As**.

Estimating LEARNING ACTIVITY

- Create a new quick filter and use it on your board.
- Search for an issue and then save the search as a favorite.

Lab 25 : Estimating

There's a huge variety of ways to estimate stories, but the objective of every approach is to be able to get better at predicting how much work the team can do each [sprint](#). In agile scrum, this translates to knowing the team's [velocity](#).

Velocity measures the number of 'estimation units' that a team usually completes from sprint to sprint. It is effectively a productivity rate based on an estimation of volume of work, and it is best worked out in a measure other than 'time'.

▼ **What are story points?**

Story points are a commonly used measure for estimating the size of an issue in scrum teams. During a typical planning session, a trivial bug fix might be estimated as a 1 or 2, and a larger feature might be anything up to a 12. Note that the scale of points does vary between scrum teams. Some use 1-12, others 1-5. The key is to use the same scale so that your velocity is consistently calculated.

If issues are estimated larger than this, they might need to be broken into smaller stories or subtasks.

The focus is on story points here because it's the more common scrum estimation method, and we use it at Atlassian. Your company may use hours or ideal days. If you do use time for estimation, you might want to have a look at [Configuring estimation and tracking](#).

"Even seasoned devs can't estimate accurately in hours/days. We just need to predict what can realistically get done with a reasonable degree of certainty. Story points all the way."

~ Atlassian Developer

▼ Q: Why use story points instead of hours?

A: It is much more useful in the long term to roughly predict how much work your team can do each sprint, than it is to try to estimate how long each story will take in time.

Story points enable the team to estimate stories in comparison to other stories, instead of forcing them to determine the time it will take to complete each story. Velocity is then worked out based on how many points the team can complete in each sprint. After a few sprints, team velocity stabilizes and estimation accuracy increases (or rather the same degree of inaccuracy is applied).

The problem with using hours to estimate is that the team is likely to estimate using the 'working hours' of the sprint (with some buffer time) and this is almost never an accurate reflection of a story's complexity or size. Accurate velocity is difficult to establish when stories are estimated in isolation, instead of by comparison.

Estimation, time and velocity are really critical things to understand. Read more in the [Estimating an issue](#) topic.

Estimate in points, track in time

Even when you estimate in story points, you can still track in time if you want. Knowing the team velocity (regardless of the measure used) enables you to roughly guess how long estimated backlog items will take to complete.

But JIRA Software does also have a couple of dedicated fields (**Remaining Estimate** and **Time Spent**) to track time while using story points.

⚠ NOTE: You need to be a board admin to make any configuration changes to a board. If you created the board, then you're already the admin.

Go to Board > Configure > Estimation.

Estimation

Issues can be estimated when in the Backlog to get an idea of how much work is being committed to in a sprint. [Read more about estimation and tracking.](#)

Estimation Statistic

Estimate issues in the Backlog by entering values for **Story Points**. Your velocity from sprint to sprint will be measured against these estimates.

Time Tracking ☒ None

Issues will burn down their **Story Points** value upon completion.

☐ Remaining Estimate and Time Spent

Track time against issues using JIRA's **Remaining Estimate** and **Time Spent** fields.

- Select the **Estimation Statistic** (unit of estimation) - choose from story points, original time estimate, and issue count.
- Switch on the **Remaining Estimate** and **Time Spent** option to get a more accurate picture of how things are tracking in time units.
- If you leave **Time Tracking** as **None**, you can still refer to reports such as the Burndown chart to monitor progress.

▼ Fun fact: Inaccuracy is good

The goal of velocity is to be able to look at a backlog of not particularly well-understood stories, and guess how many sprints it will take to complete. It requires a similar level of uncertainty for all of the estimates in the backlog. Determining accurate velocity relies on the equality of the inaccuracy, which is why you should NEVER re-estimate issues after a sprint has started.

LEARNING ACTIVITY

In your dummy project, go to **Board > Configure > Estimate** page and change the **Estimation statistic** to one of the following:

- Story points
- Original time
- Issue count

Note the differences when you create and estimate an issue.

Lab 26 : Analyze team reports

1. Plan for the team
2. Customize the team board
3. Estimate in story points
4. Analyze team reports
5. Optimize future plans

JIRA Software comes with a whole lot of [reports](#) that will help everyone in the team understand where things are at. Here are just a few.

Access them all from the left navigation pane.

Velocity for the team

Teams want to know what their velocity is. They want to know how accurately they are estimating and how much they are getting done each sprint, in order to estimate better in future.

Velocity Chart

- See if the team is over-committing or under-committing stories in sprint planning
- Check for outliers and drill into a sprint to learn from past mistakes (e.g. a grossly underestimated story)
- Monitor how velocity alters during periods of team change or growth

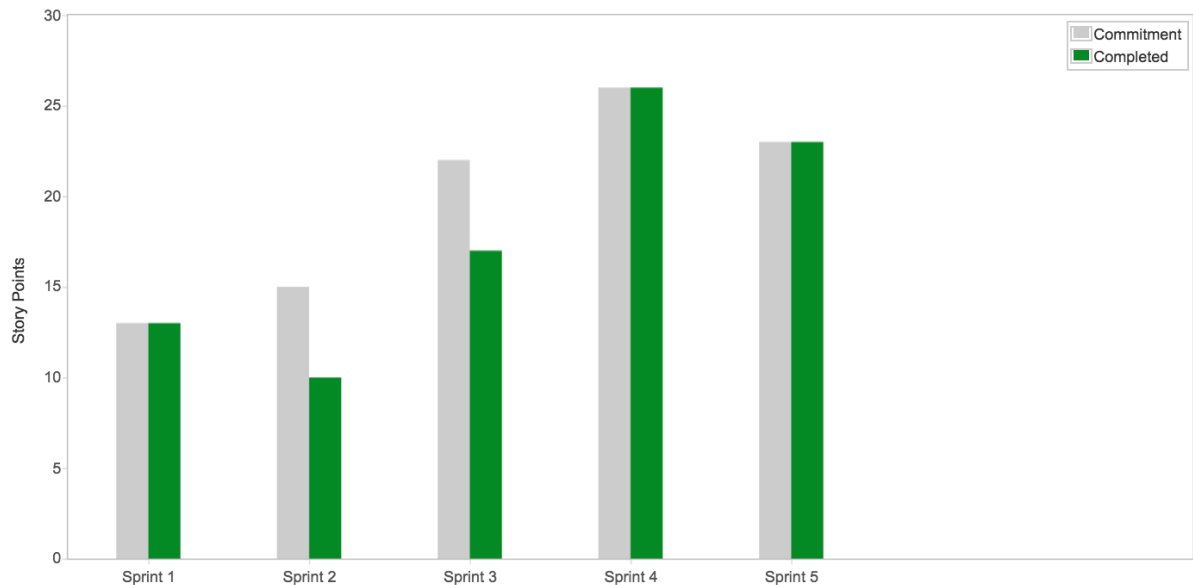


? How to read this chart

Track the amount of work completed from sprint to sprint. This helps you determine your team's velocity and estimate the work your team can realistically achieve in future sprints.

[Hide this information](#)

Velocity Chart



Sprint	Commitment	Completed
Sprint 1	13	13
Sprint 2	15	10
Sprint 3	22	17
Sprint 4	26	26
Sprint 5	23	23

Burndown for the leads

Team leads (Dev leads, Tech leads, etc.) usually spend half their time thinking ahead to future work, and the other half firmly involved in the current sprint. They need to know how things are going now to help steer the team to a successful sprint completion.

"Burndown is the fastest way for me to see what's going on and help out the team if things are looking a bit shaky."

~ Atlassian
team lead

Burndown Chart

- If the **Remaining Value** line is headed straight across and not down, you need to find out why
- A successful sprint is where the **Remaining Values** line hits the bottom on or before the **Guideline**
- If the **Remaining Value** line ends before the guideline, you may have a bunch of overachievers or under-committers on your hands!

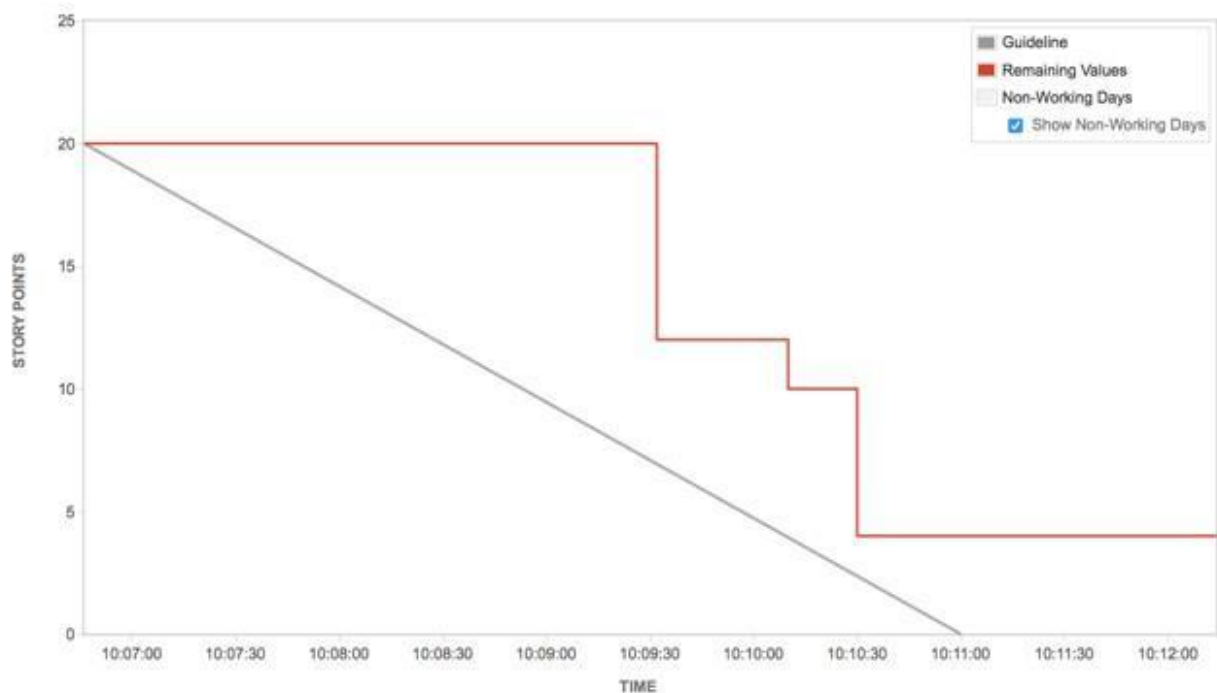
⌵ [Show Burndown Chart](#)

Burndown Chart

Spring Cleanup

Story Points ▾

? How to read this chart



Date	Issue	Event Type	Event Detail	Story Points		
				Inc.	Dec.	Remaining
01/Jul/15 10:06 AM	TIS-1	Sprint start		6		20
	TIS-2			2		
	TIS-5			8		
	TIS-8			4		
01/Jul/15 10:09 AM	TIS-5	Burndown	Issue completed		8	12
01/Jul/15 10:10 AM	TIS-2	Burndown	Issue completed		2	10
	TIS-1	Burndown	Issue completed		6	4

Epics for product managers

Product managers (or project managers, depending on your structure) like to keep an eye on the big picture stuff, so they can check if project milestones are being hit, where dev effort is being spent, and if the backlog is in need of a re-prioritization. The Epic Report is a good read for satisfying some of these demands.

Epic Report

- Easily see the how near or far an epic is from completion. Check with the team if things look unbalanced.
- Monitor unestimated issues. Chat to the dev lead to see if anything needs further clarification or breakdown.

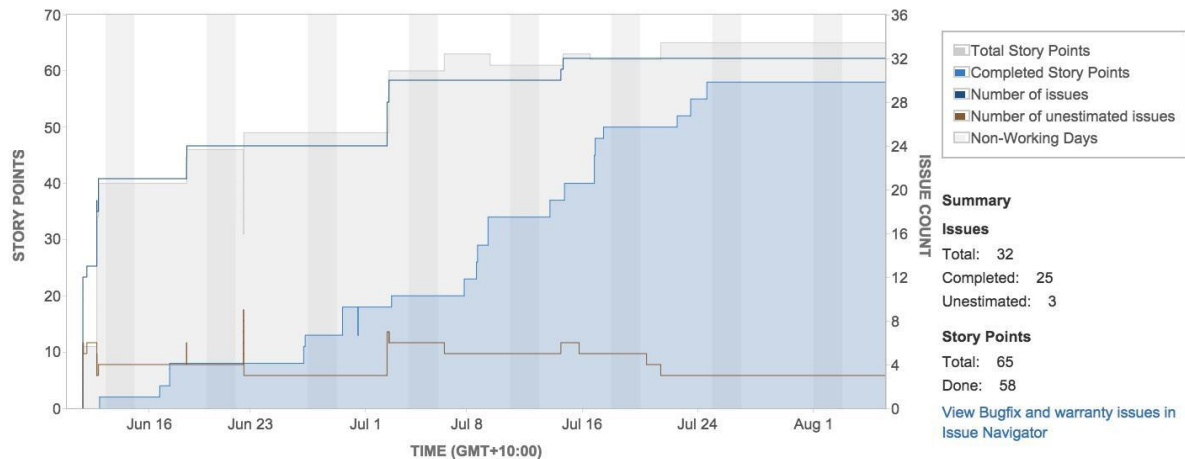
How to read this chart

Understand the progress towards completing an epic over time. This helps you manage your team's progress by tracking the remaining incomplete/unestimated work.

[Hide this information](#)

Bugfix and warranty issues

TIS-100 Bugfix and warranty issues [Linked pages](#)



Status Report

Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (58)
TIS-81	Afterburner initial design	Bug	Critical	RESOLVED	-
TIS-82	Afterburner initial prototype	Bug	Minor	RESOLVED	-

LEARNING ACTIVITY

You need to have historical sprint data in your project to view the above reports. If you don't have access to an existing project, just browse the report help topics:

- [Sprint Report](#)
- [Burndown Chart](#)
- [Epic Report](#)

Lab 27 : Optimize future plans

1. Plan for the team
2. Customize the team board
3. Estimate in story points
4. Analyze team reports
5. Optimize future plans

When things are running well, velocity is stable, the team is feeling challenged and even the product manager is happy, it is not time to stop planning.

Plan ahead

Everything learned from burndowns, reports, story estimation, sprints and standups is ready and waiting to help make planning better for future sprints and issues. Even if the team changes, the budget is cut, or the company you work for insists on a whole new focus, the data collected is still really valuable.

Here's the top things you might change next time, based on what you learn this time:

▼ [Add more story points for high performing teams](#)

A high performing team that always completes a sprint with no work to spare may need to be challenged with more points in the next sprint. But be careful of tipping the scale too far in the other direction. Teams that rarely complete a sprint without leftover points may feel like they are underperforming, even if they aren't.

▼ [Change things up to get different results](#)

When things are going well, you don't want to mess with the formula. But when things aren't going as well, it can help to change things around a bit (if you have the luxury to do so). This could take the form of swapping responsibilities for a sprint, or introducing a different kind of sprint, such as an innovation sprint or bugfix only sprint. Change can be re-invigorating, and it can still contain work!

▼ [If overcommitment persists, please see your PM](#)

Failure to meet committed story points can be disappointing for a team, especially if they don't know why. Check in with the PM or person who reported the issue to clarify details during estimation, and raise concerns during retros. If estimates are way off due to lack of basic information, for example, this needs to be addressed.

"One of the best things about estimating as a team is that it sparks some really important discussions about how we build our product and how we can get better at what we do."

~ Atlassian
Product
Manager

Don't forget the invisible things

Scrum is all about team and a lot of things happen during sprints that need to be captured and fed into future plans. Here's a few things that new starters should know and experienced operators sometimes forget.

- **Always hold a retro. Always.**
Retrospectives can give you insights that reports can't. Make sure you discuss any issues that come up, which might result in adjustments to planning. Record insights (preferably in Confluence) and brag loudly about the good stuff.
- **Never assume the team knows how they are performing**
Just because velocity reports are available and are viewed in monthly team meetings, doesn't mean the team knows how they are going. Scrum leads and team leads need to use words too. Who doesn't like being told in person they are going well.
- **Say it with meaning even if you've said it before**
Standups and planning can become a bit routine over time, especially with experienced teams who have worked together for ages. Try not to let the rituals become lip service, make sure people are saying meaningful things and are listening to each other actively.