

# GUIDELINES FOR **WORKING IN JIRA**

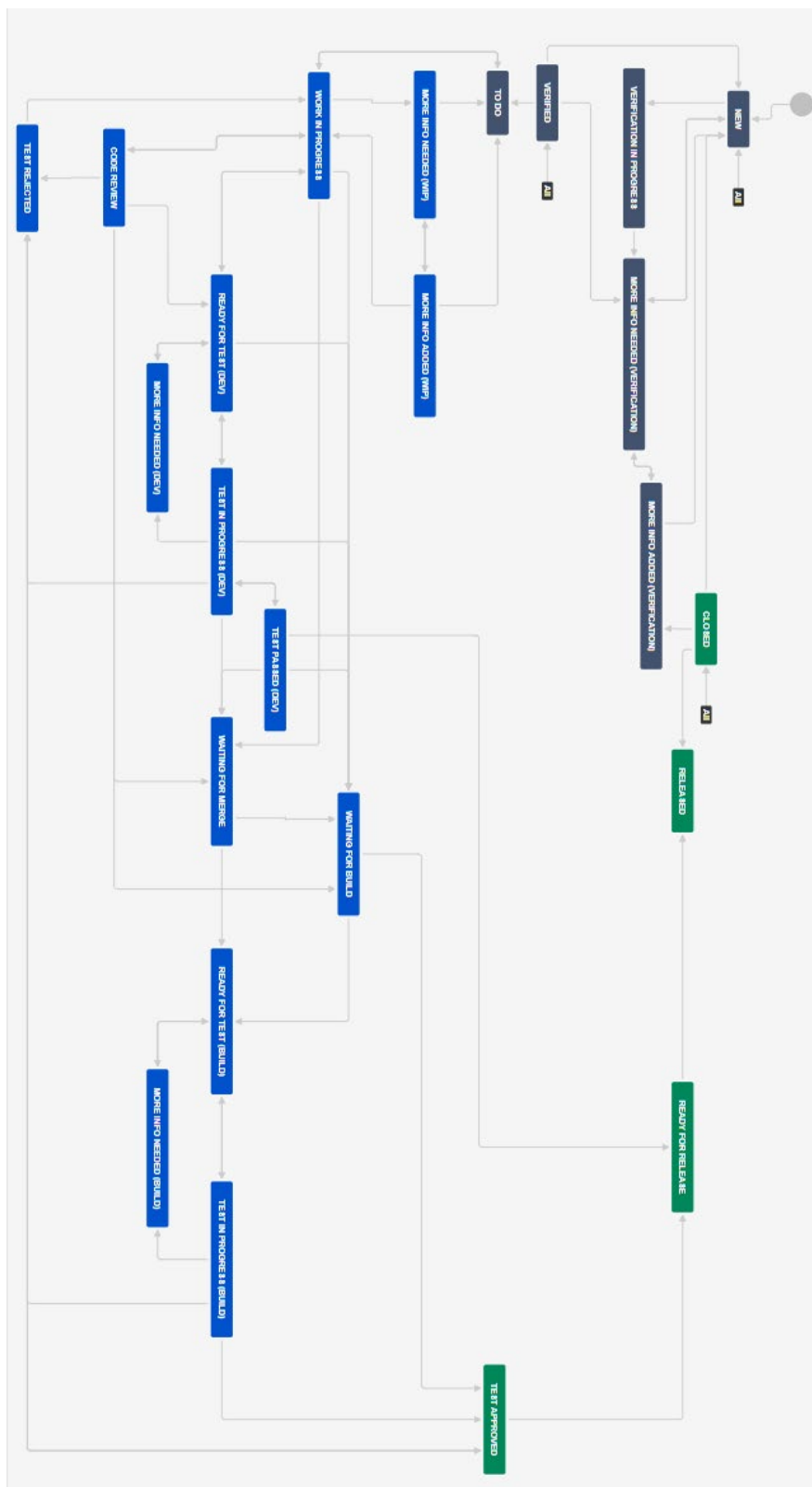
## Contents

Contents .....	1
Working in Jira .....	2
Issue Workflow .....	4
Recommendations.....	6
Configura Bot.....	6
Working with Issues Escalated from Jira Service Desk .....	7
Issue / Ticket Status Explanation .....	10
Epic Status Explanation .....	12
Issue / Ticket Resolution Explanation.....	13
Dashboards .....	14
Backlog View.....	17
Board (Kanban) / Active Sprint (Scrum) View.....	18
Scrum vs Kanban.....	18
Changing Issue Status in the Board/Active Sprint View .....	19
Releases View .....	20
Reports View .....	22
Issues View .....	23
Creating a new Issue.....	26
Moving an Issue.....	27
Editing an Issue.....	28
Searching for Issues .....	29
Components View .....	30
Submitting issues from CET Designer directly to partner's project .....	30
Appendix 1: Global Project Automation Rules .....	32
Jira Software Project.....	32
Jira Service Desk .....	34
Appendix 2: Jira Software Keyboard Shortcuts .....	35

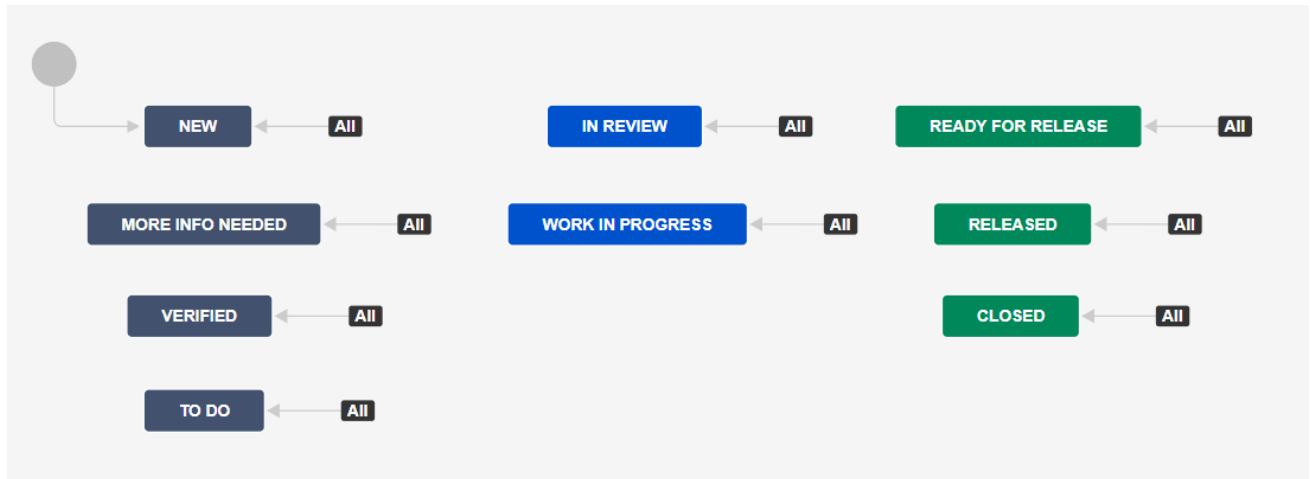
## Working in Jira

Below is a graphical overview of the workflow for working with issues in Jira. An up-to-date version of this can always be accessed by clicking the **View workflow** link next to the issue status in Jira.

Graphical overview of all issue types except Epic:



Graphical overview of Epics:



## Issue Workflow

1. *New* issues are created.
2. Set the issues to *Verified* if you have all the necessary info.
3. Product Owner to go through the new issues and prioritize the issues in the backlog view.
4. Scrum master or Product owner can group the issues under the same epic depending on the product line or project phase. Epic status can be used to indicate different project stages. (optional)
5. Scrum master or Product owner to move the issues to different sprint versions. They can either drag issues from the backlog to a sprint OR right-click on issues in the backlog and choose to send to different sprint versions.
  - Create different quick filters in the backlog (for example for different priorities, newly created issues etc.)
  - Create special “pre-sprint” backlog for higher priority issues.
  - Set the epics.
  - Note: a sub-task is always assigned to the same sprint as its parent task.
6. Developers or Scrum master to estimate the issues. Use either story point or original time estimates. Set issues to *To Do*. Issues can be estimated in the upcoming sprint and/or the special backlog. Note: developers should log their work in terms of time spent and remaining time, once they begin working.
7. When a developer or QA or product owner needs more information to verify an issue, resume work or continue testing, add a comment and set the issues to *More info needed (Verification)*, *More info needed (Work in progress)*, *More info needed (Dev)*, or *More info needed (Build)*. Make sure the assignee is set correctly.
8. An assignee of a *More info needed* issue can answer a question by adding a comment and set the issue to *More info added (Verification)*, *More info added (Work in progress)* or *Ready for Test (Dev)* or *Ready for Test (Build)*. Make sure the assignee is set correctly.
9. Developers use the scrum board to work in the sprint. Please note that moving a card up or down in a column will change its rank.
10. Developers can either drag cards from one column to another to update the issue status or use the Transition button on the View Issue screen. When dragging a card to other columns, built-in rules will provide visual feedback on where the card may be placed.
11. The *Work in Progress* status should be used when a developer starts working on the issue.
12. Before setting the status to *Ready for Test (Dev)* or *Ready for Test (Build)*, please fill out the following:
  - Description: any relevant background information and test instructions.
  - Fixed in version: feature branch or Extension version.

13. File issues during testing:
  - If the issue is straight-forward/one test case, test reject without filing any new issue.
  - If the issues have more test cases/test scenarios, file issue as separate issues and link those issues to the parent test ticket.
14. Add test result and update the status (*Test rejected* or *Waiting for build*). Test result should show the failed test cases and special notes for developer. You can add the issue number in the test result where it will become a hyperlink to that issue.
15. Sprint review and Retrospective. The Scrum master can use the reporting tool in JIRA to track and analyze the team's work throughout a project.
16. Sprint Retrospective. Create the Retrospective document in your own SharePoint.
17. After QA set an issue to *Waiting for Build*, the developer can set it to *Ready for Test (Build)* for the customer to test in the build. (optional)
18. Developer can set *Test Approved* issued to *Ready for Release* if all testing is done.
19. QA: s to run regression test and/or smoke test.
20. When the build is released, set the Release version to Released. All issues that are in that version and have status *Ready for release* will be transitioned to *Released*.
  - If you transition an issue from *Ready for release* to *Released* (either manually or by using the above feature), it will send an email to the reporter about it.
21. QA to update the regression test cases with the new changes.

*For more information on how to set up boards, please see*  
[https://downloads.configura.com/marketing/resources/Documents\\_Jira.pdf](https://downloads.configura.com/marketing/resources/Documents_Jira.pdf)

## Recommendations

- Work in the Board/Active Sprint and Backlog views. They are highly visual and provide great visual support on what is allowed, how things may move etc.
- Jira has very powerful and flexible filters that are publicly shareable and work across multiple projects.
- Use Quick Filters.
- Create a special “pre-sprint” backlog for higher priority tickets.
- Set the epics.
- Use the Bulk Change to edit multiple issues simultaneously.
- Use mentions (@name) to ask a question or share information with a specific user within Jira.

## Configura Bot

The Configura Bot will automatically perform certain actions following an action or a status transition. Should it not be setup correctly, however, you can edit the field manually.

For example: When an issue is transitioned to *Work in Progress*, the assignee will automatically be set to *Developer* if the developer field is set. If the developer field is empty, the assignee will automatically be set to current user. For more information, see Appendix 1.

## Working with Issues Escalated from Jira Service Desk

If your Jira Software Project includes issues that are escalated from Jira Service Desk, there are some things that are good to know. If your organization does not have that connection, you may skip this section.

When an issue is escalated from a Service Desk project, a linked issue is created in the Software project. Summary, Description, Troubleshooting steps, Additional and Attachments are copied to the linked issue. In Troubleshooting steps, the support agent can choose to specify which steps have been taken by support before being escalated. The Troubleshooting steps are displayed as a Comment in the Activity section.

Comments are *not* copied, but the developer can use the link to go to the Service Desk issue and view the comments:

FD-52022 [Return to queue](#)

### Test Emil

[Edit](#) [Comment](#) [Log work](#) [More](#) [Email](#) [Share](#) [Export](#)

- Details
- Description
- Attachments
- Issue Links
- Activity

People

Dates

Agile

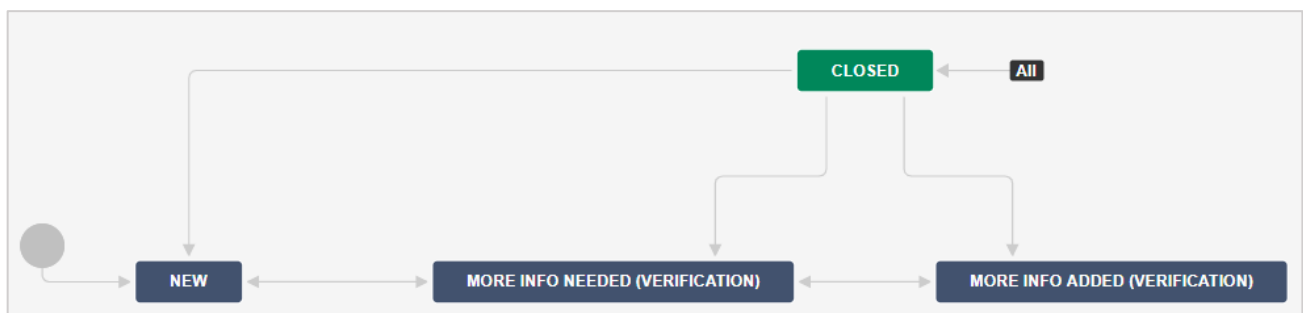
Issue Links

escalated from

FS-79 Test Emil [ESCALATED](#)

[Comment](#)

Below is a graphical overview of the workflow for **User Bug Reports** and **User Feature Requests**:





If the developer wishes to share information or ask a question, they can use @support in a comment. The comment is then added to the Service Desk issue and the issue status changes to *Waiting for support*. The status of the Software issue will automatically be changed to *More info needed*.


Vice versa, support agents can use @developer to add a comment to the Software issue. The Configura Bot will update the Jira Software issue status automatically depending on the previous issue status. For more information, see Appendix 1.

The developer can request additional files, for example log files or drawings from the Customer/Reporter through the support agent. When the Customer/Reporter provides additional files, they become available only on the Service Desk issue. The support agent can then share a comment with the developer and write "See linked issue for additional files".

A developer can access the Service Desk issue without a Jira Service Desk license, but they cannot edit the ticket except make internal comments or attach files.

When a Service desk issue is resolved and closed, there will be an automatic comment added to the Software project that the Service Desk issue has been resolved:

---

▼  Configura Bot added a comment - Just now

Support issue got resolution:  
Done

---

Please note that User Bug Reports and User Feature Requests always come in through support, and it is the responsibility of the Product Owner to review them, decide if action should be taken, and, in that case, convert them to regular issues. For more information, see the *Issue Type* and *Moving an issue* sections.

When a Software issue with a Service Desk link is set to *Released* by the developer, an automatic comment is shared with the Customer/Reporter.

Edit Comment Assign More Release Closed

Details

Type: Task

Priority: High

Affects Version/s: None

Labels: None

Status: READY FOR RELEASE (View Workflow)

Resolution: Done

Fix Version/s: Pilot

When an entire version is set to *Released* in Software, the status of each individual issue is also set to *Released*, meaning that each of the linked Service Desk issues will share automatic comments with the Customer/Reporter the same way as described above.

Releases

Manage Versions

QUICK FILTERS: Released Unreleased Archived

Merge versions

Version name
  Start date (optional)
  Release date (optional)
  Description (optional)
 Add

Version	Status	Progress	Start date	Release date	Description	Actions
Phase 2	UNRELEASED	No issues	2019-10-07	2020-04-30		...
Phase 1	UNRELEASED	No issues	2019-06-03	2020-01-01		...
Pilot	UNRELEASED	<div></div>	2019-01-21	2019-08-15		...

Atlassian Jira Project Management Software (v6.1.0#801000-sha1:2e1cd1b) · About Jira · Report a problem

ATlassian

Release  
 Build and Release  
 Archive  
 Delete  
 Edit

## Issue / Ticket Status Explanation

STATUS	Notes:
NEW	All newly created tickets should be set to new.
VERIFICATION IN PROGRESS	Person in charge is verifying an issue.
VERIFIED	Person in charge is responsible for checking and setting the tickets to verified, which can be done when they have all the necessary info.
MORE INFO NEEDED (VERIFICATION)	Person in charge needs more info before setting the status to verified.
MORE INFO ADDED (VERIFICATION)	More info added back to the questioner so that the ticket can be verified.
TO DO	Developer has estimated the ticket.
WORK IN PROGRESS	Developer is working on the ticket (fixing or implementing new features).
MORE INFO NEEDED (WORK IN PROGRESS)	Developer needs more info on how to fix the issue or implement the changes.
MORE INFO ADDED (WORK IN PROGRESS)	More info added for the questioner so that the work can resume.
CODE REVIEW	Code review for developer.
READY FOR TEST (DEV)	Ready for testing in developer version.
TEST IN PROGRESS (DEV)	QA is testing the ticket in developer version.
MORE INFO NEEDED (DEV)	QA needs more information from developers to proceed with testing in developer version.
TEST PASSED (DEV)	Testing is approved in developer version. Merging is not needed for now.
WAITING FOR MERGE	Testing is approved in developer version, and it is ready for merge.
WAITING FOR BUILD	Merge request is merged and ready for build.
READY FOR TEST (BUILD)	Ready for testing in build version.
TEST IN PROGRESS (BUILD)	QA is testing the ticket in build version.
MORE INFO NEEDED (BUILD)	QA needs more information from developers to proceed with testing in build version.
TEST REJECTED	The ticket has not passed testing and is rejected.
TEST APPROVED	The ticket has passed testing and is approved.
READY FOR RELEASE	Fixes are added in the build, but the build is not yet released.
RELEASED	The fixes or new features are released.

CLOSED

Closed the issue with different resolution.

## Epic Status Explanation

STATUS	Notes:
NEW	All newly created epics should be set to new.
VERIFIED	Person in charge is responsible for checking and setting the epics to verified, which can be done when they have all the basic information to proceed to planning stage. High level of estimation can be done during this stage (optional).
MORE INFO NEEDED	Person in charge needs more information before setting the epic status to verified.
IN REVIEW	This is a planning stage. Pre-study, adding more information, estimation and prioritization should take place at this stage.
TO DO	Developer has estimated the ticket.
WORK IN PROGRESS	Developer is working on the issues of this epic (fixing or implementing new features).
READY FOR RELEASE	All issues of this epic are added in the build, but the build is not yet released.
RELEASED	All issues of this epic are released.
CLOSED	Closed the epic with different resolution.

## Issue / Ticket Resolution Explanation

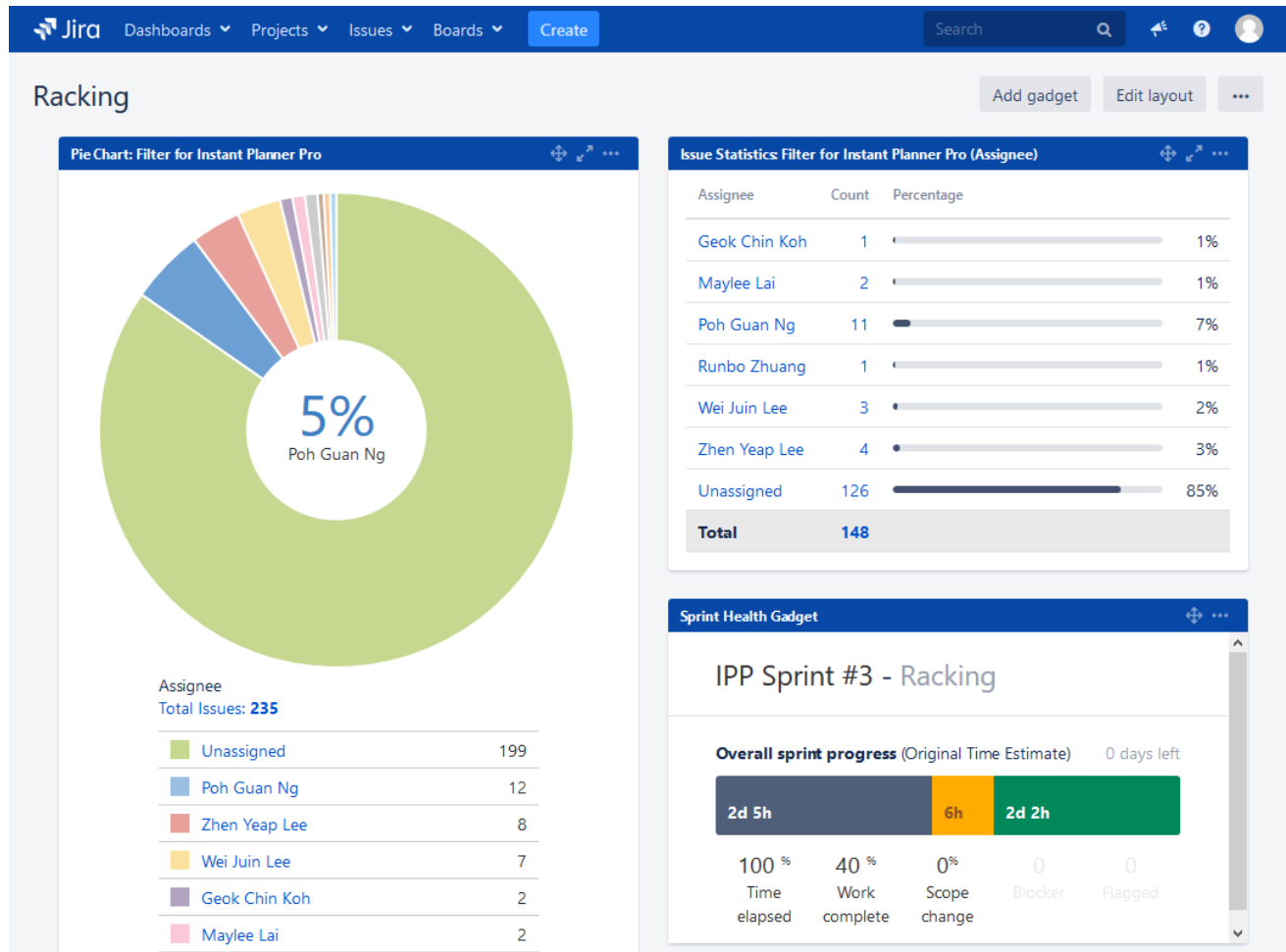
These are the different resolutions that can be used when closing an issue.

RESOLUTION	Notes:
Done	Work has been completed on this issue.
Duplicate	The problem is a duplicate of an existing issue (Existing issues should be linked to the duplicate.)
Cannot reproduce	This issue cannot be reproduced in the latest version.
Declined	This issue was rejected.
Hardware failure	This issue was caused by physical items connected to your computer, such as keyboard, mouse, RAM, or hard drive.
Software failure	This issue was caused by third party software, such as drivers, Windows, antivirus or any 3 <sup>rd</sup> party program causing havoc.
Revisit	This issue is a verified bug, feature, or task but is currently of low priority. Product owner should revisit these issues on a regular basis to reassess.
No response	There was no response from user after a request for more information.
Update / Clean restart	An update or a clean restart solved the problem.

## Dashboards

Logging on to Jira displays the **Dashboard**, which is intended to give you a high-level view of the project quickly. Multiple dashboards are supported, and they each consist of a collection of *gadgets* that you can customize and reorganize to your own needs.

### Example



To create a new Dashboard, go to the Dashboard menu and select Manage dashboards. Click the Create new dashboard button:

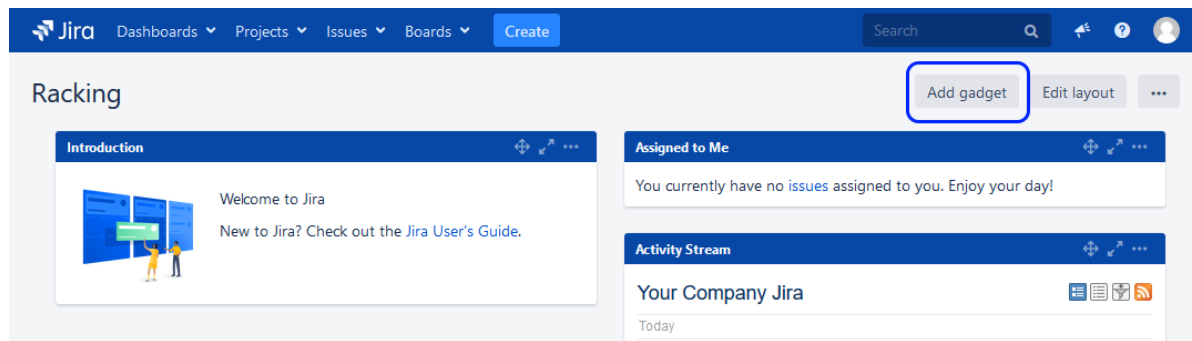
The screenshot shows the Jira interface. At the top, the navigation bar includes 'Jira', 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button. A dropdown menu for 'Dashboards' is open, showing 'Racking' and 'Manage Dashboards'. A blue circle with the number '1' highlights the 'Manage Dashboards' option. To the right, a blue circle with the number '2' highlights the 'Create new dashboard' button. Below the navigation bar, the 'Favorite Dashboards' section is visible, showing a table with one dashboard named 'Racking' owned by 'Sara Schill Saran (sarsc)'.

Enter a name and either choose an existing dashboard to start from or choose to start from a blank dashboard. There are also other parameters available, but they are optional. When done, click Add:

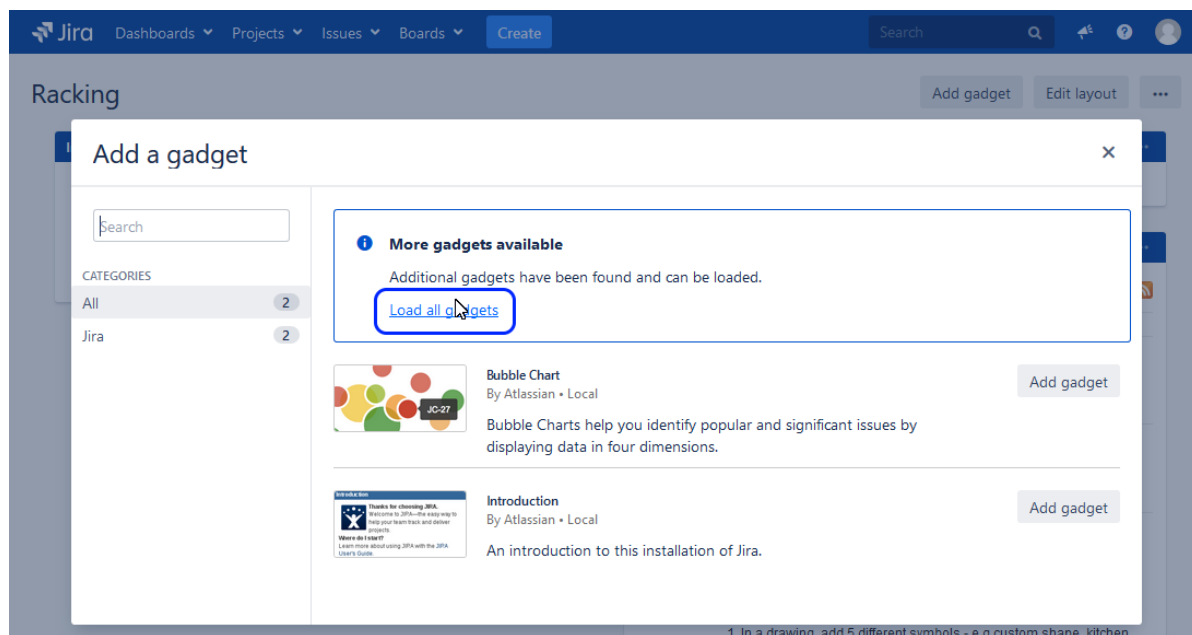
The screenshot shows the 'Create New Dashboard' form. The 'Name' field is filled with 'Racking'. The 'Description' field is empty. The 'Start From' dropdown is set to 'System Dashboard'. Below this, there is a note: 'Choose a dashboard whose gadgets will be copied to the new dashboard. Alternatively, choose 'Blank dashboard' to create a dashboard with no gadgets.' The 'Favorite' checkbox is checked. The 'Viewers' section is set to 'Not shared'. The 'Add Viewers' section shows a dropdown set to 'Group' and a text input field containing 'Confiqura Developers', with an '+ Add' button. The 'Editors' section is also set to 'Not shared'. The 'Add Editors' section shows a dropdown set to 'Group' and a text input field containing 'Confiqura Developers', with an '+ Add' button. At the bottom, there are two buttons: 'Add' (highlighted with a blue box) and 'Cancel'.



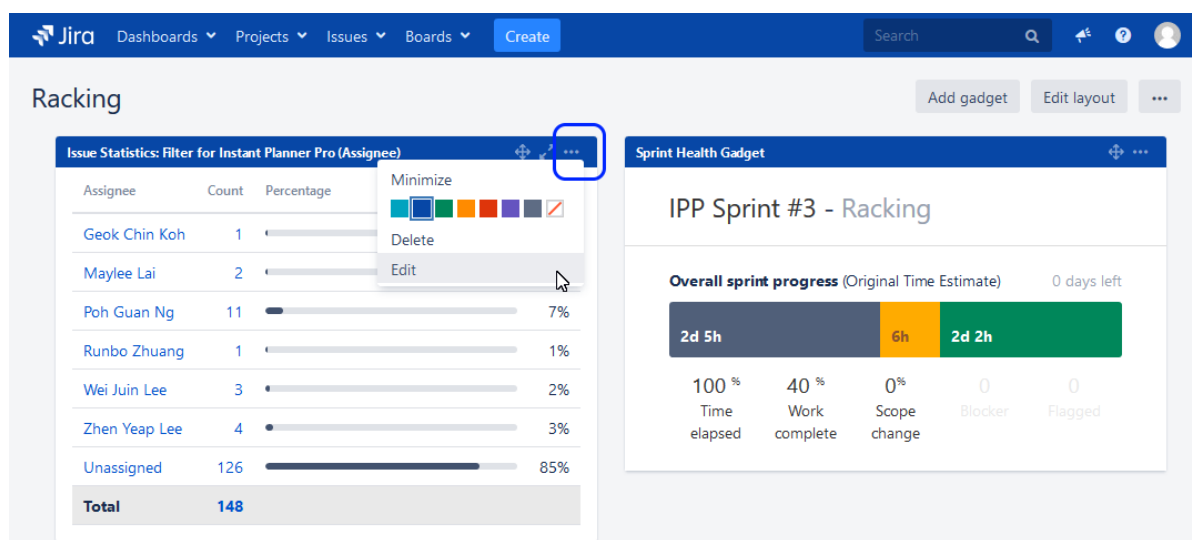
In the list, click the new dashboard that you created to open it, and then click the **Add gadget** button:



Click the Load all gadgets option to list all available gadgets, and add the ones you want:



Configure and save your gadgets. Once in place, a gadget can easily be moved by clicking the gadget header and dragging it to the desired location on the dashboard. Each gadget can also be edited:



## Backlog View

The **Backlog** view is where to:

- Create sprint versions.
- Change the sprint version for the issues.
- Start and close the sprint.
- Make an original time estimate for the sprint.

The backlog can be reordered by clicking and dragging the issues.

Clicking an issue in the backlog bring up the **Issue Detail View**, as shown here:

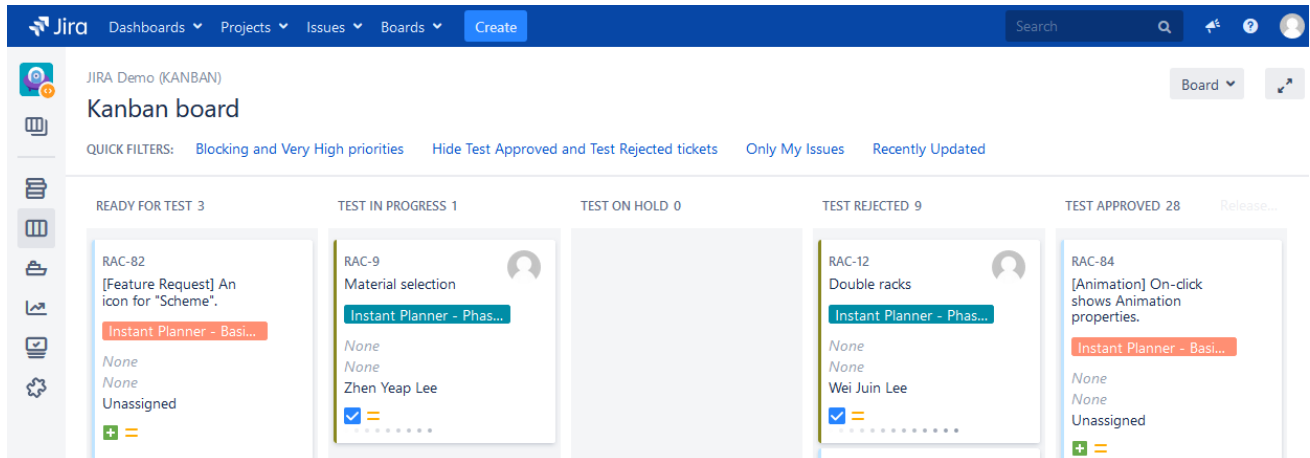
The screenshot shows the Jira Backlog view for the 'Racking' project. The left sidebar contains navigation links: Racking, Backlog, Active sprints, Releases, Reports, Issues, and Components. The main area displays the 'Backlog' view with a search bar and quick filters: Performance, Crashes, Blocking and Very High priorities, High priority, Medium priority, and ... Show more. The backlog is organized into sprints. The current sprint is 'Regression Test - Feb BETA' (5 issues, ACTIVE), which started on 28/Jan/19 5:58 AM and ends on 01/Mar/19 5:58 AM. Below this, a list of issues is shown, including 'RAC-447 REG - Racking - Smoke Test' (4h estimate, None priority, New status), 'RAC-448 REG - Racking - General Test' (5h estimate, None priority, New status), 'RAC-449 REG - Racking - Property Box' (6h estimate, None priority, New status), 'RAC-450 REG - Racking - Scheme' (5h estimate, None priority, New status), and 'RAC-438 Template does not shrink when racks are deletec' (1h estimate, None priority, Test rejected status). The bottom of the backlog shows 'IPP Sprint #3' (47 issues, ACTIVE), which started on 25/Feb/19 7:34 AM and ends on 08/Mar/19 12:37 AM. The right sidebar displays the 'Issue Detail View' for 'Racking / RAC-447'. It shows the issue title 'REG - Racking - Smoke Test', an estimate of 4h, and a remaining time of 4h. The 'Details' section includes fields for Resolution (Unresolved), Type (QA), Sprint (Regression Test - Feb BETA), Fixed in (9.5 Git ippro\_China), Team (None), Status (NEW (View Workflow)), Priority (Medium), Component/s (None), Labels (None), Affects Version/s (None), Fix Version/s (None), and Epic Link (None). The 'People' section is also visible at the bottom.

## Board (Kanban) / Active Sprint (Scrum) View

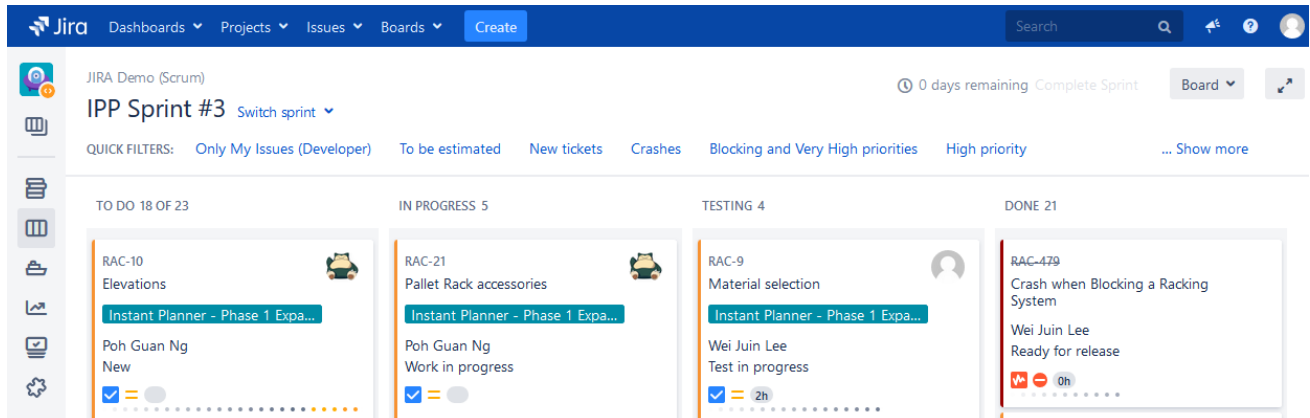
The **Board/Active Sprint** view is where the development team do their daily work. There can be multiple boards for different scrum teams or groups (currently set up by Configura), and a single board can cover multiple projects.

The Board/Active Sprint has a card layout where **Columns**, **Swimlanes** and **Quick Filters** group and filter the cards.

### Example: Board (Kanban)



### Example: Active Sprint (Scrum)



## Scrum vs Kanban

### Scrum

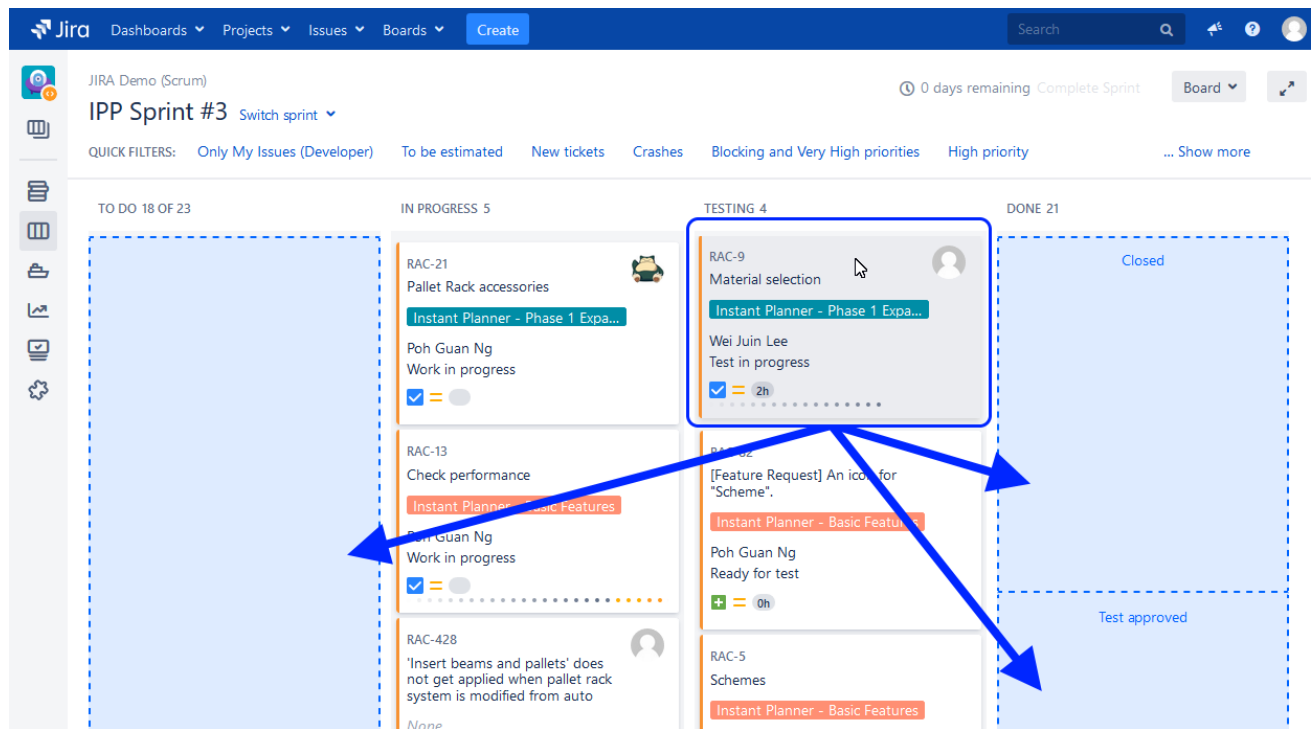
- Create sprint backlogs
- Add tasks to sprints
- More iterative

### Kanban

- Work directly from master backlog
- Pull next highest priority from the master backlog
- Less iterative

## Changing Issue Status in the Board/Active Sprint View

When working in the **Board/Active Sprint** view, issue status is changed by dragging the card from one column to another. When doing so, Jira will guide you to where the card may be moved:



## Releases View

The **Releases** view lets you know the status of a version at a glance. Hover over a progress bar for a summary, or click a version to see the complete status, including the issues, development data, and potential problems.

For each version, it is possible to generate a basic release notes draft\* via the Release Notes links as highlighted in the screenshot below:

The screenshot shows the Jira interface for a version named 'Version Fika 9.5.1' with a status of 'UNRELEASED'. The 'Release Notes' link is highlighted with a blue box. Below the version header, there is a progress bar and a summary of issues: 7 Issues in version, 1 Issues done, 6 Issues in progress, and 0 Issues to do. A table lists the issues:

P	T	Key	Summary	Assignee	Status
^	+	FO-97	Feature Request- Ability to change tile type on FIKA panels	Unassigned	CODE REVIEW
^	✓	FO-140	Auto place Fika chairs around tables	Kyle Peltier	WORK IN PROGRESS
^	■	FO-184	Office chair price defaults to 1.	Unassigned	TEST APPROVED

\*Pre-requisites to make this work are:

1. A version has to be created in the Project Settings.
2. A Fix Version has to be specified for each issue to determine which release it should be included in.

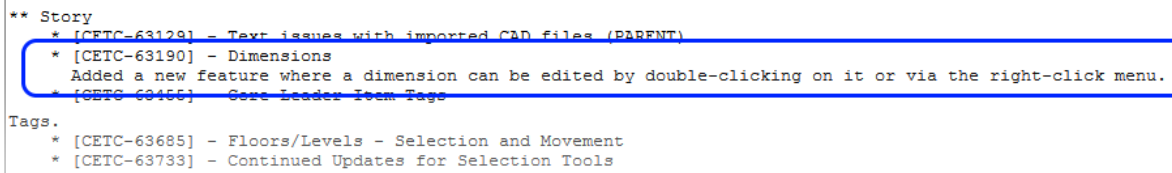
When release notes are generated using this button, they will include text that has been entered into issues' Release Notes field:

The screenshot shows the 'Edit Issue' dialog for CETC-63190. The 'Release' tab is selected, and the 'Release notes' field is highlighted with a blue box. The text in the field is: 'Added a new feature where a dimension can be edited by double-clicking on it or via the right-click menu.' Below the field, there is a 'Comment' section with a rich text editor toolbar.

In the release notes, the text input appears like this:

### Edit/Copy Release Notes

The text area below allows the project release notes to be edited and copied to another document.



```
** Story
  * [CETC-63129] - Text issues with imported CAD files (PARENT)
  * [CETC-63190] - Dimensions
    Added a new feature where a dimension can be edited by double-clicking on it or via the right-click menu.
  * [CETC-63466] - Core Leader Item Tags
Tags.
  * [CETC-63685] - Floors/Levels - Selection and Movement
  * [CETC-63733] - Continued Updates for Selection Tools
```

The screenshot shows a text editor with a light gray background. A blue rounded rectangle highlights the following text:

- \* [CETC-63129] - Text issues with imported CAD files (PARENT)
- \* [CETC-63190] - Dimensions
- Added a new feature where a dimension can be edited by double-clicking on it or via the right-click menu.
- \* [CETC-63466] - Core Leader Item Tags

Below the highlighted text, the word "Tags." is followed by two more items:

- \* [CETC-63685] - Floors/Levels - Selection and Movement
- \* [CETC-63733] - Continued Updates for Selection Tools

## Reports View

The **Releases** view offer many different report templates for reports, charts, and diagrams.

The screenshot displays the Jira Reports View interface. On the left is a sidebar with navigation options: Racking, Backlog, Active sprints, Releases, Reports (selected), Issues, and Components. Below these are project shortcuts and an 'Add link' button. The main content area is titled 'All reports' with a 'Switch report' dropdown. Under the 'Agile' section, six report templates are shown in a grid:

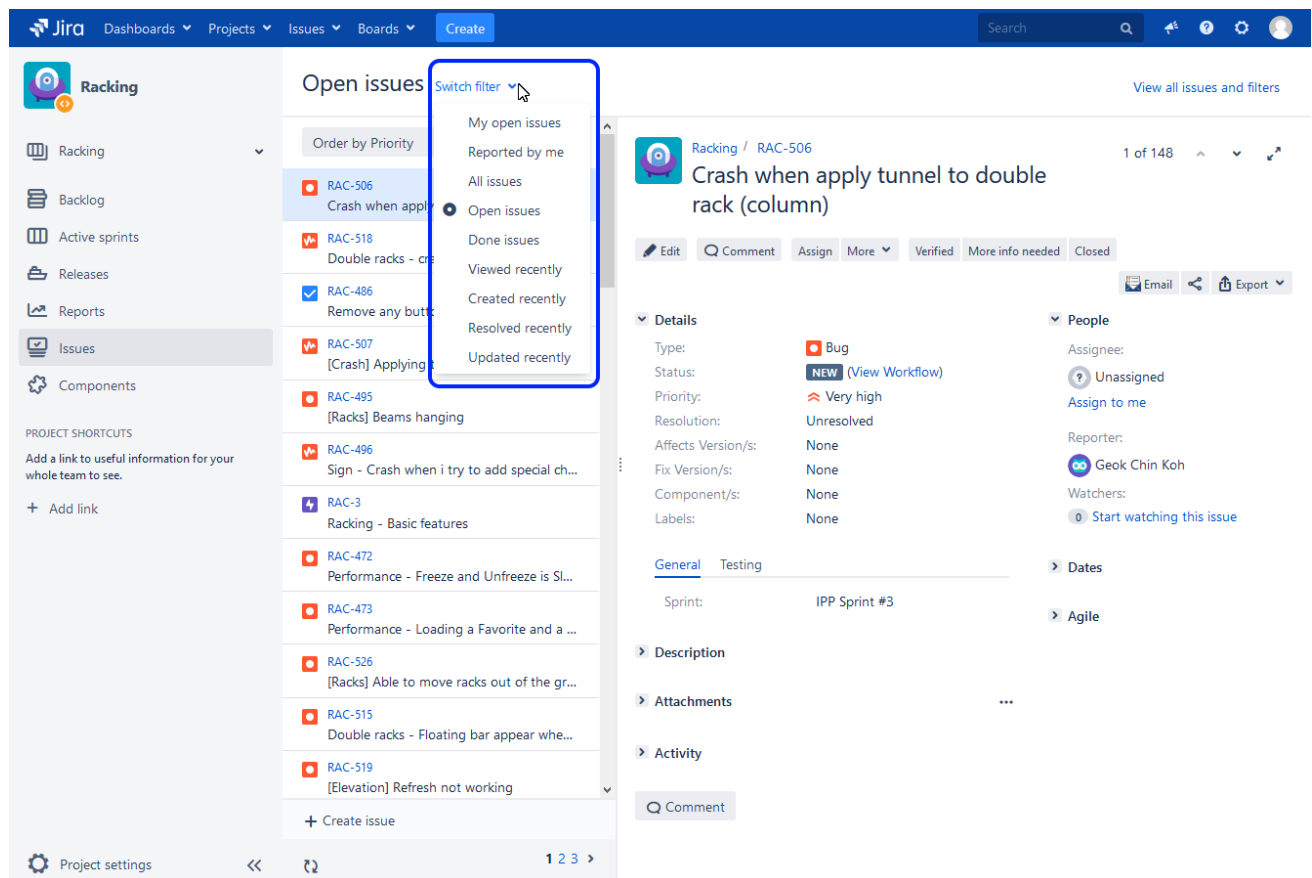
- Burndown Chart:** Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.
- Sprint Report:** Understand the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is excessive scope creep.
- Velocity 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.
- Cumulative Flow Diagram:** Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.
- Version Report:** Track the projected release date for a version. This helps you monitor whether the version will release on time, so you can take action if work is falling behind.
- Epic Report:** Understand the progress towards completing an epic over time. This helps you manage your team's progress by tracking the remaining incomplete/unestimated work.

A few examples of available reports are:

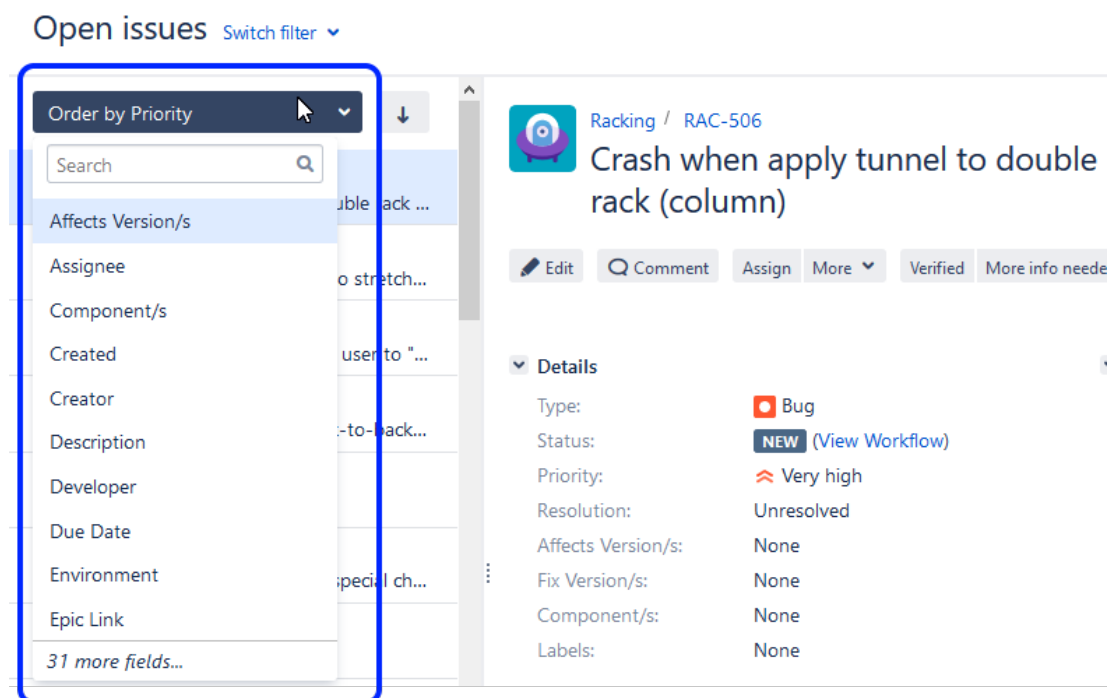
- **Sprint Report:** understand the work completed or pushed back to the backlog in each sprint.
- **Burndown Chart:** track the total work remaining and project the likelihood of achieving the sprint goal.
- **Velocity Chart:** track the amount of work completed from sprint to sprint.
- **Control Chart:** shows the cycle time for your product, version, or sprint.
- **Cumulative Flow Diagram:** shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.

## Issues View

The **Issues** view lists all open issues in the board, but can be filtered to show only certain issues:



The list can be ordered in different ways:





Selecting an issue in the list shows detailed information about it in the **Issue Detail** view:

The screenshot displays the Jira web interface. On the left is a sidebar with navigation options like 'Racking', 'Backlog', 'Active sprints', 'Releases', 'Reports', 'Issues', and 'Components'. The main area is titled 'Open issues' and shows a list of issues. Issue RAC-506, titled 'Crash when apply tunnel to double rack (column)', is selected. To the right, the 'Issue Detail' view is open, showing the issue's metadata: Type (Bug), Status (NEW), Priority (Very high), Resolution (Unresolved), and Assignee (Unassigned). The 'Details' section includes fields for Affects Version/s, Fix Version/s, Component/s, and Labels, all set to 'None'. The 'People' section shows the Reporter as Geok Chin Koh and the Assignee as Unassigned. The 'Description' and 'Attachments' sections are also visible. A blue box highlights the 'Issue Detail' view.


The issue status can be updated in the **Issue Detail** view using the **Transition** buttons. Simply click one of the buttons to update to that status:


This screenshot shows the same 'Issue Detail' view for issue RAC-506. A blue box highlights the transition buttons: 'Verified', 'More info needed', and 'Closed'. The 'Details' section shows the issue's metadata, and the 'People' section shows the Reporter and Assignee. The 'Description' and 'Attachments' sections are also visible.

If you need to communicate directly with a user who does not have a Jira account, you can use the **Email** button available in the **Issue Detail** view. If the user replies to the email, the reply is added to the issue as a comment.

Order by Priority ▾



 **RAC-506**  
Crash when apply tunnel ...

 **RAC-518**  
Double racks - crash whe...








☒ **RAC-486**  
Remove any buttons that...



Racking / **RAC-506**

## Crash when apply tunnel to double rack (column)

1 of 148 ▴ ▾ ↺

 Edit  Comment  Assign  More ▾  Verified  More info needed  Closed

 Email   Export ▾

### Details

Type:

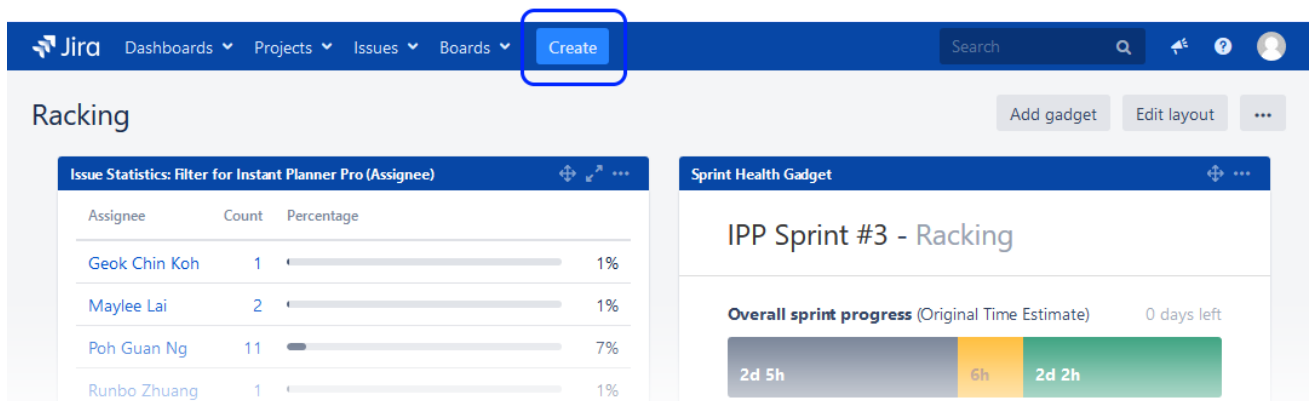
 Bug

### People

Assignee:

## Creating a new Issue

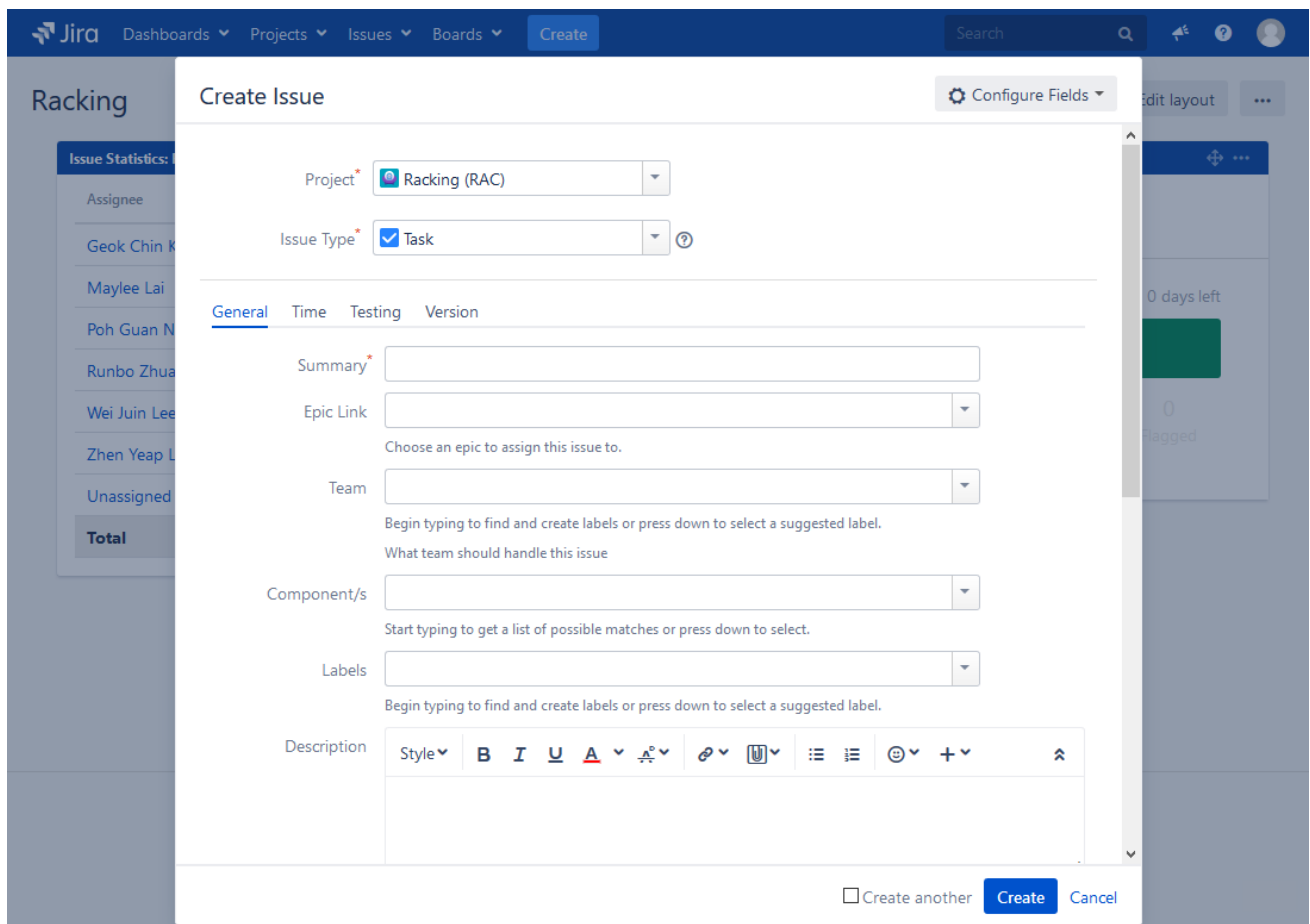
To create a new issue, click the **Create Issue** button on the menu bar, or use the keyboard shortcut C:



The screenshot shows the Jira dashboard for the 'Racking' project. The top navigation bar includes 'Jira', 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button highlighted with a blue box. Below the navigation bar, the dashboard displays two gadgets: 'Issue Statistics: Filter for Instant Planner Pro (Assignee)' and 'Sprint Health Gadget'. The 'Issue Statistics' gadget shows a table with assignee names, counts, and percentages. The 'Sprint Health Gadget' shows the 'IPP Sprint #3 - Racking' with an overall sprint progress bar.

Assignee	Count	Percentage
Geok Chin Koh	1	1%
Maylee Lai	2	1%
Poh Guan Ng	11	7%
Runbo Zhuang	1	1%

The template consists of many fields, and the more information that you include the better. The fields **Project**, **Issue Type** and **Summary** (a.k.a Title) are compulsory and cannot be left empty, and we also recommend to always add a **Description**:



The screenshot shows the 'Create Issue' form in Jira. The form is titled 'Create Issue' and has a 'Configure Fields' dropdown. The 'Project' field is set to 'Racking (RAC)'. The 'Issue Type' field is set to 'Task'. The 'Summary' field is empty. The 'Epic Link' field is empty. The 'Team' field is empty. The 'Component/s' field is empty. The 'Labels' field is empty. The 'Description' field is empty. The form has tabs for 'General', 'Time', 'Testing', and 'Version'. The 'General' tab is selected. The form has a 'Create' button and a 'Cancel' button.

Project: Racking (RAC)

Issue Type: Task

Summary:

Epic Link:

Team:

Component/s:

Labels:

Description:

Create another: ☐ Create Cancel

Components and Labels are optional but very useful for sorting and filtering issues. Components are categories available in the board, while Labels are free text that can be applied to an issue.

## Issue Types

The following **Issue Types** are available:

### Create Issue

Project\* Racking (RAC)

Issue Type\* Task

- Epic
- Story
- New Feature
- Bug
- Regression Bug
- QA
- Crash
- User Bug Report
- User Feature Request

Summary\*

Epic Link

Team

What team should handle this issue

Epics can have Stories and Tasks linked to them while Stories and Tasks can have Sub-tasks.

Please note that **User Bug Reports** and **User Feature Requests** always come in through support, and it is the responsibility of the Product Owner to review them, decide if action should be taken and, in that case, convert them to regular issues.

## Moving an Issue

You can move issues to another project or issue type. To do so, click the **More** menu and select **Move**:

CET Utilities / CETU-260

### Template (JIRA)

Edit Comment Assign More Reopen More info needed Admin

Details

Type: User Bug

Priority: High

Affects Version/s: 9.0

Component/s: None

Labels: None

General Testing

Found In: 9.0

Fixed in: 9.5dev, 9.5

Description

Test QA ticket

TestRail: Results

No test results in TestRail are linked to this issue

Attachments

Log work

Agile Board

Rank to Top

Rank to Bottom

Attach files

Stop watching

Watchers

Create sub-task

Move

Link Move this issue to another project or issue type.

Clone

Labels

Email This Issue

Delete

In the next step, move the User Bug Report to Bug (or User Feature Request to New Feature):

## Step 1 of 2: Select Project and Issue Type

Choose the project and issue type to move to

Next Cancel

**i** The change will affect **1** issues with issue type(s) **User Bug Report** in project(s) **CET Utilities**.

Move

To



CET Utilities



CET Utilities (CETU)



User Bug Report



Bug



☐ Use the above project and issue type pair for all other combinations.

Next Cancel

## Editing an Issue

Issues can be edited from the **Backlog**, **Board / Active Sprint**, and **Issues** views.

In the **Edit Issue** view, there are four tabs available for editing the issue:

The screenshot shows the Jira 'Edit Issue' page for issue RAC-13. The 'General' tab is selected and highlighted with a blue box. The interface includes the following fields and options:

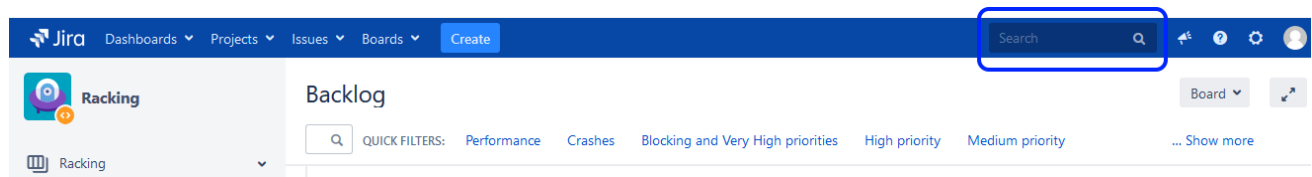
- Issue Type:** Task (selected)
- Summary:** Check performance
- Epic Link:** Instant Planner - Basic Features
- Team:** Asia-IPP
- Component/s:** Performance
- Labels:** (empty)
- Description:** (text area with formatting options)

The right sidebar shows the issue's status as 'Unresolved' and 'Task'.

Note that the test result of an issue is found on the **Testing** tab.

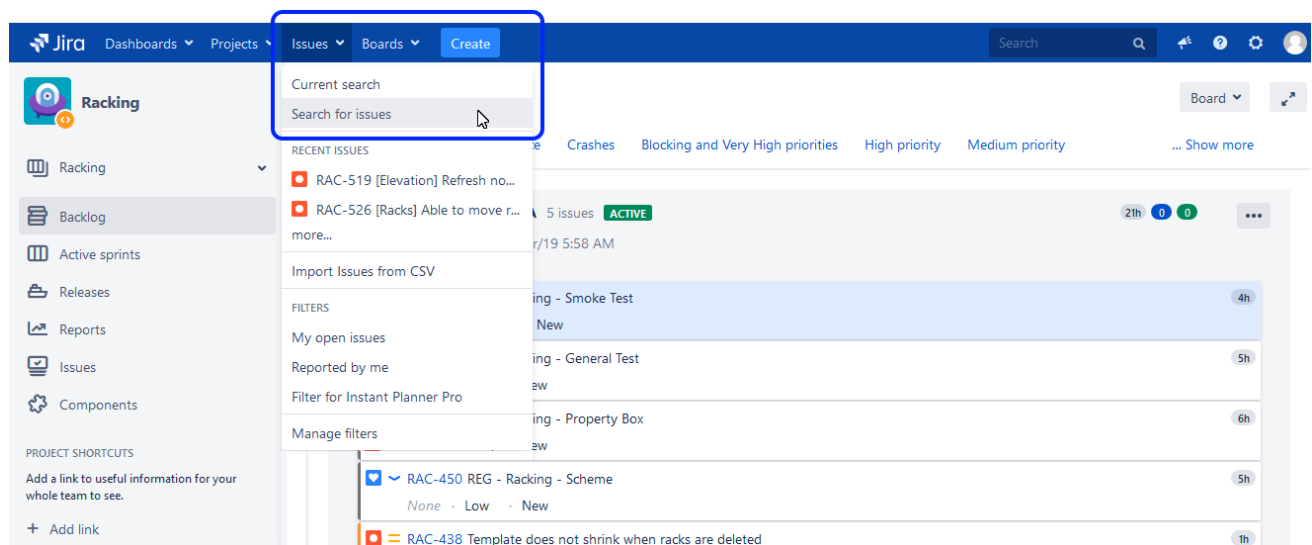
## Searching for Issues

A **Quick Search** is always available, regardless of where you are in Jira:



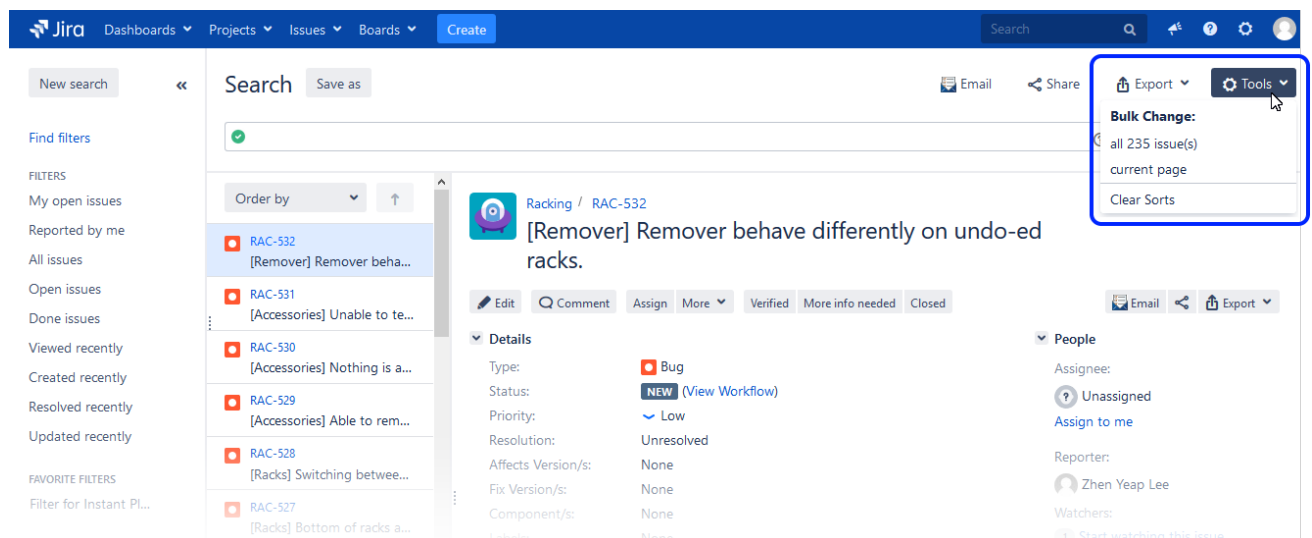
The search is dynamic, and the results will update as you continue typing.

There is also a dedicated search page where you can search for issues, and filters that others have created and shared:



Please note that when searching, the list may look incomplete. Keep writing and more suggestions will be displayed.

From the **Search** view, it is also possible to make bulk changes to multiple issues at once:



Bulk change is also possible by selecting multiple issues in the Backlog (CTRL click), right-clicking and selecting Bulk Change.

## Components View

The **Components** view is where you create and edit components, i.e. categories in the Board:

The screenshot shows the Jira interface with the 'Components' view selected for the 'Racking' project. The left sidebar contains navigation options: Racking, Backlog, Active sprints, Releases, Reports, Issues, and Components (highlighted). Below the sidebar, there are project shortcuts. The main content area displays a table of components with columns: Component, Issues, Lead, Description, Default assignee, and Actions.

Component	Issues	Lead	Description	Default assignee	Actions
Calculation	1 Issue		Issues related to parts	Project default	...
cmSym	0 Issues		Issues due to cmsym	Project lead	...
Integration	6 Issues		Issues related to integration with CET Designer core functionalities	Project default	...
Pallet Racking - Accessories	4 Issues		Issues related to accessories (Selective Racks)	Project default	...
Pallet Racking - Back-to-back racks	8 Issues		Issues related to double racks	Project default	...

Using Components is optional but very useful for sorting and filtering purposes.

## Submitting issues from CET Designer directly to partner's project

By default, all bug reports will be sent to the CET Support service desk, but there are cases where the partner's development team want to see their issues directly and not wait for the CET Support to escalate the issue.

If a development team (developers or QA:s) or UAT testers want to submit a bug report via CET Designer, they can choose to add the text **[INFO: <linked issue key>]** in the description field, as shown in the screenshot below. The linked issue key could, for example, be the parent test ticket number.

If you include this tag however, **[INFO: <issue key>]**, it will:

- Create a new bug ticket and link to the mentioned "linked issue key". Both tickets are in the same project.
- Remove the issue from the CET Support service desk.

The link type can be updated, removed, etc. as needed.


The screenshot shows a 'Send bug report' dialog box. It has a 'Problem description' section with a text area containing 'I would like to report the following issue:' and 'Testing linked ticket Tag: Info'. Below this, the text '[INFO:CETU-798]' is entered and highlighted with a blue circle. There is a 'Verify that your email address is correct' section with the email 'geokchin.koh@configura.com'. At the bottom, there is an 'Attached files' section showing three files: 'c:\Users\geoko\AppData\Local\CET Data\10.0-RC-64-bit\log0.cmtxt', 'c:\Users\geoko\AppData\Local\CET Data\10.0-RC-64-bit\log1.cmtxt', and 'c:\Users\geoko\AppData\Local\CET Data\10.0-RC-64-bit\log2.cmtxt'.

In Jira, it will look like this:

▼ Issue Links

+


more info by



[CETU-799](#) CET Issue: CET Designer 10.0-RC-64-bit (Account 6208)

▼

CLOSED



[CETU-802](#) CET Issue: CET Designer 10.0-RC-64-bit (Account 6208)

▼

CLOSED



## Appendix 1: Global Project Automation Rules

### Jira Software Project

SET THE STATUS TO	AUTOMATION / WHAT WILL HAPPEN	REMARK
New	Not available.	Assignee will not be changed.
More info needed (verification)	Show a pop-up dialog for users to set an Assignee.	
More info added (verification)	Set Assignee to Developer or remove Assignee if developer field is empty.	
Verified	Not available.	
To Do	Not available.	
Work in progress	If the developer field is set, set Assignee to Developer. If not, set both Assignee and Developer to current user.	
More info needed (WIP)	Show a pop-up dialog for users to set an Assignee.	
More info added (WIP)	Set Assignee to Developer or remove Assignee if the developer field is empty.	
Code review	Remove Assignee.	
Ready for test (Dev)	Remove Assignee.	
Test in progress (Dev)	Set Assignee to current user.	
More info needed (Dev)	Set Assignee to Developer.	
Test passed (Dev)	Not available.	
Waiting for merge	When it is transitioned from <i>Test in Progress (Dev)</i> , set QA field to current user.	
Waiting for build	When it is transitioned from <i>Test in Progress (Dev)</i> , set QA field to current user.	
Ready for test (build)	Remove Assignee.	
Test in progress (build)	Set Assignee to current user.	
More info needed (build)	Set Assignee to Developer.	
Test rejected	Set Assignee to Developer. When it is transitioned from <i>Test approved</i> , remove Resolution.	

	When it is transitioned from Test In progress (build), set QA field to current user.	
Test approved	Remove Assignee. When it is transitioned from <i>Test in progress (dev)</i> or <i>Test in progress (build)</i> , set QA field to current user. Set Resolution to Done. Set Remaining Time to zero.	
Ready for release	Set Resolution to Done.	
Released	When the fix version is set to <i>Released</i> , change the issues to Released if the previous status is <i>Ready for Release</i> . Set all duplicate issues with <i>Closed</i> status to Released too. Send out an email or shared comment with customer to the reporter in the Service Desk ticket saying that an update has been released. Close the Service Desk ticket with resolution Done.	
Closed	Set Remaining Time to zero for all resolutions except for Revisit. Add Resolution info as a comment in the linked Jira Service Desk issue.	

ACTION		
Add a comment with @support tag	If the previous status of the Jira Software ticket is <b>New</b> or <b>More info added (verification)</b> then it will be set to <b>More info needed (verification)</b> .	No changes for other statuses (Verified, Closed, etc.).
Edit priority	Add a comment “Escalated issue got priority..” on the Service Desk issue.	

## Jira Service Desk

SET THE STATUS TO	AUTOMATION	REMARK
Close	If the previous status of the Jira Software ticket is <b>More info needed (verification)</b> , then it will be set to <b>More info added (verification)</b> automatically.	No changes for other statuses (New, Verified, Closed, etc.).

ACTION		
Add a comment with @developer tag	If the previous status of the Jira Software ticket is <b>More info needed (verification)</b> or <b>Closed</b> , then it will be set to <b>More info added (verification)</b> automatically.	No changes for other statuses (New, Verified, Closed, etc.).  Comment will still be copied to the Jira Software project.
Update resolution	Add a "Support issue got resolution:" comment in the Jira Software ticket.	

## Appendix 2: Jira Software Keyboard Shortcuts

Keyboard Shortcuts	
<b>Global Shortcuts</b>	
Go to Dashboard:	<b>g</b> then <b>d</b>
Next Item:	<b>j</b>
Browse to a Project:	<b>g</b> then <b>p</b>
Go to Boards:	<b>g</b> then <b>a</b>
Find Issues:	<b>g</b> then <b>i</b>
Previous Item:	<b>k</b>
Administration Quick Search:	<b>g</b> then <b>g</b>
Go to Tempo:	<b>g</b> then <b>t</b>
Log work with Tempo:	<b>w</b>
Quick Search:	<b>/</b>
Create an Issue:	<b>c</b>
Quick Operations:	<b>.</b>
Open shortcut help:	<b>?</b>
Form Submit:	<b>Ctrl</b> + <b>Alt</b> + <b>s</b>
<b>Navigating Issues</b>	
View selected Issue:	<b>o</b> or <b>Enter</b>
Next Issue:	<b>j</b>
Previous Issue:	<b>k</b>
Dock/Undock the Filters Panel:	<b>[</b>
Next Activity:	<b>n</b>
Previous Activity:	<b>p</b>
Focus search field:	<b>f</b>
Search for issues:	<b>u</b>
Switch filter view:	<b>t</b>
Detail View Order by:	<b>z</b>
Log work on issue:	<b>w</b>
<b>Issue Actions</b>	
Edit Issue:	<b>e</b>
Assign Issue:	<b>a</b>
Comment on Issue:	<b>m</b>
Share Issue:	<b>s</b>
Edit Issue Labels:	<b>l</b>
<b>Board Shortcuts</b>	
Backlog:	<b>1</b>
Active sprints / Kanban board:	<b>2</b>
Reports:	<b>3</b>
Presentation mode:	<b>z</b>

(Source: <https://www.atlassian.com/blog/jira-software/4-ways-get-jira-keyboard-shortcuts>)