GUIDELINES FOR
WORKING IN JIRA

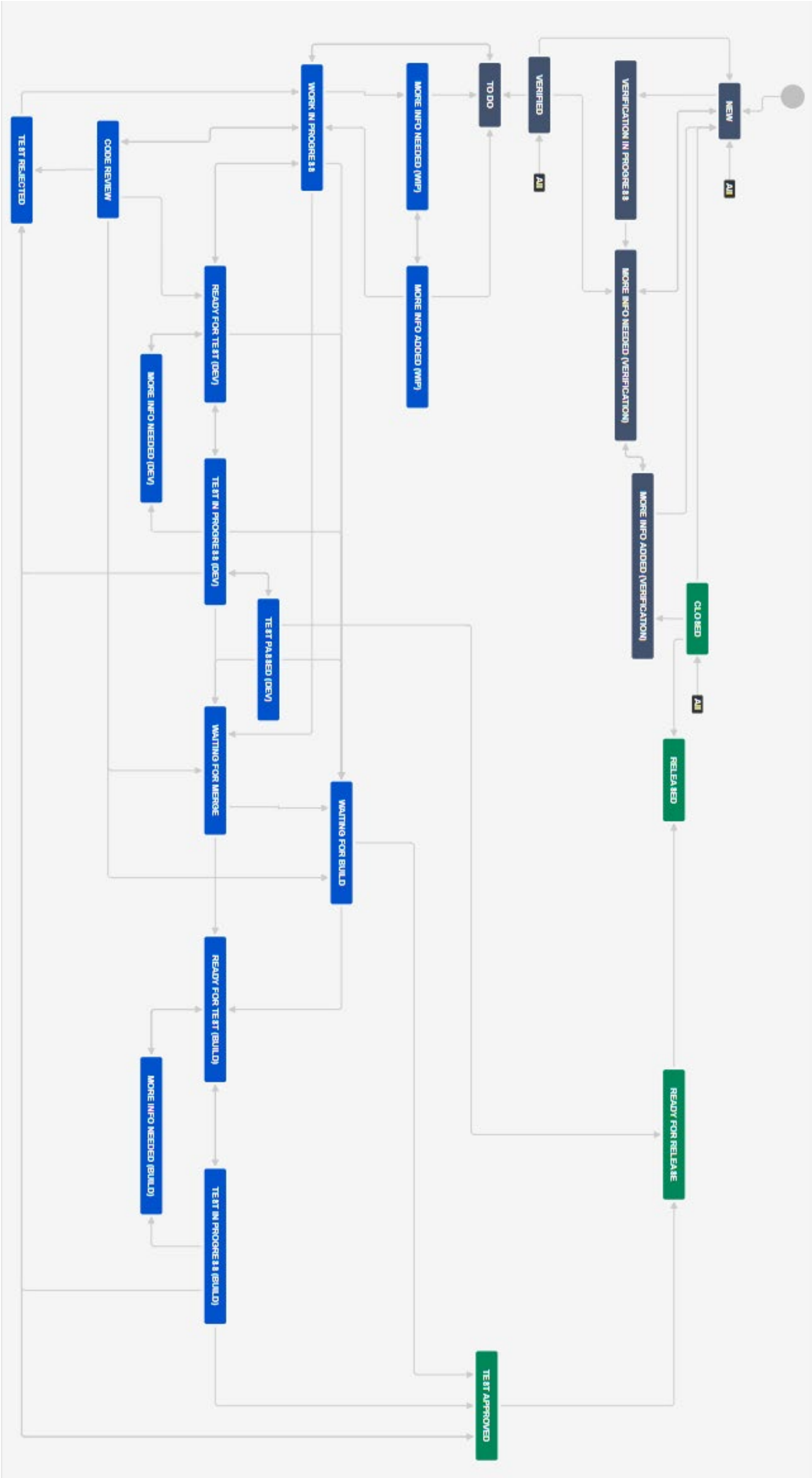
CONFIGURA

Contents

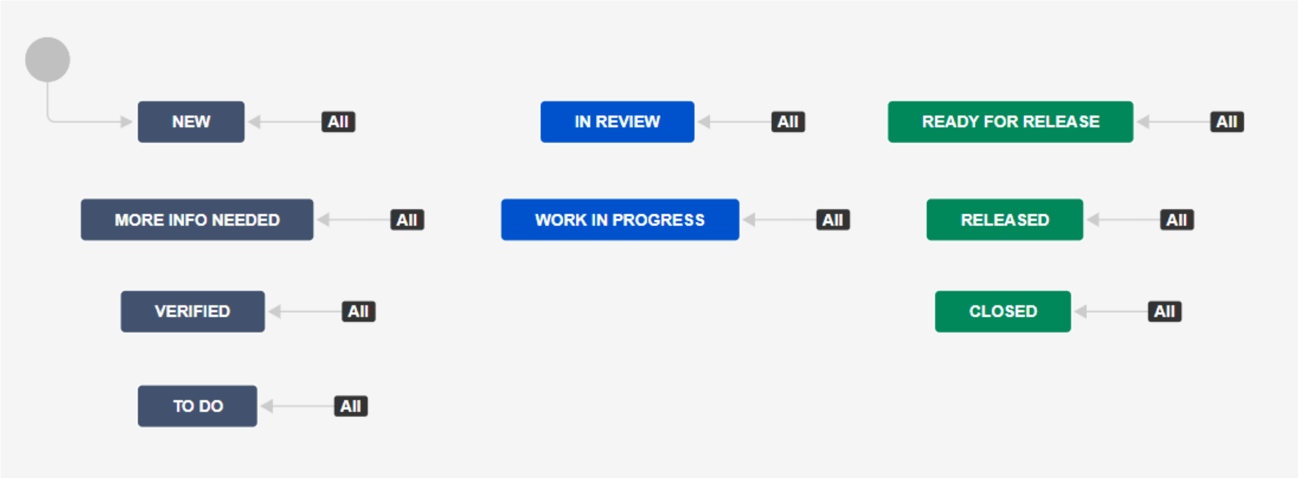
Working in Jira

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 W*aiting 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

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.

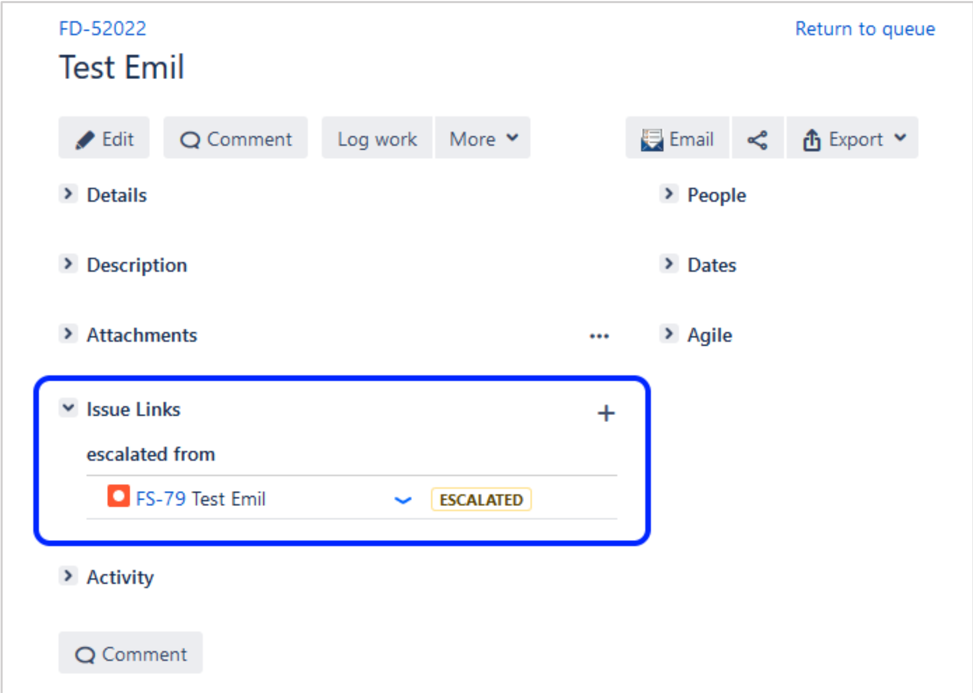
For example: When an issue is transitioned to *Work in Progress*, the assignee will automatically be set to *Developer* if the developer field is set. If the developer field is empty, the assignee will automatically be set to current user. For more information, see Appendix 1.

Working with Issues Escalated from Jira Service Desk

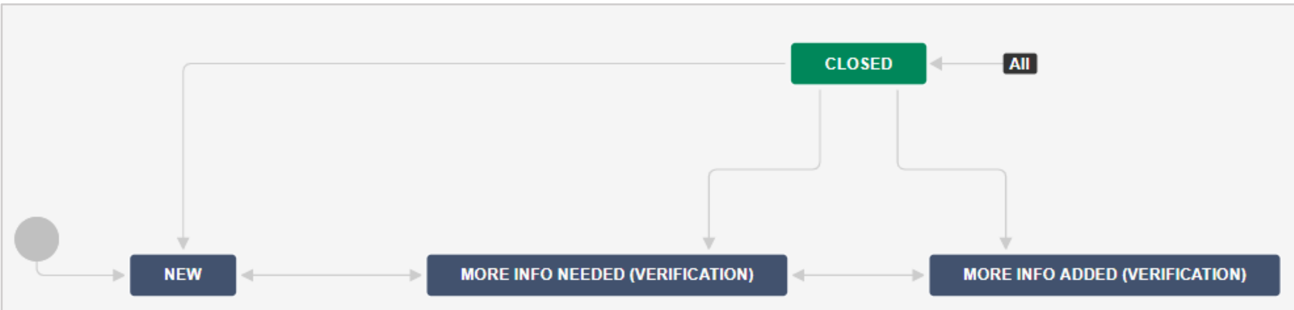
If your Jira Software Project includes issues that are escalated from Jira Service Desk, there are some things that are good to know. If your organization does not have that connection, you may skip this section.

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:



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



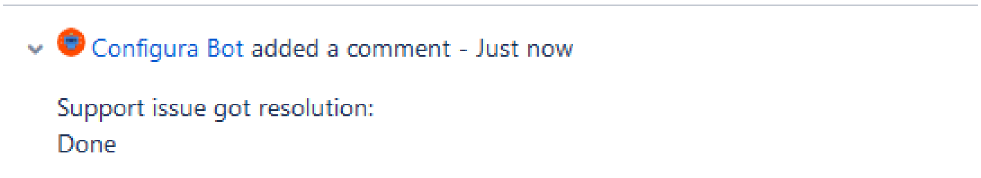
If the developer wishes to share information or ask a question, they can use @support in a comment. The comment is then added to the Service Desk issue and the issue status changes to *Waiting for support*. The status of the Software issue will automatically be changed to *More info needed*.

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".

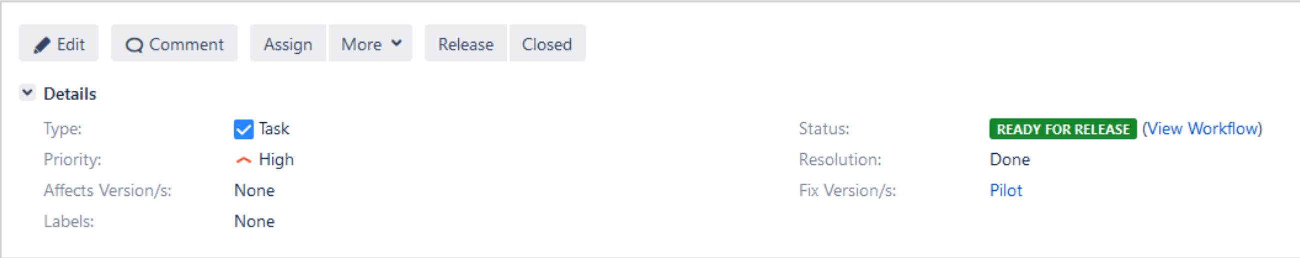
A developer can access the Service Desk issue without a Jira Service Desk license, but they cannot edit the ticket except make internal comments or attach files.

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:

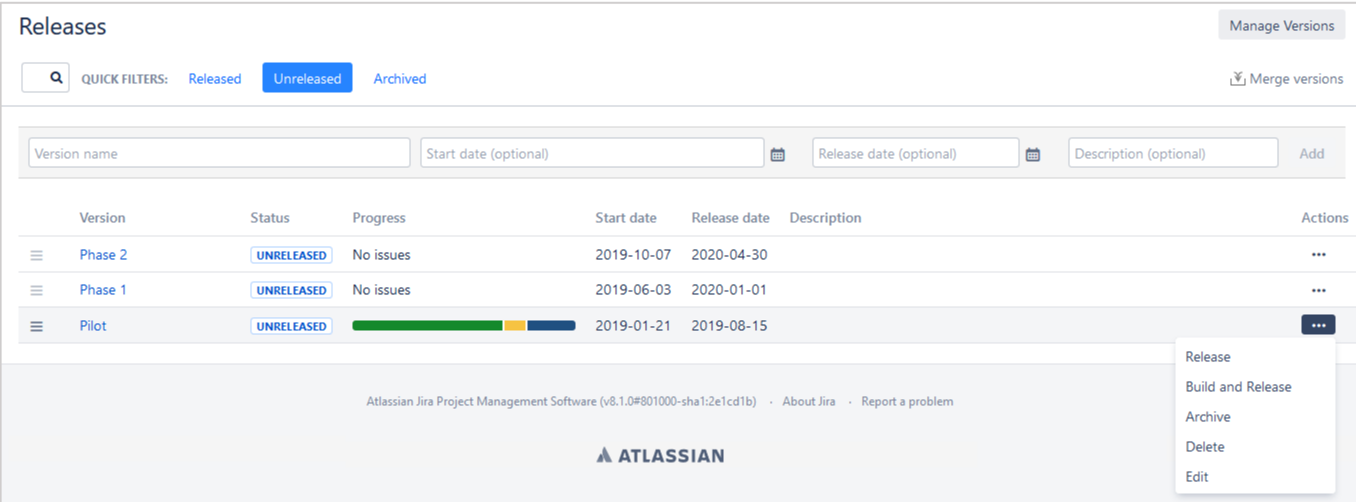


Please note that User Bug Reports and User Feature Requests always come in through support, and it is the responsibility of the Product Owner to review them, decide if action should be taken, and, in that case, convert them to regular issues. For more information, see the *Issue Type* and *Moving an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



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.



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

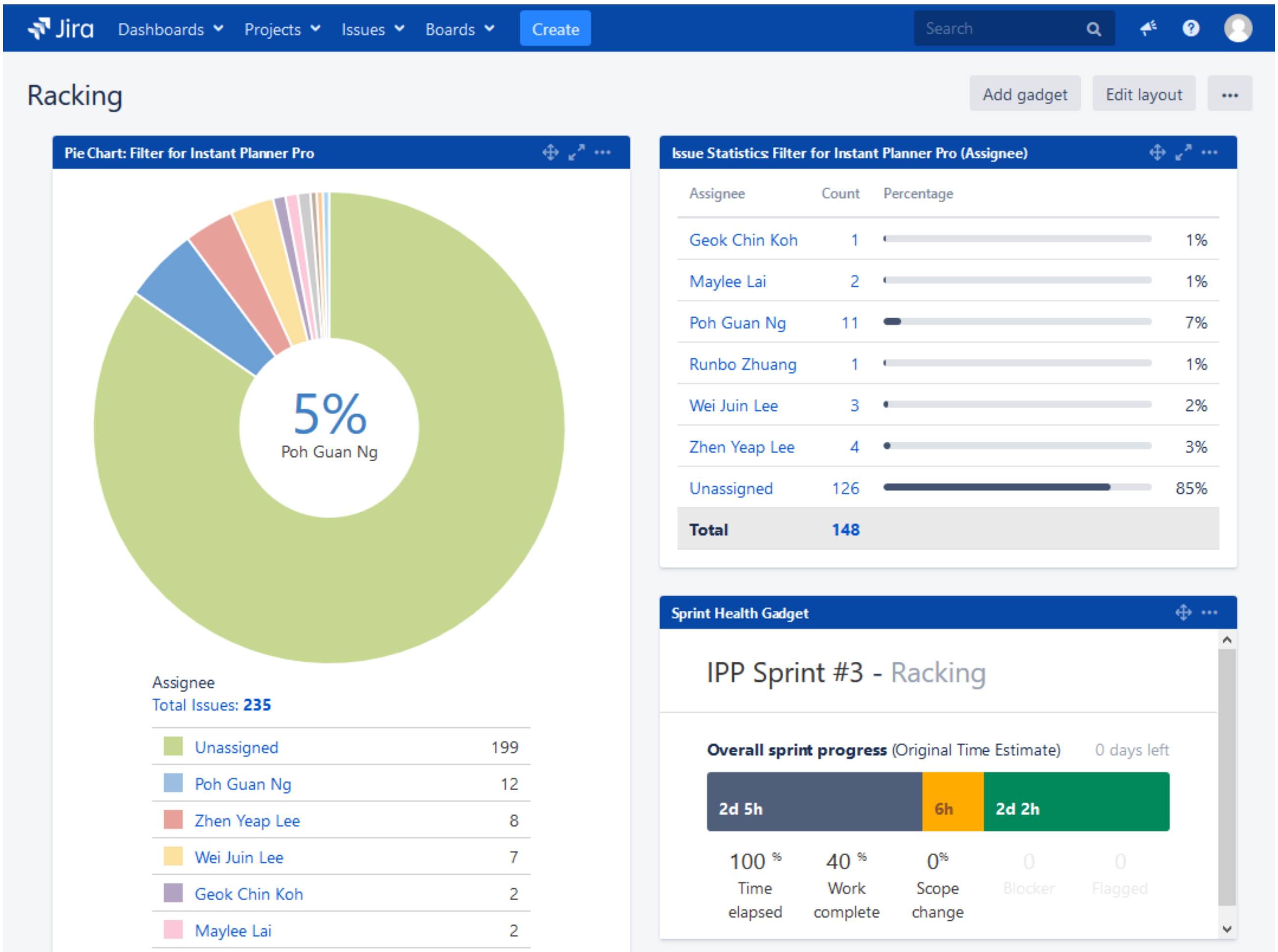
These are the different resolutions that can be used when closing an issue.

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 3rd 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.

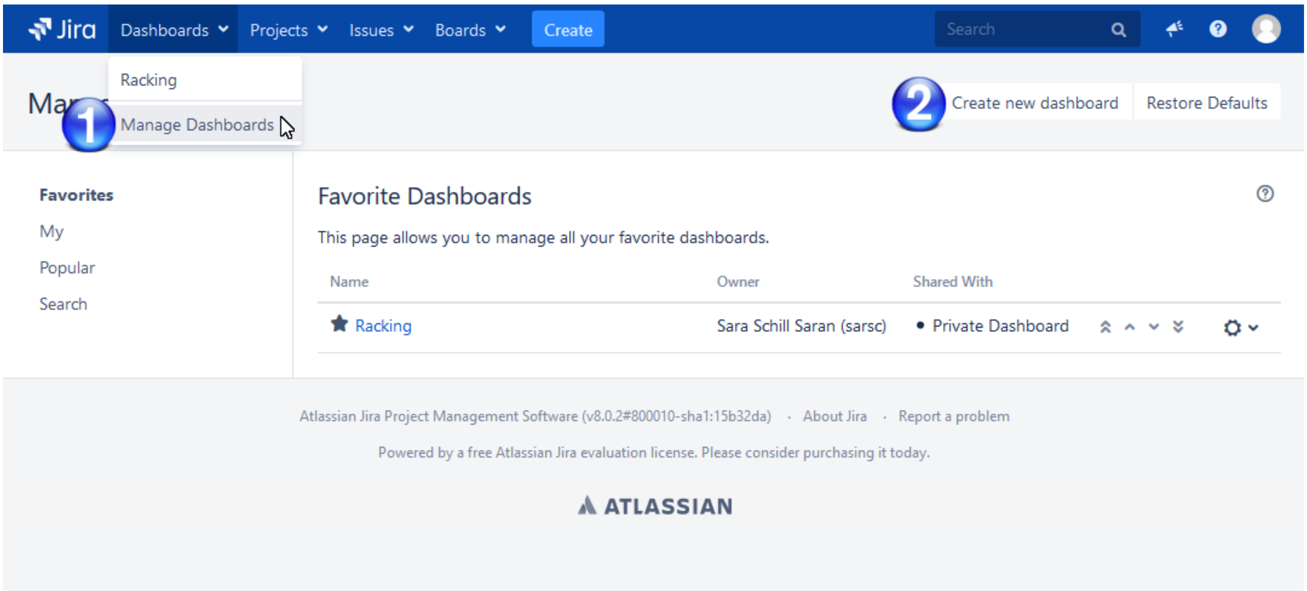
Dashboards

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.

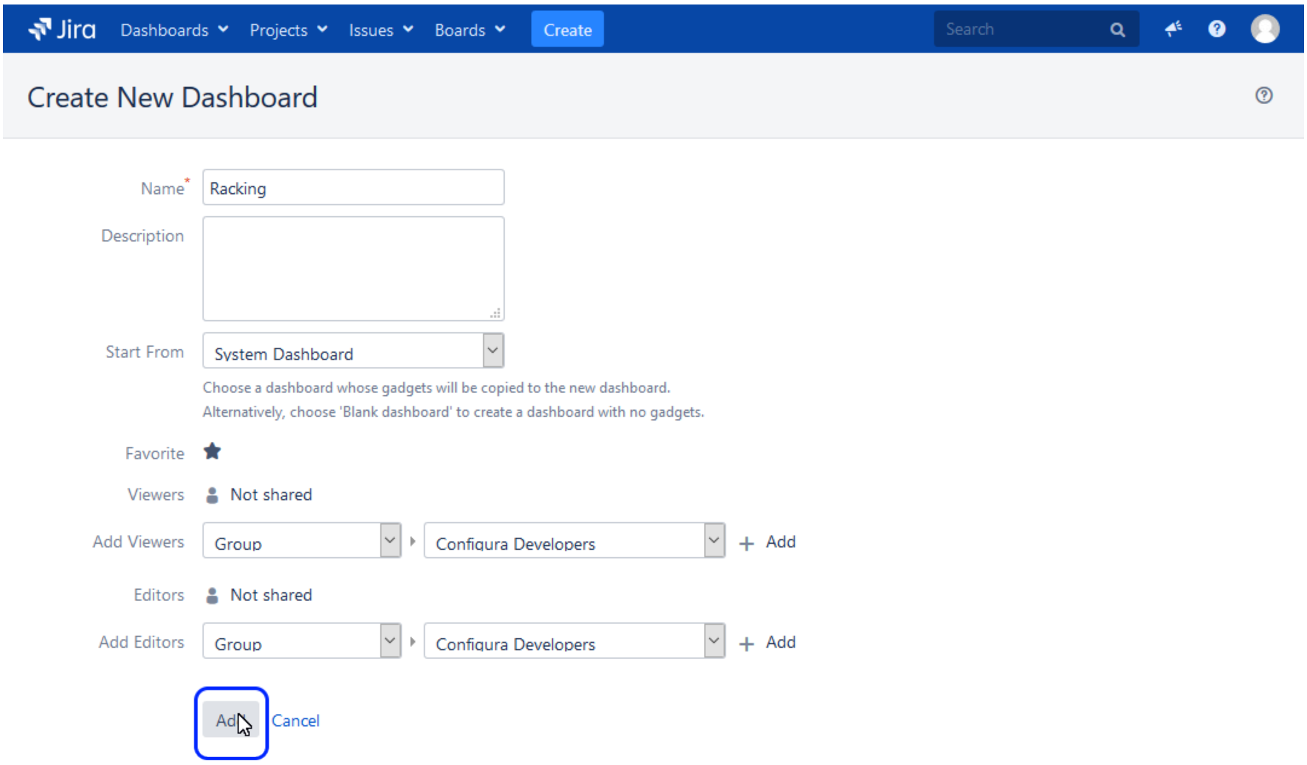
Example



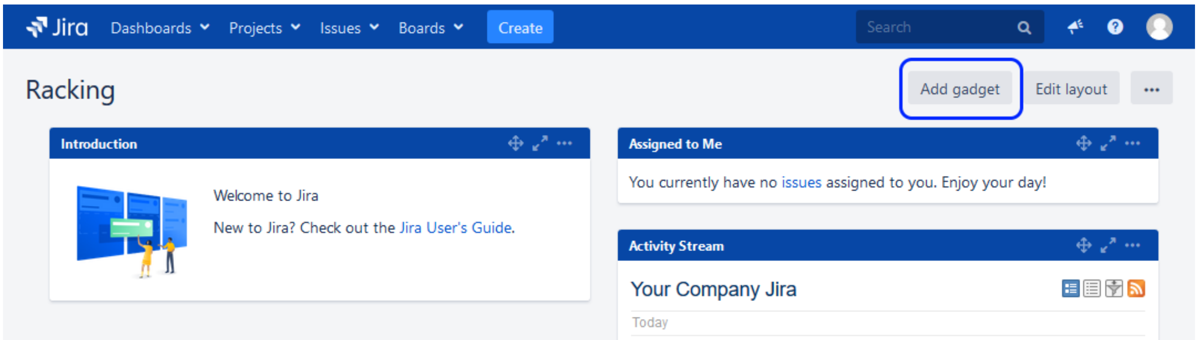
To create a new Dashboard, go to the Dashboard menu and select Manage dashboards. Click the Create new dashboard button:



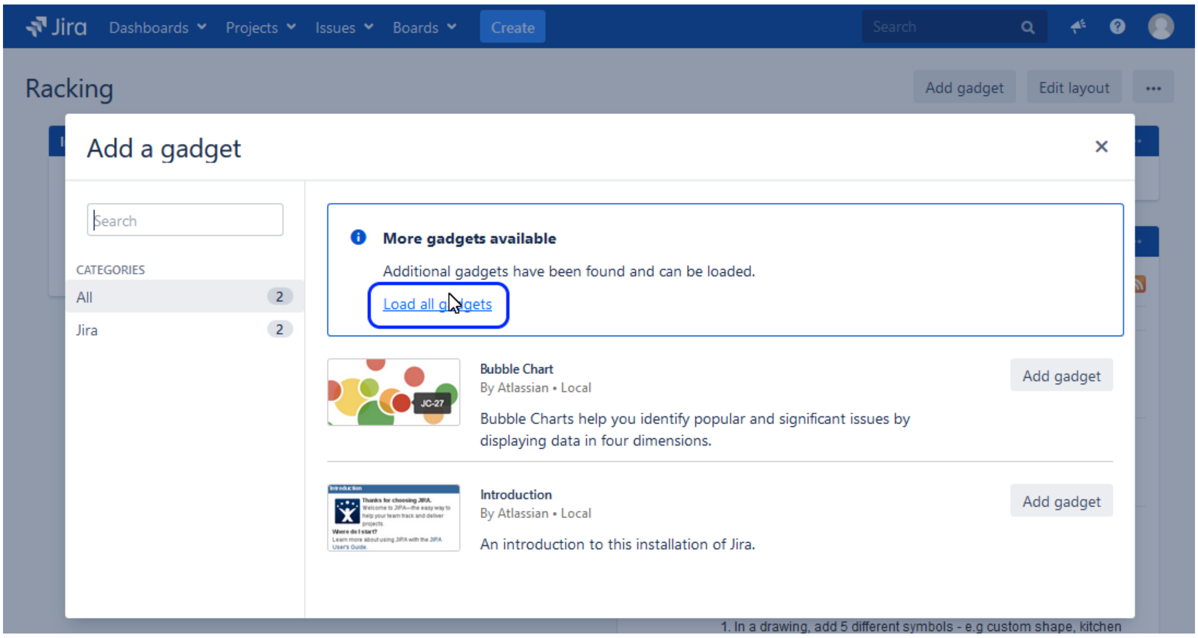
Enter a name and either choose an existing dashboard to start from or choose to start from a blank dashboard. There are also other parameters available, but they are optional. When done, click Add:



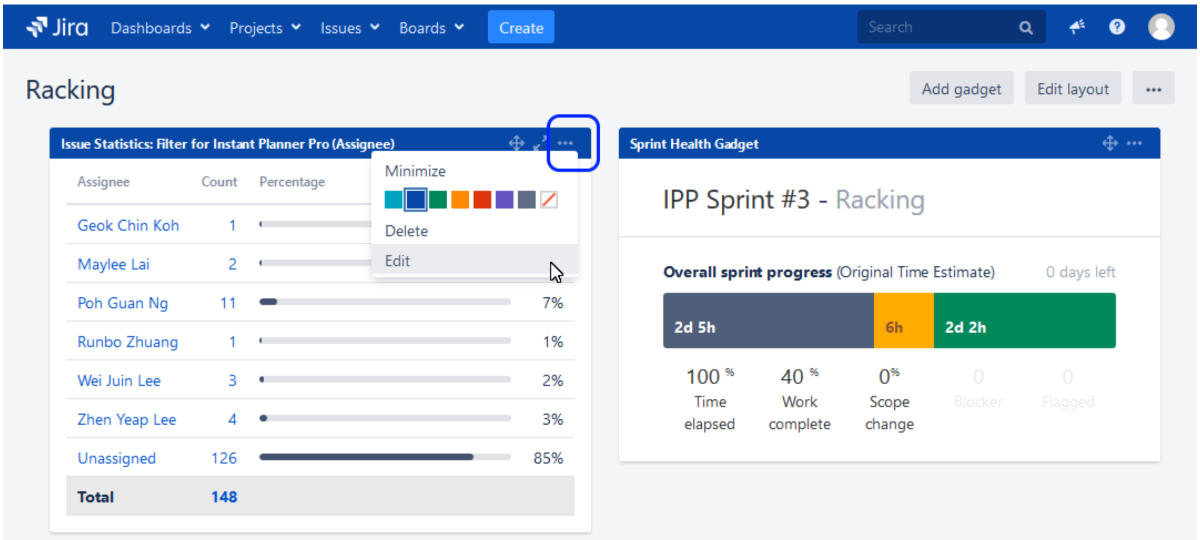
In the list, click the new dashboard that you created to open it, and then click the **Add gadget** button:



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:



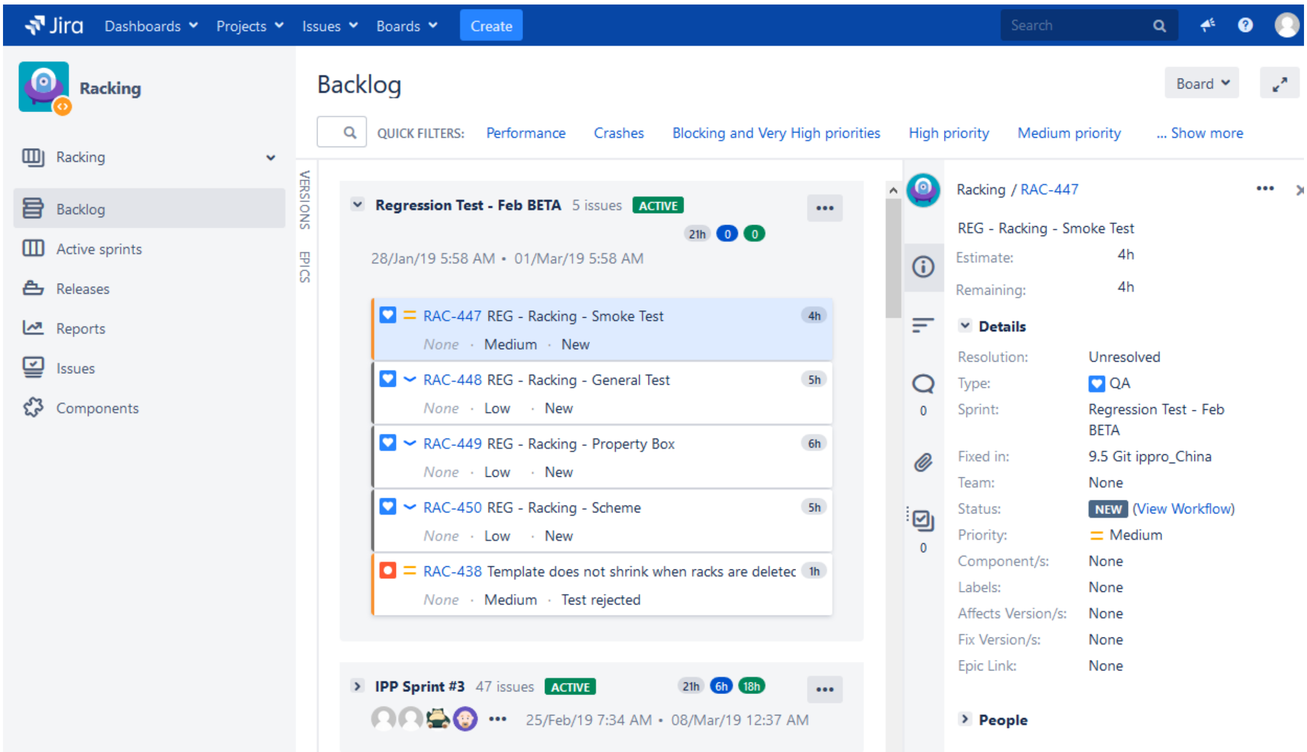
Backlog View

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:

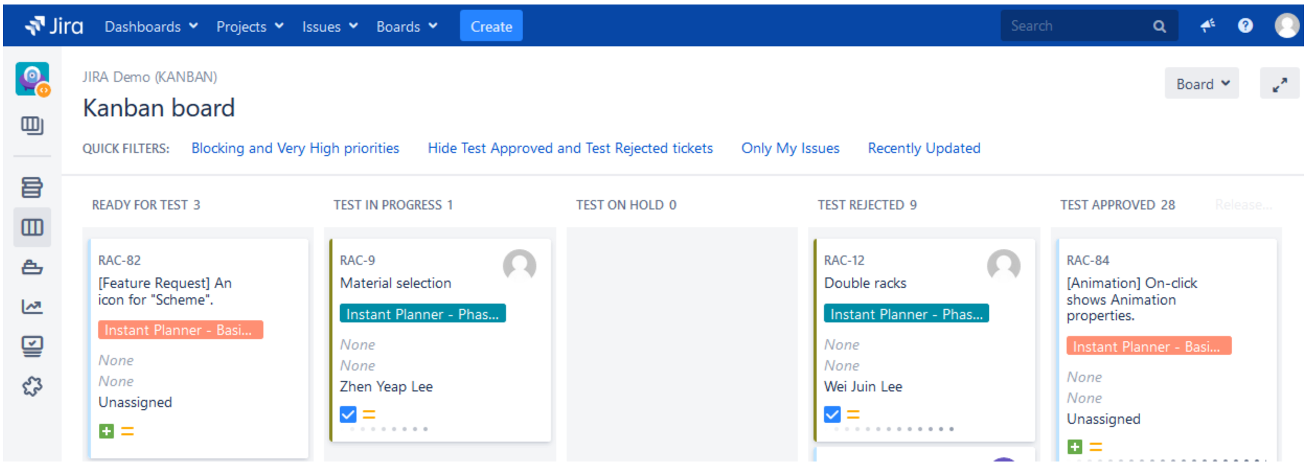


Board (Kanban) / Active Sprint (Scrum) View

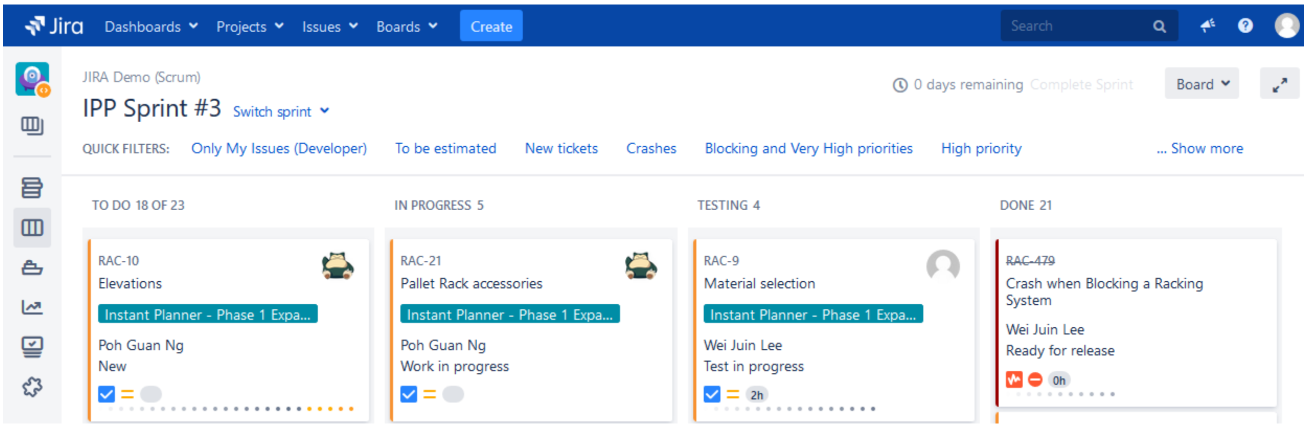
The **Board/Active Sprint** view is where the development team do their daily work. There can be multiple boards for different scrum teams or groups (currently set up by Configura), and a single board can cover multiple projects.

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)



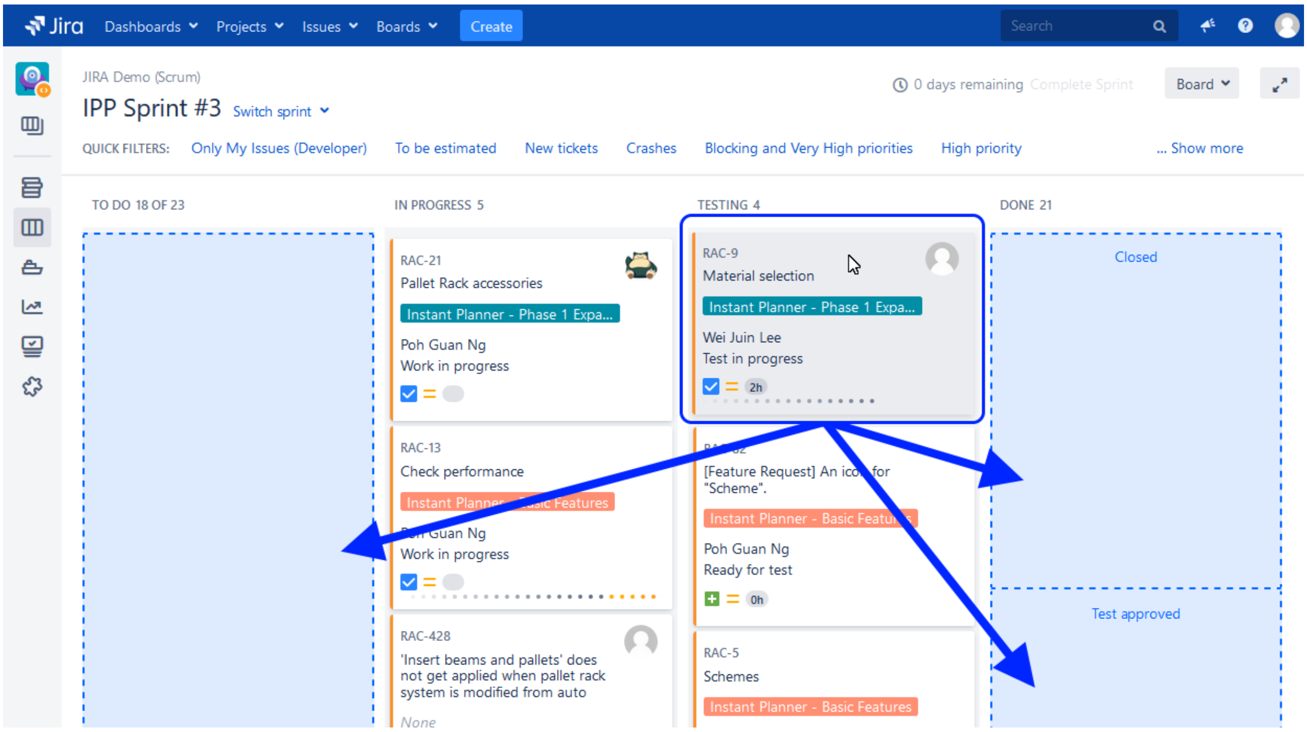
Scrum vs Kanban

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

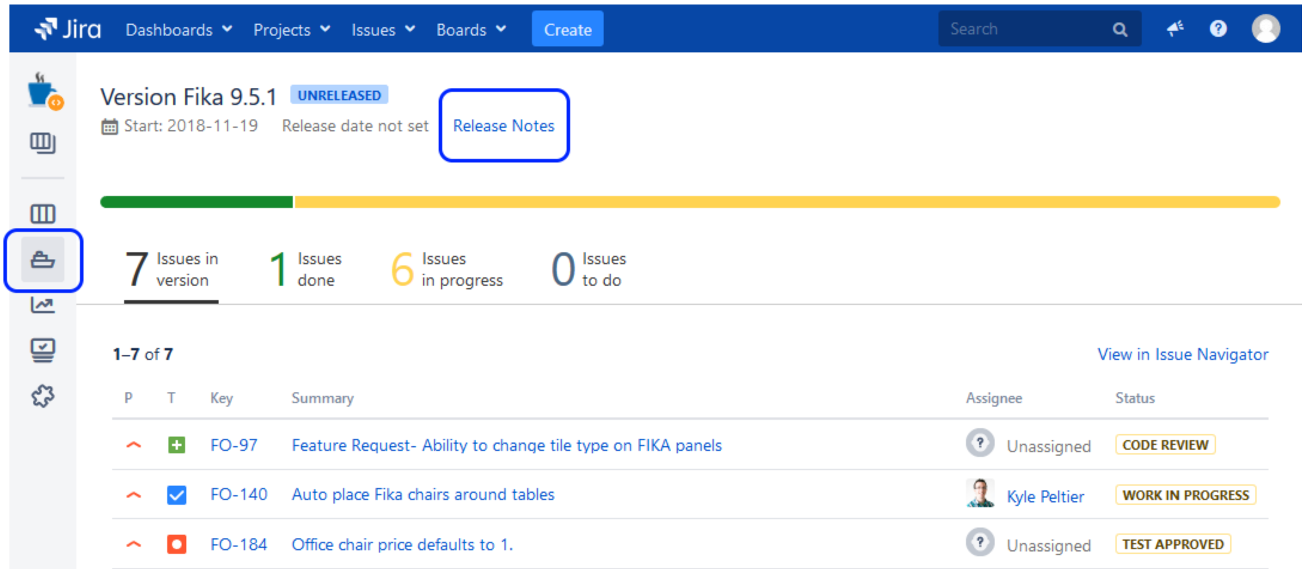
When working in the **Board/Active Sprint** view, issue status is changed by dragging the card from one column to another. When doing so, Jira will guide you to where the card may be moved:



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.

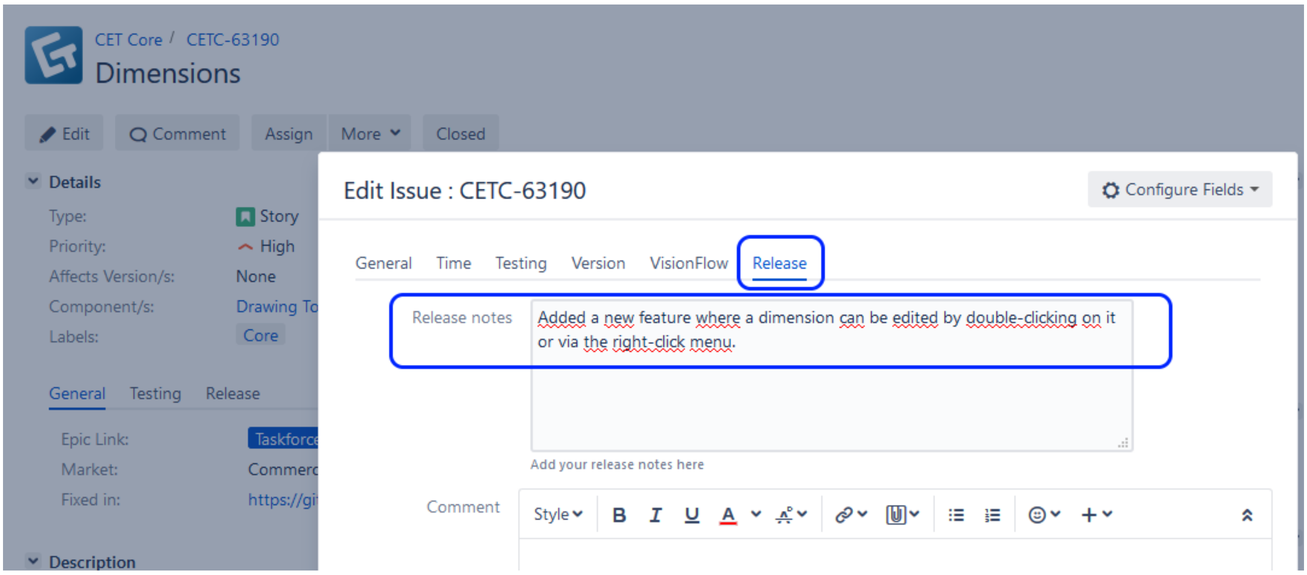
For each version, it is possible to generate a basic release notes draft* via the Release Notes links as highlighted in the screenshot below:



*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:



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

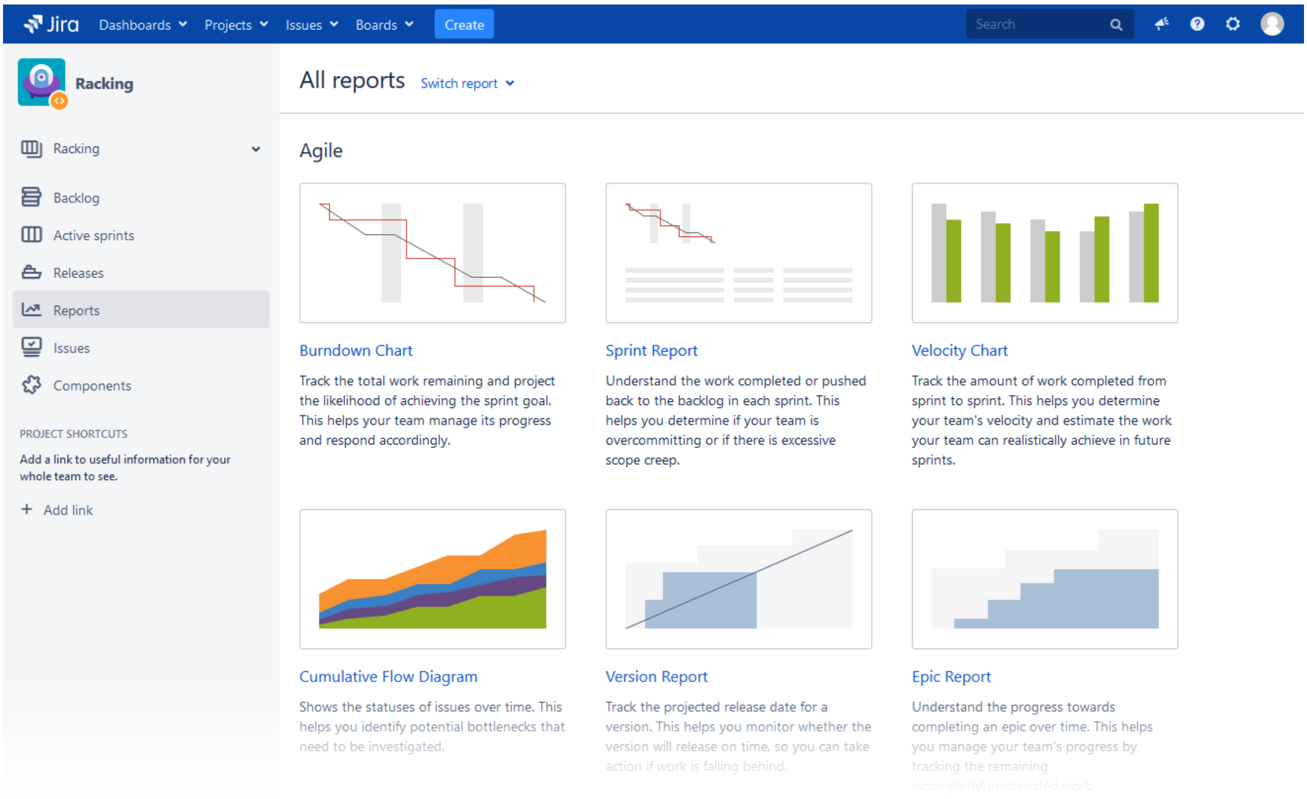
Edit/Copy Release Notes

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-63455] - Core Leader Item Tags
Tags.
    * [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.

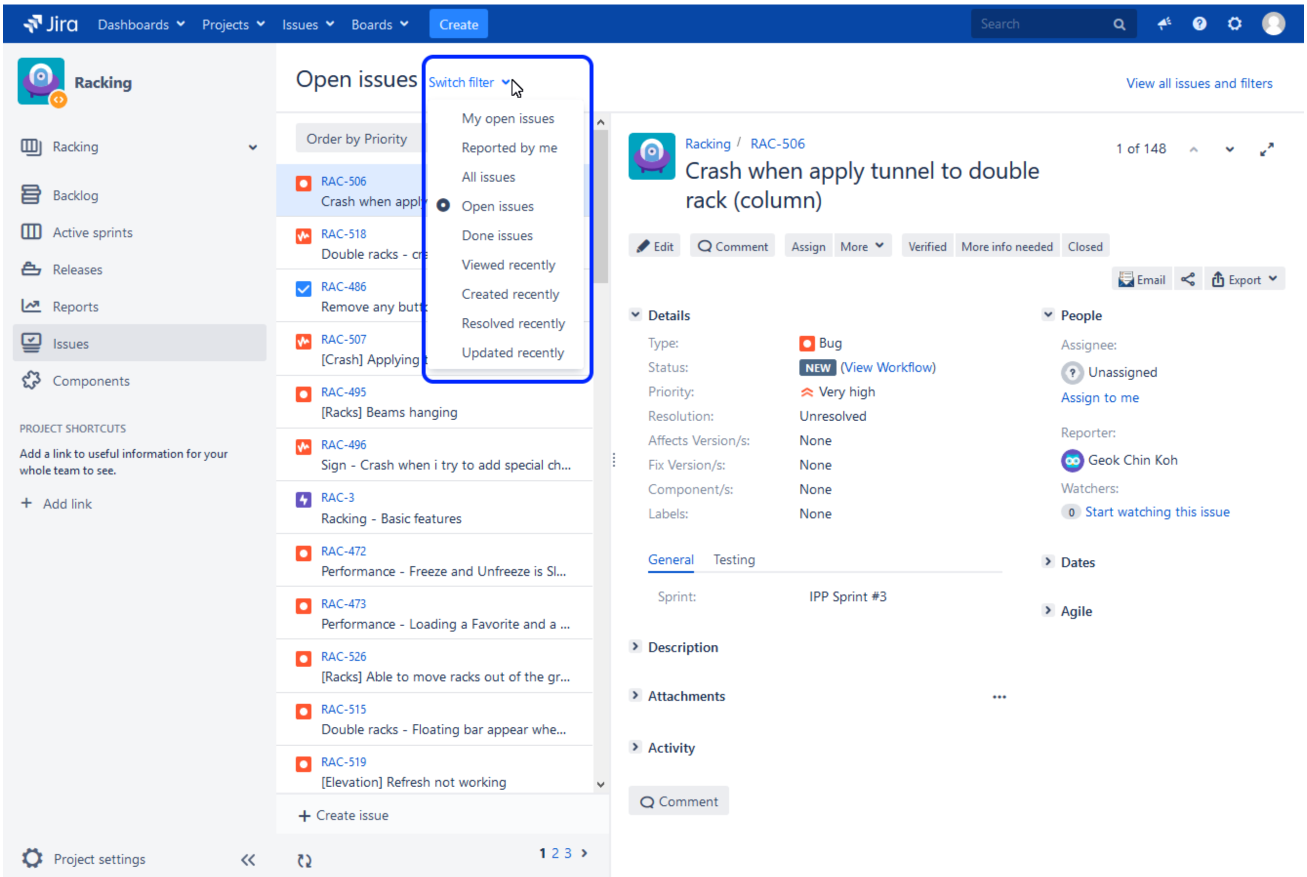


A few examples of available reports are:

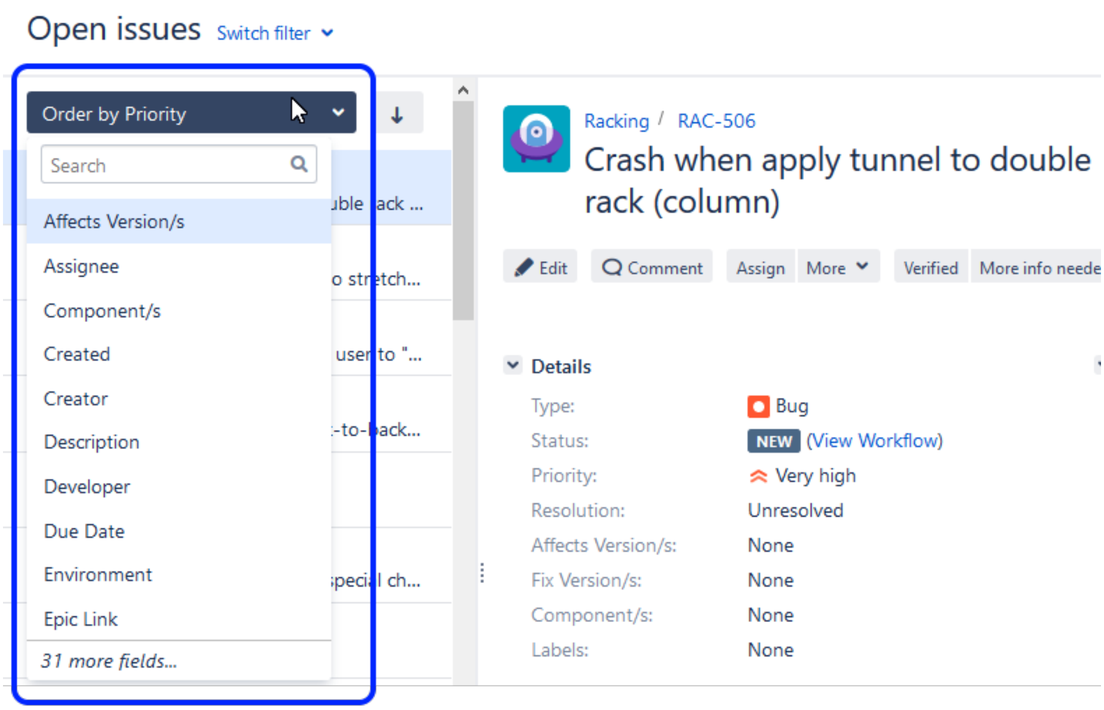
- **Sprint Report:** understand the work completed or pushed back to the backlog in each sprint.
- **Burndown Chart:** track the total work remaining and project the likelihood of achieving the sprint goal.
- **Velocity Chart:** track the amount of work completed from sprint to sprint.
- **Control Chart:** 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

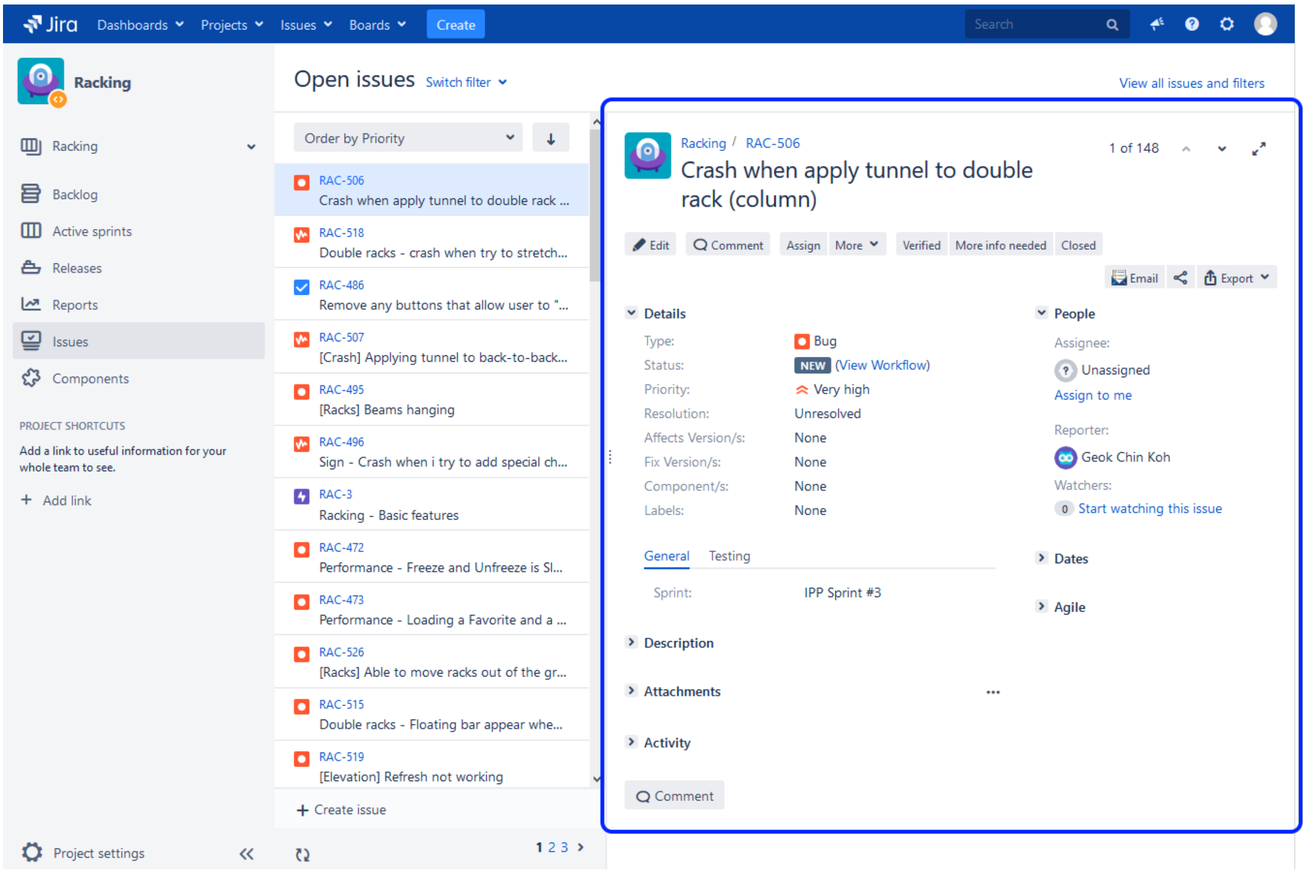
The **Issues** view lists all open issues in the board, but can be filtered to show only certain issues:



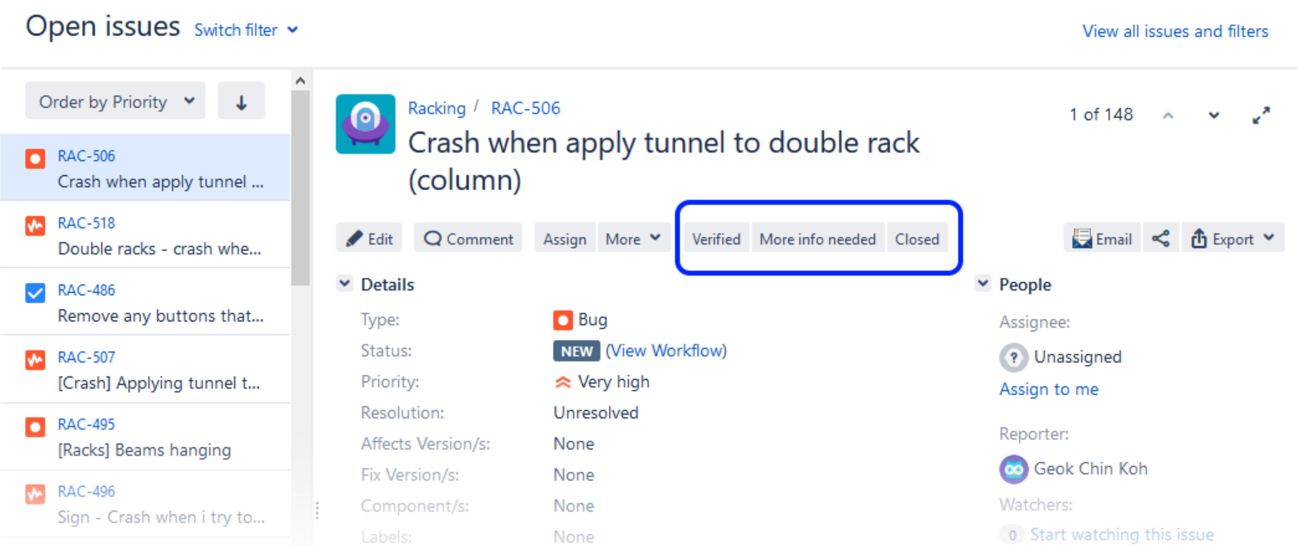
The list can be ordered in different ways:



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



The 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:

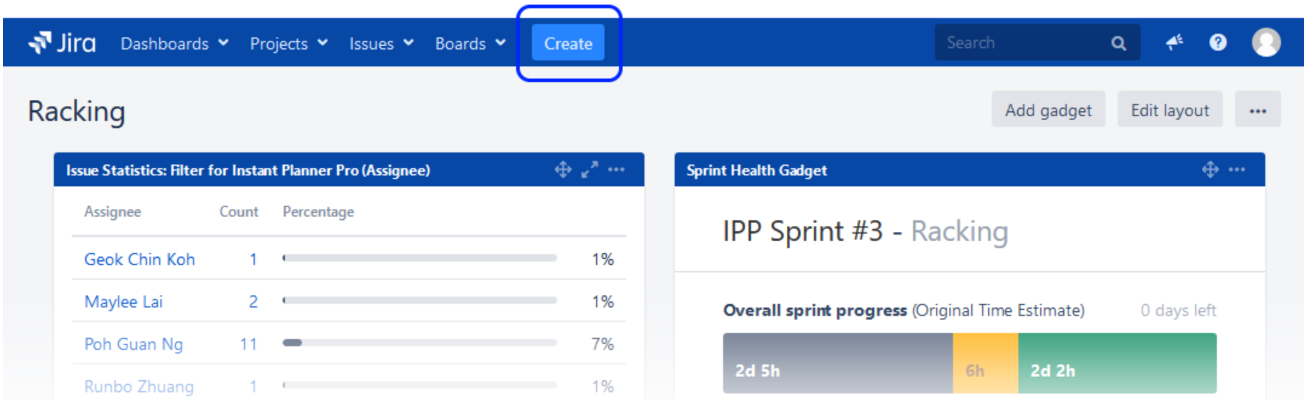


If you need to communicate directly with a user who does not have a Jira account, you can use the **Email** button available in the **Issue Detail** view. If the user replies to the email, the reply is added to the issue as a comment.

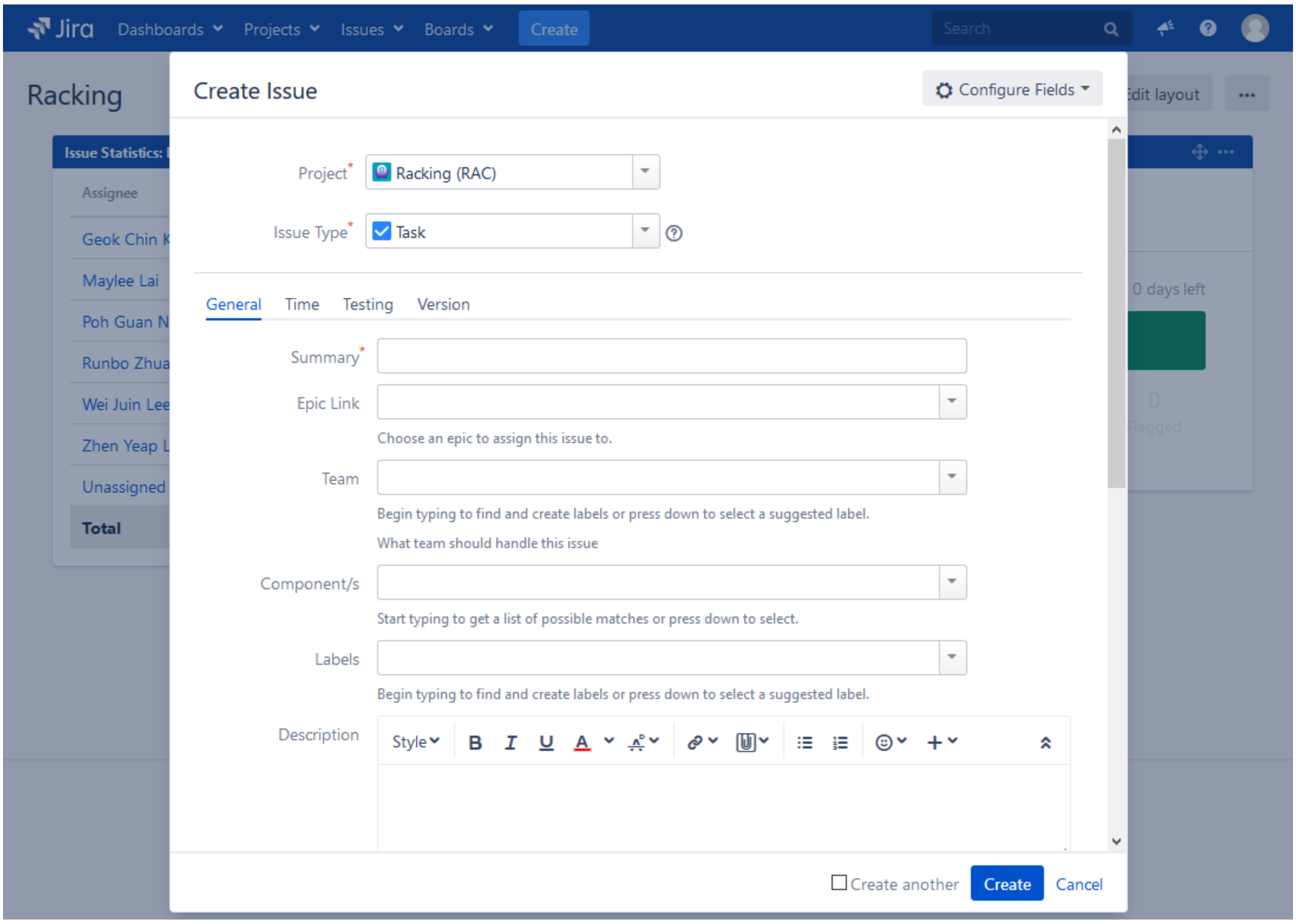

Open issues
Switch filter
View all issues and filters
Order by Priority
RAC-506
Crash when apply tunnel ...
RAC-518
Double racks - crash whe...
RAC-486
Remove any buttons that...
Racking / RAC-506
Crash when apply tunnel to double rack (column)
1 of 148
Edit
Comment
Assign
More
Verified
More info needed
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:



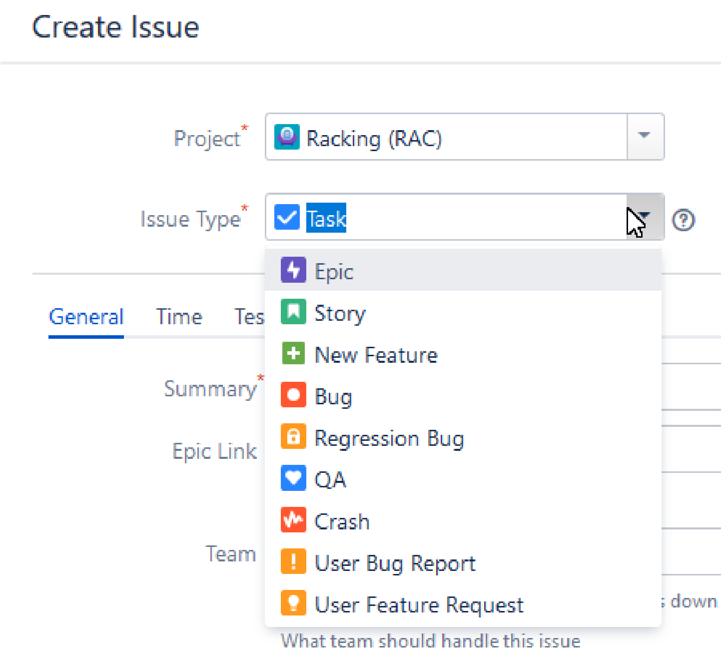
The template consists of many fields, and the more information that you include the better. The fields **Project**, **Issue Type** and **Summary** (a.k.a Title) are compulsory and cannot be left empty, and we also recommend to always add a **Description**:



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:

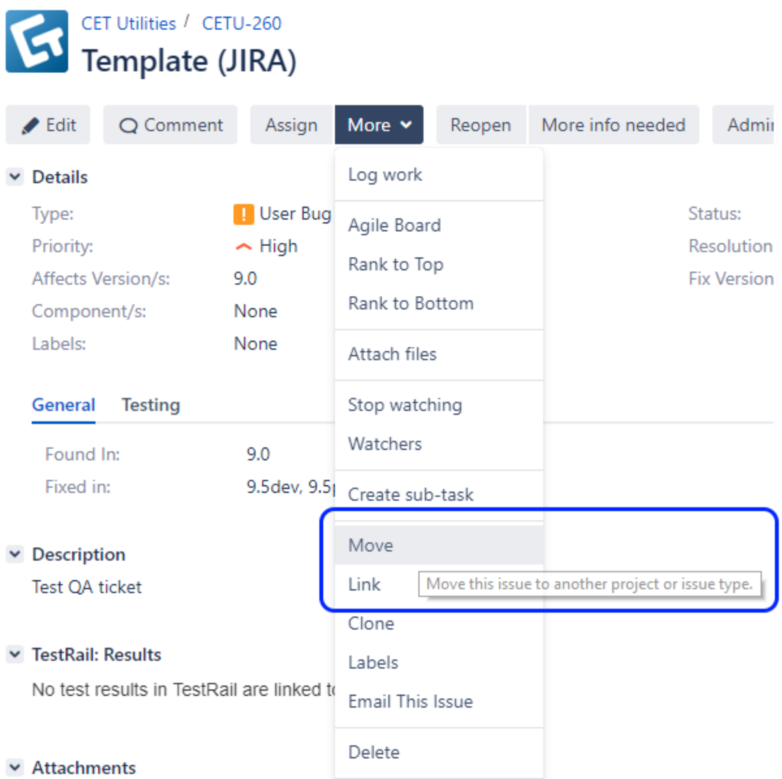


Epics can have Stories and Tasks linked to them while Stories and Tasks can have Sub-tasks.

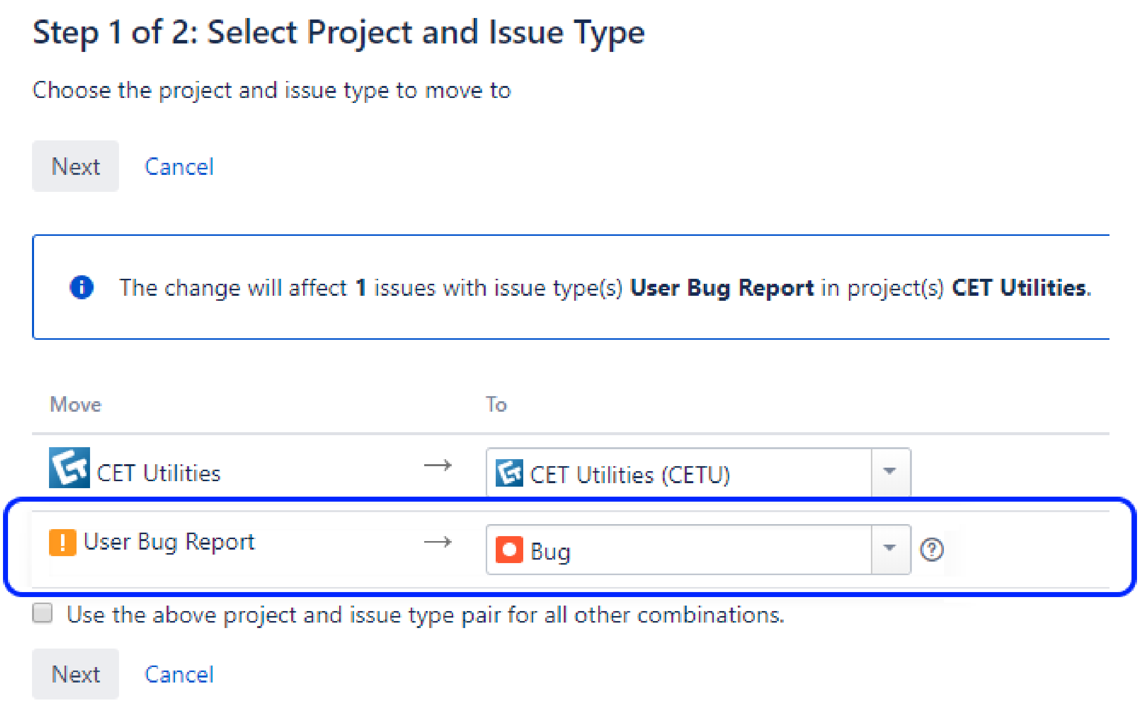
Please note that **User Bug Reports** and **User Feature Requests** always come in through support, and it is the responsibility of the Product Owner to review them, decide if action should be taken and, in that case, convert them to regular issues.

Moving an Issue

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



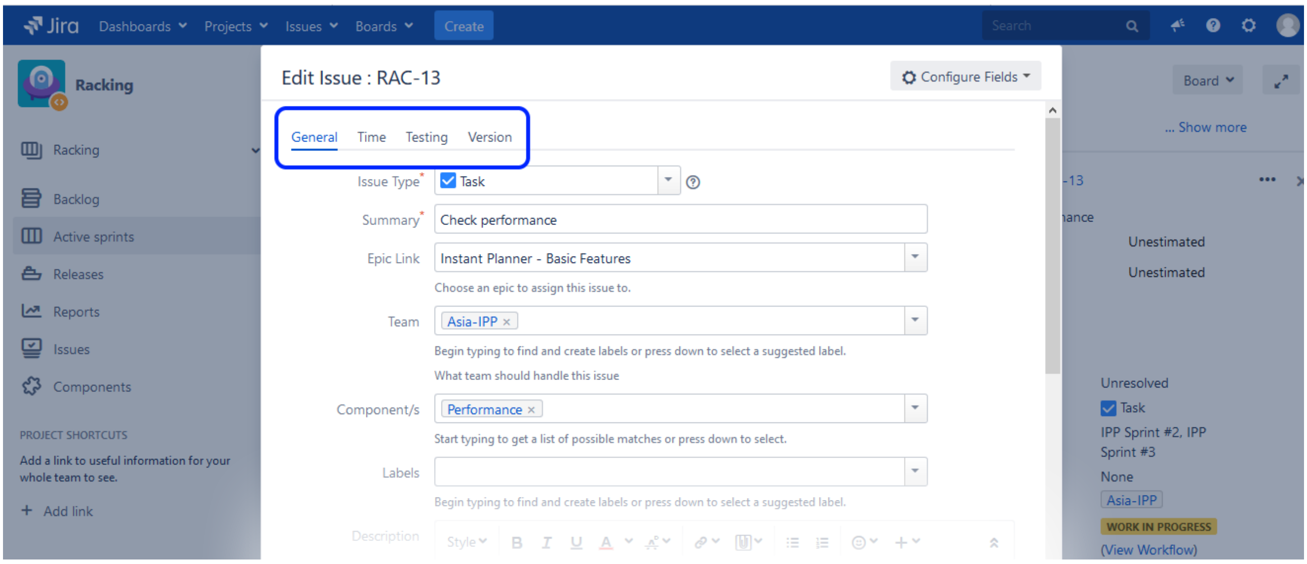
In the next step, move the User Bug Report to Bug (or User Feature Request to New Feature):



Editing an Issue

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

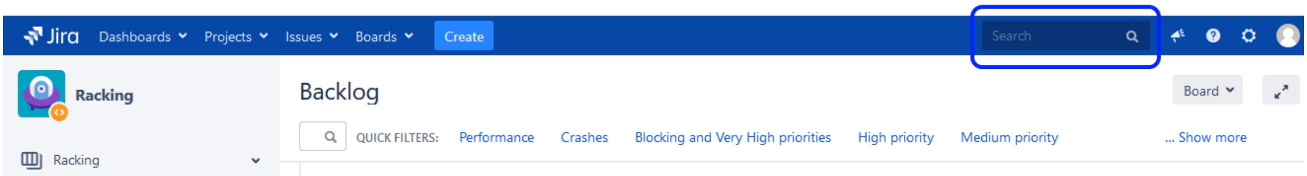
In the **Edit Issue** view, there are four tabs available for editing the issue:



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

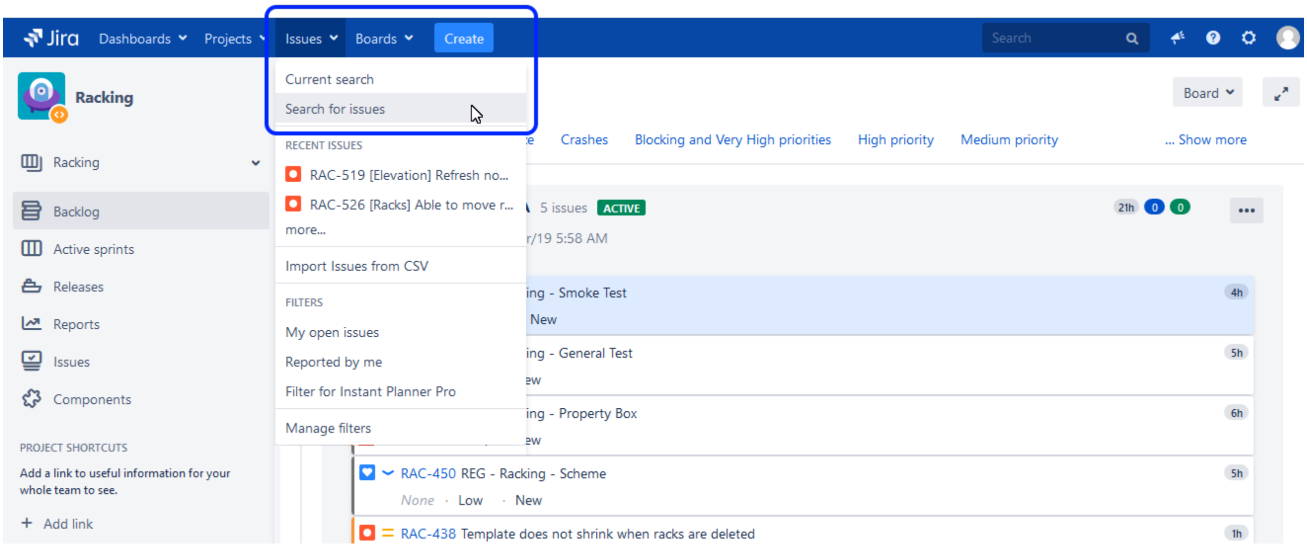
Searching for Issues

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



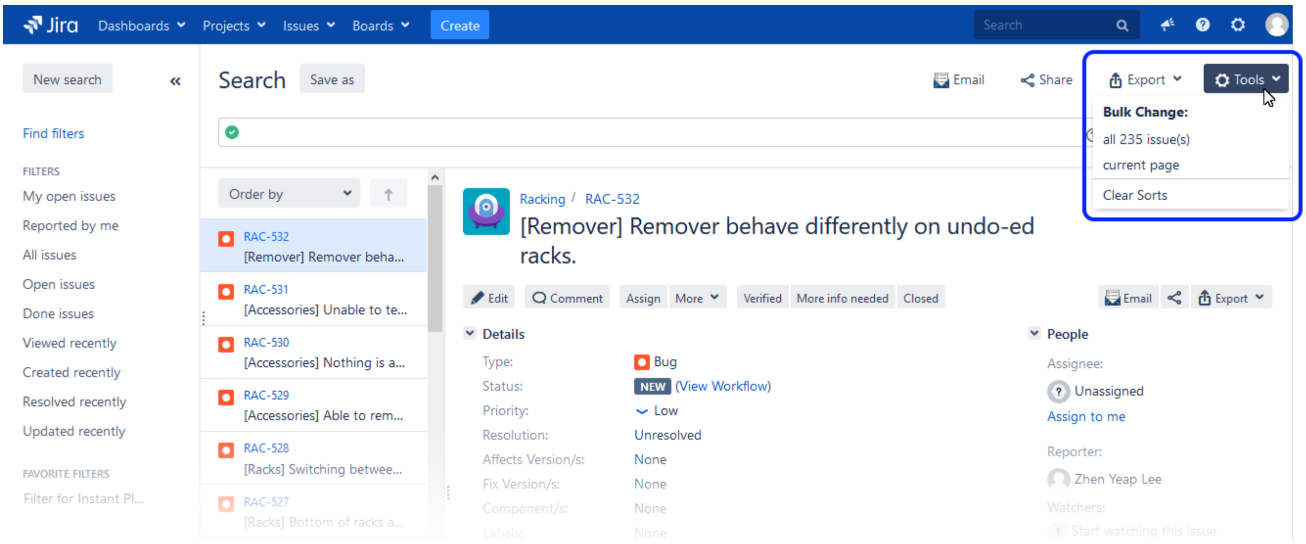
The search is dynamic, and the results will update as you continue typing.

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.

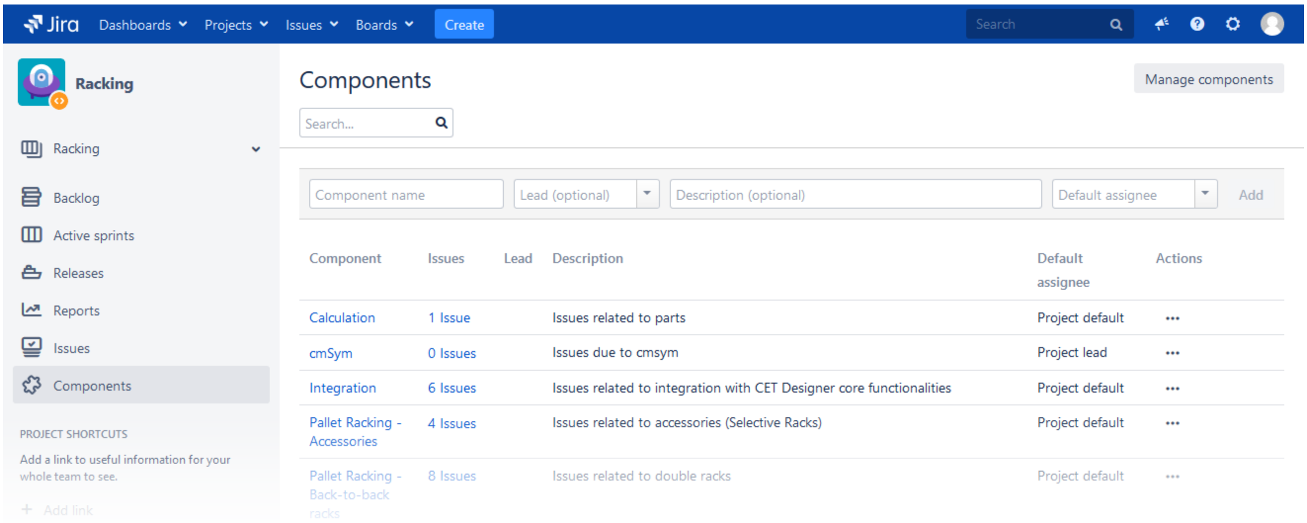
From the **Search** view, it is also possible to make bulk changes to multiple issues at once:



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:



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

Submitting issues from CET Designer directly to partner's project

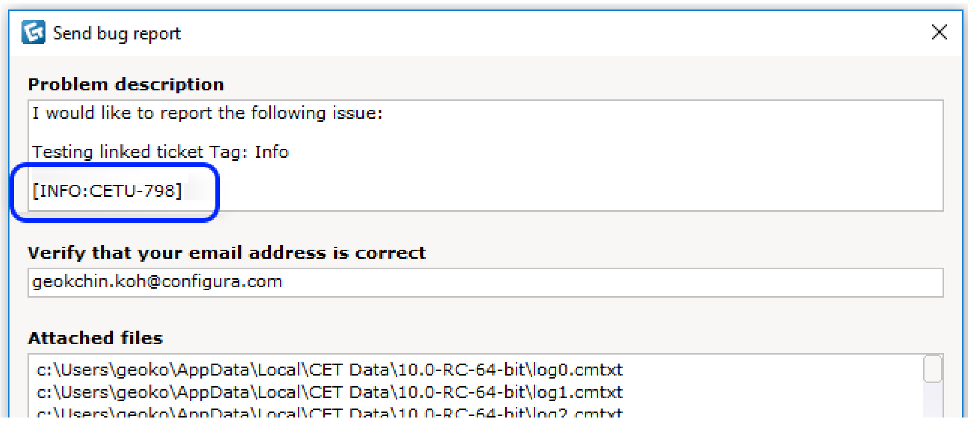
By default, all bug reports will be sent to the CET Support service desk, but there are cases where the partner's development team want to see their issues directly and not wait for the CET Support to escalate the issue.

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.

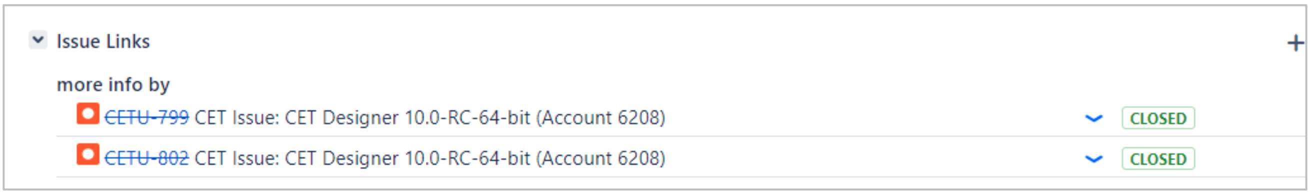
If you include this tag however, [INFO: <issue key>], it will:

- 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.



In Jira, it will look like this:



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 *Test in Progress (Dev)*, set QA field to current user.	
Waiting for build	When it is transitioned from *Test in Progress (Dev)*, 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 *Test approved*, 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 *Test in progress (dev)* or *Test in progress (build)*, 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 *Released*, change the issues to Released if the previous status is *Ready for Release*. Set all duplicate issues with *Closed* 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.	

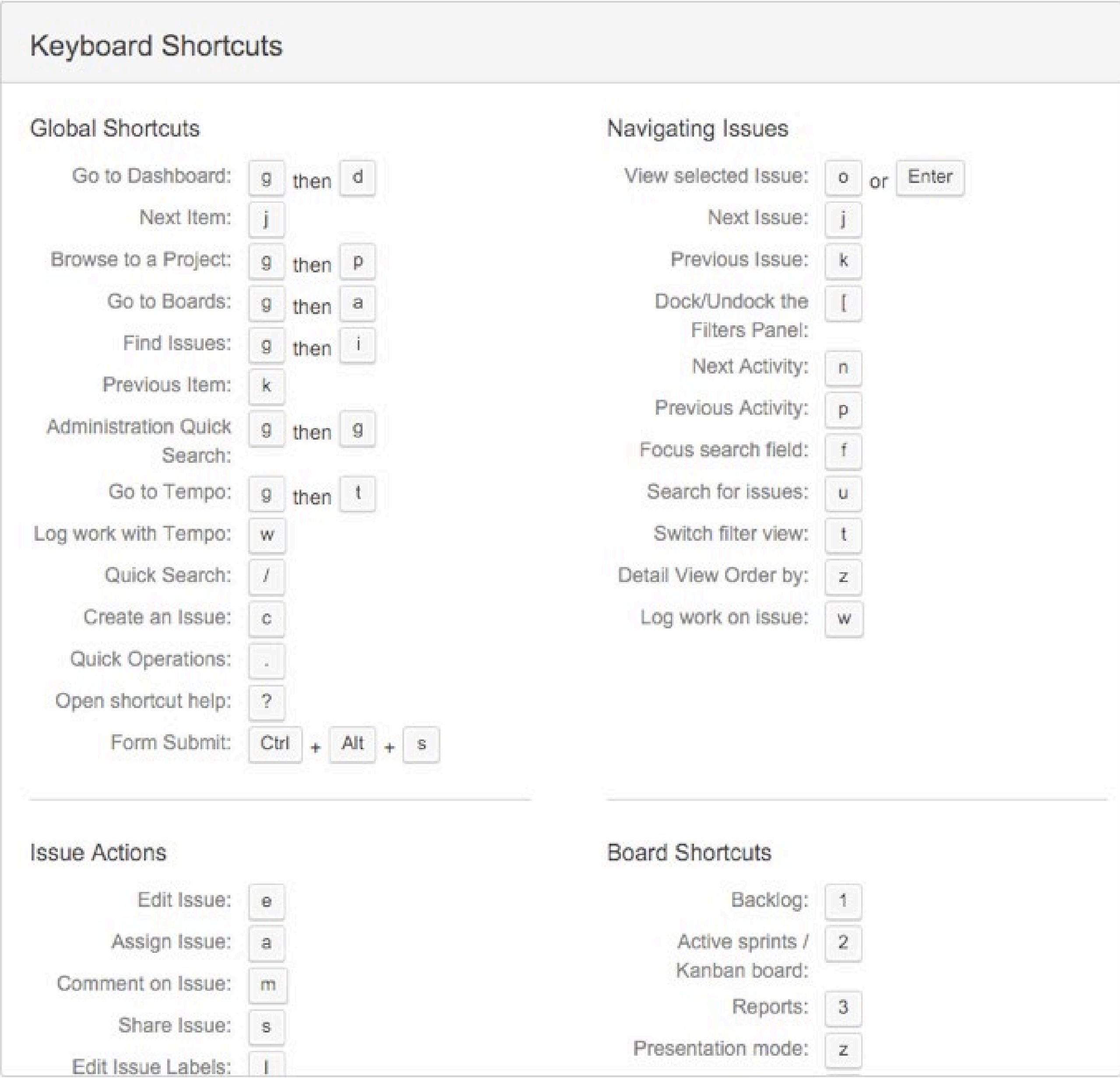
ACTION		
Add a comment with @support tag	If the previous status of the Jira Software ticket is **New** or **More info added (verification)** then it will be set to **More info needed (verification)**.	No changes for other statuses (Verified, Closed, etc.).
Edit priority	Add a comment “Escalated issue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



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