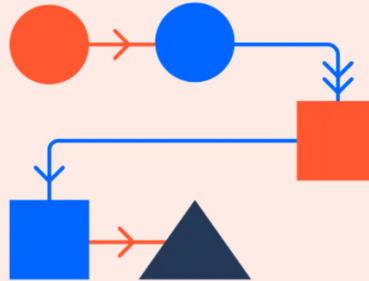


Season three of Atlassian's original podcast, *Teamistry*, is here! [Listen to episode one](#) now.

x



Building an awesome Jira workflow: concepts and examples

I'm amazed by how many different types of teams use Jira to manage their [workflows](#). Jira's customizable workflow engine lets you adapt it to all kinds of company cultures – whether small and new, or large and legacy.

As your company grows, you develop a unique culture based on how employees work together in teams. And how a team functions is reflected in its workflow: a repeatable process that makes it easier for people to work together at scale.

“

Without a doubt, customizing a killer workflow is the key to getting the most out of Jira.

[Tweet](#)

Much like trying on six pairs of shoes until you find the perfect fit, taking the time to customize your Jira workflow pays dividends down the line. So let's walk through a few concepts and lay out

PUBLISHED OCTOBER 16, 2013 IN
JIRA SOFTWARE



DAN RADIGAN

SUBSCRIBE TO WORK LIFE

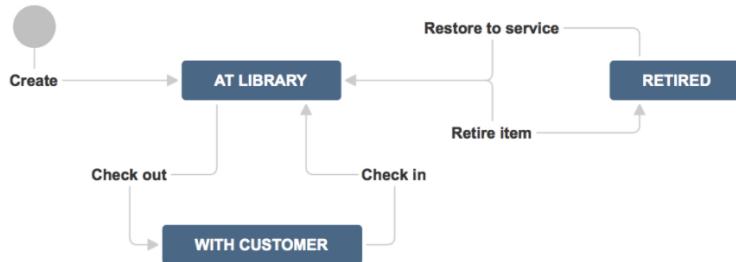
Get stories about tech
and teams in your
inbox

[Subscribe](#)

some best practices for designing a custom workflow for your team.

Anatomy of a Jira workflow

A workflow has four unique components: **statuses**, **transitions**, **assignees**, and **resolutions**. To illustrate them at a conceptual level, we'll use a well-known workflow example: that of a library. Each item a library lends out could be stored in Jira as an issue, and follow a simple workflow.



Statuses: where

Statuses indicate where an issue is within a workflow, and each status can be unique. (Your issues may flow through a given status multiple times, if you configure your transitions to be bi-directional – more on that in a moment.) In the library workflow example, a book can be in one of three places: at the library, with a customer, or retired from inventory.

Statuses come in two flavors: unresolved and resolved. At *Library* and *With Customer* would be considered open statuses, while *Retired* would be considered closed. Other common open statuses include *Open*, *In Progress*, *In Review*, *Scheduled*, *Waiting on Customer*, and *Pending Analysis*. Common closed statuses include *Shipped*, *Published*, *Fixed*, *Analysis Complete*, and *Closed*.

Transitions: what

Transitions reflect the action being taken and dictate the way an issue moves from status to status – sort of like a road connecting two cities – and can be one-way or bi-directional. You can allow issues to transition into a given status from any other status, which lets you design workflows for free-flowing processes where there are several possible next steps for an issue. Or, you can set up transitions that keep issues flowing along a tightly defined pathway to reflect more regimented processes (such as in our library workflow example).

Jira also supports workflow permissions that allow only certain people to transition an issue. At the library, books are loaned out by librarians, returned by customers, and evaluated by library staff to see if they're fit for circulation. Note that customers cannot evaluate if an item is fit for inventory – that must be decided by the library staff.

decided by the library staff.

Assignees: who

The assignee indicates the responsible party for an issue. In our library example, we would set the assignee to the borrower every time a book is loaned out so the library knows which customer has that book. When the book is returned, we would remove the assignee (in other words, set it to *Unassigned*) when we put the book back on its shelf.

Resolutions: why

Resolutions explain why an issue transitioned from an open status to a closed one. The act of retiring a book removes it from inventory (reflected by transitioning its issue to *Retired*, a closed state). So we can put a resolution on that book. We might set up resolutions like *Damaged*, *Out of Date*, or *Lost*.

Workflow “gotchas” to avoid

There are two common mistakes first-time workflow designers make.

Confusing resolution with status

Status describes where an item is in the workflow; resolution explains why an item is not in flight anymore. Think about *Retired* vs. *Damaged*. (People sometimes confuse resolution with transition, too – think *Damaged* vs. *Retire Item*.)

Pro-tip: Searching for everything that's *Retired* (a resolved status), as well as why it's been retired, provides granular reporting options. Many teams review resolved issues to ensure they are resolved for the right reasons.

Setting resolutions incorrectly

You must set a resolution when an issue moves from an unresolved state to a resolved state. Likewise, resolution needs to be removed if an issue flows back into an unresolved state. If our librarian decides to put a book back into circulation (i.e., taking an issue from *Retired* to *At Library* via the *Restore to Service* transition), they need to remove the resolution from that book. You can configure Jira to do this automatically using a transition post function, which we'll talk more about below.

Creating the best workflow

I've spent 10 years building workflows for many different types of teams. Here are a few pieces of hard-learned advice.

Gather all of your stakeholders

Workflow is about scaling



culture, and culture is about people. Whenever it comes time to build a process around a set of people, identify all of the

stakeholders in that workflow. For example, if you're trying to build a workflow that'll be used by product management, software development, and support, make sure you have one representative from each role in the meeting.

As the person designing the workflow, spend time talking with each stakeholder about what's important to them. Draw a draft workflow on the whiteboard, then walk through each case and gather their feedback.

Workflows are touchy because they govern how people work. Be patient, and your investment will pay off.

Keep your Jira workflow simple

Stakeholders often want to have statuses for each part of the workflow. That's generally a good thing, but remember: each status adds more transitions and complexity. Aim for simple and scalable instead.

Whenever adding a new status to a workflow, make sure you have no other option. Let's look at two examples.

- Code review is an important part of the software development process. Jane, the development manager, wants to add a specific status called *Code Review* so it's clear to her team which issues are under active development, and which issues are awaiting review. Reviewing code is distinctly different than writing code, so it makes sense to add a new state.
- Bill, the QA manager, wants to add new status called *Failed Verification* for all issues that don't pass review by his team. I'd advise against doing this, as the test engineers can simply send any issue that fails review back to a previous state, such as *In Progress* or *In Development*.

“

As a Jira administrator, you owe it to yourself and your users to keep workflows simple.

[Tweet](#)

//

Make every transition count

Jira has a number of options admins can take advantage of when setting up transitions:

- **Conditions** – [Conditions](#) control who can perform a transition. For example, only someone in the library can retire a book from



RELATED ARTICLE

[Discover your natural leadership style with this quiz](#)

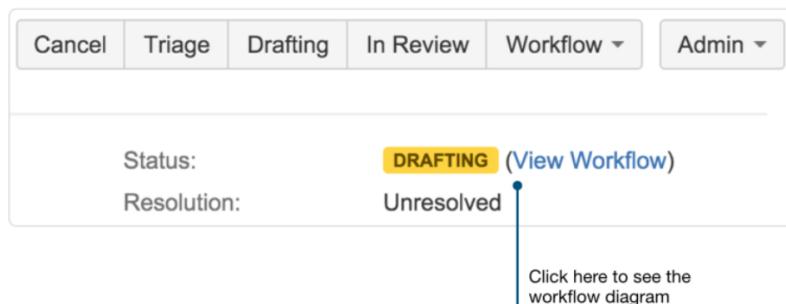
BY SARAH GOFF-DUPONT IN QUIZZES

inventory (i.e. move it from *At Library* to *Retired*). If anyone else besides the librarian is logged into Jira, the option to retire an item won't even be shown.

- **Validators** – Validators make sure a transition happens only when certain criteria are met. For example, if a book is to be retired because it's allegedly damaged by a customer, we'd like to ensure the book has been checked out at least once to verify that claim. If an issue doesn't meet the validators' criteria, the transition will be aborted.
- **Post functions** – Post functions perform additional actions on issues (besides moving it to a different status). The most common post function is to clear the resolution when transitioning an issue back to an unresolved state. In the library example, we would use a post function to clear the reason a book was retired when it got put back into service. We could also have Jira set the assignee to *Unassigned* when a customer returns a book. And speaking of...
- **Assignees** – When a transition occurs, it often means that the issue has a new owner as well. You can change assignee automatically using a post function, or surface a transition screen where the person transitioning the issue chooses who to reassign it to.
- **Properties** – Jira recognizes some properties on transitions. The most common one is to limit the resolutions displayed to the user on a given transition. For example, we might want the resolution *Scratched* to show up when retiring disk media, but not when retiring books. (Note that this example assumes you have distinct issue types for books and disc media. But don't worry: multiple issue types can share the same workflow.)

Nail the workflow diagram

Once you've built the workflow, use the [workflow designer](#) to make sure the diagram is presentation quality. You'll want to share the diagram with all your stakeholders for a final round of feedback. You should also enable the option in Jira to show where the current issue is in the workflow. All users have to do is click *View Workflow*.



Be build the change you seek

I said in the beginning that workflow is a fundamental aspect of succeeding with Jira. Listen to your customers, have empathy, and help them understand why what they're doing might be inefficient – and how new ways of working can help them scale. When people work well together they have more fun and get

more done. You don't want to be a part of that! In my next article, we'll see how Jira workflows help your team visualize data. I'll show you [five steps to configure your workflows](#) so you can put all these ideas into practice. See you there!

Workflows are only the beginning. If you haven't tried Jira yet, [check out our product tour](#) to see what else it can do for you and your team.

Ready for more Jira tips and tricks? Click below for more tips and best practices blogs.

[Bring on the Jira tips!](#)

Did you find this post helpful? Share it on your social network of choice and help your fellow Jira users create an awesome workflow!

ABOUT THIS ARTICLE

Published October 16, 2013

ABOUT THE AUTHOR

DAN RADIGAN

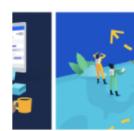


RELATED CONTENT

More in 'Jira Software'



JIRA SOFTWARE
Introducing flexible terminology for Jira Software Data Center



JIRA SOFTWARE
Advanced roadmaps is now part of Jira Software Data Center



JIRA SOFTWARE
JQL: the most flexible way to search Jira



CONFLUENCE
Link Jira issues to Confluence pages automatically

MORE COLLECTIONS

COLLECTION

Bring Your Full Self to Work

Teams are made up of individuals, and they function best when each member is happy, healthy, and productive. These stories will show

COLLECTION

Work Life Balance

Work life balance: everyone wants it, few know how to attain it. But once you do, you'll be amazed as the stress of work and life melt away, your

COLLECTION

The Future of Work

Whether you dread what the future holds for workers or embrace it with open arms, there's a lot to know and discover. We report on industry trends