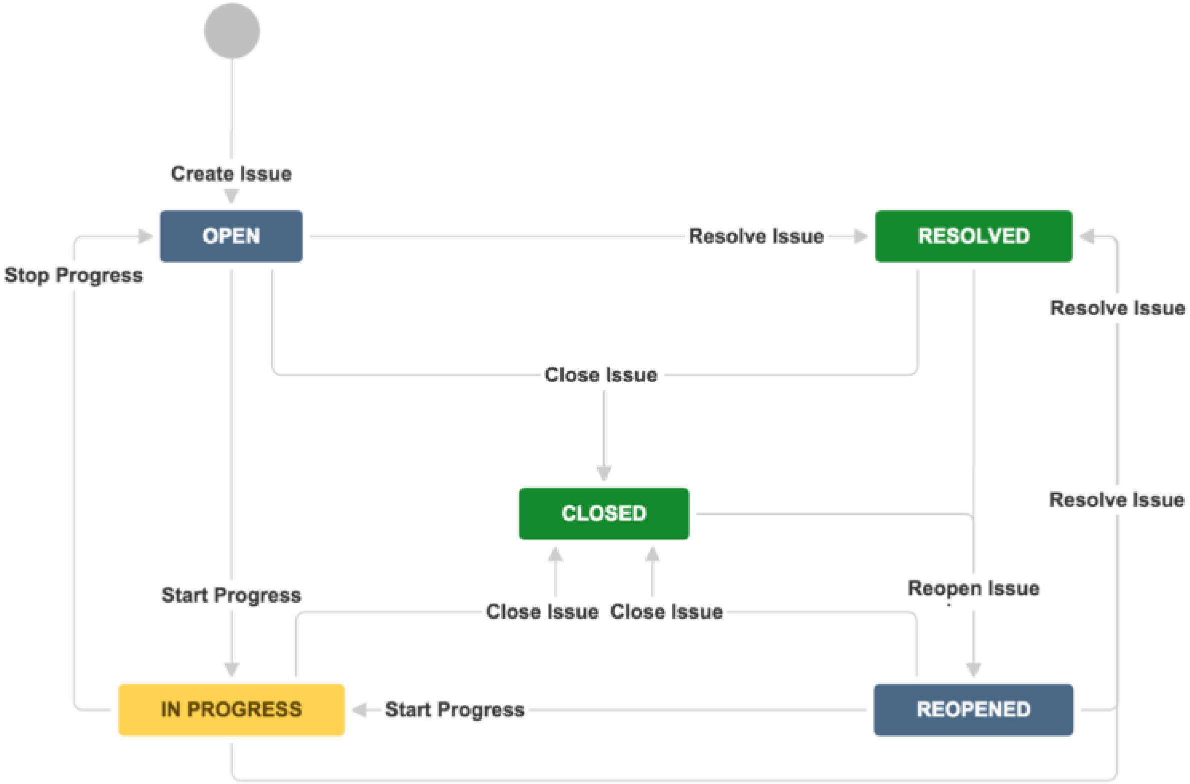
A guide to Jira workflow best practices (with examples)

June 4, 2018



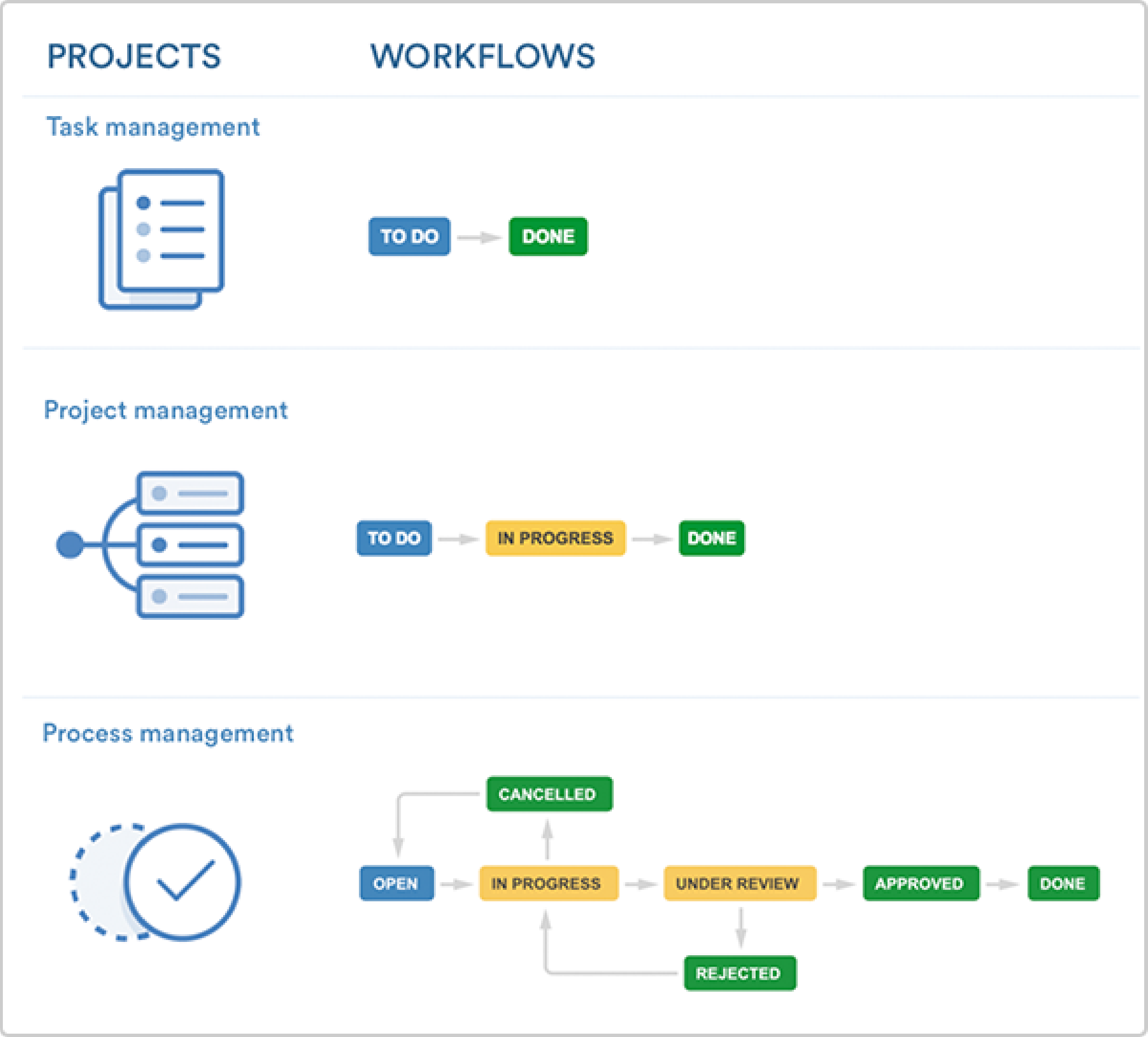
In this guide I'll explore with you the power of workflows in Jira. I'll cover Jira workflow best practices as well as dig into some practical examples.

I'll also discuss exactly why you would want to create a new Jira workflow, customized to your teams' processes and how you would go about doing so.

With Jira you can really make the system work to your needs, precisely suiting your team's processes and requirements. You can make it *yours*. And the way to do that is by building these workflow schemes.

So without further ado, let's get into workflows for Jira:

What are Jira workflows?



Workflows in Jira model your organizational processes and allow you to progress tasks in the system.

In an extremely basic workflow, that might mean:

1. creating an Issue, to reflect a new task. This will have a "To Do" status.
2. marking it as "in progress" once work is begun.

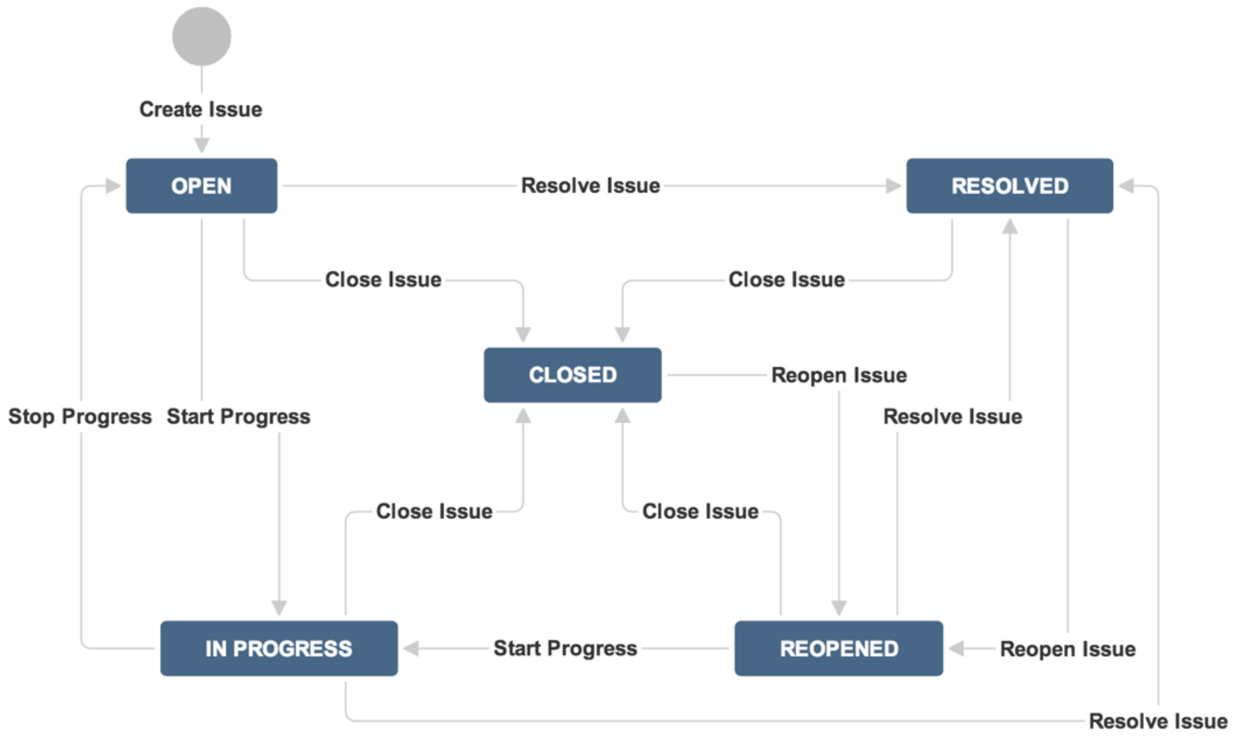
3. Then, when the task is finished, you can mark it as being "Done". And you've already gone from one end of the workflow to the other.

Jira has its own workflows built-in. These include:

- Task management (https://confluence.atlassian.com/jiracorecloud/task-management-803373915.html): The simplest possible workflow, to get tasks done as soon as possible.
- Project management (https://www.atlassian.com/project-management-software): A very slightly more complex workflow, which includes an "In Progress" status to better mark work on the task.
- Process management (https://confluence.atlassian.com/jiracorecloud/process-management-803373920.html): A structure that comes with multiple statuses and resolutions, which begins to reflect the complexity of business and development processes.

However, it might be a better choice to create your own custom Jira workflow.

Why create a new Jira workflow



Creating a new Jira workflow allows you to adapt the system to exactly how your team works.

While you might use Jira's standard workflow to be able to simply move an Issue from "Open," to "In Progress" to "Complete," things are often more complicated in the real world.

For example, if work needs to be approved then you may need to add additional statuses to reflect "Awaiting Approval," "Review In Progress" and "Review Complete".

Another reason to customize workflows is the diversity of teams. All kinds of teams use Jira – from customer support to HR to developers. Separate teams might also naturally benefit from having their own work patterns and requirements built into Jira.

Just make sure to not overdo it, as explained in our blogpost on the worst (common) advice for Jira (https://www.idalko.com/worst-common-advice-for-jira/).

So, when considering customization, it's important to think about precisely what you're doing and why.

Clear communication is essential. If you can use existing elements and features with minimal hassle, then it might present the simplest way to get ahead.

If it means that the system simply won't reflect the way that you work, though, then it may actually be essential to deploy customization. And you could see a big productivity boost as a result.

Role of Jira workflows for teams

Jira enables users to seamlessly log tasks and information.

There's no need to drop a colleague an email to let them know that a task has become available for them. It's all tracked and logged automatically.

This way you'll be mapping how tasks are processed and dealt with while wiping out redundancy.

Of course, if you're using Jira for other purposes, such as a CV database, then workflows may be even more useful. Allowing you to move Issues around the system exactly as you need.

Benefits and drawbacks to changing workflows

Creating Jira workflows to match the way you work comes with the benefit of making Jira far more useful for you and your team.

This will allow you to ensure that everything is correctly logged on the system. Hence offering new insights into exactly how your processes function.

The drawbacks however are:

- If you do want to deploy customization then your admin will need to build and maintain the changes. That requires time and resources. Overdoing it is probably not a good idea.
- They'll also have to ensure that the instance continues to function correctly (including with apps and other integrations). Which can be quite challenging if you add too many customizations.
- A third drawback of over-customization is communication disconnect between teams. If many teams start using different workflows and naming conventions, it might be pretty confusing when these teams have to start collaborating. Or when members migrate to new teams.

The challenges of getting people to use Jira workflows in a structured manner

When it comes to workflows, one of the biggest challenges is ensuring that users work in a structured manner and make proper use of the system.

That might mean that once an individual starts working on an Issue, for example, they need to mark it as being "In Progress."

If they don't make status changes as they go, then Jira won't correctly indicate the team's actual Issue flow. And there won't be a record that the Jira user is working on the item at the time.

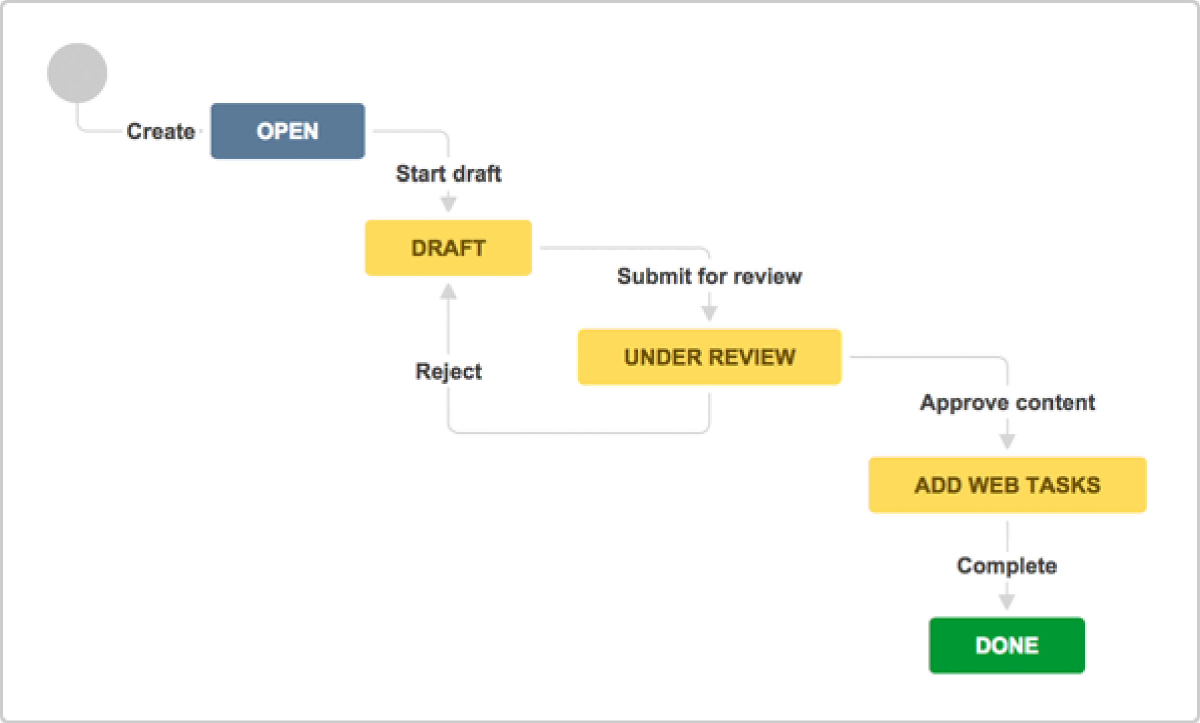
This then has a knock-on effect that there is a partial record of the team's activities. So Jira will be far less useful than it should be.

The solution is to ensure that everyone on the team makes a habit of adjusting Issue statuses whenever required. That way you'll be using workflows as they are intended and keep the system up to date.

Of course, part of the effectiveness of these workflows is also about ensuring that there is a clear and intuitive workflow in place that Jira users can understand.

To ensure teams are progressing through the workflows in Jira correctly, it's definitely a good idea to audit the process on a regular basis.

The elements of Jira workflows



There are four basic building blocks for a Jira workflow – statuses, transitions, assignees and resolutions. This answers the question of who is doing what, where they're doing it and what needs to happen next.

The workflow status

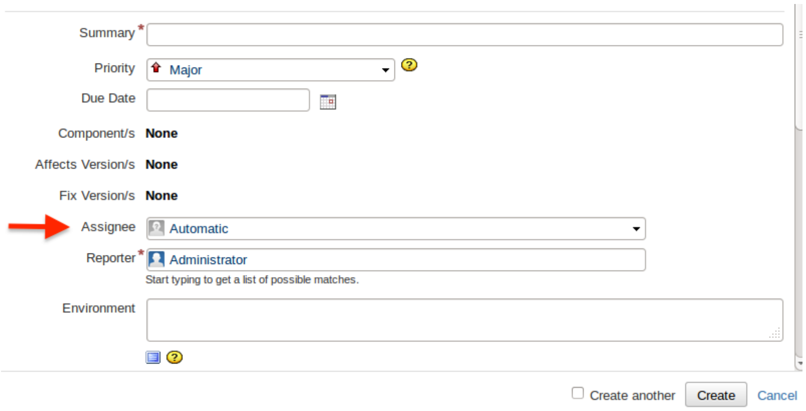
An Issue can only have one status at a time. It might be something like “In Progress” or “Complete”.

However when looking at new statuses, it’s worth considering whether a status represents an activity or a stage.

For example, the status “Under Review” might represent a stage which actually comprises of several different actions.

It may be tempting to split these actions out. But doing so will inevitably create a much more complex workflow. Which is less transferable between teams and departments.

The Jira assignee



Summary*

Priority: Major

Due Date

Component/s: None

Affects Version/s: None

Fix Version/s: None

Assignee: Automatic

Reporter*: Administrator

Start typing to get a list of possible matches.

Environment

This is who the Issue is assigned to and who is responsible for it. And it often changes between different statuses.

When the task is complete the assignee name can be removed.

There are multiple approaches to assigning issues in Jira. Which we discussed in our post Handing over issues to a new Jira assignee (the pitfalls and best practices) (https://www.idalko.com/hand-over-issues-new-jira-assignee/)

The transitions in a workflow

A transition is a one-way connection that joins two statuses. Between the status “In Progress” and “Under Review” the transition might be “Submit for Review.”

There may be conditions for who can make a transition and when.

For example, only a project admin might be able to move an Issue past the “Under Review” status.

Transitions can also be defined so as to notify specific parties when changes are made.

Ultimately, transitions lead towards a resolution.

Resolve an issue in a Jira workflow

The resolution is the final Issue state – such as “Fixed,” “Complete” or “Won’t Fix.”

If you ever want to reopen close items, you’ll need to ensure that you clear the resolution. If you don’t then the now-active issue will confusingly appear to be complete and its name will have a line through it.

How to create a new Jira workflow

To create a new Jira workflow you can either copy a default workflow or you can start from scratch.

A note of caution, the default workflows include various restrictions, such as not being able to edit resolved issues. That may mean that it’s actually easier to start with a clean slate.

To get down to work, you’ll want to use the Jira workflow designer – which you’ll need Jira Administrator permissions to use.

This tool allows you to edit the layout of your workflow and the progress path of statuses and transitions.

There are various entities to consider when creating a Jira workflow. They include:

- **Status** – where the issue is (for example, “In Progress” or “Under Review”)
- **Resolution** – why the Issue is no longer in flight (for example, because it’s completed)
- **Conditions** – control who can action a transition
- **Validators** – only allow transitions to occur when specific information is provided

- **Post Functions** – make additional changes to issues, alongside transitions (for example, removing a resolution when an issue is reopened)
- **Triggers** – automatically activating transitions when specific events take place. For example, moving an Issue from “In Progress” to “Under Review” when code is submitted for review
- **Workflow properties** – setting certain properties for transitions. For example, only displaying resolutions that are relevant to the specific Issue type
- **Workflow schemes** – determining the associations between a workflow and issue type

Jira apps for workflows

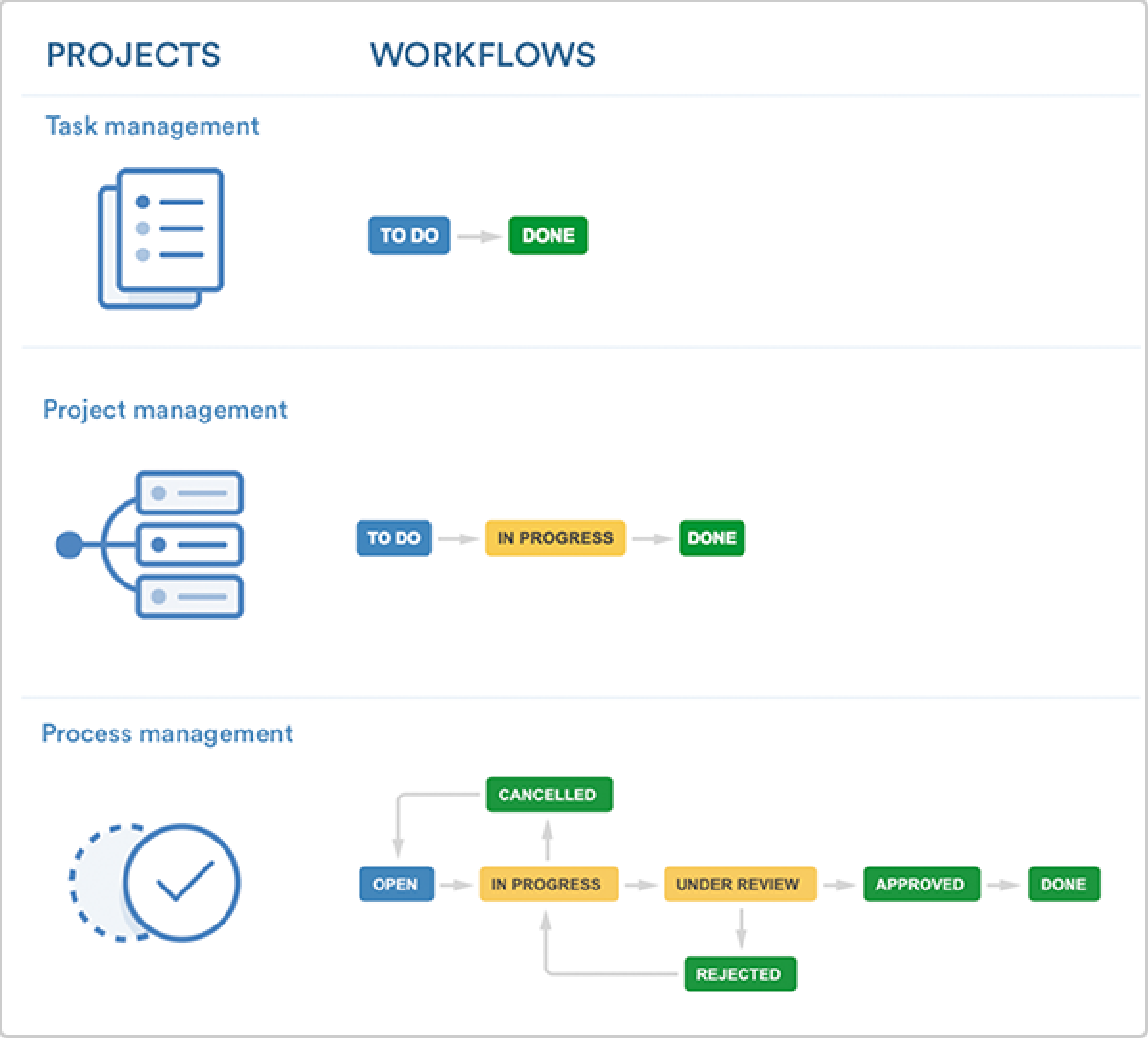
Jira apps can also assist in customizing workflows in order to get the effects that you want.

For example, you might want to prevent a Jira subtask from being reopened if the issue that it belongs to is resolved. Or to create an undo button so status changes can be reversed.

You can also use apps to create custom data fields.

To achieve this, you could either build what you need yourself or you could use an off-the-shelf app, such as AM Utils (https://marketplace.atlassian.com/apps/1211565/am-utils?hosting=server&tab=overview) or Insight (https://marketplace.atlassian.com/apps/1212137/insight-asset-management-for-jira?hosting=server&tab=overview).

Jira workflow examples

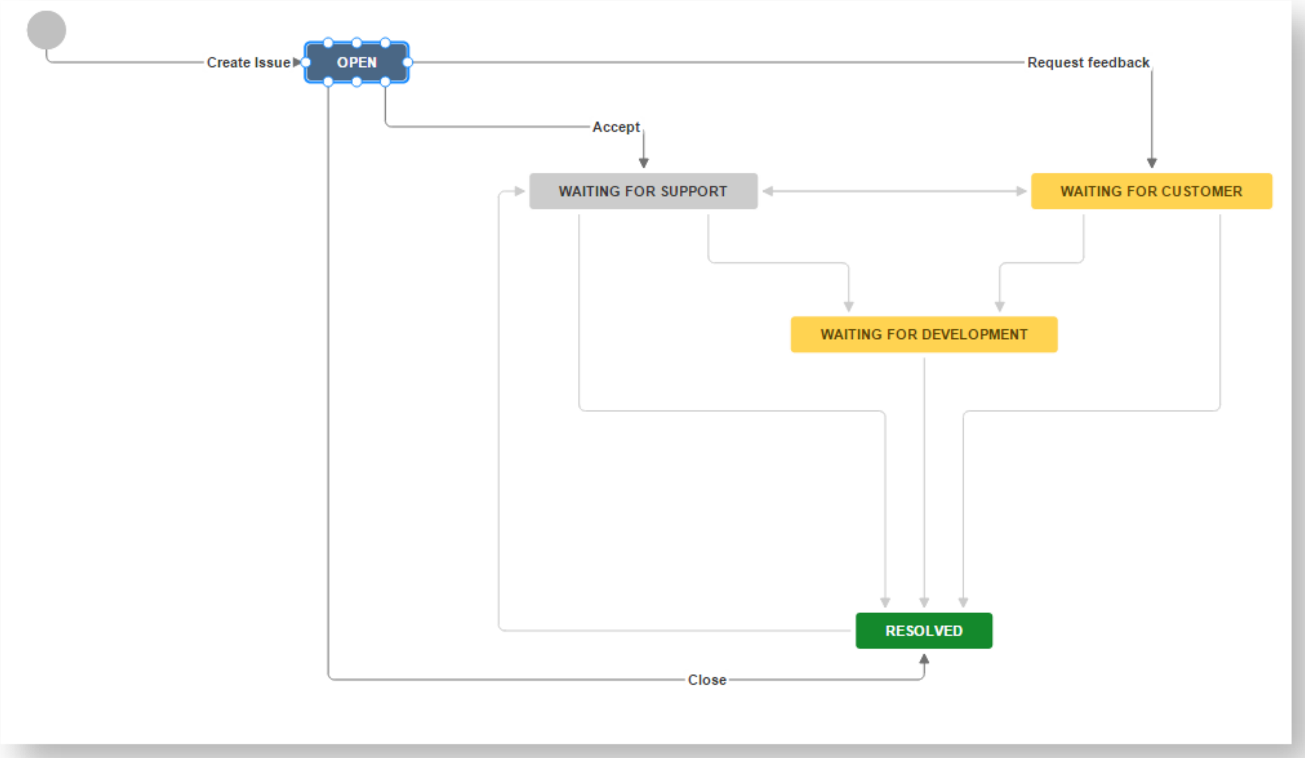


The diagram above illustrates the different ways that workflows can function depending on what is needed.

That might mean simply moving a task from one state to another.

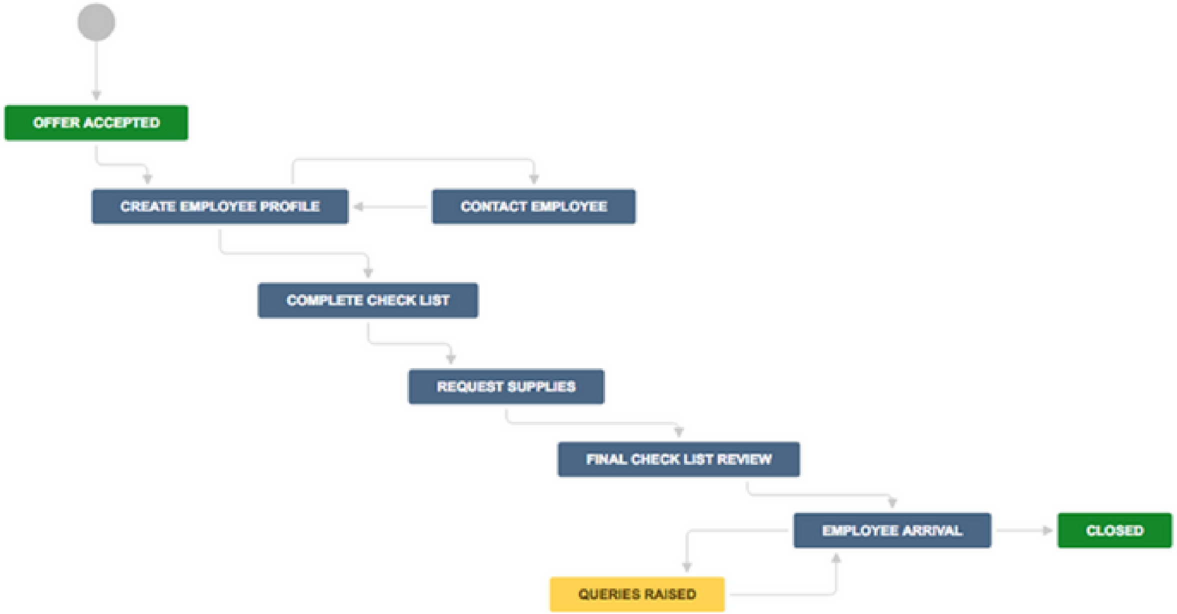
Or it might mean moving tasks between several individuals with multiple resolution states.

You can see Idalko’s support workflow below:



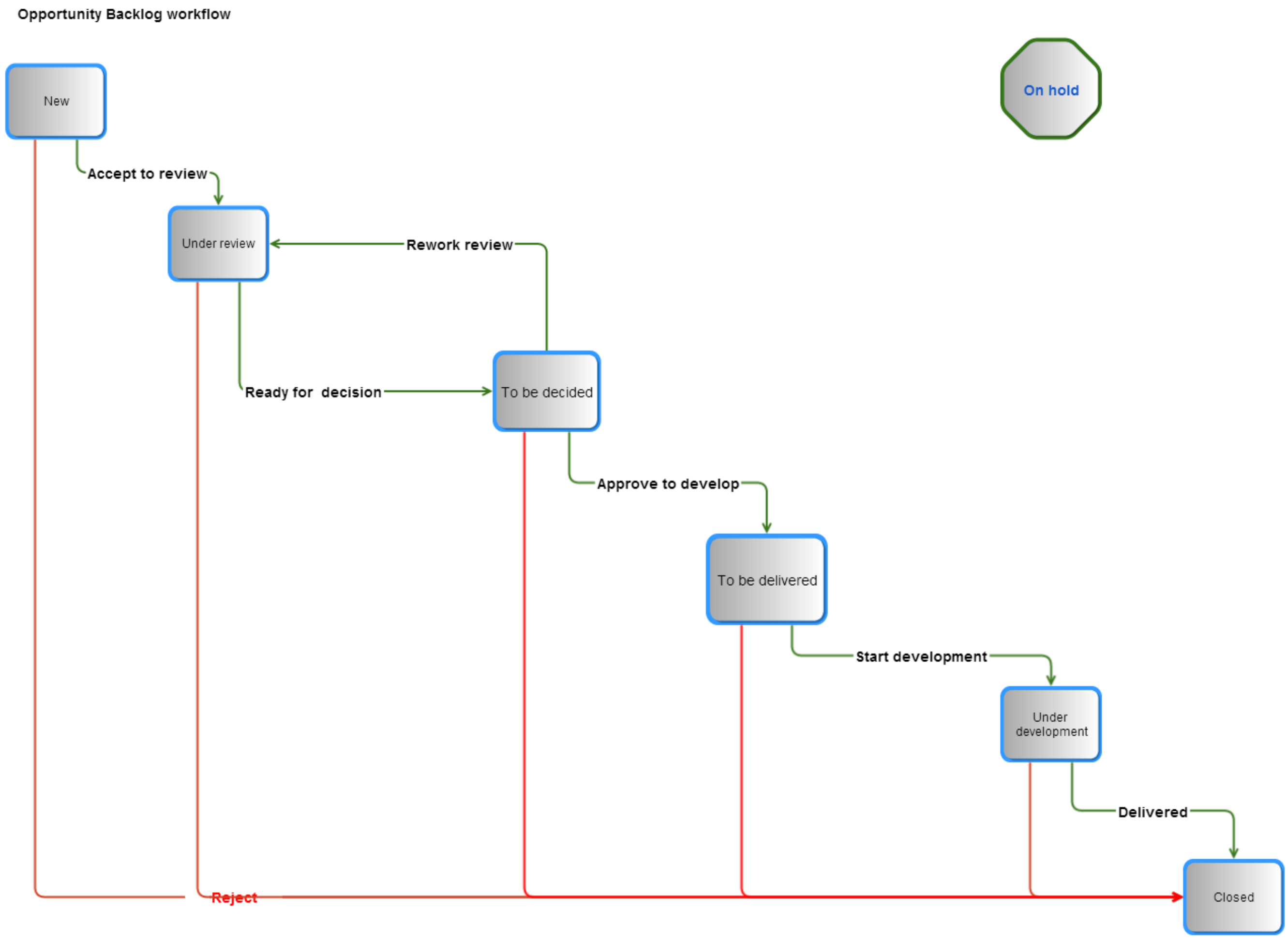
This Jira workflow allows for transitions in several directions. Which direction you will go depends on what is required.

The HR workflow below sets extremely granular steps for onboarding a new employee:



In this case, the workflow has a number of statuses marking tasks leading up to a new hire's arrival.

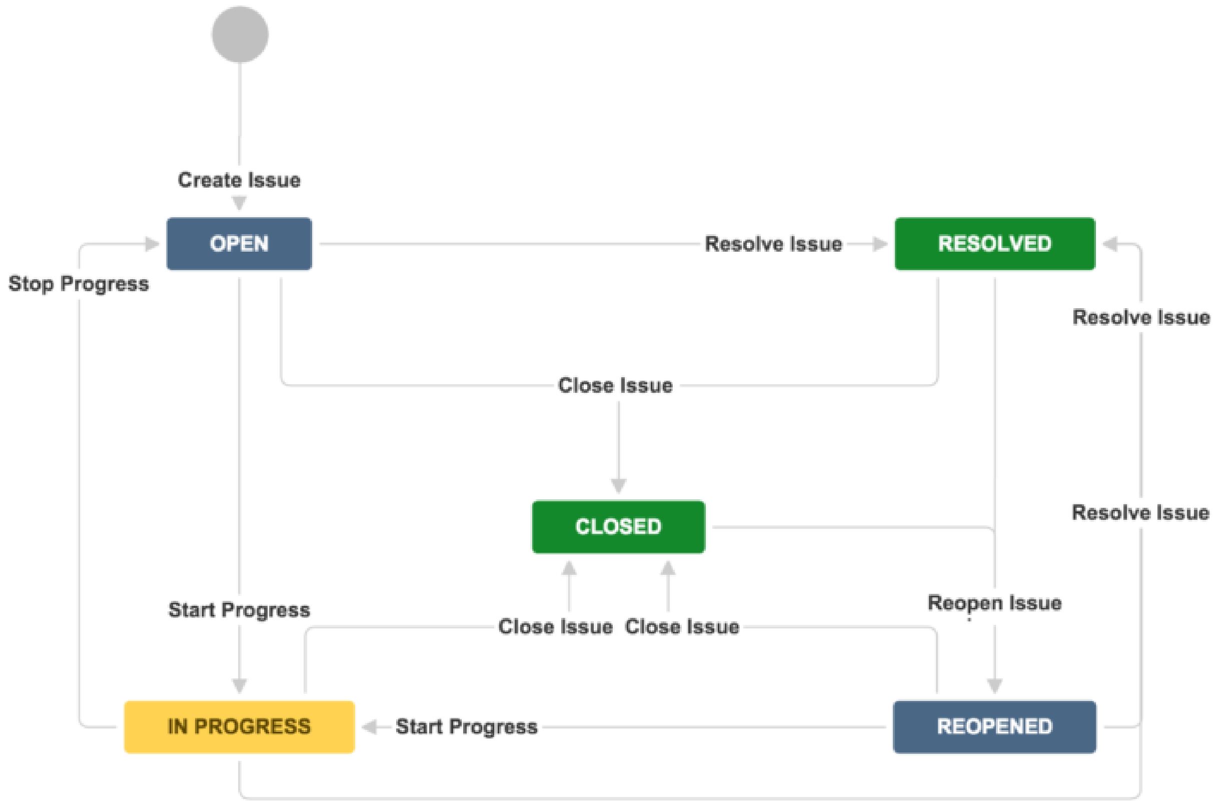
This is a workflow used for managing an opportunity backlog (https://wiki.idalko.com/display/CUS/Opportunity+Backlog).



1. New opportunities get raised and assigned to the project lead.
2. The portfolio management team decides if this opportunity should be picked up (and refined) or rejected.
3. Opportunities under review are detailed out, estimating cost to develop and value it brings.
4. Optionally the opportunity can be documented on confluence, allowing for free form discussions and requirements gathering.
5. Once that all required details are collected, opportunities are progressed into status ‘to be decided’.
6. At this point a formal decision can be taken to include the opportunity into a development plan. Of course the opportunity might to be costly
7. to develop (and be rejected) or put ‘on hold’ if the circumstances are not favorable to start its development.
8. The ‘to be delivered’ queue is used by R&D to select the opportunities which will be included in future releases.
9. The ranking of the opportunies is therefore very important as R&D will focus on the top ranked ones.

10. R&D will translate the opportunities into epics/stories in the release projects.
11. The status of the opportunity is progressed into 'Under development', and the opportunity is linked with the related epics and stories.
12. Once that the product is delivered (i.e. available for installation on customer premises), the opportunity is closed.

The image below illustrates the default Jira workflow. Which allows for transitions in multiple directions from all statuses.



This Jira workflow builds in a great deal of flexibility with how the Issue can be moved forward.

Best practices for testing workflows in Jira

Testing your new workflow is essential. And it comes in several distinct stages, starting with the planning phase.

Design phase of the new workflow

Before building the new workflow, you need to test all the different use cases that are imagined.

For example, how will urgent items move through the system? What information do you need? And how is this captured?

Asking questions and getting the detail now will save a great deal of time later.

Validation phase of the new workflow

This is a dry run of the live workflow implemented once it has been built.

To validate the work, you'll need to gather representatives of the different stakeholders who will use the workflow. (For example, each department).

Unlike testing at the design phase, validation requires you to manually run through every step in the workflow.

It also requires you to run through all possible statuses and escalations. This way, you'll ensure they will function as intended.

You can collect and collate feedback as you go.

Of course, it's far better to discover problems now rather than later. And playing games with the Jira workflow can be a fun way to see how it fits together.

Before launch, you'll also want to provide documentation for the changes.

When working at scale it can save a lot of time to have a clear, easy-to-access set of video tutorials and workshop sessions to provide on-the-job training.

Documents explaining the functionality and what is intended could be a second best option.

Production phase of the new workflow

Launch day has come and gone and your new workflow is now live on the system.

Is the job finished? Not quite.

What you'll still have to do is:

– **Define and track process deviations**: If users aren't following Jira workflows as intended, you need to identify the fact and find out why. How you go about this will depend on the environment and precisely what users are doing. For example, if developers are deploying code changes without Issues being updated, it indicates there may be a process deviation taking place.

– **Review**: Once you've discovered any issues, you'll want to find out exactly why the disconnect is occurring. Is the Jira user unaware of the correct process or is the workflow unclear?

Note: Changing workflows for large numbers of people may require a lot of training. But it's worth making sure you reach everyone.

This may mean:

- *passing information from team leads down to their teams,*
- *holding all-company meetings,*
- *creating blog items,*
- *or managing a company Wiki detailing your Jira workflow functionality.*

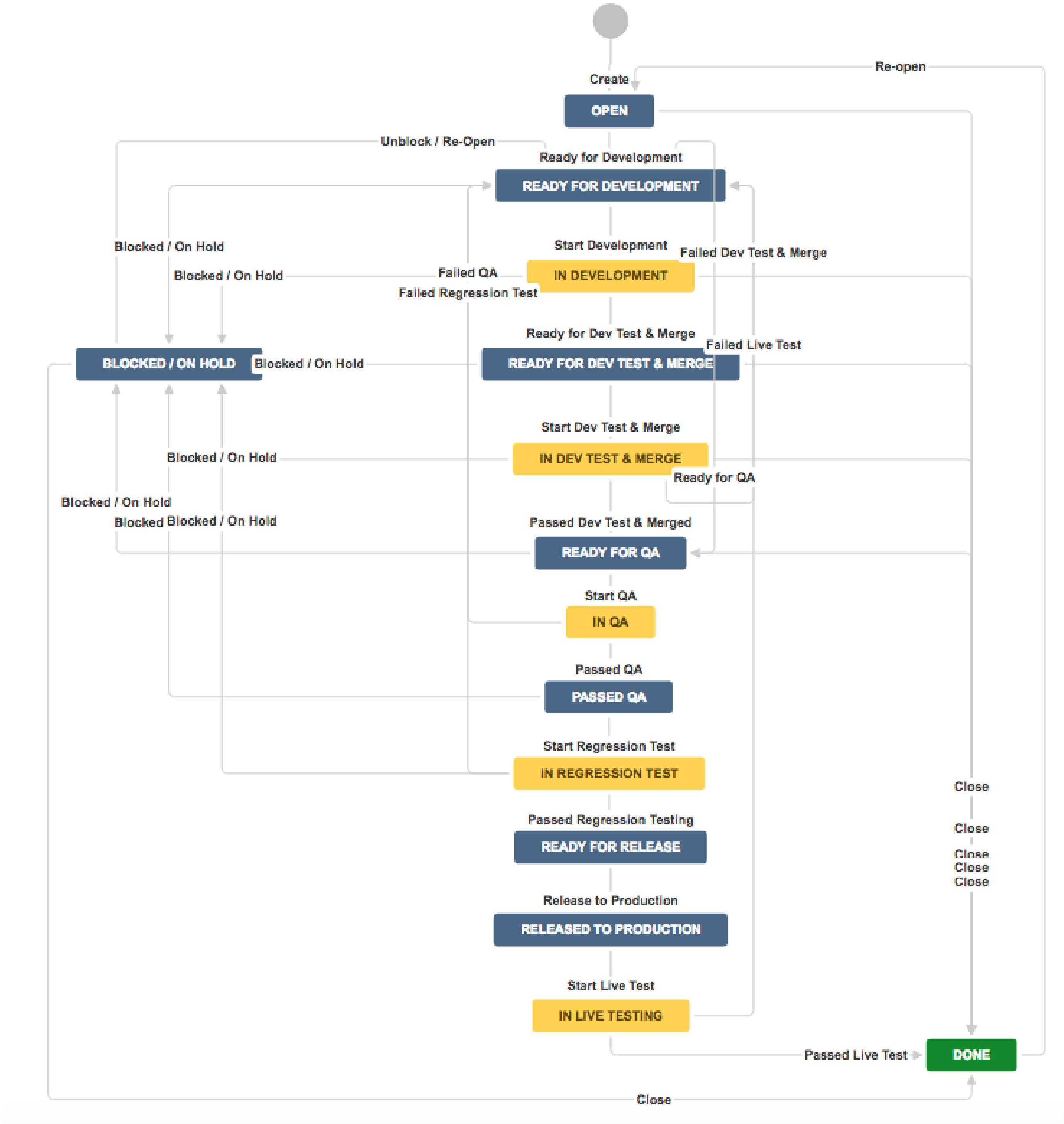
– **Discuss the impact**: Workflow changes are likely to be a big deal for the organization. It can hence be useful to connect with team leaders to review changes. This can be done quarterly or bi-annually. You'll have to ensure that new workflows function correctly, and are being used. And you'll have to discuss further changes. Admins may also allow teams to experiment with workflows to determine precisely what is needed.

Common mistakes to avoid when creating workflows in Jira

Jira workflow customization is a complex process.

So it's easy to make mistakes.

Here are a few things to keep in mind while still at the planning stages.



This is the kind of structure that you want to avoid – where there is far more detail than is needed for an effective workflow.

Creating new Jira issue statuses duplicating existing workflow functionality

It can be tempting to build on the fly, as you go.

However, if you do so, or if you let project admins extensively customize on their own, then you may find that different teams will be moving in different directions.

Customizing your Jira workflows too far

As you customize your Jira workflows, they'll evolve and become increasingly different.

If that continues, they'll eventually become different species and they won't be able to interbreed…

So you won't be able to move Projects or Issues between teams or instances.

The solution to this problem is to ensure that customization is consistent across the organization. Or at least within each department, thereby minimizing compatibility issues.

To make sure that everyone stays on the same page, it's wise to limit admin access. And it's also a good idea to ensure that customization is approved and managed with the big picture in mind.

Customizing workflows in Jira without consultation

Perhaps the most annoying discovery that an admin can make is to find that new workflow statuses aren't actually used.

To make sure you don't set out on unnecessary work, spend time talking to users and mapping out workflows.

Doing so will ensure that the users' requirements are fully understood and that the specification is both needed and reflects actual work processes. As the saying goes, less haste, more speed.

Conclusion

In this guide, we have covered:

- What workflows are – a system for progressing tasks in Jira
- Why you would want to create a custom workflow – to better match your business processes
- The various elements of workflows – statuses, assignees, transitions and resolutions
- How to test your workflow – from design to validation to deployment
- We've run you through a couple of examples of Jira workflows
- And common mistakes people make when creating workflows, including the issues of creating duplicate statuses, customizing too far and creating new workflows when it isn't required

Having read this guide, you should have a clear idea of the pros and cons of creating custom Jira workflows. And of exactly what a new Jira workflow can do for you.

Jira is an immensely powerful platform and with a little work and planning, you can ensure that it's set up to be exactly right for you. As a result you'll be able to unlock a huge boost in productivity.

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*	Email*	Website
Your Name *	Your Email *	Your Website

□ Save my name, email, and website in this browser for the next time I comment.

SUBMIT

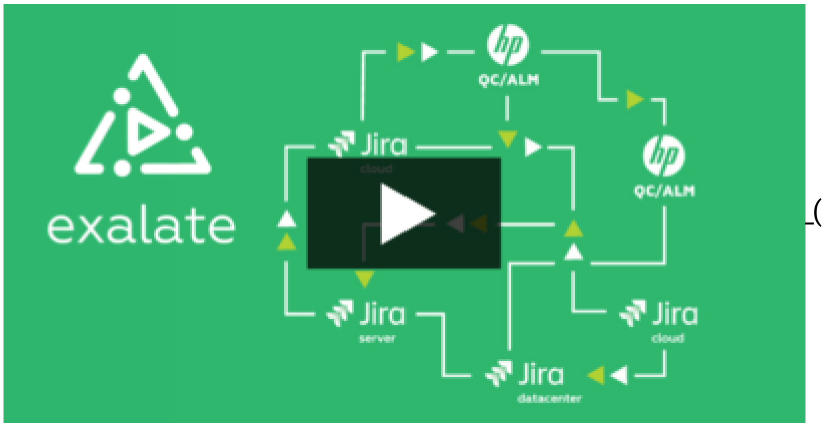
iDalko is a Platinum Atlassian Solution Partner based in Belgium, offering Atlassian services & consultancy (https://www.idalko.com/atlassian-partner/) surrounding the Atlassian toolset. The company is also know for its Jira add-ons Table Grid Editor (https://www.idalko.com/table-grid-editor/) and the Pivot Gadget (https://www.idalko.com/table-grid-editor/). Which are used by some of the top fortune 500 companies globally. iDalko is also the developer and owner of Exalate (https://www.idalko.com/exalate/), the synchronization tool for issue-trackers like Jira (server, cloud & datacenter), HP QC/ALM & more. Which is available on the Atlassian Marketplace (https://marketplace.atlassian.com/plugins/com.exalate.jiranode/cloud/overview).

☞ SEARCH

Search

★ Our Products

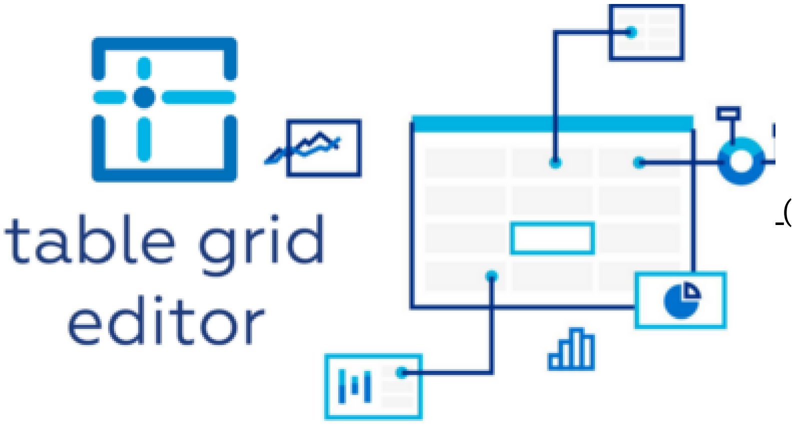
EXALATE



(/exalate/)

Connect multiple JIRA instances and other issue trackers at the flip of a switch. Learn more (/exalate/)

TABLE GRID EDITOR

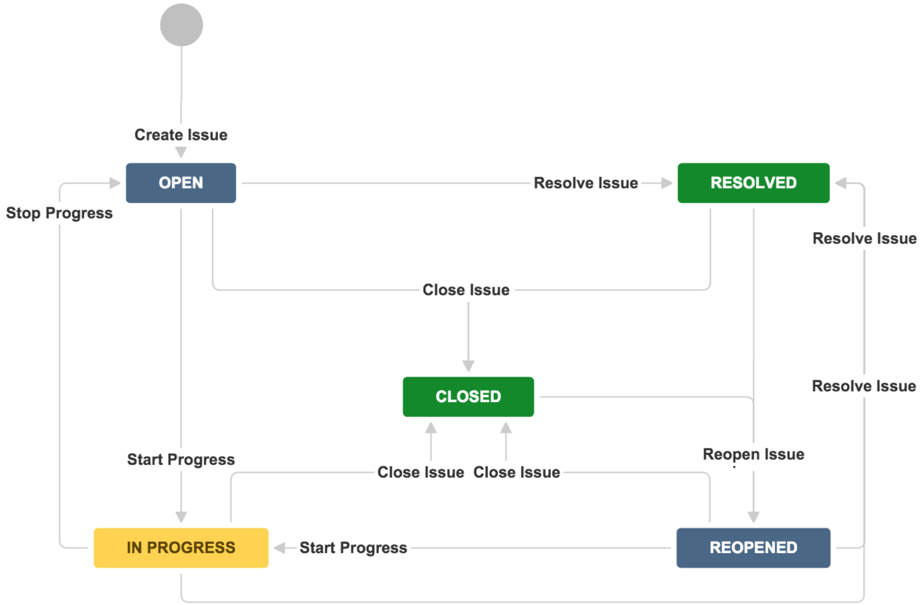


(/table-grid-editor/)

View and edit tabular data, stored in local or external databases, from within your Jira issues. Learn more (/table-grid-editor/)

Recommended Articles

Jira workflow best practices (with examples)



(/jira-workflow-best-practices/)

A step-by-step guide to migration Jira

(/jira-migration/)

Top 5 Free Jira add-ons for users in 2018

TOP PLUGINS FOR JIRA USERS
You won't believe these are totally Free

(/free-jira-plugins/)

(https://(https://twitter.com/idalkotools/)/company/