

# Using workflows for fun & profit

Everyone hates "process", but let's face it: without an established workflow, you're going nowhere fast.



BY DAN RADIGAN

## BROWSE TOPICS

- Agile manifesto
- › Scrum
- User Stories
- Estimation
- Metrics
- Gantt chart
- Program management vs. project management
- › Product Management
- › Agile at scale
- › Software development
- › Design
- › The agile advantage
- DevOps
- › Agile Teams
- › Agile tutorials
- › About the Agile Coach

[All articles](#)

Every software team has a process they use to complete work. Normalizing that process—i.e., establishing it as a workflow—makes it clearly structured and repeatable, which, in turn, makes it scalable. At Atlassian, we take an iterative approach to workflow management because it helps us meet our goals faster and exemplifies our team culture. We are experts in agile workflow management (if we do say so ourselves), and we want to help you become experts too.

## Start simple, start now

When implementing a workflow for the team, always start simple. Fight the temptation to spend weeks (over-)engineering it. Overly complex workflows are hard to understand and adopt—not to mention adapt. For software teams, we recommend these basic workflow states:



### TO DO

Work that has not been started



### IN PROGRESS

Work that is actively being looked at by the team



### CODE REVIEW

Work that is completed and awaiting review



### DONE

Work that is completely finished and meets the team's definition of done

In an issue tracker, these statuses flow from one to the next using transitions that structure the workflow.

## PRODUCT DISCUSSED

### Jira Software

Increase transparency, accountability, and productivity with Jira workflows

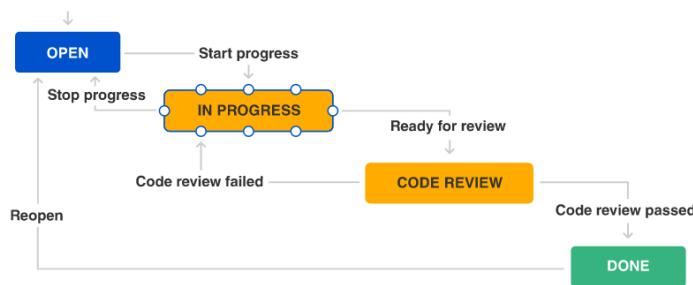
[Get it free →](#)

## SUBSCRIBE

Sign up for more articles

Email

[Subscribe](#)



Some software teams include additional states in their workflow that help them track the status of work more precisely.

**AWAITING QA**  
!

Work that has been implemented, but is still waiting for a tester review (see our article on [agile testing](#) for more details).

**READY TO MERGE**  
✓

Code that has been reviewed and is ready to merge into the main or release branch.

Each state in the workflow doesn't need to be handled by a different person. As an agile team matures, developers handle more and more of the work—from design all the way through to delivery. An autonomous team that can handle heterogeneous work is one of the hallmarks of agility, after all.



**Healthy workflows adapt to the needs of the team. Occasional pain is normal. Chronic pain is not.**

Discuss each pain point in the team's retrospective, and keep in mind that each team will have slightly different values based on their project, technology stack, and method in which they like to work. That's why it's important to choose an issue tracker that has a flexible workflow configuration. Far too many teams compromise their work style to fit a particular toolset, which is frustrating for everyone. This can lead team members to avoid using that tool altogether, compounding frustration across the team and generally wreaking havoc. And when morale falls, productivity suffers. That's a double whammy we all want to avoid!

Teams that are new to agile or that don't have cross-functional skills often end up with "mini waterfalls" in their workflow. For example, design kicks off a work item with a mockup. Development does the implementation. Test confirms quality. Each state is

blocked until the former state is complete. Sound familiar? That's waterfall. But we can do much better with agile workflows to unblock the team and make development easier.

## Optimize the workflow

When you're comfortable with the basic workflow and are ready to customize it, create statuses for each type of work in a team's process. Ideation, design, development, code review, and test are functionally different and can be individual statuses. Aim for a lean set of statuses that clearly communicate what phase a piece of work is in.

Project statuses can also be shared with the rest of the organization. When building a workflow, think about what metrics are important to report on and what non-team members might be interested in learning. For example, a well designed workflow answers the following questions:

- What work has the team completed?
- Is the backlog of work increasing or keeping pace with the team?
- How many items are in each status?
- Are there any bottlenecks that are slowing the team down?
- How long does it take to complete an average task?
- How many work items didn't pass our quality standards the first time around?

The next step in optimizing the workflow is to ensure a steady stream of work through the workflow. Work in progress (WIP) limits dictate a minimum and maximum number of issues in a particular state of the workflow, ensuring each state has enough work to keep the team fully utilized, but not so much that it loses focus from juggling priorities. Enforcing WIP limits will quickly show which processes slows overall work through the pipeline. As the team learns to optimize around its WIP limits, throughput will increase. (See the article on [WIP limits](#) for more details.)

## The challenges of scaling a workflow

Organizations that have several agile teams face special challenges with workflows. Teams often want

to optimize their own workflow to reflect their unique process and culture. This is perfectly understandable. But it can create headaches when different teams use different processes while working on the same project.

Agile teams that work together can benefit from [sharing the same workflow](#). Using the same workflow can make transitioning work between agile teams easier, because they use the same conventions for defining and delivering work. Creating a common process usually involves some give and take from both teams. That's good! They'll learn from one another and come out with a better workflow in the end.

#### PRO TIP

With Jira Software, Atlassian's issue tracker, teams can share workflows but have different representations of the process on their agile board. This ability leads to flexible visualization options without sacrificing the shared asset of workflow.

No matter what your workflow looks like, the process of developing it should be agile, too. Discuss it in retrospectives from time to time, and adapt it as the team's culture and composition changes.

SHARE THIS ARTICLE



DAN RADIGAN

Agile has had a huge impact on me both professionally and personally as I've learned the best experiences are agile, both in code and in life. You'll often find me at the intersection of technology, photography, and motorcycling.



TUTORIAL

## Learn kanban with Jira Software

Step-by-step instructions on how to drive a kanban project, prioritize your work, visualize your workflow, and minimize work-in-progress with Jira Software

[Try this tutorial →](#)



ARTICLE

## Epics, User Stories, Versions, and Sprints

These popular agile methodologies help teams strike a healthy balance between structure and flexibility

[Read this article →](#)

## Agile Topics

Agile project management  
Scrum  
Kanban  
Design

Software development  
Product management  
Teams  
Agile at scale  
DevOps

Sign up for more agile articles and tutorials.

Email

email@example.com

[Subscribe](#)



Up Next  
[Epics, stories, themes →](#)