

Kanban vs. scrum: which agile are you?

Uncover the key considerations when choosing between scrum or kanban, and what to do if you can't decide.



BY MAX REHKOPP

BROWSE TOPICS

- Agile manifesto
- › Scrum
- › Agile project management
- › Product Management
- › Agile at scale
- › Software development
- › Design
- › The agile advantage
- DevOps
- › Agile Teams
- › Agile tutorials
- › About the Agile Coach
- All articles

Summary: “Kanban vs. scrum” is a discussion about two different strategies for implementing an agile development or project management system. Kanban methodologies are continuous and more fluid, whereas scrum is based on short, structured work sprints.”

Agile is a set of ideals and principles that serve as our north star. DevOps is a way to automate and integrate the processes between software development and operations teams. When implementing agile and DevOps, kanban and scrum provide different methodologies for managing complex work.

It's easy to point out the differences between scrum practices and kanban practices, but that's just at the surface level. While the practices differ, the principles are largely the same. Both frameworks will help you build better products (and services) with fewer headaches.



Don't ask, "kanban vs scrum." Instead, ask "kanban or scrum" or even "kanban and scrum." Make it more about the principles than the practices.

So, where were we?

Agile is a structured and iterative approach to project management and product development. It recognizes the volatility of product development, and provides a methodology for self-organizing teams to respond to change without going off the rails. Today, agile is hardly a competitive advantage. No one has the luxury to develop a product for years or even months in a black box. This means it's more important than ever to get it right.

Kanban is all about visualizing your work, limiting work in progress and maximizing efficiency or

RELATED TUTORIAL

Learn kanban with Jira Software

[Try this tutorial →](#)

SUBSCRIBE

Sign up for more articles

Email

[Subscribe](#)

[WORK IN PROGRESS](#), and maximizing efficiency ([flow](#)). Kanban teams focus on reducing the time it takes to take a project (or [user story](#)) from start to finish. They do this by using a [kanban board](#) and continuously improving their flow of work.

Scrum teams commit to ship working software through set intervals called sprints. Their goal is to create learning loops to quickly gather and integrate customer feedback. Scrum teams adopt specific roles, create special artifacts, and hold regular ceremonies to keep things moving forward. Scrum is best defined in [The Scrum Guide](#).

	Scrum	Kanban
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint	Continuous delivery
Roles	Product owner, scrum master, development team	No required roles
Key metrics	Velocity	Lead time, cycle time, WIP
Change philosophy	Teams should not make changes during the sprint.	Change can happen at any time



Scrum: A structured agile

approach

With scrum, your team promises to ship some valuable increment of work by the end of each sprint. Scrum is built on empiricism, focusing on small increments of work that will help you learn from your customers and better inform what you do next. Here's how it breaks down:

Scrum cadence

Scrum moves fast, with sprints of two to at most four weeks with clear start and finish dates. The short time frame forces complex tasks to be split into smaller stories, and helps your team learn quickly. A key question is this: Can your team ship useable code that fast?

Sprints are punctuated by the [sprint planning](#), [sprint review](#), and [retrospective](#) meetings and peppered with daily [scrum\(standup\)](#) meetings. These [scrum ceremonies](#) are lightweight and run on a continuous basis.

Release methodology

Nowadays, it's common to have ad-hoc releases in scrum, but it's long been a best practice to release at the end of each sprint. Teams set an objective for each sprint, the sprint goal, and either approves it for release in the [sprint review meeting](#), or don't.

Scrum roles

Scrum has three clearly defined roles.

- The [product owner](#) advocates for the customer, manages the product backlog, and helps prioritize the work done by the development team.
- The [scrum master](#) helps the team stay grounded in the scrum principles.
- The [development team](#) chooses the work to be done, delivers increments, and demonstrates collective accountability.

Who manages the scrum team? Well, nobody. Scrum teams are self-organizing and everyone is equal, despite having different responsibilities. The team is united by the goal of shipping value to customers.

Key metrics

Velocity—the number of [story points](#) completed in a sprint—is the central metric for scrum teams. It guides

future sprint commitments, or how much work the scrum team takes on in future sprints. If the team completes an average of 35 story points per sprint (Velocity = 35), it won't agree to a sprint backlog that contains 45 points.

Change philosophy

Teams strive to not make scope changes during a sprint. Scrum teams sometimes get feedback and learn that what they're working on isn't as valuable to the customer as they thought. In such cases, the scope of the sprint *should* change to reflect the importance of shipping value to the customer first and foremost. During the sprint retrospective, scrum teams should discuss how to limit change in future, as changes put the potentially shippable increment at risk.

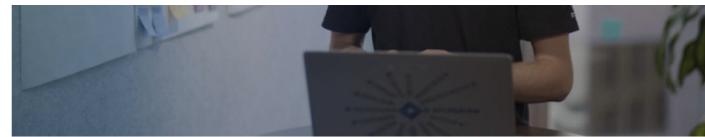
For more on scrum methodologies see [What Is Scrum?](#)



Kanban: Continuous improvement, flexible processes

Kanban helps visualize your work, limit work-in-progress(WIP) and quickly move work from "Doing" to "Done."





Kanban is great for teams that have lots of incoming requests that vary in priority and size. Whereas scrum processes require high control over what is in scope, kanban lets you go with the flow. Let's take a look at the same five considerations to help you decide.

Kanban cadence

Kanban is based on a continuous workflow structure that keeps teams nimble and ready to adapt to changing priorities. Work items—represented by cards—are organized on a [kanban board](#) where they flow from one stage of the workflow(column) to the next. Common workflow stages are *To Do*, *In Progress*, *In Review*, *Blocked*, and *Done*. But that's boring.

The best part of Kanban is making custom columns for how your team works. My team ships content, so our columns(simplified) go from *Backlog*, to *Prioritized*, to *Outlines Ready*, to *Writing*, *Designing*, *Technical Review*, and *Shipped*. Our board helped us learn that we ship about one piece of content per week, and where our bottlenecks are (looking at you *Technical Review!*).

Release methodology

In kanban, updates are released whenever they are ready, without a regular schedule or predetermined due dates.

In theory, kanban does not prescribe a fixed time to deliver a task. If the task gets completed earlier (or later), it can be released as needed without having to wait for a release milestone like sprint review.

Kanban roles

The whole team owns the kanban board. Some teams enlist an agile coach but, unlike scrum, there is no single “kanban master” who keeps everything running smoothly. It’s the collective responsibility of the entire team to collaborate on and deliver the tasks on the board.

Key metrics

Lead time and cycle time are important metrics for kanban teams. The deal with the average amount of time that it takes for a task to move from start to finish. Improving cycle times indicates the success of

kanban teams.

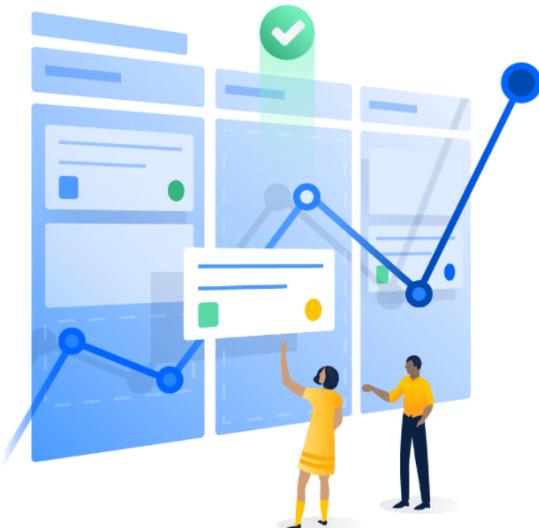
The Cumulative Flow Diagram (CFD) is another analytical tool used by kanban teams to understand the number of work items in each state. CFDs help identify specific bottlenecks that need to be resolved for better throughput.

Another way to deal with bottlenecks is through [Work In Progress\(WIP\) limits](#). A WIP limit caps the number of cards that can be in any one column at one time. When you reach your WIP limit, a tool like Jira Software caps that column and the team swarms on those items to move them forward.

Change philosophy

A kanban workflow can change at any time. New work items can get added to the backlog and existing cards can get blocked or removed all together based on prioritization. Also, if the team capacity changes, WIP limit can be recalibrated and work items adjusted accordingly. It's all about being flexible in kanban.

For more on kanban methodologies see [What is kanban?](#)



Scrum tools vs kanban tools

The agile community believes this conversation shouldn't be about the tools. We often see the tool of choice driving the framework of choice and the framework driving the principles the team adopts. We believe the decision should flow the other direction.

Once you're aligned on scrum principles and happy with the scrum framework, then it's time to find a scrum tool that serves you well. The same goes for kanban. We're biased, but as the number 1 software

development tool used by agile teams, we think Jira Software has you covered.

With Jira's dedicated project types for scrum and kanban, you can realize the principles of each framework. We're also here to help you get started with our guides on [how to do scrum with Jira software](#) and [how to kanban with Jira Software](#).

Kanban vs scrum: What if you can't choose?

Scrum and kanban are “agile by-the-books.” They work in a tried and true fashion that is quite frankly hard to argue against. Borrowing from another famed catchphrase, you might say that, “No one gets fired for choosing scrum.”

But your decision doesn't need to be so black and white. Hundreds of teams are using hybrid models influenced by both scrum and kanban. We set out to help teams do so in Jira Software, which is why we created team-managed projects.

Team-managed projects, as the name suggests, allow teams to pick and choose the agile features that make sense for them; whether that's scrum, kanban, or a mix of both. Instead of implementing one framework on day one, team-managed projects allow you to progressively layer on more and more powerful features as you learn what works for your team (and what doesn't).

You can confidently choose team-managed Scrum or team-managed Kanban knowing that both templates can evolve to suit the needs of your team.

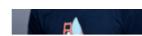
Regardless of what you choose, stick with it for a little while. Take some work from the backlog all the way to done and then ask your team what went well and what went poorly. By trying scrum and kanban and asking these questions, you're well on your way to agile bliss.

SHARE THIS ARTICLE



MAX REHKOPF

As a self-proclaimed “chaos muppet” I look to agile practices and lean principles to bring order to my everyday.



It's a joy of mine to share these lessons with others through
the many articles, talks, and videos I make for Atlassian



TUTORIAL

Create an agile board in Jira Software

Find out how to create agile boards in Jira with this step by step guide

[Try this tutorial →](#)



ARTICLE

Kanplan: where your backlog meets kanban

Kanplan adds the backlog and backlog grooming concepts of scrum to kanban, using the backlog instead of the To Do column to plan and prioritize work.

[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

[Subscribe](#)



Up Next
[Kanplan →](#)