

# Scrum

Learn how to scrum with the best of 'em

## BROWSE TOPICS

Agile manifesto

### Scrum

Scrum master

Retrospectives

Distributed scrum

Roles

Scrum of scrums

Agile scrum artifacts

Scrum metrics

Kanban

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

## What is Scrum?

Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

While the scrum I'm talking about is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. This is one of the reasons scrum is so popular. Often thought of as an agile project management framework, scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.

In this article, we'll discuss how a traditional scrum framework is comprised with the help of the [Scrum Guide](#) and David West, CEO of Scrum.org. We'll also include examples of how we see our customers stray from these fundamentals to fit their specific needs. For that, our own Megan Cook, Group Product Manager for Jira Software and former agile coach, will give tips and tricks in our Agile Coach video series:



THE AGILE COACH

What is scrum?



READ ON BELOW

## Scrum articles



ARTICLE

### Sprints

A sprint is a short, time boxed period when a scrum team works to complete a set amount of work.



ARTICLE

### Sprint Planning

Sprint Planning is an event in scrum that defines what can be delivered in the upcoming sprint and how that work will be achieved.



ARTICLE

### Four agile ceremonies, demystified

Learn how to facilitate great agile ceremonies like sprint planning, daily stand-ups, iteration review and retrospectives.



ARTICLE

### The product backlog: your ultimate to-do list

What is a product backlog in agile or scrum? Learn about the best practices for managing and prioritizing a healthy product backlog.



ARTICLE

### Three steps to better sprint reviews

Learn how sprint reviews demonstrate the hard work of the entire team: designers, developers, and the product owner.



ARTICLE

### Standups for agile teams

Learn how standups contribute to a healthy agile program and some tips and tricks for you and your team.



ARTICLE

### What is a Scrum Master?

Learn what a Scrum Master is (and what they are NOT), and how the role supports and works with other members of an agile team.



ARTICLE

### Agile retrospectives: Use the past to define the future

A retrospective helps teams perform better over time. See what the agile community is saying and learn how to run your own retrospective meetings.



ARTICLE

### Agile Scrum Roles

Learn about the responsibilities and activities associated with the three major agile scrum roles: scrum master, product owner, and development team.



**ARTICLE**

## Scrum of scrums

Scrum of scrums is a scaled agile technique that offers a way to connect multiple teams who need to work together to deliver complex solutions. Learn how to scale scrum with examples from Atlassian and others.

**TUTORIAL**

## Learn scrum with Jira Software

A step-by-step guide on how to drive a scrum project, prioritize and organize your backlog into sprints, run the scrum ceremonies and more, all in Jira.

[Try this tutorial →](#)

**PRODUCT FEATURE**

## From silo to cohesion with Jira Scrum Boards

The Jira Scrum Board is the visual display of progress during the development cycle.

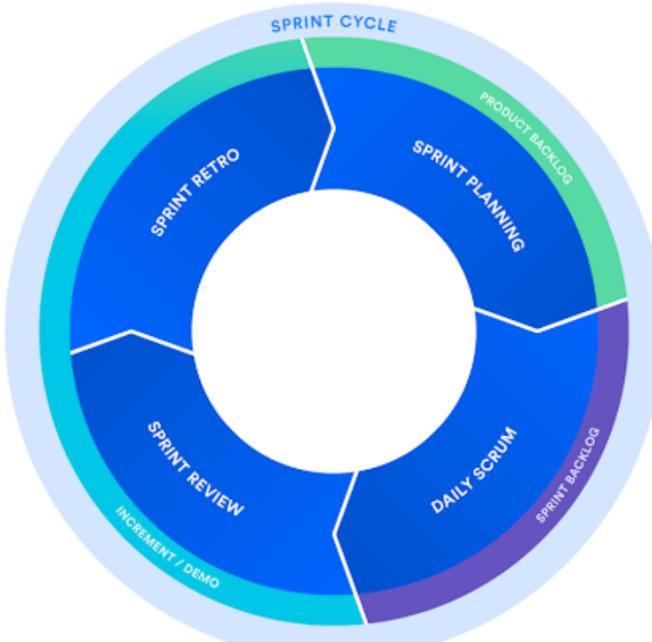
[Get it free →](#)

[CONTINUED]

## The framework

People often think scrum and agile are the same thing because scrum is centered around continuous improvement, which is a core principle of agile. However, scrum is a framework for getting work done, where [agile](#) is a mindset. You can't really "go agile", as it takes dedication from the whole team to change the way they think about delivering value to your customers. But you can use a framework like scrum to help you start thinking that way and to practice building agile principles into your everyday communication and work.

The scrum framework is heuristic; it's based on continuous learning and adjustment to fluctuating factors. It acknowledges that the team doesn't know everything at the start of a project and will evolve through experience. Scrum is structured to help teams naturally adapt to changing conditions and user requirements, with re-prioritization built into the process and short release cycles so your team can constantly learn and improve.



While scrum is structured, it is not entirely rigid. Its execution can be tailored to the needs of any organization. There are many theories about how exactly scrum teams must work in order to be successful. However, after more than a decade of helping agile teams get work done at Atlassian, we've learned that clear communication, transparency, and a dedication to continuous improvement should always remain at the center of whatever framework you choose. And the rest is up to you.

## Scrum artifacts

Let's start with identifying the three artifacts in scrum. Artifacts are something that we make, like a tool to solve a problem. In scrum, these three artifacts are a product backlog, a sprint backlog, and an increment with your definition of "done". They are the three constants in a scrum team that we continue to revisit and invest in overtime.



- **Product Backlog** is the primary list of work that needs to get done maintained by the product owner or product manager. This is a dynamic list of features, requirements, enhancements, and fixes that acts as the input for the sprint backlog. It is, essentially, the team's "To Do" list. The product backlog is constantly revisited, re-prioritized and maintained by the Product Owner because, as we learn more or as the market changes, items may no longer be relevant or problems may get solved in other ways.
- **Sprint Backlog** is the list of items, [user stories](#), or bug fixes, selected by the development team for implementation in the current sprint cycle. Before each sprint, in the sprint planning meeting (which we'll discuss later in the article) the team chooses which items it will work on for the sprint from the product backlog. A sprint backlog may be flexible and can evolve during a sprint. However, the fundamental sprint goal – what the team wants to achieve from the current sprint – cannot be compromised.
- **Increment** (or Sprint Goal) is the usable end-product from a sprint. At Atlassian, we usually demonstrate the "increment" during the end-of-sprint demo, where the team shows what was completed in the sprint. You may not hear the word "increment" out in the world, as it's often referred to as the team's definition of "Done", a milestone, the sprint goal, or even a full version or a [shipped epic](#). It just depends on how your teams defines "Done" and how you define your sprint goals. For example, some teams choose to release something to their

customers at the end of every sprint. So their definition of ‘done’ would be ‘shipped’. However, this may not be realistic of other types of teams. Say you work on a server-based product that can only ship to your customers every quarter. You may still choose to work in 2-week sprints, but your definition of ‘done’ may be finishing part of a larger version that you plan to ship together. But of course, the longer it takes to release software, the higher the risk that software will miss the mark.

As you can tell, there are lots of variations, even within artifacts, that your team can choose to define. That’s why it’s important to remain open to evolving how you maintain even your artifacts. Perhaps your definition of ‘done’ provides undo stress on your team, and you need to go back and pick a new definition.

PRO TIP



You should be just as agile with your framework as you are with your product. Take the necessary time to check in on how things are going, make adjustments if needed, and don’t force something just for the sake of consistency.

## Scrum ceremonies or events

Some of the more well-known components of the scrum framework are the set of sequential events, ceremonies, or meetings that scrum teams perform on a regular basis. The ceremonies are where we see the most variations for teams. For example, some teams find doing all of these ceremonies cumbersome and repetitive, while others use them as a necessary check-in. Our advice is to start out using all of the ceremonies for two sprints and see how it feels. You can then perform a quick [retro](#) and see where you might need to adjust.



Below is a list of all the key ceremonies a scrum team might partake in:

- 1 **Organize the backlog:** Sometimes known as backlog grooming, this event is the responsibility of the product owner. The product owner’s main jobs are to drive the product towards its product vision and have a constant pulse on the market and the customer.

Therefore, he/she maintains this list using feedback from users and the development team to help prioritize and keep the list clean and ready to be worked on at any given time. You can read more about [maintaining a healthy backlog here](#).

- 2 **Sprint planning:** The work to be performed (scope) during the current [sprint](#) is planned during this meeting by the entire development team. This meeting is led by the scrum master and is where the team decides on the sprint goal. Specific use stories are then added to the sprint from the product backlog. These stories always align with the goal and are also agreed upon by the scrum team to be feasible to implement during the sprint.

At the end of the planning meeting, every scrum member needs to be clear on what can be delivered in the sprint and how the increment can be delivered.

- 3 **Sprint:** A sprint is the actual time period when the scrum team works together to finish an increment. Two weeks is a pretty typical length for a sprint, though some teams find a week to be easier to scope or a month to be easier to deliver a valuable increment. Dave West, from Scrum.org advises that the more complex the work and the more unknowns, the shorter the sprint should be. But it's really up to your team, and you shouldn't be afraid to change it if it's not working! During this period, the scope can be re-negotiated between the product owner and the development team if necessary. This forms the crux of the empirical nature of scrum.

All the events – from planning to retrospective – happen during the sprint. Once a certain time interval for a sprint is established, it has to remain consistent throughout the development period. This helps the team learn from past experiences and apply that insight to future sprints.

- 4 **Daily scrum or stand up:** This is a daily super-short meeting that happens at the same time (usually mornings) and place to keep it simple. Many teams try to complete the meeting in 15 minutes, but that's just a guideline. This meeting is also called a 'daily stand-up' emphasizing that it needs to be a quick one. The goal of the daily scrum is for everyone on the team to be on the same page, aligned with the sprint goal, and to get a plan out for the next 24 hours.

**The stand up is the time to voice any concerns you have with meeting the sprint goal or any blockers.**

A common way to conduct a stand up is for every team member to answers three questions in the context of achieving the sprint goal:

- What did I do yesterday?
- What do I plan to do today?
- Are there any obstacles?

However, we've seen the meeting quickly turn into people reading from their calendars from yesterday and for the next day. The theory behind the stand up is that it keep distracting chatter to a daily meeting, so the team can focus on the work for the rest of the day. So if it turns into a daily calendar read-out, don't be afraid to change it up and get creative.

- 5 **Sprint review:** At the end of the sprint, the team gets together for an informal session to view a demo of, or inspect, the increment. The development team showcases the backlog items that are now 'Done' to stakeholders and teammates for feedback. The product owner can decide whether or not to release the increment, although in most cases the increment is released.

This review meeting is also when the product owner reworks the product backlog based on the current sprint, which can feed into the next sprint planning session. For a one-month sprint, consider time-boxing your sprint review to a maximum of four hours.

- 6 **Sprint retrospective:** The [retrospective](#) is where the team comes together to document and discuss what worked and what didn't work in a sprint, a project, people or relationships, tools, or even for certain ceremonies. The idea is to create a place where the team can focus on what went well and what needs to be improved for the next time, and less about what went wrong.

## Three essential roles for scrum success

A scrum team needs three specific roles: product owner, scrum master, and the development team. And because scrum teams are cross-functional, the development team includes testers, designers, UX specialists, and ops engineers in addition to developers.



### The scrum product owner

Product owners are the champions for their product. They are focused on understanding business, customer, and market requirements, then prioritizing the work to be done by the engineering team accordingly. Effective product owners:

- Build and manage the product backlog.
- Closely partner with the business and the team to ensure everyone understands the work items in the product backlog.
- Give the team clear guidance on which features to deliver next.
- Decide when to ship the product with a predisposition towards more frequent delivery.

The product owner is not always the product manager. Product owners focus on ensuring the development team delivers the most value to the business. Also, it's important that the product owner be an individual. No development team wants mixed guidance from multiple product owners.

### The scrum master

Scrum masters are the champions for scrum within their teams. They coach teams, product owners, and the business on the scrum process, and look for ways to fine-tune their practice of it.

An effective scrum master deeply understands the work being done by the team and can help the team optimize their transparency and delivery flow. As the facilitator-in-chief, he/she schedules the needed resources (both human and logistical) for sprint planning, stand-up, sprint review, and the sprint retrospective.

## The scrum development team

Scrum teams get s\*& done. They are the champions for sustainable development practices. The most effective scrum teams are tight-knit, co-located, and usually five to seven members. One way to work out the team size is to use the famous 'two pizza rule' coined by Jeff Bezos, the CEO of Amazon (the team should be small enough to share two pizzas).

Team members have differing skill sets, and cross-train each other so no one person becomes a bottleneck in the delivery of work. Strong scrum teams are self-organising and approach their projects with a clear 'we' attitude. All members of the team help one another to ensure a successful sprint completion.

The scrum team drives the plan for each sprint. They forecast how much work they believe they can complete over the iteration using their historical velocity as a guide. Keeping the iteration length fixed gives the development team important feedback on their estimation and delivery process, which in turn makes their forecasts increasingly accurate over time.

## Scrum, kanban, and agile

Scrum is such a popular agile framework that scrum and agile are often misunderstood to be the same thing. But there are other frameworks, like [kanban](#), which is a popular alternative. Some companies even choose to follow a hybrid model of scrum and kanban, which has acquired the name of "Scrumban" or "Kanplan," which is [Kanban with a backlog](#).

Both scrum and kanban use visual methods such as the scrum board or kanban board to track the progress of work. Both emphasize efficiency and splitting complex tasks into smaller chunks of manageable work, but their approaches towards that goal are different.

Scrum focuses on smaller, fixed-length iterations. Once the time period for a sprint is finalized, the stories or product backlog entries that can be implemented during this sprint cycle are then determined. In kanban, however, the number of tasks or the work in progress (WIP limit) to be implemented in the current cycle is fixed at first. The time taken to implement these features is then calculated backward.

Kanban is not as structured as scrum. Other than the WIP limit, it is fairly open to interpretation. Scrum, however, has several categorical concepts enforced as part of its implementation such as sprint review, retrospective, daily scrum, etc. It also insists on cross-functionality, which is the ability of a scrum team to not depend on external members to achieve their goals. Putting together a [cross-functional team](#) is not straightforward. In that sense, kanban is easier to adapt whereas scrum can be considered as a fundamental shift in the thought process and functioning of a development team.

## But, why scrum?

The scrum framework itself is simple. The rules, artifacts, events, and roles are easy to

understand. Its semi-prescriptive approach actually helps remove the ambiguities in the development process, while giving sufficient space for companies to introduce their individual flavor to it.

The organization of complex tasks into manageable user stories makes it ideal for difficult projects. Also, the clear demarcation of roles and planned events ensure that there is transparency and collective ownership throughout the development cycle. Quick releases keep the team motivated and the users happy as they can see progress in a short amount of time.

However, scrum could take time to fully understand, especially if the development team is acclimatized to a typical waterfall model. The concepts of smaller iterations, daily scrum meetings, sprint reviews, and identifying a scrum master could be a challenging cultural shift for a new team.

But, the long-term benefits far outweigh the initial learning curve. Scrum's success in developing complex hardware and software products across diverse industries and verticals makes it a compelling framework to [adopt for your organization](#).



CLAIRE DRUMOND

Claire Drumond is a marketing strategist, speaker, and writer for Atlassian. She is the author of numerous articles published on the Trello and Atlassian blogs and is a regular contributor to various publications on Medium including Hacker Noon, Art+Marketing, and PoetsUnlimited. She speaks at tech conferences around the world about agile, breaking down silos, and building empathy.



TUTORIAL

[Ready to get started? Try this tutorial!](#)



UP NEXT

[Kanban](#)