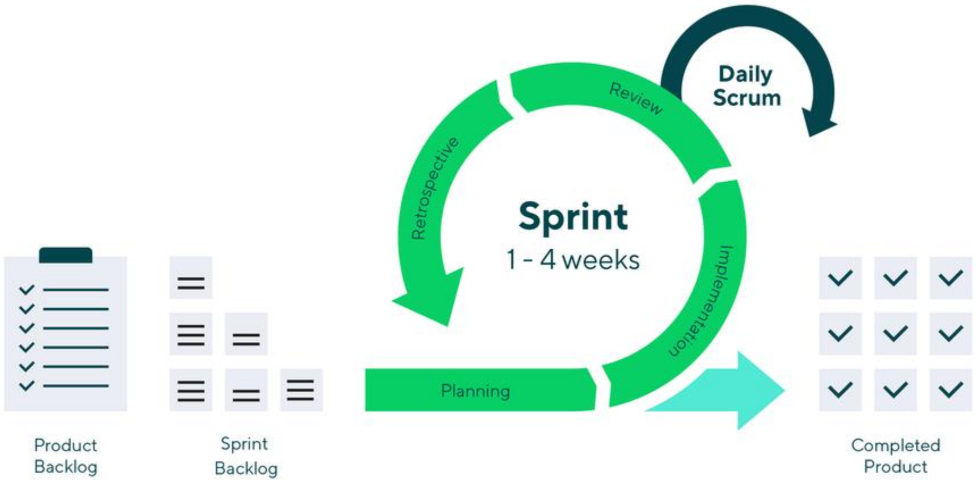
Agile

What is Agile methodology?

Agile is an approach that divides work into phases, emphasizing continuous delivery and improvement. The Agile methodology benefits teams by enabling adaptive planning, rapid execution, and ongoing evaluation, leading to more responsive and successful outcomes.

Whereas the traditional "waterfall" approach has one discipline contribute to the project, then "throw it over the wall" to the next contributor, agile calls for collaborative cross-functional teams. Open communication, collaboration, adaptation, and trust amongst team members are at the heart of agile. Although the project lead or product owner typically prioritizes the work to be delivered, the team takes the lead on deciding how the work will get done, self-organizing around granular tasks and assignments.



Agile isn't defined by a set of ceremonies or specific development techniques. Rather, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement.

The original Agile Manifesto didn't prescribe two-week iterations or an ideal team size. It simply laid out a set of core values that put people first. The way you and your team live those values today – whether you do scrum by the book, or blend elements of kanban and XP – is entirely up to you.

Why Agile?

Teams choose agile so they can respond to changes in the marketplace or feedback from customers quickly without derailing a year's worth of plans. "Just enough" planning and shipping in small, frequent increments lets your team gather feedback on each change and integrate it into future plans at minimal cost.

Agile Methodology
Step 1
More Control
Step 2
Better Productivity
Step 3
Better Quality
Step 4
Higher Customer Satisfaction
Step 5
Higher Return on Investment

But it's not just a numbers game—first and foremost, it's about people. As described by the Agile Manifesto, authentic human interactions are more important than rigid processes. Collaborating with customers and teammates is more important than predefined arrangements. And delivering a working solution to the customer's problem is more important than hyper-detailed documentation.

An agile team unites under a shared vision, then brings it to life the way they know is best. Each team sets their own standards for quality, usability, and completeness. Their "definition of done" then informs how fast they'll churn the work out. Although it can be scary at first, company leaders find that when they put their trust in an agile team, that team feels a greater sense of ownership and rises to meet (or exceed) management's expectations.

Agile yesterday, today and tomorrow

The publication of the Agile Manifesto in 2001 marks the birth of agile as a methodology. Since then, many agile frameworks have emerged such as scrum, kanban, lean, and Extreme Programming (XP). Each embodies the core principles of frequent iteration, continuous learning, and high quality in its own way. Scrum and XP are favored by software development teams, while kanban is a darling among service-oriented teams like IT or human resources.

Today, many agile teams combine practices from a few different frameworks, spiced up with practices unique to the team. Some teams adopt some agile rituals (like regular stand-ups, retros, backlogs, etc.), while others created a new agile practice (agile marketing teams who adhere to the Agile Marketing Manifesto).

The agile teams of tomorrow will value their own effectiveness over adherence to doctrine. Openness, trust, and autonomy are emerging as the cultural currency for companies who want to attract the best people and get the most out of them. Such companies are already proving that practices can vary across teams, as long as they're guided by the right principles.

Sprint planning

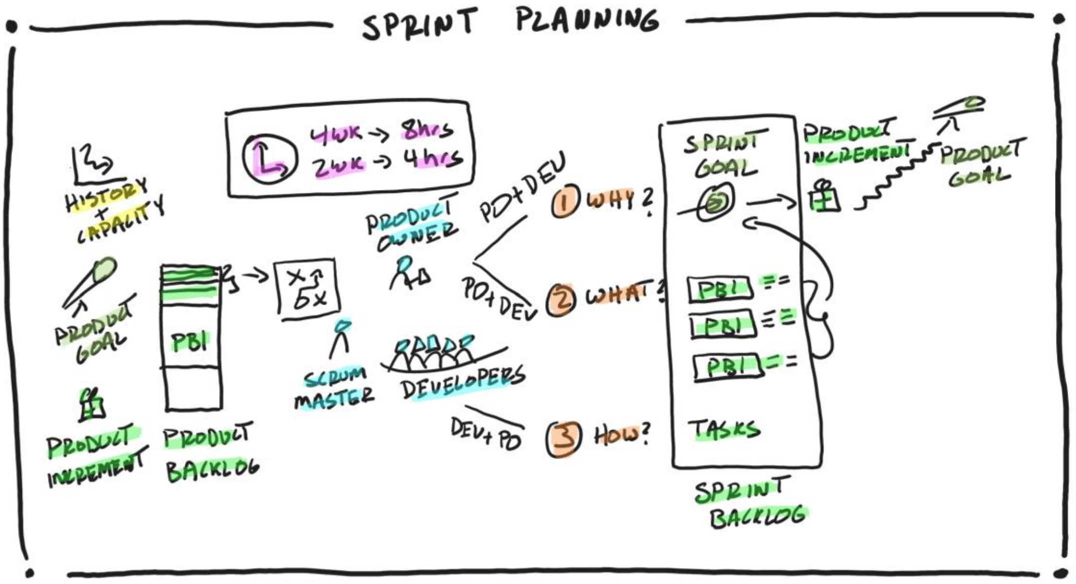
When practicing scrum, the sprint planning meeting is held at the beginning of the sprint and is where teams identify what can be delivered in the sprint and how that work will be achieved. At the end of the planning meeting, every scrum member needs to be clear on what can be delivered in the sprint and how the increment can be delivered.

Attendees: Development team, scrum master, product owner

When: At the beginning of a sprint.

Duration: Usually around one hour per week of iteration. e.g. a two-week sprint kicks off with a two-hour planning meeting.

Agile framework: Scrum. (Kanban teams also plan, of course, but they are not on a fixed iteration schedule with formal sprint planning)

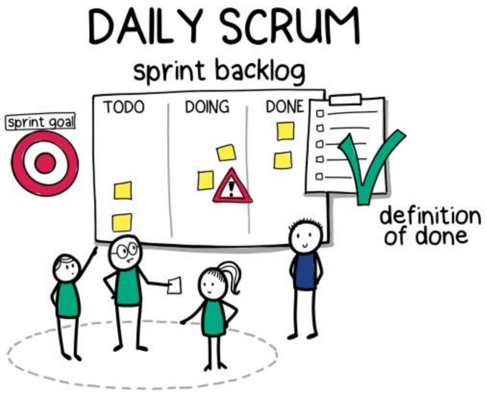


Purpose: Sprint planning sets up the entire team for success throughout the sprint. Coming into the scrum meeting, the product owner will have a prioritized product backlog. They discuss each item with the development team, and the group collectively estimates the effort involved. The development team will then make a sprint forecast outlining how much work the team can complete from the product backlog. That body of work then becomes the sprint backlog.

Use the sprint planning meeting to flesh out intimate details of the work that needs to get done. Encourage team members to sketch out tasks for all stories, bugs, and tasks that come into the sprint. Foster discussions and gather consensus on the plan of action. Effective planning significantly increases the team's chances of success by meeting the commitments of the sprint.

Daily stand-up

The daily stand-up – a.k.a. daily scrum – is a short, 15-minute (or less) daily meeting to discuss progress and identify blockers. Attendees are urged to participate while standing to help keep the meeting short.



Attendees: Development team, scrum master, product owner

When: Once per day, typically in the morning.

Duration: No more than 15 minutes. Don't book a conference room and conduct the stand-up sitting down. Standing up helps keep the meeting short!

Agile framework: Scrum and kanban.

Purpose: A daily stand-up is designed to quickly inform everyone of what's going on across the team. It's not a detailed status meeting. The tone should be light and fun, but informative. Have each team member answers the following questions:

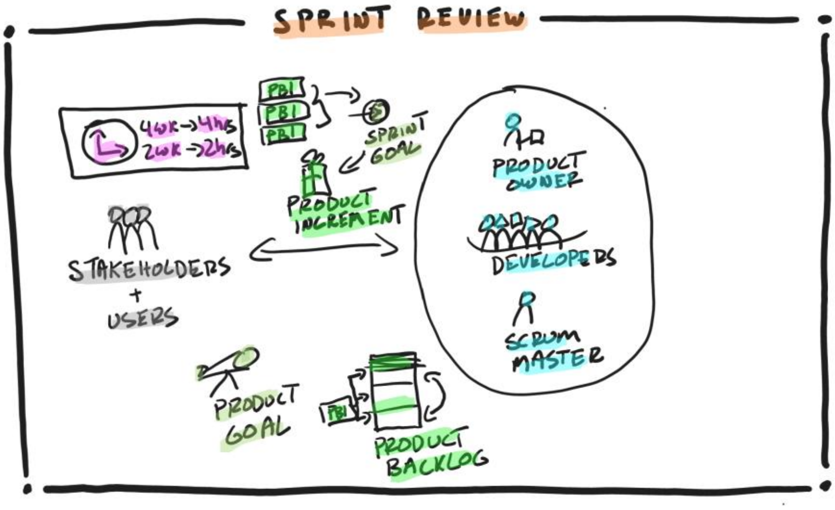
- What did I complete yesterday?
- What will I work on today?
- Am I blocked by anything?

There's implicit accountability in reporting what work you completed yesterday in front of your peers. No one wants to be the person who is constantly doing the same thing and not making progress.

Some teams use timers to keep everyone on track. Others toss a ball across the team to make sure everyone's paying attention. Many distributed teams use videoconferencing or group chat to close the distance gap.

Sprint review

The sprint review, also called an iteration review, is where the scrum team meets to reveal what was accomplished during the sprint. A development team shows which backlog items are “Done” to stakeholders and teammates, who can then give feedback.



Attendees: Development team, scrum master, product owner

When: At the end of a sprint.

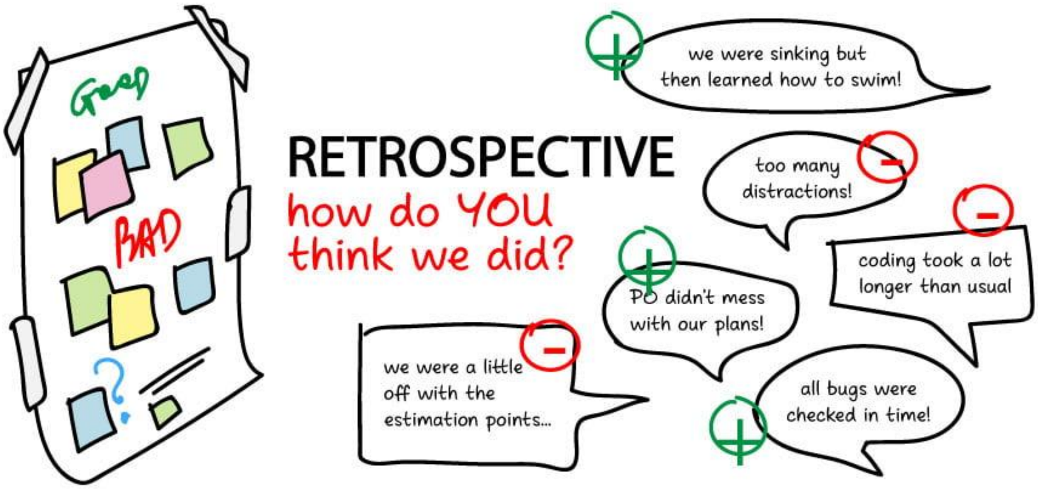
Duration: Typically 45 minutes per week of iteration - e.g. a 90-minute retrospective after a two-week sprint.

Agile framework: Scrum and kanban. Scrum teams do sprint retrospectives based on a fixed cadence. Kanban teams can benefit from occasional retrospectives, too.

Purpose: A sprint review is a time to showcase the work of the team. They can be in a casual format like "demo Fridays", or in a more formal scrum meeting structure. This is the time for the team to celebrate their accomplishments, demonstrate work finished within the iteration, and get immediate feedback from project stakeholders. Remember, work should be fully demonstrable and meet the team's quality bar to be considered complete and ready to showcase in the review.

Sprint retrospective

A sprint retrospective is a meeting to review what was successful during the sprint and what can be improved upon. Agile teams can specifically review team dynamics, processes, and tools, then create plans to improve the way the team works.


Good
BAD
?
RETROSPECTIVE
how do YOU think we did?
we were sinking but then learned how to swim!
too many distractions!
coding took a lot longer than usual
PO didn't mess with our plans!
we were a little off with the estimation points...
all bugs were checked in time!

Attendees: Development team, scrum master, product owner

When: At the end of a sprint.

Duration: Typically 45 minutes per week of iteration-e.g. a 90-minute retrospective after a two-week sprint.

Agile framework: Scrum and kanban. Scrum teams do sprint retrospectives based on a fixed cadence. Kanban teams can benefit from occasional retrospectives, too.

Purpose: Agile is about getting rapid feedback to make the product and development culture better. Retrospectives help the team understand what worked well and what didn't.

Retrospectives aren't just a time for complaints without action. Use retrospectives to find out what's working so the team can continue to focus on those areas. Also, find out what's not working and use the time to find creative solutions and develop an action plan. Continuous improvement is what sustains and drives development within an agile team, and retrospectives are a key part of that.

Even if things are going well across the team, don't stop doing retrospectives. Retrospectives provide ongoing guidance for the team to keep things going well.