Agile

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Benefits of Agile

1. Satisfied customers

By involving customers in the development process, Agile teams keep them in the loop and show that they value their opinion. Stakeholders want to be engaged throughout the project life cycle so they can offer feedback and ensure that the final product will be suited to their needs. These tailor-made deliverables will likely improve the overall user experience and boost customer retention.

2. Improved quality

Agile methodologies use an iterative approach to project management, meaning processes are improved upon each time an interval is repeated. This consistent focus on improvement and quality control is one of the core principles of Agile, and it helps to create superior products.

3. Adaptability

The central theme of Agile is flexibility. Agile teams are responsive to change, even at the last minute, and can adapt to it without much disruption. Project deliverables are not set in stone, so teams can easily reassess their plans and adjust their priorities to align with updated goals. Being adaptable means teams can deliver consistently and manage clients' changing requirements effectively.

4. Predictability

Agile teams work in short time periods, sometimes referred to as sprints. These fixed durations (e.g., two weeks) make it easier for project managers to measure team performance and assign resources accordingly. It is also easier to predict costs for shorter time periods than for a long-term project, simplifying the estimation process.

5. Reduced risk

Developers regularly assess progress during sprints, meaning they have better visibility into the project and can spot potential obstacles quickly. These minor issues can be tackled before they escalate, creating an effective risk mitigation process and giving the project a greater chance of success.

6. Better communication

Agile teams prioritize face-to-face communication and continuous interaction. They will usually conduct daily meetings to ensure everyone is on the same page and working towards the same objectives. By regularly communicating with each other, they eliminate potential confusion to successfully achieve their objectives.

Scrum

Scrum is an Agile project management methodology that offers a structure for teams to deliver incrementally while prioritizing efficient planning, collaborative execution, and continuous improvement. In this Scrum guide, you'll gain a better understanding of this Agile framework including roles, ceremonies, and essential Scrum software.

Scrum principles

Scrum's principles serve as guidelines for executing the Scrum methodology. They ensure that there is an understanding of the processes and dynamics of a Scrum environment. It is the Scrum master's responsibility to uphold Scrum principles and values. Here's a breakdown of the six Scrum principles.

1. Empirical process control

To remain effective, agile, and able to respond to change, Scrum relies on empirical process control. This means that the entire Scrum process is ruled by transparency, inspection, and adaptation.

2. Self-organization

The Scrum method encourages a level of independence from the Scrum team. When Scrum teams are described as "self-organizing," this simply means that they manage their own tasks, solve problems independently, and are accountable to themselves and each other — rather than an outside manager.

3. Time-boxing

Time-boxing is a practice where a fixed amount of time is allocated for certain activities or objectives. Time-boxing allows for activities to be completed in an optimal time frame without running too long. Time-boxing is ideal for setting time frames around activities like sprint planning and sprint retrospectives.

4. Value-based prioritization

To achieve value-based prioritization, items in the product backlog are constantly updated based on their value and importance to the end-user and stakeholders.

5. Iterative development

Because of its ongoing sprints, the objectives in product development are consistently reviewed and updated to create the best quality product and delivery process.

6. Collaboration

Scrum teams collaborate frequently and, sometimes, at great length. Daily standup meetings are an opportunity to collaborate and problem-solve, as are sprint reviews and retrospectives.

Roles on an Agile Scrum team

As mentioned earlier, the Scrum team’s goal is to build a quality product for the end-user. But who does the Scrum team consist of? The Scrum team consists of three roles. The product owner, Scrum master, and development team. Here’s a breakdown of each Scrum role and what they do.

The product owner - The product owner is in charge of maximizing value. That means they’re not just concerned about the day-to-day work on the product or planning the sprint itself. Their job is to ensure that the interests of the stakeholders and customers are represented through the prioritization of the product backlog. The product owner also ensures that the deliverables meet evolving requirements.

The Scrum master - The Scrum master is the person in charge of guiding the development team to be more self-sufficient. The Scrum master acts as a facilitator, ensuring all team members adhere to the principles and processes of the Agile Scrum methodology.

As a guide, the Scrum master encourages the team to think of creative ways to tackle pressing impediments. This is achieved by constantly asking the right questions that will nudge them to come up with solutions.

The development team - The development team is the brains behind the process. While they work with the support of the product owner and Scrum master, they have the necessary technical skills to build and deliver a great product.

Ideally, the development team consists of cross-functional team members, including QA testers, designers, and developers. To ensure optimal performance, the development team must observe Scrum best practices and rules during product development.

Scrum methodology overview: What’s involved?

Many teams choose Scrum because it is ideal for project objectives that tend to change over time. Because of this, there’s a need for constant feedback and iterations at every stage of the product development.

Ideally, the team applies Scrum processes based on a set of principles and values that must be acknowledged during the software development life cycle.

Scrum methodology includes:

- The product backlog
- Sprint planning

- A series of sprints that lead to a product increment and then a working product

The Product backlog - The product backlog includes the necessary features and functionalities that need to be added to the software. Because the product owner sets the foundation of the product development, they create and groom the backlog by adding and prioritizing items. The product owner evaluates these items by asking questions such as “what impact does this feature create in the development process?", What are the risks involved?”, “What are the costs involved?” Answering these questions helps outline the clear value of each item in the backlog.

Sprint planning - After creating the product backlog, the team begins sprint planning. This process is where teams go through the product backlog to figure out how to achieve the most important objectives and ensure they don’t spread themselves too thinly. After this, the team can get a better understanding of how the product development will progress.

Sprints - During sprints, the Scrum team completes the goals and objectives outlined for that particular time frame. Ideally, a sprint should last 1 – 4 weeks — but shouldn’t take more than a month. During the sprint, the Scrum team attends a daily standup meeting with the product owner and Scrum master. This is where they discuss what they are working on that day and talk through any issues or impediments they are experiencing. It's important to note that no changes must occur when a sprint is in progress. This is to ensure that the team can meet the sprint goal without issue.

Sprint review - At the end of every sprint, the Scrum team — including the product owner and Scrum master — gather together to review the sprint, including what went well, what was completed, and what the updated backlog will look like. A product demo may also be on the agenda.

Here, the product owner will validate the sprint work. Once this phase ends, the cycle begins again, starting with the product backlog, sprint planning, next sprint, and a product increment.

In summary, the Scrum methodology is a learning and disciplinary process that enables the Scrum team to identify ways to improve and deliver the best quality product to the end-user.

Benefits of the Scrum methodology

Scrum provides tremendous benefits to organizations, product development teams, and individuals. Here are some of the benefits of Scrum:

Timely prediction - By using the Scrum framework, you can estimate the average speed of the team. As a result, it is possible to estimate when a certain feature in the product backlog will be delivered.

Improved team morale - The Scrum method involves a concerted effort on the part of each team member. Being part of a self-organizing team enables people to be proactive, innovative, and focused. The support of a Scrum master is also vital for team morale because they guide the Scrum team, remove all impediments, and work towards improving the performance of the team.

Better quality product - Scrum provides the framework for collaboration and continuous improvement. Activities like sprint retrospectives and sprint reviews ensure the team has a chance to reflect and improve their processes.

Increased customer satisfaction - The primary objective of the Scrum team is to provide value for the end-user. This is achieved by:

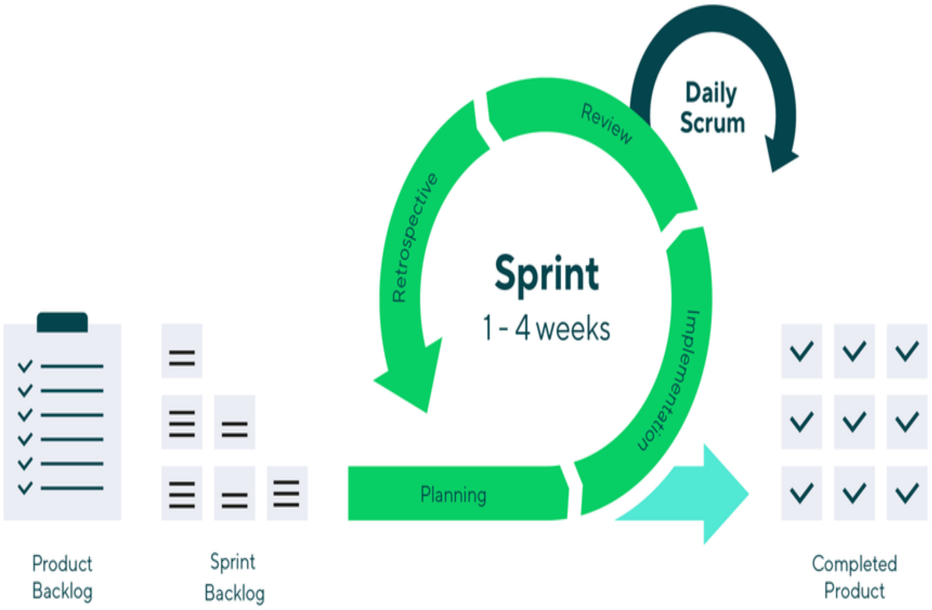
- Keeping all stakeholders and customers in the loop throughout the project development
- Having an experienced product owner that understands the scope of the project development cycle
- Constantly updating and re-prioritizing the project backlog as feedback is provided

In Agile software development, the Scrum framework encourages better collaboration and understanding amongst team members.

What is a Scrum sprint?

A Scrum sprint is a time-boxed period during an ongoing development cycle where a specific set of features or capabilities are worked on. A sprint usually runs for 1 to 4 weeks. During that time, the Scrum team's main goal is to provide a product increment — a version of the product that includes the features and backlog items prioritized and completed during the sprint.

Think of a sprint as a way of breaking down the project into bite-sized chunks. In each stage, what is worked on next is based on feedback and the features prioritized during the backlog grooming and sprint planning process.



Kanban

Agile is a project methodology that promotes tackling projects by breaking them down into smaller stages. It emphasizes constant collaboration, continuous improvement, and high levels of customer involvement. There are various frameworks teams can choose to follow to adopt Agile, Kanban being one of them. Think of Agile as being what you want to achieve and Kanban being one recipe for how to achieve it.

What are the fundamentals of Kanban?

Kanban is about more than using cards to help manage just-in-time delivery. The Kanban framework is designed to help teams reduce bottlenecks, improve efficiencies, increase quality, and boost output. Kanban is based on four principles and six core practices.

The four principles of the Kanban methodology are:

- Start with now. Focus on what you're doing now. Fully understand the processes already in place, including what works and what doesn't.
- Take an incremental approach. Look at how to slowly change your processes over time. Avoid implementing radical changes.
- Keep roles. Unlike other frameworks that promote their own unique roles (such as Scrum master), Kanban emphasizes working with the roles your team already has.
- Encourage leadership. Innovation and ideas for improvement should be promoted at all levels. Encourage every employee to act as a leader, regardless of role or title.

Kanban methodology's six core practices are:

- **Visualize the workflow -** Kanban requires using a physical or virtual board to visualize how workflows from one stage to the next.
- **Limit work in progress** - Each project team needs to set a limit to how many tasks are allowed to be in each stage of the workflow at once. If you have five reviewers, you may limit the "Review" stage to having no more than five tasks in it at once.
- **Actively manage the workflow** - As a project manager, your primary role is to monitor the workflow for bottlenecks and make adjustments to remove roadblocks and improve efficiency.
- **Create process guidelines** - Have clearly communicated guidelines on how work is completed, what "done" means, etc. This can be a checklist in each column or on each "card" outlining what is required for it to move to each stage.

- **Use feedback loops** - Use tools and processes to promote early and continual feedback. This can mean multiple review stages, or reports and metrics communicating performance.
- **Evolve** - As with other Agile frameworks, adapting, evolving, and improving your processes is encouraged. Focus on developing and implementing small changes to improve your workflow and processes.

What is a Kanban board?

The Kanban board is a physical or virtual board that maps out your project's workflow and how tasks move through it from beginning to completion. A Kanban board ensures the workflow is standardized, and that team members can easily see where each task is in the overall scheme.

The most basic Kanban board only has three workflows: To Do, In Progress, and Complete. But, columns can be added or changed to suit your project.

Each task is represented as a "card" and placed on the board in the column representing its current stage of work. As tasks progress, the card is moved throughout the workflow. Each card will contain information about the task, such as:

- A short description
- The name of the person responsible
- An estimate of how long it will take
- Requirements to move it to the next stage

Virtual cards may also contain other data, including links to relevant documents and supporting files.

How is Kanban methodology different from Scrum?

Scrum is another extremely popular Agile project framework. As both Kanban and Scrum are based on the Agile project methodology, they have similar principles and ideals. Both frameworks encourage collaboration, process improvement, and breaking projects down into phases. However, there are essential differences.

The Kanban process focuses on breaking a project down into workflow stages and managing the flow and volume of tasks through those stages. Scrum revolves around breaking a project down by time (usually 1–4-week “sprints”) and managing tasks completed in each sprint.

Kanban project management isn’t time-based. While cards may have deadlines or estimated times to complete, Kanban is viewed as a continuous flow. It’s often used by IT service desks and other teams who have a never-ending flow of tasks.

Scrum also has several unique roles, such as Scrum master, product owner, etc. While Kanban encourages keeping the roles your team already have. Generally, Scrum is better for time-sensitive projects, while Kanban better suits teams with a continuous influx of new tasks. However, many teams are adopting a fairly new framework called Scrumban that attempts to capture the best of both worlds.

What types of projects is Kanban best for?

Kanban project management is best for projects that have a lot of individual deliverables and an emphasis on workloads over delivery dates. Because an individual card represents each task, projects with a lot of interdependencies may suffer.

However, suppose you’re dealing with a large volume of tasks that have multiple discrete statuses, with a different person responsible for each one. In that case, Kanban can help you effectively monitor each stage of the process.

- Some examples of projects that do well with Kanban are:
- A marketing campaign requiring many separate ads
- A content creation project, where each blog, chapter, or story is a separate task
- A service project to resolve bugs and close out customer tickets