**Scrum in Software Development: Roles, Processes, and When to Use It?**

Scrum is a project management technique that is actively used in software development. Based on Scrum, it is possible to implement a flexible approach to complex product development.

Many people call Scrum a methodology, but actually, it is a framework. The rules in both concepts differ: a methodology provides ready instructions on how to achieve results, but the framework does not. People follow the rules of Scrum, but they make decisions on various work processes themselves.

### What is Scrum? Definition and Meaning

Scrum is a method of putting Agile principles into practice. It is based on an autonomous team (5 to 9 people) who make independent decisions about how to carry out a task.
It should be a cross-functional team, meaning that its members should possess enough knowledge and skills to fully perform the project from the step where an idea is voiced to the final stage where the ready product is released.

Two more people have to join the team: a product owner and a Scrum master. The former connects the team with the customer and monitors the project development – this person is not a formal team leader but rather a curator. The latter helps the former implement a business process.

This is not a rigid pattern of work on a product, but a technology that allows changes to be made in the work process and provides the customer with the work result in steps based on importance. The product features are built step by step, meaning that the most necessary and basic things are done first, and all the rest improvements follow them.

### Scrum Artifacts and Rituals (Processes)

The Scrum approach divides the work process into equal sprints – they are usually time periods from a week to a month, depending on the project and team. A sprint is always limited in time (1 to 4 weeks) and has the same duration throughout the entire product life. When the sprint is over, a new working build of the product is ready.

Sprints are very convenient for comparison and allow managing work efficiency. In turn, retrospective meetings allow keeping everything under control.

The team creates artifacts in Scrum. They are special objects which help the team manage their work and launch a working product:

* Product Backlog
* Sprint Backlog
* Sprint Goal
* Sprint Burndown Chart.

The team regularly updates artifacts based on changes occurring in the market, feedback, or new ideas.

For visualization, Scrum uses boards: physical and electronic. The board is used to make the project more transparent, to plan the tasks and set limits. This approach is crucial when the team has no formal leader.

The board is divided into columns. Each column is a task stage (“Development”, “Testing”, and “Release”). The number of columns depends on the project, but the fewer the better.



The cards are tasks. Each has a description, importance, and priority. When the task goes through the next stage, it is pasted into the appropriate column. Taking a quick glance at the board, you comprehend how things are going with the project as a whole and with each separate task.

### What are Scrum Roles?

There are three main roles in Scrum, and the entire work process is built on the interaction among them.

1. **The Scrum team** is a well-organized group of professionals who understand the principles of Scrum and manage their work based on them. The team should be small: 7 – 9 people. If there are more people, the team members will spend too many resources on communication.
2. **The product owner** is the person who is responsible for final product reliability and performance; they understand how this product should look and work. They work on the customer side, prioritize tasks, and interact with the team.
3. **The Scrum master** is a person who can be called a project manager, although this is not exactly the case. They monitor the correct method application, support the morale of the team, and manage the work processes.

The Scrum master is essentially a guardian of traditions and a ceremonial master. He ensures that the team follows the language of Scrum and the spirit of Agile.

### Agile vs Scrum: What’s the difference?

Agile is not a separate methodology but a whole software development philosophy. Scrum is one of the Agile frameworks, a formalized methodology for working on projects.
We can visually present similarities and differences between Scrum and Agile in the following table:



### Advantages and Disadvantages of Scrum in Software Development

It is a win-win approach, from which both the team and the customer will benefit. Here are its distinct advantages:

* **High transparency level.** The team practices open data exchange, and everyone feels involved in a common goal. The customer is always up to date;
* **Team autonomy.** The team members decide how to work on the project themselves; the freedom of action and responsibility motivate them;
* **Motivating result.** Each team member can see their own and general achievements on a daily basis. The customer gets an increase in functionality with every subsequent iteration;
* **Market risks minimization.** The team quickly responds to changing project requirements and does not do any extra work;
* **Financial risks minimization.** Little time and money are spent to eliminate bugs and add new functionality; no one exceeds the budget.

However, Scrum is a formalized approach, and for some projects, it is not easy to apply. Here are its obvious disadvantages:

* not suitable for projects with vague requirements for the final product, as the customer may want to endlessly increase the functionality;
* it is difficult to learn to properly prioritize and estimate tasks;
* project success is highly dependent on the Scrum master;
* it is difficult to use Scrum in large-scale projects;
* high-performance teams can feel at ease and stop improving. The problem is clearly seen when the performance dynamics of sprints is decreased; the Scrum master has to show the team how serious the problem is.

### What Type of Work is Scrum Most Suitable for?

Each methodology solves its particular problem. Consequently, the choice always depends on the goals you pursue in each particular project.
Scrum is a good choice for:

* nontrivial, long, and complex projects developing in the process;
* when it is better to develop the product stagewise; no one knows precisely what is necessary, but in the end, the clients want to get what they need and not what turned out;
* startups, as it allows you to avoid spending several months creating a solution, but quickly launch the [MVP (Minimal Viable Product)](https://lvivity.com/minimum-viable-product) and quickly enhance it after feedback from users is received;
* when everyone knows that some components of the future product and priorities may change in the development process (new ideas appear, new opportunities are discovered, the customer appears to have other needs).

Scrum is a really cool development process if the customer trusts and understands the team and if the team works in a sound environment. However, in a real-case scenario, separate methods are rare in their pure form. Most often, companies combine those parts of flexible systems which are most suitable for them.

Scrum allows developing such valuable qualities in the employees as proactivity, self-sufficiency, self-discipline, interpersonal skills, and vision. Resorting to the methodology, the team gets the rhythm and understands whether it met the deadline or not. The real work speed becomes well understood.