Agile software development is a set of principles for software development in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. Agile itself has never defined any specific methods to achieve this, but many have grown up as a result and have been recognized as being 'Agile'.

What is Scrum?

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a process.

In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented.

This is why in Scrum development, for example, a sprint planning meeting is described in terms of the desired outcome (a commitment to a set of features to be developed in the next sprint) instead of a set of Entry criteria, Task definitions, Validation criteria, Exit criteria (ETVX) and so on, as would be provided in most methodologies.

Scrum relies on a self-organizing, cross-functional team. The scrum team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved. Those are issues that are decided by the team as a whole.

And in Scrum, a team is cross functional, meaning everyone is needed to take a feature from idea to implementation.

Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level.

The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

Scrum Development: What's involved?

The Scrum model suggests that projects progress via a series of sprints. In keeping with an agile methodology, sprints are timeboxed to no more than a month long, most commonly two weeks.

Scrum methodology advocates for a planning meeting at the start of the sprint, where team members figure out how many items they can commit to, and then create a sprint backlog – a list of the tasks to perform during the sprint.

During an agile Scrum sprint, the Scrum team takes a small set of features from idea to coded and tested functionality. At the end, these features are done, meaning coded, tested and integrated into the evolving product or system.

On each day of the sprint, all team members should attend a daily Scrum meeting, including the ScrumMaster and the product owner. This meeting is timeboxed to no more than 15 minutes. During that time, team members share what they worked on the prior day, will work on that day, and identify any obstacles to progress.
The Scrum model sees daily scrums as a way to synchronize the work of team members as they discuss the work of the sprint.

At the end of a sprint, the team conducts a sprint review during which the team demonstrates the new functionality to the PO or any other stakeholder who wishes to provide feedback that could influence the next sprint.

This feedback loop within Scrum software development may result in changes to the freshly delivered functionality, but it may just as likely result in revising or adding items to the product backlog.

Another activity in Scrum project management is the sprint retrospective at the end of each sprint. The whole team participates in this meeting, including the ScrumMaster and PO. The meeting is an opportunity to reflect on the sprint that has ended, and identify opportunities to improve.

Scrum Process: Main Artifacts

The primary artifact in Scrum development is, of course, the product itself. The Scrum model expects the team to bring the product or system to a potentially shippable state at the end of each Scrum sprint.

The product backlog is another artifact of Scrum. This is the complete list of the functionality that remains to be added to the product. The product owner prioritizes the backlog so the team always works on the most valuable features first.

The most popular and successful way to create a product backlog using Scrum methodology is to populate it with user stories, which are short descriptions of functionality described from the perspective of a user or customer.

In Scrum project management, on the first day of a sprint and during the planning meeting, team members create the sprint backlog. The sprint backlog can be thought of as the team's to-do list for the sprint, whereas a product backlog is a list of features to be built (written in the form of user stories).

The sprint backlog is the list of tasks the team needs to perform in order to deliver the functionality it committed to deliver during the sprint.

Additional artifacts resulting from the Scrum agile methodology is the sprint burndown chart and release burndown chart. Burndown charts show the amount of work remaining either in a sprint or a release, and are an effective tool in Scrum software development to determine whether a sprint or release is on schedule to have all planned work finished by the desired date.

The Agile Scrum Project: Main Roles

Even if you are new to Scrum, you may have heard of a role called the ScrumMaster. The ScrumMaster is the team's coach, and helps Scrum practitioners achieve their highest level of performance.

In the Scrum process, a ScrumMaster differs from a traditional project manager in many ways, including that this role does not provide day-to-day direction to the team and does not assign tasks to individuals.

A good ScrumMaster shelters the team from outside distractions, allowing team members to focus maniacally during the sprint on the goal they have selected.

While the ScrumMaster focuses on helping the team be the best that it can be, the product owner works to direct the team to the right goal. The product owner does this by creating a compelling vision of the product, and then conveying that vision to the team through the product backlog.

The product owner is responsible for prioritizing the backlog during Scrum development, to ensure it’s up to par as more is learned about the system being built, its users, the team and so on.

The third and final role in Scrum project management is the Scrum team itself. Although individuals may join the team with various job titles, in Scrum, those titles are insignificant. Scrum methodology states that each person contributes in whatever way they can to complete the work of each sprint.

This does not mean that a tester will be expected to re-architect the system; individuals will spend most (and sometimes all) of their time working in whatever discipline they worked before adopting the agile Scrum model. But with Scrum, individuals are expected to work beyond their preferred disciplines whenever doing so would be for the good of the team.

One way to think of the interlocking nature of these three roles in this agile methodology is as a race car.
The Scrum team is the car itself, ready to speed along in whatever direction it is pointed. The product owner is the driver, making sure that the car is always going in the right direction. And the ScrumMaster is the chief mechanic, keeping the car well-tuned and performing at its best.