

What is Scrum?

Scrum is an iterative and incremental agile software development method for managing software projects and product or application development.

Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.

It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).

The business sets the priorities. Our teams self-manage to determine the best way to deliver the highest priority features.

Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance for another iteration.

What is the difference between agile and scrum development?

Agile and Scrum are terms used in project management. The Agile methodology employs incremental and iterative work beat that are also called sprints. Scrum, on the other hand is the type of agile approach that is used in software development. Agile itself is not a specific way of working, it is generic set of principles or priorities which are outlined in the Agile Manifesto. Scrum is a specific flavor of Agile, specifically it is referred to as an agile project management framework. It draws on the principles of the Agile Manifesto but goes into detail to define day-to-day activities and how to manage a project in a specific way.

What are the advantages of Scrum?

The sprint process allows for "good enough" development that results in a saleable product even while the project is in full swing. This incremental delivery system shortens the time to market and may result in higher revenue, as each completed backlog represents a new release of the product. In addition, reviewing each sprint before moving to the next means that testing is conducted throughout the process, which allows teams to change the scope or direction of the project at any point. Although the deadline and budget are fixed variables, the project requirements are not. In fact, stakeholders and participants anticipate changes along the way. The product owner's involvement in the project management process facilitates these changes.

Well, Scrum enables Agility. Three key benefits of Scrum adoption for you are ability to

- Respond to changes, while minimizing risk
- Increase ROI (return on investment)

- Continuously improve

What are the dis-advantages of Scrum?

It can be difficult for the Scrum master to plan, structure and organize a project that lacks a clear definition. In addition, frequent changes, frequent product delivery and uncertainty regarding the precise nature of the finished product make for a rather intense project life cycle for everyone involved. Furthermore, the daily Scrum meetings and frequent reviews require substantial resources. A successful project depends on the maturity and dedication of all participants, and their ability to maintain consistently high levels of communication through each backlog and review.

What is the difference between Scrum and Extreme Programming?

- There are major four differences between Scrum and Extreme Programming:
1. Scrum teams typically work in iterations (called sprints) that are from two weeks to one month long. XP teams typically work in iterations that are one or two weeks long.
 2. Scrum teams do not allow changes into their sprints. Once the sprint planning meeting is completed and a commitment made to delivering a set of product backlog items, that set of items remains unchanged through the end of the sprint. XP teams are much more amenable to change within their iterations. As long as the team hasn't started work on a particular feature, a new feature of equivalent size can be swapped into the XP team's iteration in exchange for the unstarted feature.
 3. Extreme Programming teams work in a strict priority order. Features to be developed are prioritized by the customer (Scrum's Product Owner) and the team is required to work on them in that order. By contrast, the Scrum product owner prioritizes the product backlog but the team determines the sequence in which they will develop the backlog items. I've never seen a Scrum team not choose to work on the highest-priority item. And a Scrum team will very likely choose to work on the second most important. However, at some point one of the high priority items may not be a good fit for the sprint being planned—maybe a key person who should work on it will be swamped by work on higher priority items. Or maybe it makes sense to work on a slightly lower priority item (let's say #10 on the product backlog instead of #6) because the team will be working in the code where #10 would be implemented.
 4. Scrum doesn't prescribe any engineering practices; XP does. I love the XP engineering practices, particularly things like test-driven development, the focus on automated testing, pair programming, simple design, refactoring, and so on. However, I think it's a mistake to say to the team "you're self-organizing, we trust you, but you must do these specific engineering practices." This sends a mixed message to the team that causes confusion. I love the XP practices but don't like mandating them. I want teams to discover the value on their own.

What is a Sprint?

A sprint is the basic unit of development in Scrum. Sprints last between one week and one month, and are a "timeboxed" (i.e. restricted to a specific duration) effort of a constant length. Each sprint is preceded by a planning meeting, where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made, and followed by a review or retrospective meeting, where the progress is reviewed and lessons for the next sprint are identified.

What are types of roles in scrum?

Scrum teams consist of two types of roles:

1. Main / Core Roles
2. Ancillary Roles

Main / Core roles are often referred to as **pigs** and ancillary roles as **chickens** (after the story "The Chicken and the Pig")

What are main / core roles in Scrum?

The three main / core roles in Scrum are:

1. **Scrum Master:** Who ensures the process is followed, removes impediments, and protects the Development Team from disruption.

The ScrumMaster acts as a facilitator for the Product Owner and the team. The ScrumMaster does not manage the team. Instead, he or she works to remove any impediments that are obstructing the team from achieving its sprint goals. In short, this role helps the team remain creative and productive, while making sure its successes are visible to the [Product Owner](#). The ScrumMaster also works to advise the [Product Owner](#) about how to maximize ROI for the team.

2. **Product Owner:** Who represents the stakeholders and the business

There are three fundamental roles in the Scrum method of [agile](#) software development: the Product Owner, the ScrumMaster, and the team. I'll begin by discussing the Product Owner because it is the most demanding of the roles.

In Scrum, the Product Owner is the one person responsible for a project's success. The Product Owner leads the development effort by conveying his or her vision to the team, outlining work in the [scrum backlog](#), and prioritizing it based on business value. Of course, he or she must also consider the stakeholders (to make sure their interests are included in the release) and the team (to make sure the release is developed by the deadline and within budget). As such, the Product Owner must be available to the team to answer questions and deliver direction.

But this combination of authority and availability to the development team makes it hard for the Scrum Product Owner not to micro-manage. Scrum values self-organization and, as a result, the Product Owner must respect the team's ability to create its own plan of action. This means that a Product Owner is forbidden to give the team more work in the middle of the sprint. Even if requirements change or a

rival organization unveils a new product that renders the team's work all for naught, the Scrum Product Owner is discouraged from altering the sprint until the next sprint planning meeting. However, the Product Owner may *cancel* a Sprint when necessary. One Product Owner I know cancels Sprints once or twice per year tops.

Furthermore, it is the Product Owner's responsibility to consider which activities will produce the most business value. This means making tough decisions—that the team might not appreciate—during sprint planning. However, the Product Owner is the one person who must face the music if the project crashes and burns.

Therefore, he or she must aggressively determine which features of a product are most important, when they are developed, etc. Just as the development team must produce the negotiated work for the Product Owner, the Product Owner must deliver the product to the customer.

3. **Development Team:** A cross-functional, self-organizing team who do the actual analysis, design, implementation, testing, etc.

In the Scrum methodology, the team is responsible for completing work. Ideally, teams consist of seven cross-functional members, plus or minus two individuals. For software projects, a typical team includes a mix of software engineers, architects, programmers, analysts, QA experts, testers, and UI designers. Each sprint, the team is responsible for determining how it will accomplish the work to be completed. This grants teams a great deal of autonomy, but, similar to the Product Owner's situation, that freedom is accompanied by a responsibility to meet the goals of the sprint.

What are Ancillary roles?

The ancillary roles in Scrum teams are those with no formal role and infrequent involvement in the Scrum process—but nonetheless, they must be taken into account.

- **Stakeholders (customers, vendors):**

These are the people who enable the project and for whom the project produces the agreed-upon benefit[s] that justify its production. They are only directly involved in the process during the sprint reviews.

- **Managers**

People who control the environment.

What are the types of meetings in scrum?

There 6 types of meetings in scrum:

- Daily Scrum / Standup
- Backlog grooming: storytime
- Scrum of Scrums
- Sprint planning meeting
- Sprint review meeting
- Sprint retrospective

What are Daily Scrum / Standup?

Each day during the sprint, a project status meeting occurs. This is called a daily scrum, or the daily standup.

What are 3 questions in Daily Scrum / Standup?

The three questions in Daily Scrum / Standup are:

- What have you done since yesterday?
- What are you planning to do today?
- Any impediments/stumbling blocks?

What is Backlog grooming / story time?

The team should spend time during a sprint doing backlog grooming. This is the process of: estimating the existing backlog using effort/points, refining the acceptance criteria for individual stories, and breaking larger stories into smaller stories.

What is Scrum of Scrums?

It is a meeting each day normally after the Daily Scrum.

- These meetings allow clusters of teams to discuss their work, focusing especially on areas of overlap and integration.
- A designated person from each team attends.

The agenda will be the same as the Daily Scrum, plus the following four questions:

- What has your team done since we last met?
- What will your team do before we meet again?
- Is anything slowing your team down or getting in their way?
- Are you about to put something in another team's way?

What is Sprint planning meeting?

At the beginning of the sprint cycle (every 7-30 days), a Sprint planning meeting is held.

- Select what work is to be done
- Prepare the Sprint Backlog that details the time it will take to do that work, with the entire team
- Identify and communicate how much of the work is likely to be done during the current sprint
- Eight-hour time limit
 - (1st four hours) Product Owner + Team: dialog for prioritizing the Product Backlog
 - (2nd four hours) Team only: hashing out a plan for the Sprint, resulting in the Sprint Backlog

What is Sprint review meeting?

At the end of a sprint cycle, the ½ Sprint Review Meeting ½ is held.

- Review the work that was completed and not completed
- Present the completed work to the stakeholders (a.k.a. ½ the demo ½)
- Incomplete work cannot be demonstrated
- Four-hour time limit

What is Sprint retrospective?

At the end of a sprint cycle, two meetings are held: the ½ Sprint Retrospective ½ is held.

- All team members reflect on the past sprint
- Make continuous process improvements
- Two main questions are asked in the sprint retrospective: What went well during the sprint? What could be improved in the next sprint?
- Three-hour time limit

What is Product Backlog?

The product backlog is an ordered list of "requirements" that is maintained for a product. It contains Product Backlog Items that are ordered by the Product Owner based on considerations like risk, business value, dependencies, date needed, etc. The features added to the backlog are commonly written in story format. The product backlog is the ½ What ½ that will be built, sorted in the relative order it should be built in. It is open and editable by anyone, but the Product Owner is ultimately responsible for ordering the stories on the backlog for the Development Team. The product backlog contains rough estimates of business value and development effort, these values are often stated in story points using a rounded Fibonacci sequence. Those estimates help the Product Owner to gauge the timeline and may influence ordering of backlog items. For example, if the ½ add spellcheck ½ and ½ add table support ½ features have the same business value, the one with the smallest development effort will probably have higher priority, because the ROI (Return on Investment) is higher.

The Product Backlog, and business value of each listed item is the responsibility of the Product Owner. The estimated effort to complete each backlog item is, however, determined by the Development Team.

What is Sprint Backlog?

The sprint backlog is the list of work the Development Team must address during the next sprint. The list is derived by selecting stories/features from the top of the product backlog until the Development Team feels it has enough work to fill the sprint. This is done by the Development Team asking "Can we also do this?" and adding stories/features to the sprint backlog. The Development Team should keep in mind the velocity of its previous Sprints (total story points completed from each of the last sprints stories) when selecting stories/features for the new sprint, and use this number as a guide line of how much "effort" they can complete.

The stories/features are broken down into tasks by the Development Team, which, as a best practice, should normally be between four and sixteen hours of work. With this level of detail the Development Team understands exactly what to do, and potentially, anyone can pick a task from the list. Tasks on the sprint backlog are never assigned; rather, tasks are signed up for by the team members as needed during the daily scrum, according to the set priority and the Development Team member skills. This promotes self-organization of the Development Team, and developer buy-in.

The sprint backlog is the property of the Development Team, and all included estimates are provided by the Development Team. Often an accompanying task board is used to see and change the state of the tasks of the current sprint, like "to do", "in progress" and "done".

What is Increment?

The increment is the sum of all the Product Backlog Items completed during a sprint and all previous sprints. At the end of a sprint, the Increment must be done according to the Scrum Team's definition of done. The increment must be in useable condition regardless of whether the Product Owner decides to actually release it.

What is Burn down?

The sprint burn down chart is a publicly displayed chart showing remaining work in the sprint backlog. Updated every day, it gives a simple view of the sprint progress. It also provides quick visualizations for reference. There are also other types of burndown, for example the release burndown chart that shows the amount of work left to complete the target commitment for a Product Release (normally spanning through multiple iterations) and the alternative release burndown chart,[16] which basically does the same, but clearly shows scope changes to Release Content, by resetting the baseline.

What is User Story?

A feature that is added to the backlog is commonly referred to as a story and has a specific suggested structure. The structure of a story is: "As a I want to so that " This is done so that the development team can identify the user, action and required result in a request and is a simple way of writing requests that anyone can understand. Example: As a wiki user I want a tools menu on the edit screen so that I can easily apply font formatting.

A story is an independent, negotiable, valuable, estimatable, small, testable requirement ("INVEST Acronym"). Despite being independent i.e. they have no direct dependencies with other requirements, stories may be clustered into epics when represented on a product roadmap or further down in the backlog.

What is Theme?

A theme is a top-level objective that may span projects and products. Themes may be broken down into sub-themes, which are more likely to be product-specific. Themes can be used at both program and project level to drive strategic alignment and communicate a clear direction.

What is Epic?

An epic is a group of related stories, mainly used in product roadmaps and the backlog for features that have not yet been analyzed enough to break it down into its component stories, which should be done before bringing it into a sprint so to reduce uncertainty. Epics can also be used at a both program and project level.

What is Spike?

A time boxed period used to research a concept and/or create a simple prototype. Spikes can either be planned to take place in between sprints or, for larger teams, a spike might be accepted as one of many sprint delivery objectives. Spikes are often introduced before the delivery of large epics or user stories in order to secure budget, expand knowledge, and/or produce a proof of concept. The duration and objective(s) of a spike will be agreed between the Product Owner and Delivery Team before the start. Unlike sprint commitments, spikes may or may not deliver tangible, shippable, valuable functionality. For example, the objective of a spike might be to successfully reach a decision on a course of action. The spike is over when the time is up, not necessarily when the objective has been delivered.

What is Tracer Bullet?

The tracer bullet is a spike with the current architecture, current technology set, current set of best practices which results in production quality code. It might just be a very narrow implementation of the functionality but is not throw away code. It is of production quality and rest of the iterations can build on this code.

What is Point Scale/Effort/Story points?

Relates to an abstract point system, used to discuss the difficulty of the story, without assigning actual hours. The most common scale used is a rounded Fibonacci sequence (1,2,3,5,8,13,20,40,100), although some teams use linear scale (1,2,3,4...), Powers-of-2 (1,2,4,8...), and Clothes size (XS, S, M, L, XL).

What are Tasks?

Added to the story at the beginning of a sprint and broken down into hours. Each task should not exceed 12 hours, but it's common for teams to insist that a task take no more than a day to finish.

What is Definition of Done (DoD)?

The exit-criteria to determine whether a product backlog item is complete. In many cases the DoD requires that all regression tests should be successful.

What is Velocity?

The total effort a team is capable of in a sprint. The number is derived by adding all the story points from the last sprint's stories/features. This is a guideline for the team and assists them in understanding how many stories they can do in a sprint.

What is Impediment?

Anything that prevents a team member from performing work as efficiently as possible.

What is Sashimi?

A report that something is "done". The definition of "done" may vary from one Scrum team to another, but must be consistent within one team.

What is Abnormal Termination?

The Product Owner can cancel a Sprint if necessary. The Product Owner may do so with input from the team, scrum master or management. For instance, management may wish to cancel a sprint if external circumstances negate the value of the sprint goal. If a sprint is abnormally terminated, the next step is to conduct a new Sprint planning meeting, where the reason for the termination is reviewed.

What is Planning Poker?

In the Sprint Planning Meeting, the team sits down to estimate its effort for the stories in the backlog. The Product Owner needs these estimates, so that he or she is empowered to effectively prioritize items in the backlog and, as a result, forecast releases based on the team's velocity.

What is Scrum-ban?

Scrum-ban is a software production model based on Scrum and Kanban. Scrum-ban is especially suited for maintenance projects or (system) projects with frequent and unexpected user stories or programming errors. In such cases the time-limited sprints of the Scrum model are of no appreciable use, but Scrum's daily meetings and other practices can be applied, depending on the team and the situation at hand. Visualization of the work stages and limitations for simultaneous unfinished user stories and defects are familiar from the Kanban model. Using these methods, the team's workflow is directed in a way that allows for minimum

completion time for each user story or programming error, and on the other hand ensures each team member is constantly employed.