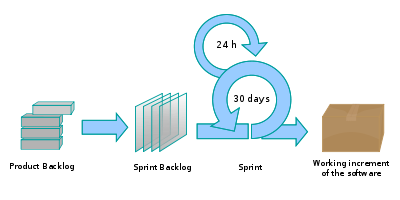
[](http://en.wikipedia.org/wiki/File:Scrum_process.svg)

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

**Characteristics**

Scrum contains sets of methods and predefined roles. The main roles in Scrum are:

1. the "Scrum Master", who ensures the process is followed, removes impediments, and protects the Development Team from disruption
2. the "Product Owner", who represents the stakeholders and the business
3. the "Development Team", a cross-functional, self-organizing team who do the actual analysis, design, implementation, testing, etc.

**Sprint**

A sprint is the basic unit of development in Scrum. Sprints last between one week and one month and are a "[timeboxed](http://en.wikipedia.org/wiki/Timeboxing" \o "Timeboxing)" (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.

During each sprint, the team creates finished portions of a product. The set of features that go into a sprint come from the product *backlog*, which is a prioritized list of requirements. Which backlog items go into the sprint is determined during the sprint planning meeting. During this meeting, the Product Owner informs the team of the items in the product backlog that he or she wants completed (the ones with the highest priority). The team then determines how much of this they can commit to complete during the next sprint, and records this in the sprint backlog. During a sprint, no one is allowed to change the sprint backlog, which means that the requirements are frozen for that sprint. Development is [timeboxed](http://en.wikipedia.org/wiki/Timeboxing) such that the sprint must end on time; if requirements are not completed for any reason they are left out and returned to the product backlog. After a sprint is completed, the team demonstrates how to use the software.

Scrum enables the creation of self-organizing teams by encouraging co-location of all team members, and verbal communication between all team members and disciplines in the project.

A key principle of Scrum is its recognition that during a project the customers can change their minds about what they want and need (often called requirements churn), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team’s ability to deliver quickly and respond to emerging requirements.

Like other agile development methodologies, Scrum can be implemented through a wide range of tools. Many companies use universal tools, such as spreadsheets to build and maintain artifacts such as the sprint backlog. There are also open-source and proprietary packages dedicated to management of products under the Scrum process. Other organizations implement Scrum without the use of any tools, and maintain their artifacts in hard-copy forms such as paper, whiteboards, and sticky notes.

**Roles**

Scrum teams consist of three core roles and a range of ancillary roles—core roles are often referred to as *pigs* and ancillary roles as *chickens* (after the story The Chicken and the Pig).

**Core roles**

The core roles in Scrum teams are those committed to the project in the Scrum process—they are the ones producing the product (objective of the project).

Product Owner

The Product Owner represents the voice of the customer and is accountable for ensuring that the team delivers value to the business. The Product Owner writes customer-centric items (typically user stories), prioritizes them, and adds them to the product backlog. Scrum teams should have one Product Owner, and while they may also be a member of the development team, it is recommended that this role not be combined with that of Scrum Master.

Development Team

The Development Team is responsible for delivering potentially shippable product increments at the end of each Sprint. A Development Team is made up of 3–9 people with cross-functional skills who do the actual work (analyze, design, develop, test, technical communication, document, etc.). The Development Team in Scrum is self-organizing, even though they may interface with project management organizations (PMOs).

Scrum Master

Scrum is facilitated by a Scrum Master, sometimes written as *Scrum Master*, who is accountable for removing impediments to the ability of the team to deliver the sprint goal/deliverables. The Scrum Master is not the team leader, but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum process is used as intended. The Scrum Master is the enforcer of rules. A key part of the Scrum Master’s role is to protect the Development Team and keep it focused on the tasks at hand. The role has also been referred to as a *servant-leader* to reinforce these dual perspectives.

**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.

**Agile project management with scrum**

Scrum has not only reinforced the interest in project management, but also challenged the conventional ideas about such management. Scrum focuses on project management institutions where it is difficult to plan ahead. Mechanisms of *empirical process control*, where feedback loops constitute the core management technique are used as opposed to traditional command-and-control oriented management.] It represents a radically new approach for planning and managing projects, bringing decision-making authority to the level of operation properties and certainties.

**Meetings**

**Daily Scrum**

Each day during the sprint, a project status meeting occurs. This is called a *daily scrum*, or *the daily standup*. This meeting has specific guidelines:

* The meeting starts precisely on time.
* All are welcome, but normally only the core roles speak
* The meeting length is set to 15 minutes
* The meeting should happen at the same location and same time every day

During the meeting, each team member answers three questions:

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

It is the role of the Scrum Master to facilitate resolution of these impediments, although the resolution should occur outside the Daily Scrum itself to keep it under 15 minutes.

**Backlog grooming: storytime**

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.

* Meetings should not be longer than an hour
* Meeting does not include breaking stories into tasks
* The team can decide how many meetings are needed per week.

**Scrum of Scrums**

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?

**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

At the end of a sprint cycle, two meetings are held: the “Sprint Review Meeting” and the “Sprint Retrospective”

**Sprint review meeting**

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

**Sprint retrospective**

* 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

**Artifacts**

**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 (See terminology below). The product backlog is the “What” 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 both 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 “add spellcheck” and “add table support” 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.

**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.

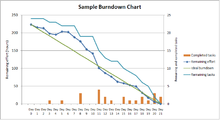
The stories/features are broken down into tasks by the Development Team, which, as a best practice, should normally be between four and sixteen hours of work. With this level of detail the Development Team understands exactly what to do, and potentially, anyone can pick a task from the list. Tasks on 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”.

**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.

**Burn down**

[](http://en.wikipedia.org/wiki/File:SampleBurndownChart.png)

[http://bits.wikimedia.org/static-1.20wmf2/skins/common/images/magnify-clip.png](http://en.wikipedia.org/wiki/File:SampleBurndownChart.png)

A sample burn down chart for a completed iteration, showing remaining effort and tasks for each of the 21 work days of the 1-month iteration.

*Main article: burn down chart*

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**, which basically does the same, but clearly shows scope changes to Release Content, by resetting the baseline.

It should not be confused with an earned value chart.

**Terminology**

The following terminology is used in Scrum:

**Roles**

Scrum Team

Product Owner, Scrum Master and Development Team

Product Owner

The person responsible for maintaining the Product Backlog by representing the interests of the stakeholders, and ensuring the value of the work the Development Team does.

Scrum Master

The person responsible for the Scrum process, making sure it is used correctly and maximizing its benefits.

Development Team

A cross-functional group of people responsible for delivering potentially shippable increments of Product at the end of every Sprint.

**Artifacts**

Sprint burn down chart

Daily progress for a Sprint over the sprint’s length.

Product backlog

A prioritized list of high-level requirements.

Sprint backlog

A prioritized list of tasks to be completed during the sprint.

**Others**

Sprint

A time period (typically 1–4 weeks) in which development occurs on a set of backlog items that the team has committed to. Also commonly referred to as a Time-box or iteration.

(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 <user type> I want to <do some action> so that <desired result>" 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.

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.

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 it's 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.

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.

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.

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](http://en.wikipedia.org/wiki/Fibonacci_number) 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)

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.

Definition of Done (DoD)

The [exit-criteria](http://en.wikipedia.org/wiki/Exit-criteria) to determine whether a product backlog item is complete. In many cases the DoD requires that all regression tests should be successful.

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.

Impediment

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

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.

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.

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.