

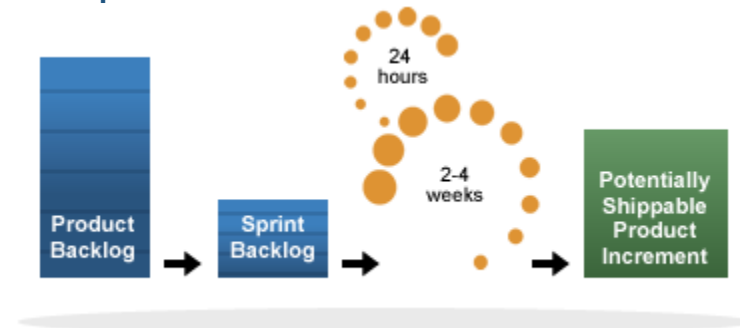
# What is Scrum?

From [http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum)

## A lean approach to software development

Scrum is an agile software development framework. Work is structured in cycles of work called **sprints**, iterations of work that are typically two to four weeks in duration. During each sprint, teams pull from a prioritized list of customer requirements, called **user stories**, so that the features that are developed first are of the highest value to the customer. At the end of each sprint, a **potentially shippable product** is delivered.

## A simple framework



Scrum is made up of three roles, three ceremonies, and three artifacts.

**Three roles:** the Product Owner, who is responsible for the business value of the project; the ScrumMaster, who ensures that the team is functional and productive; and the self-organized team.

**Three ceremonies:** the sprint planning meeting, daily scrum meetings, and sprint review meetings.

**Three artifacts** for prioritizing and tracking tasks: the product backlog, the sprint backlog, and a burndown chart.

## A time-tested process

When Jeff Sutherland created the scrum process in 1993, he borrowed the term "scrum" from an analogy put forth in a 1986 study by Takeuchi and Nonaka, published in the *Harvard Business Review*. In that study, Takeuchi and Nonaka compare high-performing, cross-functional teams to the scrum formation used by Rugby teams. Ken Schwaber formalized the process for the worldwide software industry in the first published paper on Scrum at OOPSLA 1995.

Since then, Scrum has become one of the leading agile development methodologies, used by Fortune 500 companies around the world, as you can see by glancing through the list of Scrum Alliance members and their companies.

# Scrum Roles

Scrum has three roles: Product Owner, ScrumMaster, and Team.

The **Product Owner** has the following responsibilities.

- Define the features of the product;
- Decide on release date and content;
- Be responsible for the profitability of the product (ROI);
- Prioritize features according to market value;
- Adjust features and priority every 30 days, as needed; and
- Accept or reject work results.

The product owner is responsible for the first of the three Scrum **ceremonies**: Scrum Planning.

The **ScrumMaster** is a facilitative team leader working closely with the Product Owner. He must:

- Ensure that the team is fully functional and productive;
- Enable close cooperation across all roles and functions;
- Remove barriers;
- Shield the team from external interferences; and
- Ensure that the process is followed, including issuing invitations to Daily Scrum, Sprint Review and Sprint Planning meetings.

The ScrumMaster has three primary responsibilities in addition to leading the Daily Scrum meeting:

1. The ScrumMaster needs to know what tasks have been completed, what tasks have started, any new tasks that have been discovered, and any estimates that may have changed. This makes it possible to update the Burndown Chart which shows the cumulative work remaining day by day. The ScrumMaster must also look carefully at the number of open tasks in progress. Work in progress needs to be minimized to achieve lean productivity gains.
2. The ScrumMaster needs to surface dependencies and blocks which are impediments to the Scrum. They need to be prioritized and tracked. A remediation plan needs to be implemented for impediments in priority order. Some can be resolved with the team, some can be resolved across teams, and others will need management involvement as they may be company issues that block all teams from achieving their production capacity. For example, a telecom company recently implemented Scrum and found eighteen items on their impediment list, only two of which were directly related to Scrum teams. The others were company issues that needed management attention.
3. Last but not least, the ScrumMaster may notice personal problems or conflicts within the Scrum that need resolution. These need to be clarified by the ScrumMaster and be resolved by dialogue within the team, or the ScrumMaster may need help from management or the Human Resources. Certified ScrumMaster James Coplien developed over 200 case studies of notable projects while working at ATT Bell Labs. He reports that over 50% of productivity losses were caused by personnel issues. The ScrumMaster must pay attention to them to ensure the team is fully functional and productive.

The **Team**:

- Is cross-functional, with seven (plus/minus two) members;
- Selects the Sprint goal and specifies work results;
- Has the right to do everything within the boundaries of the project guidelines to reach the Sprint goal;
- Organizes itself and its work; and
- Demos work results to the Product Owner.

## Scrum Ceremonies

Scrum has three ceremonies: Sprint Planning, Sprint Review, and the Daily Scrum Meeting.

### Sprint Planning Meeting

Preparation for a Scrum sprint begins when the Product Owner develops a plan for a product or a project. The Product Owner can be a customer representative or a customer proxy. For product companies, the customer is a market, and the Product Owner serves as a proxy for the market. A Product Owner needs a vision for the product that frames its ultimate purpose, a business plan that shows what revenue streams can be anticipated from the product in which timeframes, and a road map that plans out several releases, with features ordered by contribution to return on investment (ROI). S/he prepares a list of customer requirements prioritized by business

value. This list is the **Product Backlog**, a single list of features prioritized by value delivered to the customer. The Scrum begins when enough of the Product Backlog is defined and prioritized to launch the first thirty-day sprint. A Sprint Planning Meeting is used to develop a detailed plan for the iteration. It begins with the Product Owner reviewing the vision, the roadmap, the release plan, and the Product Backlog with the Scrum team. The team reviews the estimates for features on the Product Backlog and confirms that they are as accurate as possible. The team decides how much work it can successfully take into the sprint based on team size, available hours, and level of team productivity. It is important that the team "pull" items from the top of the Product Backlog that they can commit to deliver in a thirty-day sprint. Pull systems have been shown to deliver significant productivity gains in lean product development.

When the Scrum team has selected and committed to deliver a set of top priority features from the Product Backlog, the ScrumMaster leads the team in a planning session to break down Product Backlog features into sprint tasks. These are the specific development activities required to implement a feature and form the Sprint Backlog. This phase of the Sprint Planning Meeting is time-boxed to a maximum of four hours.

## Daily Scrum Meeting

Once planning is complete, the Sprint begins its thirty-day cycle. Each day the ScrumMaster leads the team in the Daily Scrum Meeting. This is a fifteen-minute meeting designed to clarify the state of the Scrum. Each team member speaks to three questions: what did I do yesterday, what did I do today, and what impediments got in my way? While anyone can attend this meeting, only team members who have committed to deliver work to the Scrum are allowed to speak. The goal is to get a global snapshot of the project, discover any new dependencies, address any personal needs of committed individuals, and adjust the work plan in real time to the needs of the day.

## Sprint Review Meeting

At the end of a sprint, a Sprint Review Meeting is held. This meeting is time-boxed to a maximum of four hours. The first half of the meeting is set aside to demonstrate to the Product Owner the potentially shippable code that has been developed during the sprint. The Product Owner leads this part of the meeting and invites all interested stakeholders to attend. The state of the business, the market, and the technology are reviewed. The Product Owner determines which items on the Product Backlog have been completed in the Sprint, and discusses with the Scrum team and stakeholders how best to reprioritize the Product Backlog for the next sprint. The goal for the next sprint is defined.

The second half of the Sprint Review Meeting is a retrospective for the Scrum team that is led by the ScrumMaster. The team assesses the way they worked together in the sprint and identifies positive ways of working together that can be encouraged as future practice. The team also identifies the things that could work better and develops strategies for improvement.

After the Scrum Review Meeting, the process begins again. Iterations proceed until enough features have been done to complete or release a product.

## Scrum Artifacts

Scrum has three artifacts: the Product Backlog, the Sprint Backlog, and a Burndown Chart.

### Product Backlog

At the beginning of the project, the product owner prepares a list of customer requirements prioritized by business value. This list is the Product Backlog, a single list of features prioritized by value delivered to the customer. The

Scrum Team contributes to the product backlog by estimating the cost of developing features.

The Product Backlog should include all features visible to the customer, as well as the technical requirements needed to build the product. The highest priority items in the Product Backlog need to be broken down into small enough chunks to be estimable and testable. About ten developer-days of work is a good size for a Product Backlog item that can be ready for implementation in the next iteration. Features that will be implemented further out in time can be less detailed.

## Sprint Backlog

The Sprint Backlog is an artifact of the Sprint Planning Meeting. When the Scrum Team has selected and committed to deliver a set of top priority features from the Product Backlog, the Product Backlog's features are broken down into a Sprint Backlog: a list of the specific development tasks required to implement a feature. These tasks are broken down into pieces that will require less than two days (or sixteen developer-hours) of work. When the Sprint Backlog is complete, the total work estimated is compared with original estimates from the Product Backlog. If there is a significant difference, the team negotiates with the Product Owner to get the right amount of work to take into the Sprint with a high probability of success.

## Burndown Chart

The Burndown Chart shows the cumulative work remaining in a Sprint, day-by-day.

At the Sprint Planning Meeting the Scrum Team identifies and estimates specific tasks that must be completed for the Sprint to be successful. The total of all Sprint Backlog estimates of work remaining to be completed is the cumulative backlog. When tasks are completed as the Sprint proceeds, the ScrumMaster recalculates the remaining work to be done and the Sprint Backlog decreases, or burns down over time. If the cumulative Sprint Backlog is zero at the end of the Sprint, the Sprint is successful.

The Product Backlog items brought into the Sprint are fixed for the duration of the Sprint. However, the Sprint Backlog may change for several reasons:

- The development team gains a better understanding of work to be done as time progresses and may find that they need to add new tasks to the Sprint Backlog to complete the Product Backlog items selected.
- Defects may be identified and logged as additional tasks. While these are viewed primarily as unfinished work on committed tasks, it may be necessary to keep track of them separately.
- The Product Owner may work with the team during the Sprint to help refine team understanding of the Sprint goal. The ScrumMaster and Team may decide that minor adjustments that do not lengthen the Sprint are appropriate to optimize customer value.

The Burndown Chart is used as a tool to guide the development team to successful completion of a Sprint on time with working code that is potentially shippable as a product.

## Scrum Benefits

Successful Scrum implementations have many benefits for teams and management. Scrum does, however, require a change from the status quo.

A well-functioning Scrum will deliver the highest business value features first and will avoid building features that will never be used by the customer. Since industry data shows that about half of the software features developed are never used, development can be completed in half the time by avoiding waste, or unnecessary work.

In most companies, development is slowed down by issues identified as impediments during the daily meetings or planning and review meetings. With Scrum, these impediments are prioritized and systematically removed, further

increasing productivity and quality. Well-run Scrums achieve the Toyota effect: four times industry average productivity and twelve times better quality.

Scrum removes management pressure from teams. Teams are allowed to select their own work, and then self-organize through close communication and mutual agreement within the team on how best to accomplish the work. In a successful Scrum, this autonomy can significantly improve the quality of life for developers and enhance employee retention for managers.

The simple rules of Scrum allow for continual inspection, adaptation, self-organization, and emergence of innovation. This can produce an exciting product for the customer, develop high team spirit and satisfying work, generate high productivity and customer satisfaction, and achieve the market and financial goals of the company. As a result, Scrum is being widely adopted worldwide in companies large and small, localized or distributed, open source or proprietary, for virtually any type or size of project.

Certified ScrumMaster Trainer Craig Larman notes that "Scrum is arguably the oldest and most widely applied agile and iterative method, with an emphasis on iterative and adaptive PM practices." He goes on to say that Scrum has been applied in thousands of organizations and domains since the early 1990s, on projects large and small, from Yahoo to Medtronic to Primavera, with great results. These results can only happen, though, when leadership commits to the required changes: teams that adopt Scrum must move away from command-control and wishful-thinking-predictive management. Larman says that Scrum can be easily integrated with practices from other iterative methods, such as those from the Unified Process and Extreme Programming, including test-driven development, agile modeling, use cases, user stories, and so forth. Larman concludes, "On the surface, Scrum appears to be simple, but its emphasis on continuing inspect-adapt improvement cycles and self-organizing systems has subtle implications."