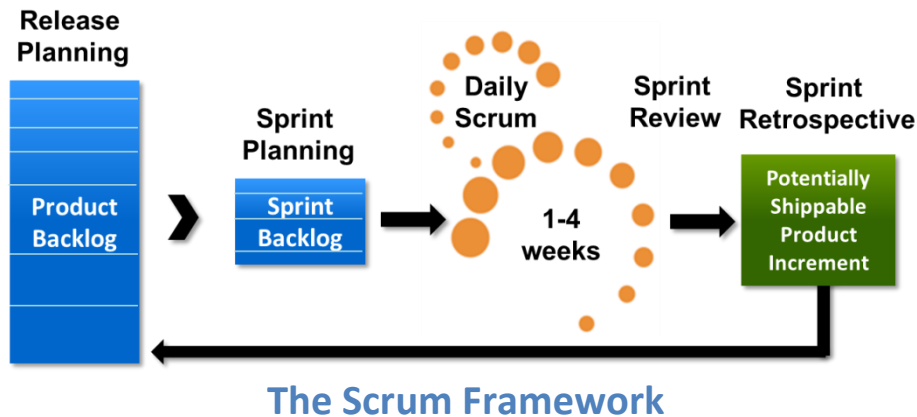


Introduction to Scrum

Scrum is framework for managing the work done by a team of people to build a product. It is typically associated with Agile software development but has been applied to many other types of products. Work is performed in iterations called Sprints. During each Sprint, teams pull from a prioritized list of customer-desired functionality. The highest value items are implemented and delivered first. At the end of each Sprint, a potentially shippable product is available for delivery.



Sprint length is determined early and kept constant for the duration of the product development. Typical lengths are 1 to 4 weeks.

Scrum has three roles: the Product Owner, who is responsible for the business value of the product; the ScrumMaster, who guides the team and facilitates the process; and the self-organized Development Team.

Scrum has four meetings: Sprint Planning, Daily Scrum, Sprint Retrospective and Sprint Review.

Scrum has three artifacts: the Product Backlog, the Sprint Backlog, and the Potentially Shippable Product Increment.

Origins

When Jeff Sutherland and his team created the Scrum framework in 1993, he borrowed the term "scrum" from an analogy put forth in a 1986 paper by Takeuchi and Nonaka, published in the *Harvard Business Review*. In that study, the authors 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.

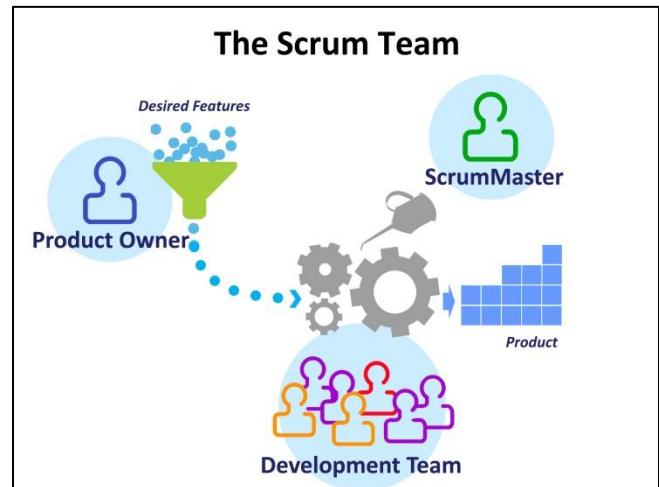
Since then, Scrum has become one of the leading agile development methodologies, used by Fortune 500 companies around the world.

Scrum Roles

Scrum has three roles: Product Owner, ScrumMaster, and Development Team. Together, these three roles form the **Scrum Team**.

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

- Define the features of the product
- Decide on release date and content
- Achieve maximum profitability of the product (Return on Investment - ROI)
- Prioritize features according to market value
- Adjust features and priority as needed
- Accept or reject work results
- Interface with business stakeholders
- Prepare the most important items on the Product Backlog for consideration at the Sprint Planning meeting



The **Team**, also known as the Development Team or Delivery Team is responsible for building the product with high quality. The Team

- Is cross-functional, with seven (plus/minus two) members
- Works together for many months, most members dedicated full-time to the Team
- Selects the Sprint goal in collaboration with the Product Owner
- Is empowered to make decisions on how work is done, who does it and how the process will evolve
- Organizes itself and its work
- Demonstrates work results to the Product Owner and Stakeholders

The **ScrumMaster** is a team guide and coach who works closely with the Product Owner and Development Team. Duties include

- Help the Team to be fully functional and productive
- Enable and enhance collaboration between all Scrum Team members
- Manage impediments to Team work flow
- Protect the Team from external interference
- Help the Team define and excel at their chosen process for getting work done
- Facilitate meetings

Scrum Preparation

Preparation for the first Scrum Sprint begins when the Product Owner develops a vision for a product or a project. The Product Owner represents the desires of the end-customer and/or business stakeholders for the product to be created or enhanced. For product companies, the customer is a market and the Product Owner serves as a proxy for the market. The Product Vision describes in simple terms its

ultimate purpose, who it is for, how it will create value and what benefits it will provide. The vision may encompass several Releases in a Product Roadmap.

The Development Team is identified and formed together with the Product Owner and ScrumMaster. Chartering of the Scrum Team is often done through a series of exercises to identify shared goals and agreements.

With help from the Development Team, ScrumMaster and other Stakeholders, the Product Vision is expressed in more detail in a Product Backlog, a single list of features prioritized primarily by value delivered to the customer. Creation of the initial Product Backlog may happen during an event called Release Planning. It is then revised continuously as priorities change during development. Items on the Product Backlog (Product Backlog Items or PBIs) are typically estimated by size during Release Planning to aid both long term and Sprint planning.

The Sprint

The Sprint is the time-box that contains the Scrum meetings and development work. A time-box is a fixed length of time allocated to accomplishing a specific goal. The Sprint is over when the time is up.

The bulk of the Team's time in a Sprint is spent building and validating the product. For example, in a 2 week Sprint there is one day of planning and review, nine days of product development.

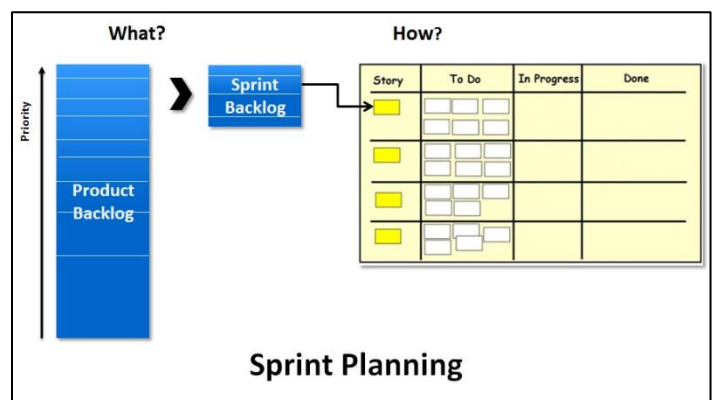
Note that the Sprint length does not vary. It is fixed by the Team and changed only in unusual circumstances. A fixed length allows the Team to develop a steady rhythm and to measure how much work can be accomplished in a Sprint.

Scrum Meetings

Scrum has four meetings, also called ceremonies: Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective.

Sprint Planning Meeting

The first Sprint begins when enough of the Product Backlog is defined and prioritized to give the Team sufficient work to do in the chosen time period. It starts with the Sprint Planning Meeting in which the Team develops a plan of what they will work on and how they will accomplish it. The meeting begins with the Product Owner reviewing the vision, the roadmap, the release plan and the Product Backlog with the Scrum team.



The team decides how much work it can successfully take into the Sprint based on available hours and the amount of product developed in previous Sprints, tracked using a measure called “velocity”. Sprint Planning is a “pull” process in which the Team plans for as much work as it can accomplish with high quality and at a sustainable pace.

When the Scrum team has selected and committed to give their best effort to deliver a set of top priority features from the Product Backlog, the ScrumMaster guides the team in a planning session to break the features into Sprint tasks. These are the specific development activities required to implement a feature. The resulting plan, called the Sprint Backlog, contains the features to be implemented and the tasks they will require. Sprint progress is then tracked in comparison to the plan as tasks and features are completed. Sprint Planning typically takes two to four hours depending on the length of the Sprint and the maturity of the Team.

Daily Scrum Meeting

Once planning is complete, the Sprint work begins. Each day the Team holds the Daily Scrum, a fifteen-minute meeting designed to clarify the state of the work. Each team member answers three questions:

- what did I do yesterday?
- what do I plan to do today?
- what impediments are blocking me?

While anyone can attend this meeting, only members of the Scrum Team are allowed to speak. The goal is to get a quick snapshot of the work, discover any new dependencies, address any personal needs of the Team Members 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 or “Demo” is held. The Development Team demonstrates the completed product increment to the Stakeholders to get their feedback. The Product Owner has already reviewed and accepted the work being demonstrated.

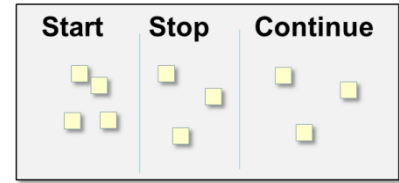


The Product Owner updates Stakeholders to the current progress of the Release. This meeting may also be used to review the state of the business, the market and the technology to assess possible shifts in emphasis or priority for the Release Plan. The Sprint Review typically takes 1 to 2 hours depending on the length of the Sprint.

Note: Some of the Scrum literature defines the Sprint Review as a single meeting that includes the Demo and the Retrospective. Since the two meetings have different purposes and attendees, most people now treat them as two separate meetings.

Sprint Retrospective Meeting

The final meeting is the Sprint Retrospective, facilitated by the ScrumMaster. In this private meeting the Team reviews their process and interactions during the Sprint to identify potential improvements. New ideas, interpersonal conflicts and impediments impacting the whole Team may be discussed. The goal is to identify ways to improve the work of the Team. A list of potential process changes is created and a plan of action is drawn up for the top few opportunities discovered. A typical meeting length is 1.5 – 2 hours for this meeting.



Gathering Retrospective Data

After the Scrum Retrospective, the process begins again with Sprint Planning from the current Product Backlog. Sprints proceed until enough functionality has been created to complete or release a product.

Scrum Artifacts

Scrum has three artifacts: the Product Backlog, the Sprint Backlog, and the Potentially Shippable Product Increment.

Product Backlog

At the beginning of product development, the Product Owner prepares a list of desired customer features prioritized by business value. This list is called the Product Backlog. The Development Team contributes to the Product Backlog by breaking features into smaller packets of value, estimating the size of each and offering feedback on development order from a technical viewpoint.

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. Features that will be implemented further out in time can be less detailed.

Scrum Teams often define the product features using User Stories, a device borrowed from Extreme Programming, another Agile development framework. User stories describe functionality in a simple, high-level format:

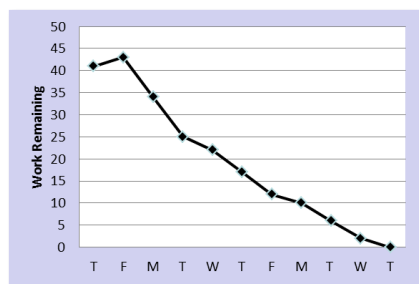
User Story

User stories remind us of two important concepts that help to maximize the ROI: each product increment must deliver value to the user and smaller chunks of work gives us a smoother flow from concept to customer feedback.

Items on the Product Backlog are typically estimated in units of relative size as an input to Sprint Planning and longer term Release Planning. A common unit is Story Points, a simple number that compares each PBI with all others in terms of size and complexity. The number of Story Points a Team actually completes in each Sprint is called “velocity”, a measure that can be averaged to forecast the amount of Product Backlog that is likely to be implemented in some future time period.

Sprint Backlog

The Sprint Backlog is the artifact created in the Sprint Planning Meeting. It is a subset of the Product Backlog, elaborated to more detail. When the Scrum Team has selected and set their goal to deliver a set of top priority features from the Product Backlog, the features are broken down into specific development tasks required for implementation. The tasks are defined in chunks that will likely take 1 to 8 hours of work for a Development Team member.



Sprint Burndown

The work remaining on the Sprint Backlog after each day of the Sprint is tracked on a Burndown Chart. When tasks are completed during the Sprint, the estimate of remaining work decreases, or burns down over time. If all work is completed in the Sprint, the cumulative Sprint Backlog is zero at the end. The Sprint Burndown chart is a simple but powerful device for seeing progress and trends as well as showing the Team how they are tracking toward the planned Sprint goal.

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 selected PBIs.
- 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 Scrum Team may decide that minor adjustments fitting within the Sprint time are appropriate to optimize customer value.
- The Scrum Team may add or remove PBIs to meet the Sprint goal at a sustainable pace.

Potentially Shippable Product Increment

The new product functionality created during a Sprint is called the Potentially Shippable Product Increment. Not all Scrum Teams actually ship the product after every Sprint. Rather, they accumulate a few Sprints worth of new functionality into a release that has sufficient value to justify the cost of packaging and deploying to the end-users. Each Sprint outcome must meet the high quality bar desired for Scrum. Each PBI must be completed to the satisfaction of the Team’s Definition of Done, a process and quality checklist, and achieve a part of the Product Owner’s vision. When these criteria are true, the product is potentially shippable, i.e. it works.

Release Planning

Release Planning is a process of building a Product Backlog with sufficient information to allow forecasting the answer to one of these questions:

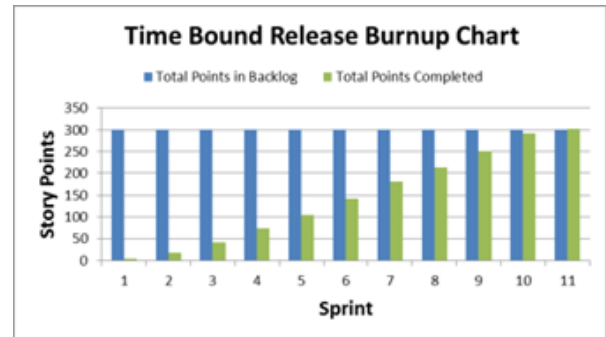
- When can we ship this backlog?
- How much of this backlog can we ship by a specified date?

Information needed for the forecast is developed during a Release Planning meeting that typically lasts from half a day to 2 days depending on how far into the future we are forecasting and how many teams are going to be working on the backlog. A typical agenda for this meeting is:

- Define and study the Product Vision
- Identify relevant User Roles for the product
- Brainstorm User Stories for those roles
- Prioritize the User Stories
- Estimate the size of the User Stories
- Forecast Team velocity

The forecast is then made by dividing the size of the backlog by the forecast velocity, realizing that there will be a fairly wide margin of error initially before true team velocity is known. A total labor cost for the release is then simple to calculate since Sprints have a constant cost.

The Release Plan is revised each sprint based on actual Team velocity and changes to the Product Backlog. Release progress is tracked on a Release Burnup or Burndown chart.



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 Team 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 has been widely adopted worldwide in companies large and small, localized or distributed, open source or proprietary, for virtually any type or size of project.

Agile Principles

Scrum works by implementing basic values and principles described in the Agile Manifesto at www.agilemanifesto.org.

The Agile principles originated in the Toyota tradition, now commonly called “Lean Product Development”. The following key principles at work in Scrum come from the Lean experience, Systems Thinking and Queuing Theory.

Agile Manifesto Values

Individuals and interactions *over* processes and tools
Working software *over* comprehensive documentation
Customer collaboration *over* contract negotiation
Responding to change *over* following a plan

- Pull systems self-regulate by limiting work demand to available capacity with sufficient slack to allow for smooth flow and innovation. Pull systems have been shown to deliver significant productivity gains over “push” systems in lean product development. Pushing a system to full capacity or beyond will cause turbulence, delay, breakage and unpredictability.
- Process improvement is a continuous goal. There is always a way to do better.
- Small batches move through a system more quickly than large batches. An item of work is completed in the shortest time (cycle time) when taken from start to finish with no wait states (single piece flow).
- The more work in progress, the longer each work item will take to accomplish. Multi-tasking is inefficient because of the overhead spent switching between tasks.
- Low quality products that require maintenance will have a higher total lifecycle cost and therefore a lower ROI. It is actually quicker and more profitable to build quality in from the start.
- The people doing the work are the best experts on how to improve the work. Management sets the strategy, Team Members determine the tactics. This is called “de-centralized control”.
- Teams can do things that no one individual can do. Synergy enhances productivity, knowledge growth and innovation.
- Complex systems are self-organizing. To respond quickly, systems need feedback loops with minimal delay times. Traditional product development has long feedback cycles that are ineffective for learning what works and what needs improvement.
- People-based processes are dependent on trust and respect, both of which take time to achieve.

Teams need time to mature.

This article is a significant revision by Roger Brown, CSC, CST of a series of pages that once existed at www.scrumalliance.org, original authors unknown.

References

- Ken Schwaber [Agile Project Management with Scrum](#)
- Mike Cohn [Succeeding with Agile: Software Development Using Scrum](#)
- Scrum Alliance www.scrumalliance.com
- Agile Atlas www.agileatlas.org
- Ken Rubin [Essential Scrum](#)
- The New, New Product Development Game <http://agilecrossing.com/wp-content/uploads/2011/10/New+New+Prod+Devel+Game.pdf>