

What is Scrum?

Even projects that have solid, well-defined project plans encounter some degree of change. Shifting market conditions, budget cuts, staff restructuring, or any number of influences will disrupt the best plan. Projects that begin with changing or unclear requirements make it sometimes difficult to even establish basic project expectations.

Scrum is a disciplined agile development process that allows teams to deliver usable product increments periodically throughout the life of the project, absorbing change and new requirements as the project proceeds.

Below is a high level picture of how Scrum works, commonly referred to as the Scrum Framework. The large input blocks represent work requests that are fed into the repeating cycles of work. Requests are decomposed and planned out for the cycle by a team of people. The team meets every day during the cycle to update each other on their progress. At the end of the cycle they share their output with others to receive feedback, and then retrospects on how to improve.



The more specific details follow, and we encourage you to also read the foundational reference document, “The Scrum Guide,” at scrumguides.org,...

Basic Terminology:

Three **roles**:

- The **Product Owner**, who represents stakeholders and sets direction on what the **Development Team (Dev Team)** should be working on;
- The **Development Team**, those who actually carry out the work of delivery;
- The **ScrumMaster**, who oversees the flow of work and ensures everyone has what they need to be successful.

When we have a **Product Owner**, a **Development Team**, and a **ScrumMaster** working together, we have a **Scrum Team**. So, we have a team within a team, but in some ways it doesn't matter since this is a team-based delivery approach, and all members are part of a team.

Three **artifacts**:

- The **Product Backlog**, an ordered list of desired features that have been estimated;
- The **Sprint Backlog**, a list of activities and tasks that represent the plan for the cycle of work, or **Sprint**;
- The **Increment** of delivery, what the **Dev Team** actually produced during the **Sprint**;

While not a formal artifact, we would also expect some form of visual progress (e.g., a Burndown Chart).

Five **events**:

- **Sprint Planning** forms the plan for the **Sprint**
- The **Daily Scrum** allows team members to update and connect with each other;
- The **Sprint Review** where the team shares with others what they have produced;
- The **Sprint Retrospective** to discuss improvement opportunities.
- **The Sprint** itself! This is where the four events listed above are contained.

While not a formal event, we would also expect some time spent getting the Product Backlog ready. We call this set of activities Product Backlog Refinement. We could also expect ongoing Product Backlog Refinement activities as the **Sprints** progress. With this perspective we ensure our **Product Backlog** is in a healthy state for that next **Sprint Planning** event.

Also while not a formal event we might want to spend time in Release Planning. The purpose of Release Planning is to establish a plan for expected product functionality and potential release dates that can be communicated across the organization.

The Product Backlog

The starting point of a Scrum project is the **Product Backlog**. This is simply a list of features and functions that we expect to be developed during the project. Compared with a more traditional method, we might say these are the Business Requirements.

Please note that this **Product Backlog** is a list where each entry has a brief description of the

item#	description	estimate	priority
D-001	build credit interface to ABC processor	25	1
D-004	accept credit cards on profile page	18	2
D-005	add processing for new AVS mismatches	22	3
D-007	changes to allow for international zips	5	4
D-008	new card creation app	10	5
D-009	XML update feed acceptance	7	6
D-010	Financial Recon report 1	6	7
D-011	Financial Recon report 2	12	8

feature or deliverable we desire for this project, an estimate of effort required (represented in days for this example), and an ordering (priority). We've also included an item# that's used as a reference. There are variations of how this can be represented, and some organizations will include additional columns of information to help them manage their work, but if you have these basic items you have a healthy **Product Backlog**.

Product Backlog Refinement

This is an initial and possibly ongoing activity where the **Product Backlog** is created, verified, and refined. This is not a formally defined event of **Scrum** and therefore does not have a specific recommended timebox since every endeavor and every organization is different in deciding what level of detail and confidence they wish to achieve in establishing their **Product Backlog**. Because we wish to begin execution as soon as possible, it would be wise to evaluate the cost of more extensive planning balanced against the benefits of learnings that can only be achieved by allowing the team to begin work.

The Product Owner

The **Product Backlog** is owned and represented by the **Product Owner** who has authority regarding this list and its priorities. There may be many interested stakeholders for this project, but the **Product Owner** is the one voice who has final say over the content of the **Product Backlog**. Here are some of the characteristics of a **Product Owner**:

- ☞ Typically the internal or external client, can be a delegate or liaison, but is only *one* person even if there are many interested stakeholders

- ~ Responsible for the **Product Backlog**, but they will need to call on others for help in establishing estimates or understanding technical requirements
- ~ Establishes and promotes the vision of the product so the **Development Team** can make decisions as they proceed with their work
- ~ Responsible for the ROI (return on investment) of the project by prioritizing the work
- ~ Monitors progress against goals
- ~ Makes decisions regarding implementations

The Development Team (or, Dev Team)

Before we can start a **Scrum** project, we need a **Development Team**: a small, cross-functional group of developers. The word “developer” is used here in a generic sense: anyone who has committed to and is contributing to the development of the project, which can go way beyond application developers and is certainly not restricted to software products. Here are some of the basic characteristics of a **Development Team**:

- ~ Recommended 3-9 people, cross-functional, & “full time,” meaning they are not working on multiple projects simultaneously
- ~ Responsible for the **Sprint Backlog** (see below) and ensuring work is decomposed into small, manageable tasks, typically 4-8 hours of effort per task; some teams can allow for larger task that get broken down during the Sprint
- ~ Manages their own work and self organizes around how to reach their commitments within the limits of established standards and procedures
- ~ Creates working agreements and a definition of “done” for expected behaviors
- ~ Responsible for the actual doing of the work required to accomplish the commitments, with some ability to outsource to other departments or teams if the **Dev Team** does not possess the needed skill
- ~ Demonstrates their work at the close of the **Sprint** (see below)

The Sprint

Each iteration of work for our project is called a **Sprint**. The **Sprint** is a repeatable, fixed period of time, up to four weeks in length but typically two weeks, dedicated to the delivery of potential shippable pieces of functionality, or product increments.

The Sprint Planning

Once we have a healthy **Product Backlog** by spending time in Product Backlog Refinement as well as a **Development Team** of people who can work on the project we can enter into the cyclical pattern of the Scrum framework. The Scrum framework, represented by the first graphic above, provides us with the guidance needed to deliver features and functions incrementally as the project progresses.

The entry point to this cyclical delivery method is the **Sprint Planning** event. **Sprint Planning** is a time boxed event that officially marks the beginning of the **Sprint**. This event is broken into two discussions: 1) the **Scrum Team** determines *what* features and functions will be worked on during the next **Sprint**, and 2) the **Team** decomposes those features and functions into small, manageable tasks that represent *how* they get the work done. For a four week Sprint we would typically allocate a full day for this event (eight hours) even if we do not consume all the time allocated; proportionally shorter time-boxed recommendations would be appropriate for shorter Sprints.

The **Product Owner** and the **Development Team** join together during **Sprint Planning** to review the **Product Backlog**. They also invite whoever else is needed to properly plan out the work opportunity ahead of them. They review which features and functions have the highest priority to ensure the **Development Team** has a good understanding of what's expected. They also determine which of the highest priority items can be worked on during the next **Sprint**. They do this by understanding their capacity for work during the coming **Sprint**.

The Sprint Backlog

The items selected for work during **Sprint Planning** are moved from the **Product Backlog** to the **Sprint Backlog**. We then move to the second part of **Sprint Planning** where the selected **Product Backlog** items are broken down into smaller, manageable pieces of work.

item#	description	estimate
D-015		40
D-015.10	enhance 0100 msg as per spec (new bit)	8
D-015.20	c_cred table for logging transactions	12
D-015.25	test logging of 0100 msgs	4
D-015.40	script to pre-populate c_cred w/test data	16
D-014		150
D-014.10	new integration test parameters	4
D-014.20	verify different card types	7

In this sample we've included an item# that simply refers back to the original **Product Backlog** item, a description of tasks necessary to complete the **Product Backlog** feature, and an estimate that is small enough to be managed in a day or two, but not so small that we spend an inordinate amount of time in decomposition. In our example our estimates are represented in hours, but a relative point-based system is common.

The Daily Scrum

Once we have agreed upon the **Sprint Backlog** content, the associated goal of the Sprint, and have committed to their best effort for the work therein, **Sprint Planning** is

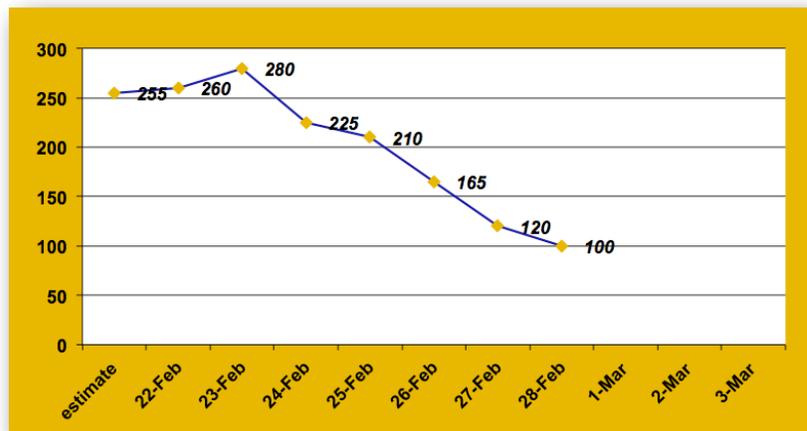
complete. For the length of the **Sprint** we do not want to see any disruptions to the team that would prevent them from achieving their plan for the **Sprint**. The **Development Team** will use the **Sprint Backlog** to guide them through the new iteration, and they are now ready to begin working on the development of the increment.

To help the **Development Team** understand their progress, they meet daily during the **Sprint** at the **Daily Scrum**. While others can observe, this daily event with a time box of 15 minutes expects **Development Team** members update each other, commonly by answering the following three questions:

- 1 – what did I work on since our last Daily Scrum that’s contributing to the goal of the Sprint?
- 2 – what am I planning on working on next that will contribute to the goal of the Sprint?
- 3 – what impediments are hindering my ability to achieve the goal of the Sprint?

The **Daily Scrum** is the first great opportunity to “inspect and adapt” on a regular basis, allowing the team to consider ways to improve performance and ensure delivery of the **Sprint** objectives.

We have seen two Scrum artifacts so far: the **Product Backlog** and the **Sprint Backlog**. While not a formal artifact, we expect some form of a visual progress, often the Burndown Chart. This chart is used by the team during the **Daily Scrum** to understand how much work is remaining in the **Sprint**, day by day.



You will see that in our example the x-axis represents time in terms of days of the **Sprint**, and the y-axis represents effort estimated for the **Sprint**. In our example we are representing effort in hours. Each day the **Dev Team** “burns down” remaining effort as they work towards the end of the Sprint, completing tasks as they look to have as close to zero remaining effort as possible at the end of the allotted time. The **Burndown** chart not only serves as a progress indicator for the team, but is externally available to all interested stakeholders so they may also understand progress.

The Sprint Review

At the end of the **Sprint**, the **Development Team** meets with the **Product Owner** and other stakeholders at the **Sprint Review** to review the work that was completed. The **Team** reiterates the goal of the **Sprint**, and then proceeds with a demonstration of the work delivered. Decisions are made during this event regarding potential deployment of what has been developed so far. For a four week Sprint we would typically allocate no more than four hours for this event.

This is the second great “inspect and adapt” opportunity, allowing the **Dev Team** and **Product Owner** to consider ways to improve the value of the project by reevaluating the **Product Backlog** for new items, changes to items, or re-prioritization. This is also the opportunity for the **Dev Team**, **Product Owner**, and other stakeholders to explore the good and bad of the previous **Sprint** and how they can improve.

The Sprint Retrospective

The **Scrum Team** also holds a periodic **Sprint Retrospective**. This is the third great “inspect and adapt” opportunity, allowing the **Scrum Team** to consider ways to improve their overall performance above and beyond the project itself, while addressing ongoing obstacles to productivity. What’s going well? What could be better? Answers to these and other questions are logged for reference and future action.

In addition to the **Sprint Retrospective** taking place at the end of every **Sprint** where for a four week **Sprint** we would typically allocate no more than three hours for this event, we may also expect a **Retrospective** at major intervals or special occasions, such as after a major delivery, after a major disruption to the **Scrum Team** or project, or at the end of the project funding.

We then loop back to the next **Sprint Planning** event where we identify and select items to work during the next **Sprint**.

The ScrumMaster

Who oversees this framework and helps ensure that participants are following **Scrum** principles? That would be the **ScrumMaster**. Some of the basic characteristics of a **ScrumMaster** include:

- ~ Responsible for **Scrum** values and practices
- ~ Encourages open communication, teamwork, and collaboration
- ~ Responsible for ensuring the **Development Team** has what they need to be successful
- ~ Seeks ways to increase productivity; removes obstacles to productivity
- ~ Establishes the key, few **Scrum** events:
 - ~ Product Backlog Refinement, if the **Product Owner** needs assistance
 - ~ **Sprint Planning**

- ~ **Daily Scrum**
- ~ **Sprint Review**
- ~ **Retrospective**
- ~ Protects the **Development Team** from interruptions
- ~ Assists with record keeping for visual progress and other artifacts

In addition to the process details we've just outlined, **Scrum** creates an environment that expects and promotes self-managed teams, iterative processing, continuous improvement, empirical thinking, and a high degree of visibility into the project and the organization. Implementing **Scrum** is much more than changing a process: it is changing the way we think about our work. We look to the **ScrumMaster** to be the change agent in helping others maneuver through these different ways of thinking. This is indeed a very difficult job, but can be some of the most rewarding work on the project.



An Introduction to Scrum

There are many additional sources of information about Scrum, but a great introductory read is *Agile Project Management With Scrum*, written by Ken Schwaber, or *Succeeding With Agile*, by Mike Cohn.

Helpful links:

<http://www.winnowmanagement.com>

<http://www.scrumalliance.org>

<http://groups.yahoo.com/group/scrumdevelopment>