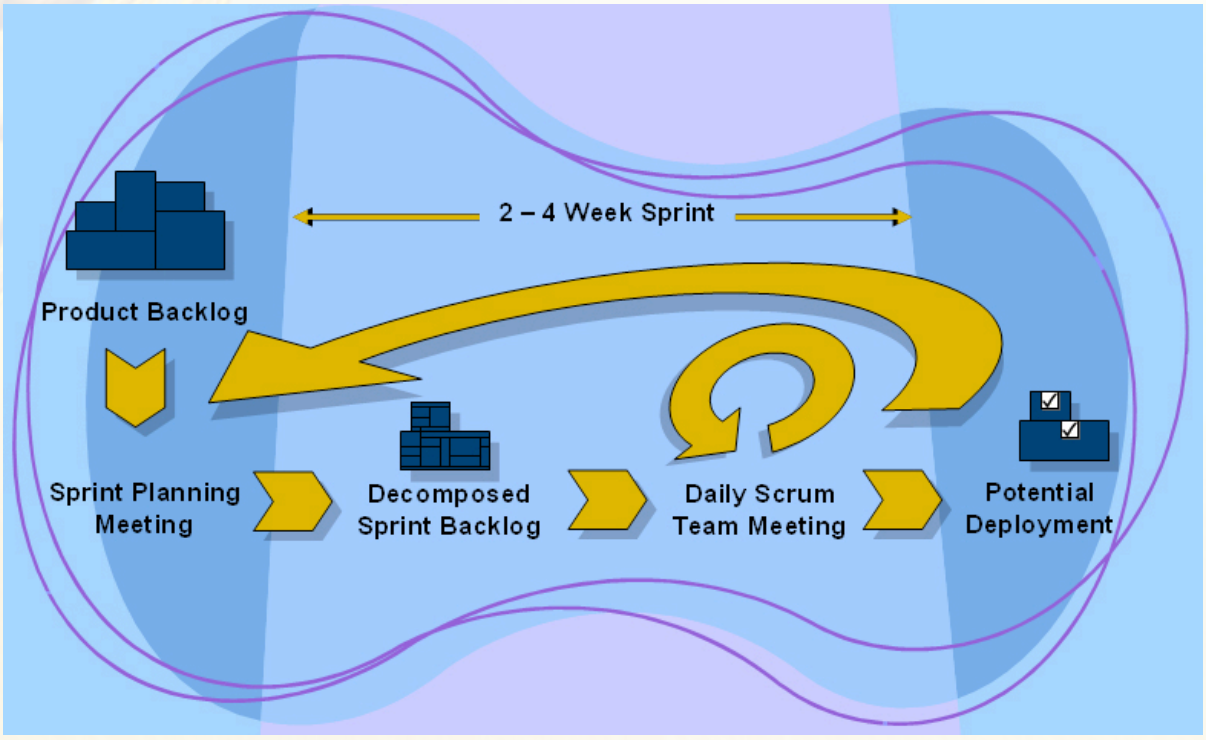
An Introduction to Scrum

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 the agile development process that allows teams to deliver usable software periodically throughout the life of the project, absorbing change and new requirements as the project proceeds.

Here's a high level picture of how Scrum works, also referred to as the Scrum Framework:



Basic Terminology:

The players:
- The **Product Owner**
- The **Development Team**
- The **ScrumMaster**

The artifacts, or documents:
- The **Product Backlog**
- The **Sprint Backlog**
- The **Burndown Chart**

The meetings, or ceremonies:
- The **Sprint Planning Meeting**
- The **Daily Scrum**
- The **Sprint Review**, or **Sprint Demo**
- The **Retrospective**

Additionally, we would expect 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.

When we have a **Product Owner**, a **Development Team**, and a **ScrumMaster** working together, we have a **Scrum Team**.

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.

item#	description	estimate	priority
D-001	build credit interface to ABC processor	25	1
D-004	accept credit cards on purchase page	18	2
D-005	add AVS mismatch processing	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

Please note that this **Product Backlog** is a list where each entry has a brief description of the feature or deliverable we desire for this project, an estimate of effort required (represented in days for this example), and a 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**.

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, the "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 would include application developers, testers, DBAs, etc. Here are some of the basic characteristics of a **Development Team**:

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	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	verification code against card types	7

- Typically 3-7 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 tasks that fall in the range of 4-16 hours of effort
- Manages their own work and self organizes around how to reach their commitments within the limits of established standards and procedures
- Creates Ground Rules for expected behaviors
- Responsible for the actual doing of the work required to accomplish the commitments, with some ability to outsource to other departments if the **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, typically 2 to 4 weeks, dedicated to the delivery of pieces of functionality of the project.

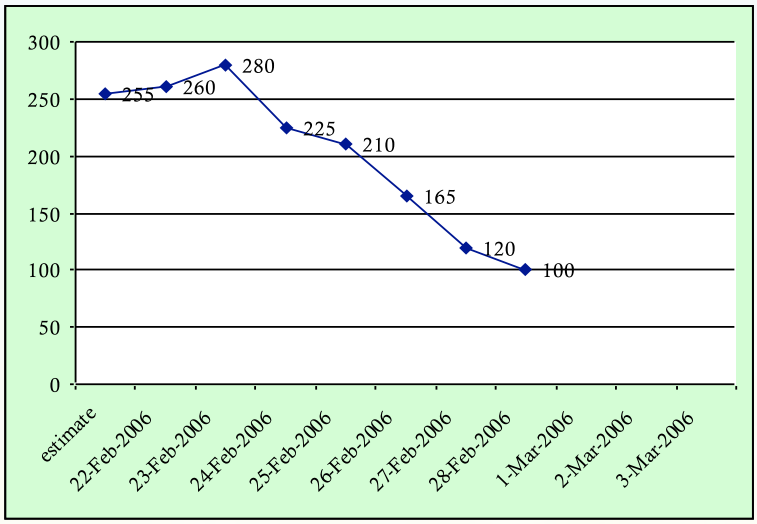
The Sprint Planning Meeting

Once we have a healthy **Product Backlog** and a **Development Team** of developers who can work on the project, we can enter into 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 Meeting**. The **Sprint Planning Meeting** is a time boxed event that officially marks the beginning of the **Sprint**. This meeting is broken into two activities: 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. Typically we allocate a full day for this meeting even if we do not consume all the time allocated.

The **Product Owner** and the **Development Team** join together during the **Sprint Planning Meeting** 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 the **Sprint Planning Meeting** are moved from the **Product Backlog** to the **Sprint Backlog**. We then move to the second part of the **Sprint Planning Meeting** where the selected **Product Backlog** items are broken down into smaller, manageable pieces of work.

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.

The Daily Scrum Meeting

Once we have agreed upon the **Sprint Backlog** content and have committed to the work therein, the **Sprint Planning Meeting** is complete. For the length of the **Sprint** we will not discuss changes in **Product Backlog** priority, but instead focus in those items selected for delivery that **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 project.

To help the **Development Team** and others understand progress while providing an opportunity to evaluate objectives, the **Scrum Team** meets daily during the **Sprint** at the **Daily Scrum Meeting.** During this daily meeting, which is time boxed at 15 minutes, each **Development Team** member is responsible to communicate answers to the following three questions:

1 – what did I work on since our last Daily Scrum Meeting?
2 – what am I planning on working on next?
3 – what obstacles are hindering my productivity?

The **Daily Scrum Meeting** 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**. The third artifact is the **Burndown Chart**. This chart is used by the **Team** during the **Daily Scrum Meeting** to understand how much work is remaining in the **Sprint**, day by day.

You will see that 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 **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 track progress.

The Sprint Review Meeting

At the end of the **Sprint**, the **Development Team** meets with the **Product Owner** at the **Sprint Review Meeting** to review the work that was completed. Other stakeholders are invited as well. The **Team** reiterates the goal of the **Sprint**, and then proceeds with a demonstration of the work delivered. Decisions are made during this meeting regarding potential deployment of what has been developed so far.

This is the second great "inspect and adapt" opportunity, allowing the **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 **Team**, **Product Owner**, and other stakeholders to explore the good and bad of the previous **Sprint** and how they can improve.

The 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 if necessary.

The **Sprint Retrospective** is held at periodic intervals or special occasions. For example: at the end of every **Sprint** (or every other **Sprint** if we are working with short **Sprint** cycles), after a major delivery, after a major disruption to the **Team** or project, or at the end of the project.

We then loop back to the next **Sprint Planning** meeting 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** meetings:
  - **Sprint Planning**
  - **Daily Scrum**
  - **Sprint Demo**
  - **Retrospective**
- Protects the **Development Team** from interruptions
- Assists with record keeping for **Burndowns** 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, 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.

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

Helpful links:

http://www.winnowmanagement.com

http://www.scrumalliance.org

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

http://www.agilealliance.org