scruminc.

The Basics of Scrum
An introduction to the framework

Introduction

Scrum, the most widely practiced Agile process, has been successfully used in software development for the last 20 years. While Scrum has been mostly practiced in a commercial software environment, the methodology has been successfully applied to education, manufacturing and an array of other industries.

Like other Lean and Agile methods, Scrum optimizes limited resources and creates efficiencies. Scrum also empowers teams to self-organize and work at a sustainable pace, free from outside interference. This helps teams reach their full potential and frees leadership to focus on the company's vision rather than on day-to-day management.

This document is meant to familiarize the reader with Scrum's core principles. Many additional layers of Agile practices and patterns have been developed over the years that enable Scrum to address the inevitable complexities of real life. However, what follows is a description of the basic Scrum terms and elements.

Scrum in a Nutshell

Divide and Conquer. Scrum divides complex work into simple pieces, large organizations into small teams and far-reaching projects into a series of short time horizons called sprints.

When complex work is divided into simple pieces it is easier to map out what needs to be done. With a clear roadmap the team can start working immediately, know what items need to be worked on together and understand when the job has been completed. Small bites of work, just like food, are easier to chew, swallow and digest.

By dividing large organizations into small teams, Scrum enables organizations to perform like small ones. Small teams make it easier to maintain focus because less time is spent communicating details. Coordinating is much easier for a team of five people than it is for a team of 20.

The divide and conquer principle also works when sectioning large projects into a series of sprints. It is easier to plan for short periods of time because there is less involved. Think in terms of writing a book. Writing a chapter at a time is easier than writing an entire novel at once. This iterative process also allows the author

to apply what he or she has learned writing the first chapter to writing the next.

Inspect and Adapt. When work is divided into simple pieces it can be finished in a shorter period of time. By accelerating the development process and getting a functioning product in the hands of people who will use it, the team can gather feedback more quickly than it otherwise would have. Feedback helps the team improve the product based not only on what they have learned during development, but also from people interacting with the product. To carry the book analogy forward, after writing the first chapter and receiving feedback from an editor, the author can make adjustments. Without this feedback, the author may have headed in the wrong direction and wouldn't have known that until he or she finished writing the book. At that point, the author would have to re-write the entire book rather than just the first chapter. With timely feedback, the author is able to make minor adjustments to the book's outline early, avoiding mass amounts of re-work and waste.

Transparency. Scrum makes all work transparent**.** There are no secrets about what needs to be done, who is doing it, and how it is being accomplished. When the entire organization is armed with this information, it can address problems as they arise and correct course at the earliest possible moment. Stakeholders and leadership are able to make more informed strategic decisions when they have an honest and clear idea of how a project is progressing.

To achieve transparency Scrum uses a number of shared tools to track activity over the length of the sprint. This makes it easy for everyone to know how things are moving along. Is the team on track to finish all the work? Where is work held up and why?

Scrum also employs a series of meetings, known as ceremonies, to increase transparency. These ceremonies happen in a regular cadence and are designed to facilitate team communication. Shared tools make the status of any particular activity visible and ceremonies give the team qualitative context about how an item or activity is being accomplished. A more detailed explanation of the ceremonies can be found in the *How Scrum Works* section of this document, but here is a quick overview:

- **Sprint planning** is when the team decides what it needs to get done in the coming sprint and how much it can accomplish.
- **Daily scrum** is when the team touches base to raise any impediments that are affecting the sprint. It also helps the team stay focused.
- The **sprint review** that takes place at the end of the sprint is designed to gather actionable feedback on what the team completed that sprint

and to provide a chance to inspect the overall plan for the product.
- The **sprint retrospective** helps the team inspect and adapt its process by asking what went well during the sprint and what could have been better.

The Scrum Framework

Scrum employs two shared tools, three roles and four meetings within a regular period of time called a sprint. This framework promotes transparency and communication saturation, which together enable the team to regularly inspect and adapt to change.

Sprints traditionally vary from one to four weeks, and are simply an iteration of time that is long enough for the team to produce a meaningful increment of the product, yet short enough to keep it focused.

Shared tools. The most important shared tool in Scrum is the **product backlog.** Simply put, the product backlog is a list of everything that could be done over the lifetime of the project. The product backlog is divided and defined into items that can be delivered independently of each other. To use the book example again, each chapter should be able to be written independently from the next chapter. The ability to conquer the project in independent chunks drives the product owner's ability to inspect and adapt the plan.

The product owner (more on this role below) orders the product backlog according to the business value of each item. Items with a high business value are placed at the top, items with a low business value are found toward the bottom. As a general rule, 80% of a product's value to the end-user is in 20% of its features. Ordering the product backlog according to business value assures the most needed features are built first. Once the first set of features has been released, end-user feedback will give the product owner more information about what features to include in the next iteration of the product. Armed with this data, the product owner can build a strong case that many of the features requested by the customer, the 80%, aren't necessary.

Items at the top of the backlog are worked on first so they need to be well defined, or ready. In Scrum, any item in the product backlog meets the definition of ready if the team has discussed it thoroughly and has a clear idea of what it will take to complete the item. Items farther down the backlog are less defined because the understanding of these items will evolve as the project moves forward and the team learns from its experience.

The **sprint backlog** includes all items that need to be completed by the end of the sprint and reflects the priority order of the product backlog. Once the sprint backlog is finalized, changes should not be made except in rare circumstances. This restriction helps the team maintain focus by limiting interruptions and changes to scope mid-sprint.

The final sprint backlog is moved onto the **scrum board**. A traditional scrum board has three columns: To Do, Work in Progress, and Done. Each activity in the sprint backlog is first placed in the "To Do" column. Activities in this column retain the same order of priority as established in the product backlog by the product owner. Team members begin the sprint by choosing an item from the top of the "To Do" column and moving it into the second column, "Work in Progress". When completed, the item is moved to the third and final column, "Done".

Scrum boards can be physical or digital, but it is important that they be visible to the entire team so that the sprint's status can be seen in real-time. This allows the team to change course if an item in the sprint backlog is in danger of not being completed.

The Scrum board allows the team not only to monitor the sprint's status, but also to track the team's output. In Scrum we measure the team's output in each sprint using a metric known as **velocity.** Velocity is measured by assigning each item in the sprint backlog a relative point value. At the end of the sprint, points for all items in the "Done" column are added together. This total is the team's velocity for the sprint it just completed. Tying a numerical value to the amount of work the team produces each sprint allows it to inspect and adapt in several ways.

First, velocity creates a baseline for team performance. From this baseline, the team is able to empirically determine how much work it can expect to get done in future sprints. For example, if the team completed 50 points of work last sprint, it most likely can complete at least 50 points in the next sprint. This data driven baseline helps the team avoid pulling too much work into the sprint and/or forcing the team to work at an unsustainable pace.

Secondly, velocity allows the team to see whether or not a process change has helped or hurt. For example, if the team has failed to a finish a particular activity several sprints in a row, they might decide to have two people work on it rather than one. If the process change is successful, the team will complete the activity and its velocity will go up. If less gets done because one activity is tying up two people, velocity will go down. This cycle of inspection and adaptation based on

real data helps the team continuously improve by keeping only changes that had a positive effect on velocity.

Finally, velocity allows the team to forecast a project's completion date. If the team knows how much work it can get done each sprint, it can measure that against the total estimated point value of the product backlog. So, if the team has a velocity of 50 points per sprint and the product backlog has 500 points worth of work, the team can reasonably expect to complete the project in 10 sprints. If sprints are two weeks long, the team can forecast completion of the project in 20 weeks.

Roles

The **team** is everyone who works toward the completion of the product. It is ideally made-up of three to nine people including the scrum master, the product owner, and the core team of workers. It is critical that the team includes all skill-sets necessary to deliver the product. In Scrum, this is called a cross-functional team. Additionally, there should be at least two people on the team that can complete any one item in the product backlog. This helps eliminate dependencies on a specific team member's skill set in case that co-worker gets sick, goes on vacation or moves on to another job.

There is no team manager in Scrum. Instead, the team organizes and manages itself with a very high degree of both autonomy and accountability. The team, with guidance from the **scrum master**, decides what it can accomplish in a sprint and how it can be done.

The scrum master is a coach, not a boss. He or she has no authority over the team. Rather the scrum master assures that the team is functioning to the best of its ability by helping to identify and remove impediments. An impediment is anything that slows or prevents the team from completing the sprint. The scrum master also makes sure the team is communicating well and using the framework to its greatest effect.

This brings us to the **product owner** (PO). The product owner is the team member responsible for the commercial success of the product. As such, the PO must know what the customer wants and the relative business value of those wants. He or she translates those wants into the **product backlog** and orders it according to business value (with the most valuable items on top). To assure the success of the product, the PO must control the order of product backlog.

However, the team is responsible for the sprint backlog. For this reason, the product owner should spend about half his or her time working with the team to make sure the work in the sprint backlog reflects the vision of the product backlog. The other half of the product owner's time should be spent consulting with customers and stakeholders to make sure the product backlog reflects their needs.

How Scrum Works

Scrum begins with the product vision. The product owner translates the vision into the product backlog. Once the product backlog has been established, the team can start sprinting. To start a sprint, the team must first conduct **sprint planning**. Sprint planning should be limited to no more than two hours for every week of sprint. So, for a one-week sprint the team should meet for no more than two hours.

The idea behind sprint planning is to have one comprehensive meeting that maps out what needs to be done by the end of the Sprint. The ceremony starts with the team pulling as many items from the top of the product backlog that it can reasonably finish within the sprint.

The team should make sure that all items brought into the sprint backlog during sprint planning are clearly defined and are ready to be worked on. This is a fundamental concept in Scrum known as the **definition of ready**. An item in the backlog is ready if it is independent, actionable, has been assigned a point value and has a clear definition of the criteria that means it is done. This in turn is referred to as the **definition of done** which ensures everyone knows exactly what is expected of an item when it is delivered.

The team creates a sprint goal once the sprint backlog has been created. This goal should articulate the high-level purpose of the items in the sprint backlog. In simpler terms, the goal provides context for why the team is working on the selected backlog items. A goal may be as simple as reaching a certain level of functionality. "By the end of this sprint the team will demonstrate how the program can save e-mail addresses automatically."

After sprint planning the team gets to work and meets every day for the **daily scrum.** All team members working on the sprint backlog need to be at the daily scrum and attendees should stand to help keep the meeting short (no longer than 15 minutes). During the daily scrum, each team member answers three questions:

1. What did I do yesterday that helped the Team meet the Sprint Goal?
2. What will I do today to help the Team meet the Sprint Goal?
3. Do I see any impediment that prevents me, or the Team from meeting the Sprint Goal?

The daily scrum is not a status report. This ceremony aligns the team and helps maintain open communication. It is an opportunity to inform the team about what everyone is doing. For example, if someone has an idea or suggestion that could make a backlog item easier to complete they can share it with the team at the daily scrum. If an impediment is surfaced during the daily scrum that is too big to resolve during the meeting, the team should coordinate outside of the meeting to address it. The Scrum Master is the team member responsible for removing impediments.

In addition to the daily scrum, the team should spend 5-10% of its time looking ahead and refining the items at the top of the product backlog. This is called **backlog refinement.** It is not an official Scrum ceremony, but it is a best practice. During refinement, just as in sprint planning, the team focuses on the top of the product backlog to make sure those items are ready to be brought into the next sprint.

At the end of each sprint, the team invites the stakeholders and customers to a demonstration of what it has completed. This ceremony, called **sprint review,** is designed to elicit actionable feedback from the stakeholders and customers, which the team can then incorporate into the product backlog. The demonstration produces a conversation between the team and the stakeholders about how to make the product better. The product owner incorporates the lessons learned during the conversation into the product backlog. This completes one of what is hopefully many inspect and adapt cycles.

After sprint review, the team gathers for the final ceremony of the sprint, **sprint retrospective**. The retrospective is the team's opportunity to inspect and adapt its processes. It is a time to reflect on how to improve the way it works by identifying one process change that can help increase velocity. The scrum master and the team need to be present. The product owner is welcome, but not required.

There are many ways to conduct retrospectives, but regardless of form a good retrospective will include hard truths, honesty, and debate. A common way to structure a sprint retrospective is to have each team member answer the following questions:

- What went well?
- What could have been better?
- What things can we try to improve in the coming sprint?

The discussion about what could have been better often leads to an analysis of what the underlying cause might be. A consensus about what improvement to make in the next sprint usually starts to emerge. This single process change is added to the next sprint's backlog and given a definition of done.

At the next sprint review the issue is revisited to see what difference, if any, the process change has made. If team performance improved, the change should be kept. If performance did not, the team needs to develop another hypothesis on what the root cause may be. It's important to just choose one change. If the team tries to implement multiple changes, it will be difficult to figure out which process change solved the real issue and led to increased performance.

If a team is able to consistently have strong retrospectives, which produce relevant process improvement *that build sprint to sprint*, it will continuously improve. Continuous improvement is the end goal of Scrum.

Conclusion

Outlined above is the most basic description of the Scrum development process. Since its inception, Scrum has incorporated many other important practices and techniques. Scrum is not some silver bullet to solve all your team's problems but it does create a framework and common vocabulary that enables teams to inspect and adapt. It is scalable and allows for a broad array of metrics that help track changes empirically. If you are interested in learning more about how Scrum works please contact Scrum Inc. and its extensive network of partners. From coaching, consulting, leadership workshops, and assessments to continuing education, we can tailor services to fit your needs.