

Scrum Methodology

Sakshi Sachdeva

trehansakshi21@gmail.com

Abstract:

The agile methods, such as Scrum and Extreme Programming (XP), have been a topic of much discussion in the software community over the last few years. While these have gained importance in the industry because of their approach on the issues of human agility and return on investment, usually within a context of small-to-medium size projects with significant requirements volatility, the ones who do not support these methods have expressed serious concerns about the effectiveness of the methods. Scrum attempts to build the work in short iterations where each iteration consists of short time boxes. This paper reviews several papers on Scrum, its framework including its artifacts and the ceremonies which are involved. This paper gives an insight of the Scrum Methodology to any beginner.

Keywords: Agile methods, methodology, Scrum, software process, Sprint, Backlog, Artifacts

Introduction

Scrum is an incremental and iterative agile software development framework for managing product development. It was first defined as "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka in the *New Product Development Game*. This strategy challenges assumptions of the "traditional, sequential approach" to product development, and encourages teams to self-organize by encouraging close online collaboration of all team members, as well as face-to-face communication among all team members and disciplines in the project.

A key principle of scrum is "requirements volatility" i.e. it recognizes the fact that during production processes, the customers can change their minds about what they want and need. These unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner and thus it's an advantage of scrum/ agile methodology. Scrum adopts an empirical approach—accepting that the problem cannot be fully defined, instead focuses on responding to emerging requirements and to adapt to evolving technologies and changes in market conditions. [5]

Scrum employs real-time decision-making processes based on actual events and information. This requires specialized teams capable of self-management, communication and decision-making. Scrum is an agile methodology that can be applied to nearly any project; however, it is most commonly used in software development. The Scrum process is suited for projects with rapidly changing or highly emergent requirements. [1], [6]

What's so special about scrum?

Scrum is a tool, a framework that can be used to build complex products. . It is "a flexible product development strategy. It does not prescribe any of the common engineering, people, risk management, or other practices. Since Scrum does not explicitly describe any engineering practices, it is possible and at times desirable too to consider non-Scrum practices that may be tightly linked to Scrum success. For example, test-driven

Development has been proved successful for agile projects but is not an explicit Scrum practice. Scrum is frequently implemented in conjunction with methods such as Extreme Programming which influences the success of Scrum.

What Scrum does provide is feedback so that someone using Scrum can improve the results. For instance, if someone wants productivity and quality and can have a co-located team, Scrum will help figuring out the best possible way. If the person starts with a dispersed team and compares its productivity to another co-located team, conclusions can be reached. The person can then intelligently take decision based on the conclusions received and can make changes accordingly. Thus, Scrum specialty lies in continuous process improvement.

Using Scrum correctly means following all of its rules, which expose everything (transparently) for inspection and adaptation. An intelligent person would then inspect what Scrum is making transparent and make changes to optimize the results. Presumably, the changes are cost justified.

Scrum can be used perfectly and ignoring what is made transparent.

Scrum can be used imperfectly and taking into account some of the things that have been made transparent.

Someone who uses Scrum perfectly and acts more intelligently on what has been made transparent will out-compete anyone else. [4]

Scrum Practices and Roles

The Scrum Alliance has published a “Scrum Guide” giving the formal definition of the method [SA09]. Sutherland and Vodde created Tests to assess the status of teams claiming to use Scrum. Silver, another Scrum Alliance member, has also identified crucial characteristics and practices for Scrum. These descriptions of Scrum and its practices are elaborated and clarified in various reports and training, as well as related books. The following brief description of Scrum practices and roles highlights the specific aspects of Scrum that may be investigated to verify that a Scrum implementation is a valid one.

In many cases Scrum is adopted as a whole with little change, but in some cases it is adopted in a “tailored” form. This tailoring may, or may not, represent a reasonable adaptation of the original method. Inappropriate Scrum variations are known as “ScrumButs.” Scrum is a “bundle” of knowledge that is best adopted as a whole; piecemeal adoption or inappropriate tailored adoption of Scrum practices is unlikely to achieve the expected behaviors and benefits of the method. [4]

With scrum, the product is built in a series of fixed-length iterations called sprints. Sprints are fixed duration cycles during which the product is built and is delivered for feedback. A Sprint is one iteration of a month or less that is of consistent length throughout a development effort. Only the Product Owner has the authority to cancel the Sprint. Milestones—i.e., the end of a sprint—come frequently and at regular intervals, bringing with them a feeling of tangible progress with each cycle that energizes everyone and continuously inspire the team and also helps find out the shortcomings or misunderstood requirements at an early stage. Short iterations also reinforce the importance of good estimation - a recurring struggle in waterfall projects. [7]

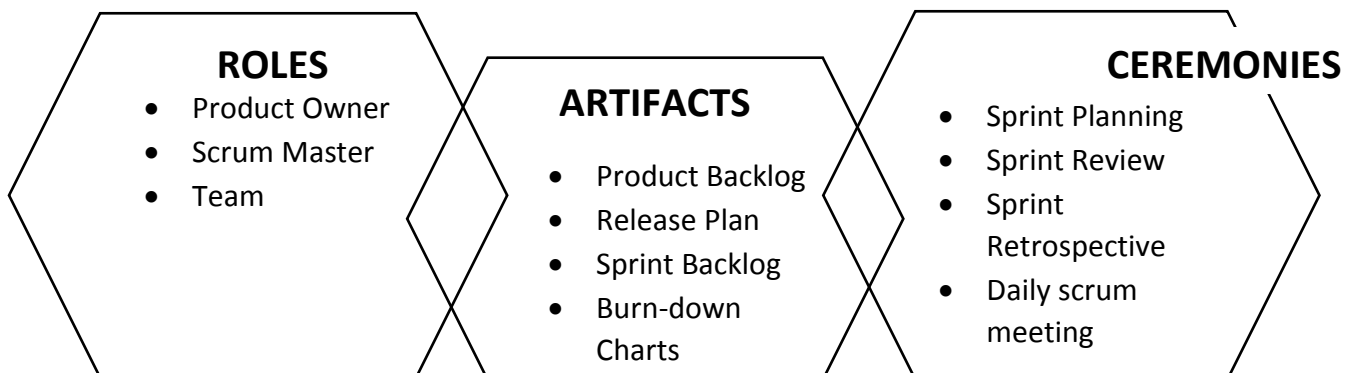


Figure 1. Scrum Framework
Scrum Roles

In contrast to classical project management methods, Scrum doesn't have and doesn't need a product manager, a task manager or a team leader. A scrum team has a slightly different composition than a traditional waterfall project, with three specific roles:

- Product Owner,
- Scrum Master, and
- Development Team.

These three roles are coequal and all of them have certain responsibilities.

The Product Owner is responsible for the vision of the product, the gathering and the prioritization of the requirements, control over the budget and the ROI. The Scrum Master takes care of the problems, takes responsibility that the rules of Scrum are appropriately followed and he coaches the team too. The team of Scrum is a self-organized group of people, responsible for the creation and the quality of the product. And because scrum teams are cross-functional, "the development team" includes testers, designers, and ops engineers in addition to developers. Besides these three roles there exist some more Stakeholders, who e.g. serve as an observer or a counselor. [2], [7]

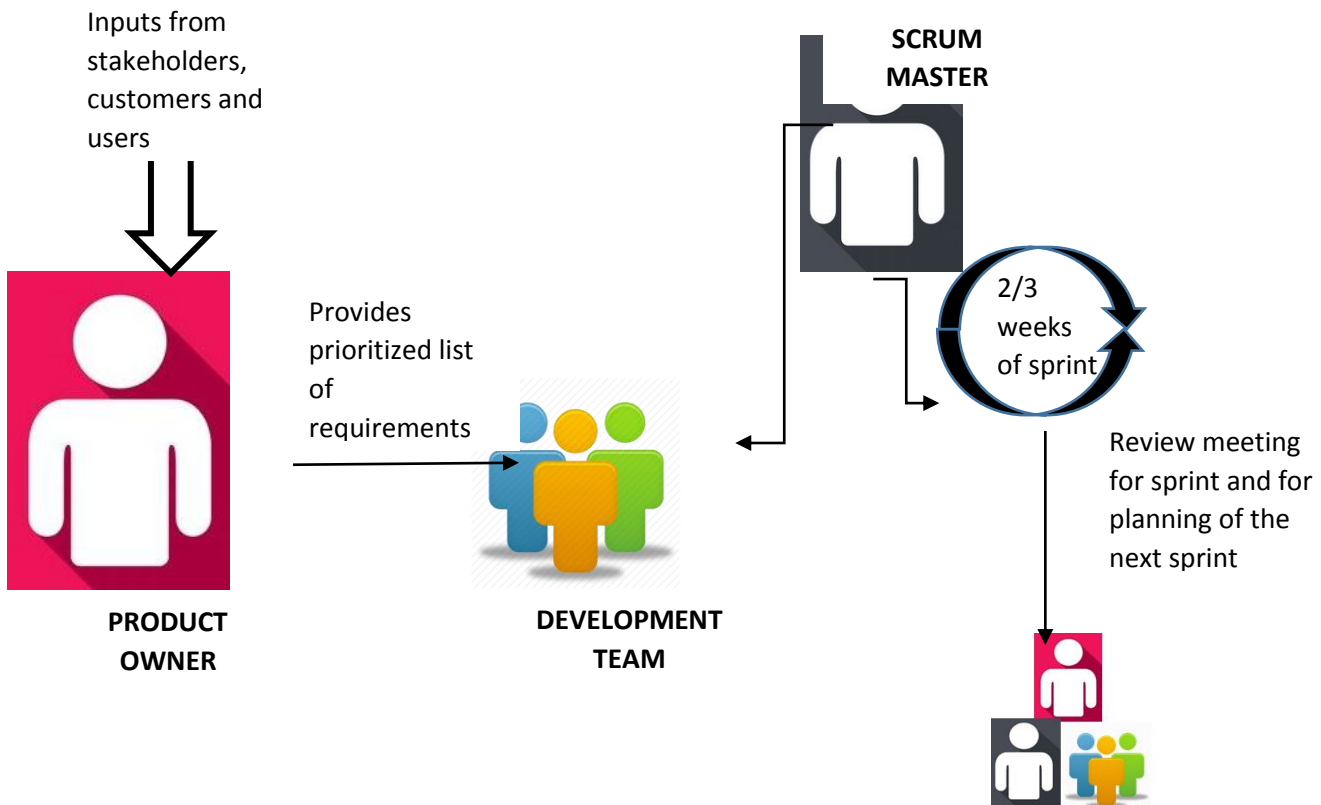


Figure 2. Scrum Roles and their relationship.

Product Owner

One of the most important things for the success of scrum is the role of the Product Owner, who serves as a mediator between the team and other involved parties (stakeholders). Product owners are not managing the status of the program. A product owner is not a project manager. They ensure that the development team delivers the most value to the business. Also, it's important that the product owner be an individual. No

development team wants mixed guidance from multiple product owners. Moreover having multiple product owners may create confusion for both the team and the stakeholders and can lead to delays in the sprint cycle. It can be said that in companies that use scrum, the tasks and responsibilities of the particular Product Owner are never the same. The role of the Product Owner is the most complex one regarding the procedure being followed.

Product owners are the champions for their product. They are focus on understanding the business and market requirements, then prioritizing the work to be done by the engineering team accordingly. Effective product owners:

- Build and manage the product backlog
- Sets the priority for each item in the Product Backlog
- Work closely and communicate well with the business and the team to ensure everyone understands the work items in the product backlog
- Give the team clear guidance on which features to deliver next
- Decide when to ship the product with the predisposition towards more frequent delivery

The Product Owner may represent multiple customer constituencies but has the responsibility and authority to reconcile conflicting requirements and determine the business value associated with each item in the Product Backlog.

Often the Product Owner has to “fight” on both sides. Whereas the team can work a certain fraction of time “protected” by the Scrum Master, the Product Owner often needs to deal with marketing, management or the customers in order to be able to present the software requirements quite precisely to the team.

He also has to decide over the importance of every single feature of the product in order to prioritize these and he has to tell the team what the product should look like in the end. He validates the solutions and verifies whether the quality is acceptable or not from the end-users’ point of view. The Product Owner must react fast on call-backs so as the enable the team work effectively. Hence he fulfills the role of a communicator, as he must be in contact with all stakeholders, sponsors and the team throughout a project.

Furthermore the Product Owner is responsible for the return on investment (ROI). After all it is his task to coordinate the financial side of the product development, in which he is successful through his continuous work and prioritizing the advancing tasks (Product Backlog). All these diverse roles demonstrate how important the selection of the “right” person for the role of the Product Owner is for the success of a project.

As the name says, the Product Owner is not only the manager of a product, but also the Owner and therefore he is the one responsible for the correct creation of a product. Being a Product Owner means:

- You are responsible for the success of the outcome of the product delivered by the team.
- You prioritize the business requirements and take important Business decisions.
- You deliver the vision of the product to the team.
- You prepare the User Stories for the team for development.
- You should possess severe domain knowledge.
- You validate the product against the user’s expectation and test for its quality.
- You react promptly on callbacks.
- You communicate with all Stakeholders, financiers and the team.
- You take good care of the financial side of the project.

Scrum Master

Scrum masters are the champion for scrum within their team. The Scrum Master is the specific individual responsible for ensuring that Scrum values, practices and rules are enacted and enforced. They coach the team, the product owner, and the business on the scrum process and look for ways to fine-tune their practice of it. The

Scrum Master is sometimes characterized as the project manager who leads by coaching, teaching and supporting the Team rather than directing and controlling.

A Scrum Master is not a project manager. Project managers don't really have a place in the scrum methodology. The project manager role within Scrum ceases to exist as its responsibilities are moved to the other Scrum roles. A scrum team controls its own destiny and self-organizes their work. Some Scrum projects may have both a Scrum Master and a project manager and in some cases where the project is too large using a Scrum of Scrums approach might have a program manager working with multiple Scrum Masters.

- An effective scrum master deeply understands the work being done by the team and can help the team optimize their delivery flow.
- As the facilitator-in-chief, they schedule the needed resources (both human and logistical) for sprint planning, stand-up, sprint review, and various other scrum practices.
- Scrum masters also look into the impediments and distractions for the development team and resolve them.
- They insulating the development team from external disruptions whenever possible.
- Scrum master's job includes another important task of observing that the team obeys the rules and realizes the method of Scrum entirely. For example: Some teams new to scrum try to change the scope of the sprint after it has already begun.

Product owners will sometimes ask, "Can't we get this one more super-important little thing into this sprint?" But keeping scope air tight reinforces good estimation and product planning—keeping in mind that this does not becomes a source of disruption to the development team.

The most obvious difference between a team leader and a Scrum Master is represented by the name itself though.

- Whereas one is leading the team and sets the tasks, the other one is in charge of observing that the team obeys the rules and realizes the method of Scrum entirely.
- The Scrum Master does not interfere into the decisions of the team regarding specifically the development, but rather is there for the team as an advisor. He only interferes actively when anyone within the team or any other participant of a project (Stakeholder) does not obey the rules of Scrum. Whereas a team leader often gives requirements and takes responsibility for the completion of those.

An experienced Scrum Master gives only impulses and advises to the team to lead the correct way, to use the right method or to choose the right technology and sees that all these resources are made available to the team whenever they are required. In specific, the Scrum Master acts more like a Team Coach than a team leader.

Scrum Master and Impediments

Another important task of the Scrum Master is to take into account and get rid of all possible impediments that might disturb the work of the team or slow down the pace of the sprint. Such problems can be classified in three different categories:

- The first one is problems that the team cannot solve. E.g. the team cannot do any kind of performance-tests because the hardware is not in place, the IT-department does not provide Bug tracker, or the ordered software just still did not reach the team. These problem may slow down the pace of the sprint. Another impediment that the team cannot remove by themselves could be that the marketing or sales manager was there again demanding that another feature gets integrated “quickly”.
- The second one regards impediments that result through the organizational structure or strategic decisions. Maybe the office is not capable of handling the important meetings or teamwork – e.g. may be because there is no media.

- The third problem refers to the individuals within the team someone needs a hand with the debugging. Another one cannot solve a task alone and needs someone else for the pair programming. Someone else has to reset his computer....

Even though a Scrum Master can't fulfill all or some of the requirements by himself, he is still responsible for solving and getting rid of problems. This task often takes up a lot of time and requires great authority. The Scrum Master has to create an optimal working-condition for the team and is responsible for this condition to be retained, in order to meet the goals of every sprint – i.e. for a short sprint the defined requirements.

One mistake that occurs quite often regards the problem that the Scrum Master is seen as the personnel responsible for the team members. This is often because of the classical role of a project leader, but using Scrum it may lead to conflicts of interests and is strongly against its major principle: The team owns a management role in the method of Scrum and is therefore coequal with the Scrum Master and the Product Owner. [2], [7], [4]

Development Team

Scrum teams are the champions for sustainable development practices. The most effective scrum teams are tight-knit, co-located, and is typically seven people, plus or minus two. Teams are cross functional, having all the skills needed to create an increment i.e. team members have differing skill sets, and cross-train each other so no one person becomes a bottleneck in the delivery of work. Strong scrum teams approach their project with a clear "we" attitude. All members of the team help one another to ensure a successful sprint completion.

Agile teams use pull models where the team pulls a certain amount of work off the backlog and commits to completing it that sprint, which is very effective in maintaining quality and ensuring optimum performance of the team over the long-term. Neither scrum masters nor project managers nor product owners *push* work to the team which, in turn never hampers quality or morale.

The changed role perception is one of the most important aspects, when someone wants to understand Scrum and with the intent to introduce it in their own company. Different from other methods, in Scrum a team is self-dependent.

- The Scrum team is not just the executive organ that receives its tasks from the project leader, it rather decides which requirements or User Stories it can accomplish in one sprint.
- It constructs the tasks and is responsible for the completion of those and thus the team becomes a manager. This new self-conception of the team and the therewith aligned tasks and responsibilities necessarily change the role of the team leader/project leader.
- The Scrum Master does not need to delegate all the work and to plan the project, he rather takes care that the team meets all conditions in order to reach the self-made goals.
- He cleans off any impediments.
- He provides an ideal working environment for the team and coaches the team.
- He is responsible for the observation of Scrum.
As mentioned above, the scrum team drives the plan for each sprint.
- They try to forecast how much work they believe they can complete over the iteration using their historical speed as a guide.
- By keeping the iteration length fixed the development team gets the important feedback on their estimation and delivery process, which in turn makes their forecasts increasingly accurate over time. [2], [7], [4]

Scrum Artifacts

Product Backlog

The Product Backlog lists the requirements for the product being developed in user story format.

- It is the master list of all functionality desired in the product.
- Each item in the Product Backlog has a description, a priority and an estimate of the effort needed to complete it.
- Managed by the product owner.
- Changes as fits to product owner and business value.

- Every product has a single product backlog.

Release Plan

The Release Plan describes

- the goal of the release,
- the highest priority items in the Product Backlog,
- the major risks, and the overall features and
- Functionality that the release will contain.

It establishes a probable delivery date and cost, assuming that nothing changes.

Sprint Backlog

The Sprint Backlog is an output of the Sprint Planning Meeting. It consists of:

- The tasks for the Sprint derived from the Product Backlog by breaking stories into tasks.
- Tasks estimated and assigned.
- “Done” defines what the Team means when they commit to “doing” a Product Backlog item in a Sprint.
- Owned and modified only by the team.

Burn-down Charts

During a sprint, visual artifacts like task boards and burn down charts, visible to the team and spectators alike, act as powerful motivators.

- They create a spirit of "we're doing this!".
- Having the opportunity to show off new work at the sprint demo is equally motivating, and the consistent, incremental feedback the team gets from stakeholders at each demo creates a powerful way to develop products.
- The Sprint Backlog Burn down is a graph of the amount of Sprint Backlog work remaining in a Sprint across time in the Sprint.
- The Release Burn down graph records the sum of remaining Product Backlog estimated effort across time. [3], [7]

Scrum Ceremonies

Scrum calls for four ceremonies that bring structure to each sprint:

- Sprint Planning
- Sprint Review
- Sprint Retrospective
- Daily scrum meeting

Sprint Planning

The Sprint Planning Meeting is when the iteration is planned.

- It is time boxed to eight hours (for a one month Sprint)
- Has two parts: determining what will be done in the Sprint and how the Team is going to build the product increment during the Sprint.
- Facilitated by Scrum Master
- Team commits with available capacity and conclude sprint goal

Sprint Review

- The Sprint Review meeting is a two to four hour fixed time period meeting (for one-month Sprints)
- It is held at the end of a Sprint where the Team presents the functionality done in the iteration to the Product Owner and other stakeholders.
- Facilitated by Scrum Master.
- The Team demonstrates and discusses the work done in the Sprint.
- Product owner gives a transparent feedback.
- Inputs for next sprint.

Sprint Retrospective

- The Sprint Retrospective meeting is a one to two hours' time boxed meeting.
- Facilitated by Scrum Master.
- Both the team and the product owner attend.
- Discuss: What went well (continue), What require improvement and what is not working (remove).

Daily scrum meeting

- Time boxed to 15 min.
- Facilitated by Scrum Master.
- Used only for the team.
- Other stakeholders may join but have to stay mute.
- Each Team member explains: What he or she has done since the last Daily Scrum; what he or she is going to do before the next Daily Scrum; what obstacles are in his or her way.
- Update task progress on tool or visual board. [2], [3], [7]

Conclusion

Software development using agile methodologies is becoming a bigger reality in the daily life of software development companies. Agility brings quality to the software development and management process. In order to add value to the final software, one must have a well structures team that follows the methodology and uses correct strategies. The hybridization of scrum with other software development methodologies is common as scrum does not cover the whole product development lifecycle; therefore, organizations find the need to add in additional processes to create a more comprehensive implementation. Various authors and communities of people who use scrum have also suggested more detailed techniques for how to apply or adapt scrum to particular problems or organizations. Also, projects where the developers are geographically separated are less suitable for the scrum approach. Because scrum sprints are short, less time is available for iterative testing, making it difficult to maintain quality control for such projects when using a scrum approach.

References

- [1] "Scrum"; <http://searchsoftwarequality.techtarget.com/>, Feb 2007, [Online]. Available:
- [2] <http://searchsoftwarequality.techtarget.com/definition/Scrum>.
- [3] "Scrum Roles", <http://www.agile42.com/>, [Online], Available: <http://www.agile42.com/en/agile-info-center/scrum-roles/>.
- [4] "THE SCRUM ROLES-CEREMONIES-ARTIFACTS", <http://www.agiletrick.com/>, [Online], Available:

- [5] <http://www.agiletrick.com/about-agile/agile-frameworks/>
- [6] Dr. Mark C. Paulk, Noopur Davis, " On Empirical Research Into Scrum" <https://www.scrumalliance.org>, [Online]. Available: https://www.scrumalliance.org/resource_download/989.
- [7] "Scrum_ (software_development)", <en.wikipedia.org>, 22 May 2016. [Online].
- [8] Available: [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)).
- [9] "Scrum", <https://www.mountangoatsoftware.com>. [Online].
- [10] Available: <https://www.mountangoatsoftware.com/agile/scrum>.
- [11] "A brief introduction to scrum" <https://www.atlassian.com>. [Online].
- [12] Available: <https://www.atlassian.com/agile/scrum>.