Scrum Methodology Intended for Meetings

Dhanamma Jagli[1], Jehan Vapiwala[2]

[1]*Assistant professor, Department of MCA, V.E.S., Institute of Technology, University Of Mumbai, India.*
[2]*Final Year Student, Department of MCA, V.E.S., Institute of Technology, University Of Mumbai, India.*
Corresponding Author: Dhanamma Jagli

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-tomedium 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.

I. 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.

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.

II. 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, Scum 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 outcompete anyone else.

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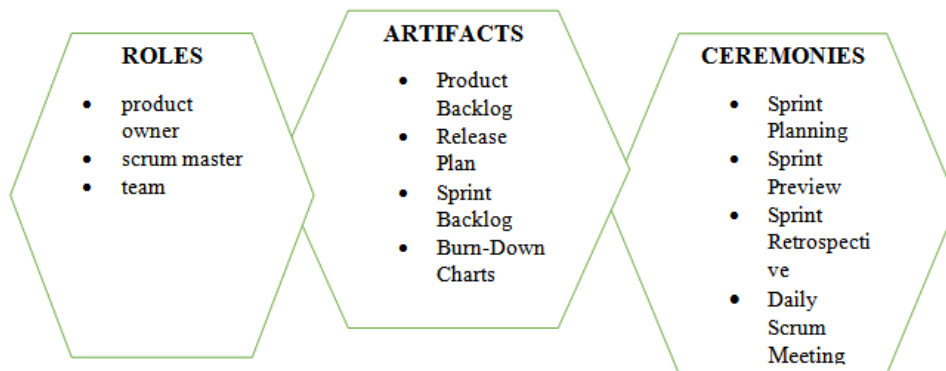
Sprints are fixed duration cycles during which the product is built and is delivered for feedback. A Sprint is one iteration of a month or less that is of consistent length throughout a development effort. Only the Product Owner has the authority to cancel the Sprint. Milestones–i.e., the end of a sprint–come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

IV. 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
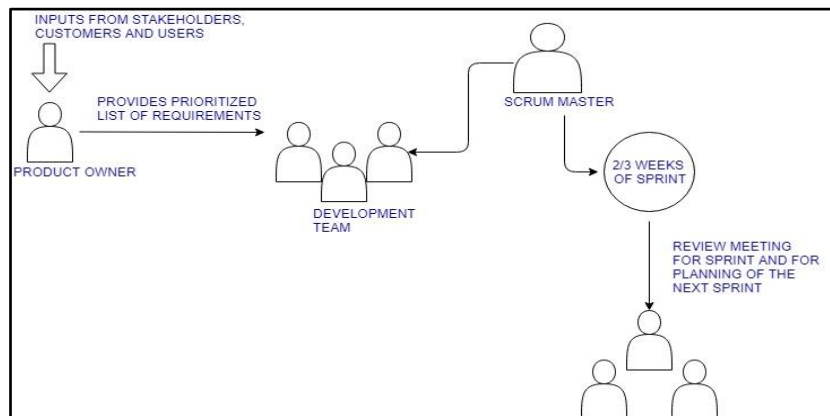• Scrum Master and
• Development Team.

Figure 1: Scrum Roles

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 counsellor.

V. 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 Scum 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 itsresponsibilities 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.

VI. 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: The Art of Doing Twice the Work in Half the Time by Jeff Sutherland (Book)
[2]. Scrum Guides: https://www.scrumguides.org/
[3]. Agile Scrum Methodology:https://www.softwaretestinghelp.com/agile-scrum-methodology-for-development-and-testing/
[4]. Scrum and Agile:https://www.mountaingoatsoftware.com/agile/scrum
[5]. Definition of Scrum:https://searchsoftwarequality.techtarget.com/definition/Scrum