**Short Paper Assignment**

*The Agile requirements refinery: Applying Scrum principles to software product management*

**SWEN 645 9040 – Spring 2021**

Version 3.0

Prepared By: Debashis Jena

03/13/2021

University of Maryland Global Campus



# Introduction

The introduction of Agile methodology and the principles have brought a revolution to the software development lifecycle. After having decades of working experience with the classic Waterfall model, the Agile methodologies introduced a different aspect of software development that had never been utilized earlier. As the Agile method and its manifesto were adopted, numerous variances were innovated. Few examples of Agile method variance are Scrum, Kanban, XP (Extreme Programming), DSDM, etc. The main objective of these processes is working software. The processes, tools, and documentation are given lesser priority. The most popular Agile method of all has been Scrum methodology.

“Software product management [SPM] is the discipline of building, implementing, and managing software or digital products, taking into account life-cycle considerations and an audience”. (Wikipedia)

Agile methodologies help the SPM in “managing requirements, defining releases, and defining products in a context where many internal and external stakeholders are involved” (Kevin et al., 2010). This short paper will elaborate on how Scrum's Agile methodology can handle SPM.

# SPM in Agile

Scrum was introduced by Ken Schwaber in 1995 (ibid.). It was a result of many issues with the long-practiced Waterfall model, where it was realized that software development “can not be planned, estimated or completed successfully" (ibid.). The Scrum method was added to this by using many flexible methods during the development process. The only things that are constant in a Scrum software development model are planning and closure (ibid.). The whole project is based on multiple sprints which are partly plan-based and partly feedback-based. Scrum introduced the concept of the backlog which replaced the heavy and rich requirement documents that are typically used in the Waterfall model. There can be two types of backlogs in Scrum; Product Backlog (PB) and Development Sprint Backlog (DSB).

* PBs are lists of all items customer requires to implement. They can be bugs, new requirements, or change requests.
* DSBs are the list of items that need to be addressed in each sprint across various teams participating in the development process.

When it comes to the Agile method as a whole, it adds another variance of the backlog which is “Product Management Sprint Backlog [PMSB]” (ibid.). “The PMSB contains all items that need to be completed within the sprint by each product manager” (ibid.).

The main input and the reference for any software development process are the requirements. The Agile methodology does not overload the process with heavy requirement documents, rather it keeps on feeding the process with a steady flow of requirements. Having PB and PMSB, in sprints facilitates the Agile method with the requirements. Sprints typically span between 2 to 4 weeks and they are a core part of the Scrum software development process. There can be several charts and matrices to measure and track the progress of a sprint; such as velocity chart, burndown chart. The objective of sprints is to deliver a piece of the requirement (story/task) that is "complete", meaning it can be released to production or can be used at the end of the sprint. The size of the story/task can be small which should fit in the span of the sprint. To address all the requirements, there are high frequent sprints. The frequency of the sprints makes it easy to accommodate change requests easily making the process flexible. Each working day within a sprint is called as Scrum which starts with a Scrum meeting (ibid.).

Diagram

Description automatically generated

Since the sprints have the objective to create shippable products, they are typically small. Scrum follows a set of guidelines and stages, to address requirements with “different sizes of granularity” (ibid.). The stages are:

1. Vision – The project sponsor brings up an idea, which becomes the vision of the project.
2. Theme – Describes the business problem or the idea behind the vision and “should be limited to a page of text” (ibid.).
3. Concept – “Themes are broken down into smaller pieces called Concepts. Concepts consist of product drivers, product constraints, and the scope." (ibid.).
4. Requirement Definition – Typically requirements are called stories. The requirements contain a set of acceptance criteria. Requirements should be discussed with the software architects and lead developers to determine the compatibility of them.

As part of the iterations, there are few recurring events. Some of them are:

1. Prepare and attend product board: The stakeholder meetings happen periodically to have the product management team present what was developed and what is going to be done in the next couple of sprints (ibid.).
2. Sprint review: Typically a sprint review meeting happens to ensure the project team is in the right direction. While the Agile method encourages accountability and taking ownership of specific requirements, team members end up working in silos.
3. Team retro meeting: Retro meetings happen at the end of every sprint and it facilitates the team in reviewing things that went well and things that need to be improved. Many feedbacks are taken which is planned to implement in the future sprints.
4. Team allocation overview: Based on the capacity and resource availability the team allocations can be changed. The team allocation reviews need to take place where the requirements assignments can be reviewed and they might be transferred from one team to another (ibid.).
5. Problem and change management: Agile method was designed originally to address the problems and changes efficiently.
6. Backlog preparations: The product backlogs are created at the beginning of the project. They are broken down into sprint backlogs.
7. Sprint planning: The planning meetings are required to have the requirements discussed and assigned to the development team. The requirement assignments are voluntarily done. Sprint planning is also done to define the scope of the requirements.
8. Demo Stories: When a story has addressed all the acceptance criteria, it will be marked as “done”. At the end of a sprint, the “done” stories are demoed to the stakeholders, to validate whether the story has been implemented as it was expected.

A few take aways from the article:

1. Sprints are performed continually to ensure requirements are well thought, prioritized, groomed, and ready before the development starts.
2. Requirements are split into multiple smaller stories or tasks if they are complex enough to fit in a sprint. That way it becomes easy to determine the scope and estimate it appropriately.
3. Scrum meetings happen daily, to discuss the things that were done on the previous day, what is going to be done today and if there are any impediments.
4. Backlogs are required to be continuously reviewed and reprioritized based on the urgency, failing to which it becomes hard for the development team and the SPM team to determine which story has a higher priority than others.

# Critique

The article tells us that, Scrum enables us to effectively collaborate among multiple teams and team members who are working on complex projects (Kevin et al.,2010). Typically a sprint review meetings happen to ensure the developers are in the right direction as a team. While the Agile method encourages accountability and taking ownership of specific requirements, there is a chance that the team members end up working in silos. While it helps project stakeholders to visually see the stories delivered, there are measures taken to ensure the team moves together towards the goal. Scrum way of building the projects is faster and effective.

* Product backlogs are the major part of the Scrum method. The product owners discuss the stories with the stakeholders and that helps in prioritizing the stories.
* Scrum encourages the project teams to emphasize more on the results than the processes and some of the unnecessary documentation.
* With the active participation of the project stakeholders, the communication between the customers and the development team becomes smooth and much faster.
* Having Vision, Theme, Concept, and Requirement definition well laid out helps the project planning very efficiently. It gives great visibility of the progress of the projects.
* Change requests in Scrum methodology are easily accepted without a lot of formalities.
* The Scrum ceremonies such as sprint reviews, retro meetings, team allocation overview, problem and change management, backlog preparation, sprint planning, and demoing stories complement each other.
* Daily Scrum meetings help the development team and project stakeholders to understand the progress of the project with the most granular information.
* Roles used in Scrum: Scrum Master, Product Owner, and Development team.

**Recommendations for Future Study**

1. With many teams working simultaneously, it becomes hard for the development teams o to manage the stories and address them properly. It might be a challenge for the Scrum method to manage the project with multiple architecture teams, multiple development teams across regions, with thousands of tasks delivered on a given sprint.
2. While the Scrum method is used mainly in software development, it is also used in other industries like automobile and healthcare. It would be interesting to see how the method performs in a non-IT group.

**Conclusion**

The article was really useful when it comes to the way the Scrum method is performed. The typical ceremonies within a Scrum team of 2 to 8, are backlog grooming, planning, sprint review. Each of these processes or ceremonies complement each other. Each of these ceremonies ensures the project runs smoothly towards the goal. Various charts and matrices like Burndown charts can be used to know the progress of the sprint.

**Reference**

No Author. (n.d.). Software product management. Retrieved from *https://en.wikipedia.org/wiki/Software\_product\_management*.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2010). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.7769&rep=rep1&type=pd.