

Agile-Scrum: A Smooth Project Execution or Challenging Bucket

Mr. Sumit Srivastava¹, Prof (Dr.) Baldev Singh²

¹Research Scholar, Ph.D., ²Dean Engineering, VGU

¹²VGU, Jaipur

Abstract- Loyal and steady customer base alone can keep the organizations successful in the current turbulent business environment. In the current era of software engineering, the success of a business process is measured in terms of 'customer satisfaction'. Day by day, customers are turning out to be more demanding, as their expectations from the software are growing. In order to achieve customer satisfaction in a meaningful way, software engineers are looking for more effective development models. "Agile" is one such model, that fits the bill and therefore industry is looking at with interest. Unlike traditional upfront-documented, plan-driven software development methods, Scrum methodology requires tight and close collaboration of all team members within the team. But the aim of this research paper is to bring the challenges and issues in Scrum implementation. This paper highlights a few challenges with Agile-Scrum and gives an insight to the user whether the agile is a smooth project execution or it is a bucket of challenges.

Keywords- Agile, Scrum, Sprint, limitation, challenges, software development, project execution

I. INTRODUCTION

Software Engineering is the technique for the development of software step by step to produce the high quality product at the end. It provides different ways to implement the software with the help of policies, procedures and processes which is called software development methodology (SDM). There are various software development methodologies and each methodology has its own software development life cycle (SDLC). When a team adapts any SDLC for the software development, it actually adapts the policies, procedures and policies within the team to take best outputs of the development life cycle such as quality and reliability. SDLC itself does not guarantee the success of the project but helps the team in success. There are several SDLC models e.g. Waterfall, Spiral and Agile but this research is focused on Scrum which is the further classification of Agile Software Development. The insights presented in the paper can be used in organizations that are in the process of agile software development using Scrum. While Scrum plays an important role in the success of Agile development, it does come with its own challenges or limitations. Scrum, in the context of this paper, refers to an Agile methodology with emphasis on project

management structure and communication between all stakeholders including clients and business representatives, regularly setting sprint time limits for software completion, reviewing changes and applying retrospection before working on the next product backlog requirements [1]. Scrum was developed in the early 1990's by Ken Schwaber and Jeff Sutherland [9]. Scrum is currently the most widely adopted Agile methodology, based on the 2017 Version One survey [4]. The reason for high adoption rates of scrum, could be its simplicity, as it can be easily understood by both business and Scrum technical teams alike. Agile adoption (which includes Scrum) has its challenges such as work specialization, organizational culture, resistance to change, and lack of communication.

What is evident from the reviewed literatures is that, whilst there are common problems and challenges identified, there are very few empirical studies on the Scrum adoption challenges experienced by individuals.

To get an overview of adoption challenges faced by individuals within software organizations, it is necessary to get to the core of the challenges, i.e.

- What are the Scrum adoption challenges experienced by project development team and stake holders, in practice?
- Is there a relationship to be discovered with the adoption challenges and Scrum adoption results?
- Will the knowledge and understanding point to a potential correlation or causal outcome?

Scrum: The Scrum software development process is an agile process that can be used to manage and control complex software and product development using iterative and incremental practices [1] and is an enhancement of iterative and incremental approach to delivering object-oriented software [9]. The origin of term "Scrum" came from the popular sport Rugby, in which fifteen players on two teams compete against each other. The first used rugby strategies to describe hyper productive development processes in Japan. Three strategies from rugby including a holistic team approach, constant interaction among team members, and unchanging core team members are adopted into Scrum management and control processes.

Framework of Scrum: The framework of Scrum consists of three components including roles, ceremonies, and artifacts. There are three distinct roles in the Scrum that are Product Owner, Scrum Master and Scrum Team. The Product Owner is

responsible for getting initial and on-going funding for the project by creating the project's overall requirements, return on investment (ROI) objectives, and release plan. The Team is responsible for implementing the functionality described in the requirements. Teams should be self-managing, self-organizing, and crossfunctional to maximize team performance. The ScrumMaster (SM) is responsible for ensuring that Scrum values, practices, and rules are enacted and enforced. The SM represents management and the team to each other. SM also tries to remove any impediments imposed on developers. There are several ceremonies in the Scrum process including the Daily Scrum Meeting, the Daily Scrum of Scrums Meeting, the Sprint Review Meeting and the Sprint Planning Meeting. The Daily Scrum Meeting (TDSM) is a 15-minute status meeting to talk about what has been accomplished since the last meeting, what items will be done before the next meeting, and what obstacles developers have. The Daily Scrum of Scrums Meeting (TDSSM) is another short daily meeting and follows the same format as a regular TDSM. The main reason for having TDSSM is to synchronize the work between multiple Scrum teams. The Sprint Planning Meeting (TSPM) is a monthly meeting, where the Product Owner and Team get together to discuss what will be done for the next Sprint which lasts usually for 30 days. In TSPM, team members break a project into a set of small and manageable tasks so that all the tasks can be completed in one Sprint. The Sprint Review Meeting (TSRM) is another monthly meeting which is held at the end of the Sprint.

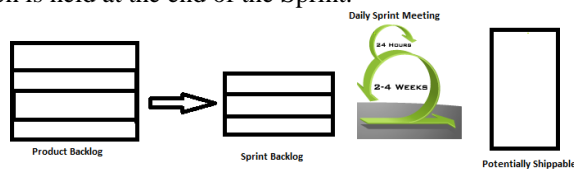


Fig.1: scrum framework

In addition to the Scrum roles and ceremonies, the Scrum process provides three artifacts namely the Product Backlog, the Sprint Backlog, and the Burndown Chart. The Product Backlog is a collection of functional and non-functional requirements, which are prioritized in order of importance to the business. The items in the Product Backlog are created and maintained by the Product Owner. The Sprint Backlog is created by team members from the Product Backlog in a way that the high priority items in the Product Backlog are first selected and broken into a set of smaller tasks. When the Product Backlog items are divided into small tasks, team members estimate the completion time for each task. Team members try to make tasks as small as possible so that every task can be accomplished within three days. The Sprint Backlog consists of these small tasks. The Burndown Chart is a graphical presentation where work remaining is tracked on the vertical axis and the time periods tracked on the horizontal

axis. The Burndown Chart should be accessible by every member who participates in the project.

Challenges:The Scrum Guide written by Schwaber and Sutherland [1]states the following about Scrum: “A framework withinwhich people can address complex adaptive problems, whileproductively and creatively delivering products of thehighest possible value. Scrum is:

- Lightweight
- Simple to understand
- Extremely difficult to master”

Scrum is a value-driven method (as opposed to a plandrivenmethod such as the waterfall method) which isiterative and incremental development [9]. The Scrum valuedrivenmethod continuously reassesses the problem whilemaking small software feature increments in short timeblocks within small teams. Scrum is so flexible andabstract in its definition and implementation that it is oftenused outside of the Software Engineering (SE) practice [2].

Adoption challenges, in the context of this paper, refers tothe challenges faced by software organizations whenchoosing and following an Agile methodology. Asmentioned in the introduction we used the narrative reviewmethod to generate the custom model's independentvariables. Before we were able to generate the independentvariables, we first had to identify the adoption challengesfaced within the global and SA context. The Scrum andAgile adoption challenges were acquired through thenarrative review method, which was as follows;

- Data sources was relatively recent i.e. all except onepaper was less than ten years of age.
- It has been cited in other literature.
- If not cited, the source must have been published byan accredited publisher, e.g. Springer, Pearson,Institute of Electrical and Electronics Engineers(IEEE), International Journal of EnvironmentalScience and Technology (IJEST), etc.
- If not published by an accredited publisher, thesource must have been presented at a knowninstitution, e.g. Agile Africa conference,Johannesburg Centre for Software Engineering(JCSE).
- Alternatively, the source is a recent dissertation orthesis paper.
- The sources where carefully perused and relevantliterature was ear marked for further investigation.
- These pre-selected literature sources were filteredbased on the content it provided, i.e. Do the literature sources contain challenges and issuesexperienced during Scrum and Agile adoption? Oris the literature describing adoption challenges onirrelevant Software Development Methodologies(SDM)?
- Identified challenges within the literature wascollated and the frequency of occurrence wasrecorded.

Table I is a consolidated list of global Scrum and Agile adoption challenges taken from twenty-one literature studies [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], with publication years ranging from 2008 to 2017. The listed challenges, the one that is very peculiar comes from the mixed mode study which recorded that cross functional generalist teams were not believable in the environment. This is contradictory to the Scrum philosophy of well-balanced redundant knowledge teams with the ability to work on various aspects of projects without the dependency of team member specialization.

S.No.	Global Scrum and Agile Adoption Challenges	Frequency
1	Lack of knowledge/training/skills	11
2	Organizational culture/mindset	9
3	Teamwork/communication issues	9
4	Lack of documentation	5
5	Budget and schedule constraint	2
6	Escalating commitment	2
7	Hard to scale	2
8	High management overhead	2
9	Lack of senior support	2
10	Work specialization	2
11	Cross functional generalist teams	1
12	Increase stress and workload	1
13	Lack of quality	1
14	Lack of top management support	1
15	Long time to market	1
16	Low user satisfaction	1
17	Over engineered solutions	1
18	Over optimistic task estimates	1
19	Project team size	1
20	Requirements creep	1
21	Retrospective inadequacy	1
22	Too many meetings	1

Table: Global Scrum and Agile Adoption Challenges

Identified Issues: Several issues are identified from SLR (Systematic Literature review). These issues are directly affecting Scrum implementation and are related to management, development and release process. Here is a detailed description of each [10].

- **Quality Items Pileup:** Due to agility of work in Scrum, teams have pressure to present something in a short time. Because of this sometimes teams ignore quality of software and create a pileup of quality related items.

- **Module Integration Issues:** As the products are released frequently during a sprint module integration testing cannot be performed properly all the times, as it requires a lot of time for testing and quality assurance. Inadequate time allocation for QA of large or complex systems can easily generate critical issues.
- **Code Quality:** Teams have a short term deadlines due the agility of the teams. To cope up with a lagging deadline, the developer needs to put extra hours. This will create Code Quality related issues. However, one cannot write bug free code all the time especially when he is working under pressure.
- **Disruption in Team Work:** From survey results, it is noted that Product Owner and Scrum Master interfere with team members by asking statuses like they used to ask in traditional SDLC. Mature vs. Immature Scrum: This is also proved from results that teams which are mature in Scrum implementation have relatively fewer issues than immature teams.
- **Sprint Duration:** Sprint duration plays an important role in effectiveness of Scrum. Scrum Master has to wisely select sprint duration to get maximum benefits of this framework. Results clearly highlight the number of issues at different sprint duration.
- **Lack of Scrum Training:** Results also revealed that 50% team members lack formal training of scrum and are unaware of the scrum process. The knowledge that they have gained is either because of the other team fellows or from their scrum masters.
- **Release Process:** Another major issue in scrum is the Release process/ deployment process. Agility in work is introduced by Scrum; sprint deployment is the major concern for every team.
- **Backlog Management:** Scrum provides insufficient guidance with respect to the structure of the backlog. The scrum management tools that are available in the market are either too complex (which in turn are more expensive) or too simple and are therefore not useful for the team.
- **No Technical Practices:** Although scrum shows a good project management capability, there aren't any technical practices that can be called best.
- **Multiple Teams:** The survey showed that working with multiple teams is a tough job, unlike the traditional SDLC methods because scrum doesn't have strong advices on it.
- **Metrics:** Burn down charts and velocity is used for measure metrics in Scrum. These metrics can only be useful in the presence of a Scrum Master or Product Manager who would help in analyzing the burn down chart for its peaks.
- **Risk Management:** There is no plan or strategy in scrum to mitigate risks. It is an important factor for any project and hence work is required in this area of Scrum.
- **Documentation:** Agile Scrum believes in no documentation, however this phenomenon is still not successful in real environment where things come in and out through email or any other source, which is quite difficult to track.

- **Too Idealistic:** Scrum assumes teams are self-managing and self-disciplined. As per rules of Scrum, Scrum master job is to remove all impediments for his team and product owner job is to provide the requirements in stories which will help to develop the project incrementally. However, in practical this scenario is not valid.
- **Communication/Scrum Ceremonies:** Due to increase in communication, some team members feel disturbance in their work. They are not able to concentrate on their work as number of meeting increases.

Conclusion: Scrum is the most widely used SDM at present, providing many organizations with a simple to understand methodology to complete project management tasks. While the advantages to using this methodology are easily noticed by adopters, the challenges during the adoption stage are recurrently not quantitatively detected. Most research on the adoption challenges primarily focused on qualitative measures for detection with case studies being the most implemented strategy. The inability to easily detect these adoption challenges can lead to teams and individuals within software organizations not using Scrum correctly or not adopting Scrum altogether, which could potentially limit the successful outcomes of a project. The extant Scrum adoption challenges were acquired through a narrative review of Scrum adoption challenges, both within the global context. The SACDM was developed to detect Scrum adoption challenges with the objective of equipping adopters with the knowledge and awareness to overcome them.

II. REFERENCES

- [1]. Asnawi, A.L., Gravell, A.M. & Wills, G.B. 2011. Empirical Investigation on Agile Methods Usage: Issues Identified from Early Adopters in Malaysia. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 192–207.
- [2]. Santos, R., Flentge, F., Begin, M.-E. & Navarro, V. 2011. Agile Technical Management of Industrial Contracts: Scrum Development of Ground Segment Software at the European Space Agency. In A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo, eds. Agile Processes in Software Engineering and Extreme Programming. Madrid: Springer Berlin Heidelberg: 290–305.
- [3]. Fægri, T.E. 2010. Adoption of Team Estimation in a Specialist Organizational Environment. In A. Sillitti, A. Martin, X. Wang, & E. Whitworth, eds. Agile Processes in Software Engineering and Extreme Programming. Trondheim: Springer Berlin Heidelberg: 28–42.
- [4]. Marchenko, A. & Abrahamsson, P. 2008. Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges. In Agile 2008 Conference. IEEE: 15–26. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4599449>.
- [5]. Overhage, S., Schlauderer, S., Birkmeier, D. & Miller, J. 2011. What Makes IT Personnel Adopt Scrum? A Framework of

Drivers and Inhibitors to Developer Acceptance. In 2011 44th Hawaii International Conference on System Sciences. IEEE: 1–10.

- <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5718964>.
- [6]. Heikkilä, V.T., Paasivaara, M. & Lassenius, C. 2013. ScrumBut, But Does it Matter? A Mixed-Method Study of the Planning Process of a Multiteam Scrum Organization. In 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE: 85–94. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6681341>.
 - [7]. Dong, L. 2008. Exploring the impact of top management support of enterprise systems implementations outcomes: Two cases. Business Process Management Journal, 14(2): 204–218.
 - [8]. Allisy-Roberts, P., Ambrosi, P., Bartlett, D.T., Coursey, B.M., DeWerd, L.A., Fantuzzi, E. & McDonald, J.C. 2017. The 11th Annual State of Agile Report. Journal of the ICRU, 6(2): 7–8. <https://academic.oup.com/jicru/article-lookup/doi/10.1093/jicru/ndl025>.
 - [9]. Schwaber, K. & Sutherland, J. 2011. The Scrum Guide. Scrum.org, October, 2: 17. <http://www.Scrumalliance.org/>.
 - [10]. Sumit Srivastava, Prof. Baldev Singh, “Study on Agile Software development, benefits and challenges, IJRECE, vol 6 issue 2, June 2018”

Bibliography



The author is having 20 years of IT experience in multi-functional areas with specialization in software project, product and program management. Extensively using agile methodologies like XP, Kanban, Scrum etc. An eminent speaker and presented various training, papers and lectures on Scrum.



This author is a Dean Engineering in VGU, Jaipur. His research interest areas are Neural Networks, Cloud computing. More than 20 books he has published. More than 20 research papers has been published by him or he is the co authored of them.