

AGILE AND SCRUM IN A SMALL SOFTWARE DEVELOPMENT PROJECT- A CASE STUDY

Shanmuganathan Vasanthapriyan

Department of Computing and Information Systems, Sabaragamuwa University of
Sri Lanka, Sri Lanka

priyan@appsc.sab.ac.lk

ABSTRACT

Agile software development practices, like Scrum, has been adopted for many software development projects, due to its ability to deal with changing product requirements, while waterfall methods are better suited to dealing with projects that have clearly defined requirements. Agile allow teams to focus on delivering product and improved communication has made it one of the easiest and best software development techniques. On the other hand, such agile methods have been designed for collocated software development and are thus not directly applicable to distributed agile development. In this research, we present recommendations from case study conducted in Sri Lankan software company where they practice scrum development for projects. We also highlight some of the challenges and benefits and unique lessons learned from them.

Keywords: Agile development, Scrum, Software development with agile, Software product management

1. INTRODUCTION

Agile, as a concept, came to prominence two decades before through the endeavors due to its ability to deal with changing product requirements, while traditional methods are better suited to dealing with projects that have clearly defined requirements ((Agile Manifesto, 2017; Dybå, & Dingsøyr, 2008; Conboy & Morgan, 2011). Agile software development practices, like Scrum, that allow teams to focus on delivering product and improved communication has made it one of the easiest and best software development techniques. Agile teams tend to be small comprising of varied specialists capable of engaging in several types of activities such as analysis, development and testing. Chow (Chow & Cao, 2008) highlighted the differences connecting agile and traditional software development methods in software development and described the well-known agile method as scrum framework. Agile advancement is predictable for considering the challenges related to the software developers, which could influence its achievement considerably (Cao et al., 2009).

The term “scrum” was coined by Ken Swaber in 1995(Schwaber,1997). This methodology was designed to solve the repeated changes in business requirements. Scrum methodology is only used to simplifying the project during easy procedure, easy to update documentation and highly mutual relationship



between the members of the team.

Scrum methodology has been adopted for many software development projects in many software development companies. In this research, we conducted a case study in a small size software development company which practices scrum methodology in their software development (Jalali, & Wohlin, 2010).

Unfortunately, authors found that presently there is a gap in literature concerning the agile and scrum in a small software development project practices in software companies in the context of Sri Lanka. Therefore, it is not possible to conduct an analysis or review for agile and scrum practices in software companies in Sri Lanka. Such that, this study aims to reduce this gap by evaluating current status of agile and scrum practices in a Sri Lankan software company using an empirical investigation.

The contribution of this paper consists of practical recommendations, which could be useful for several academic researchers and other software development companies that are planning to use Scrum and/or Agile practices.

2. CONTEXT OF A CASE STUDY

This targets for a hardware store company (HSC) that wants to implement a Business intelligence system to analyze their company insights. A business Case Study (Runeson & Höst, 2009) project was started in June and was planned to be finished in middle of September by using Agile project development principles. We carried out the investigation using a qualitative approach, namely in-depth semi-structured interviews. Further, qualitative research provides the understanding of phenomena within their real-life context.

Interviews were carried out based on grounded theory methodology (Corbin & Strauss 2014). The primary reasons for choosing this methodology in this study are: (a) Software engineering is a knowledge intensive and collaborative activity where software engineers are involved, and grounded theory is best fitted for studying people-related issues. (b) grounded theory is an advisable method for studying a phenomenon that is not studied in deep, and agile software development has not been studied yet in Sri Lankan industries. (c) Several grounded theory studies in software engineering context have led to great findings.

Originally it is planned to finish the project in two months in three iterations (Stage 0 – Stage 2) and then support the finished system for about 3-9 months.

The Stage 2 was planned mostly for the refactoring, final testing and bugs fixing only. When the project was started, the development team found inconsistencies between graphical design, list of requirements and project specification which was based on the previous version of the project.

2.1 Core Scrum roles and relation to the project team members

In order to define what is being done by whom, let's start with the definition of roles that they have used in Scrum (Agile Manifesto, 2017)

- a. Product Owner - It is the core role from the customer side. He will be in-charge of defining what they want to include in the project, which is the priority for any functionality, which are the conditions to consider the functionality as fully working and development acceptance. He also collaborates with the development team to define, analyze, and detail every requirement.
- b. Development Team: A Development Team is made up of 5-7 people with cross- functional skills who can perform analysis, design, develop, test, technical communication and documentation. The Development Team in Scrum is self-organizing, even though they may interface with project management organizations (PMOs)". At the end of the development they analyze how to improve the development process for the next requirements. Inside this team, they defined different sub roles, such as the solution architect, programmers, testers, functional analysts, graphical designer, etc.
- c. Scrum Master -The Scrum Master is the person in charge of coordinating all the developments, ensuring that development team understands and follows the Agile principles. He also trains the team in order to help for its auto organization; help to remove locks and impediments; and try to avoid risky interruptions that can disturb the team from its main task.

2.2 Specific Scrum Meetings

- a) Daily meeting: As commented in the general approach, the development team will have a 15-minute daily meeting in the morning to comment on the ongoing tasks. In this meeting, everybody in the team should answer three questions:
 - a. What have you done since yesterday's meeting?
 - b. What are you doing today?
 - c. Which blocking points have I found or I think can appear?
- b) Sprint zero meeting: The aim of this meeting or meetings is to define

what the project scope is and which functionalities are expected by the user to define the overview of project requirements. From this meeting, they extract the Product Backlog, a list of requirements to develop in order to be able to deliver a working version of the project.

- c) Sprint planning meeting: In this meeting, they choose an agreement with the Product Owner, the list of Product Backlog Items that they are going to develop in the sprint, also considered as Sprint Backlog. These items have been split into tasks in order to have a task size that is manageable by the team. This meeting is split in two parts: the first one including the product owner and the development team and the second one just including the development team.
- d) Product backlog refinement: This meeting can be scheduled at any point of a sprint but should be done for all sprints, in order to review which, the remaining Product Backlog is, if there are any added tasks to include it in the list, some priority change, etc. As in the sprint zero meeting, it's required that the Product Owner and the development team assist at the meeting and the expected output is the Product Backlog reviewed and updated.
- e) Sprint review: Once the sprint has finished they will review with the Product Owner and the stakeholders all the PBIs implemented. This will give their customer the possibility to review his expectative, reprioritize tasks, change anything that he considers, or add new requirements based on a real result about what their tool can do. Then during the product backlog refinement meeting he will be able to reschedule their work.
- f) Sprint retrospective: This is an internal development team meeting that will allow the team to review they have done, how they have solved the issues that have appeared, what they can do to avoid problems to appear again, and how to avoid locks that can freeze their development. This meeting will allow the continuous improvement of the whole team improving anyone's productivity. The output of this meeting is the list of improvements that they should apply to get a better performance of their development phase.

2.3 Sprint

The sprint concept or iteration is one of the most important concepts in Scrum, also in most of Agile methodologies. It can be also considered as iteration. Sprint is the single cycle of development inside a project that will provide a fully usable version of the project available for the customers to be tested. It is a period of between 2 and 6 weeks trying, depending on the task size of the project, with the intention of being reduced to the minimum as possible as shown in Figure 1.

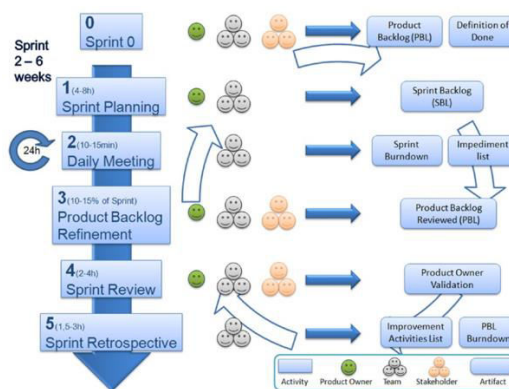


Figure 1: Sprint of 2-6 weeks

2.4 Project Artifacts

The main deliverable produced using the Scrum framework is the product itself, and Scrum expects to build a properly tested product increment (in a shippable state) at the end of each sprint (Diebold & Dahlem, 2014). The product backlog is another deliverable, and it is maintained and prioritized by the product owner. It is a complete list of the functionality (user stories) to be added to the product, and there is also the sprint backlog which is the list of functionalities to be implemented in the sprint. Other deliverables are the sprint burnout and release burnout charts, which show the amount of work remaining in a sprint or release, and indicate the extent to which the sprint or release is on schedule. From the sprint explanation, we have seen some tools that helped them to go ahead with their Scrum methodology. It is one of the most popular software for development management from an Agile perspective called JIRA.

3. LESSONS LEARNED

In order to provide a better understanding of the context and characteristics of this study, their agile development processes, and their compliance with agile

practices, we here summarize the experiences and lessons learnt.

- a. In general, the most common mentioned activities to achieve Value were related to agile practice, optimizing and automating development processes, in particular the testing processes and strategies. For example, to achieve the Value Aspect (VA) of Delivery Process w.r.t. time, the majority of the answers were about optimizing the processes and different agile practices such as iterations, backlog meetings, development loops, and scrum of scrum.
- b. For the agile manifesto's value of customer collaboration, we found the related decision characteristics show decisions depend heavily on the technical experience level of the customer or their representative, e.g. BA, whose team interaction meant less customer collaboration.
- c. The best approach could be for Proper Scrum training for team members can be held during the Daily Scrum meetings and/or more often sprint review meetings, when everybody could suggest or advise modifications for the development process. Support from other team members during meetings is crucial.
- d. Customer relationship- The Scrum development team collaborated with the BA, the customer representative, rather than the actual customer. Scrum development team calls for more collaboration with the customer, and in this case the BA worked very closely with the customer and represented him/her to the team as a direct surrogate.
- e. Scrum development teams work in short iterations with significant decisions made each iteration. A decision process challenge is that decisions are made for a specific period and then ignored or changed during that time.
- f. Based on a few retrospective meetings, a common challenge for decision intelligence was that decisions made mid-iteration are not tracked or documented.
- g. Using the whiteboard to record and track tasks is a common process in Scrum development teams, but the team noted inconsistencies between the task "updates" captured by the physical Post-Its on the white-board and the task data entered into JIRA.
- h. Team: Agile methods require knowledge and experience of the team.
- i. Collaborative tools facilitate the adoption of agile methods.
- j. Some of the barriers were identified for achieving value.
 - a. Somewhat vague requirements
 - b. Different level of knowledge within the customers
 - c. Deadlines/delivery time to customer/Insufficient time and focus

- d. Multiple communication tools must be used by the team

4. CONCLUSION AND RECOMMENDATIONS

In this paper, we reported a case study about adopting agile methods especially Scrum methodologies in a software company. However, not all the implications of adopting agile methods in the software company are widely known. Therefore, we interviewed Agile software development teams from software company, aiming at expanding "what is known about the adoption of agile methods in their company", proposing a set of recommendations for adopting agile methods in this context.

- a) Daily SCRUM meetings are essential: The daily stand-ups, or SCRUM meetings, that are essential within the SCRUM development method, are also valued highly within the agile Software product management method. The 15-min meeting at the start of each day is experienced as a positive, helpful aspect of the process. By providing constructive critique, potential problems can be avoided and existing problems can be solved.
- b) Early collaboration promotes reuse and integration: Since product managers in a SCRUM team cooperatively work on a Product Management Sprint Backlog and discuss requirements before they have been implemented, reuse and integration opportunities can be spotted at an early stage.
- c) The experiences of the case study company have shown that, to ensure effective agile Software product management, several factors should be taken into account, such as task size, backlog structure and willingness to keep the backlog up-to-date. By providing the specific lessons that Plan on has learned during its experience with agile Software product management and SCRUM.

REFERENCES

Agile Manifesto, (accessed 11th of September 2017). [link]. URL <http://agilemanifesto.org>.

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4), 332-343.

Chow, T. and Cao, D. (2008). A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6), 961-971.

Conboy, K. and Morgan, L. (2011). Beyond the customer: Opening the agile systems development process. *Information and Software Technology*, 53(5), 535-542.

Corbin, J. and Strauss, A. (2008). Basics of qualitative research: Techniques and procedures for developing grounded theory.

Diebold, P. and Dahlem, M. (2014), May. Agile practices in practice: a mapping study. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 30-44.

Dybå, T. and Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9),833-859.

Jalali, S. and Wohlin, C. (2010). August. Agile practices in global software engineering-A systematic map. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, 45-54.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2), 131-143.

Schwaber, K. (1997). Scrum development process. In *Business object design and implementation*. 117-134.