



Increase productivity of team members in a Scrum project

Mustafa Nizamul Aziz

Information Systems Discipline, East West University, Dhaka, Bangladesh

ARTICLE INFO

Article history

Accepted 13 July 2020

Online release 23 July 2020

Keyword

Scrum, sprint
Team productivity
Software development project

*Corresponding Author

M.N. Aziz

✉ mustafa.nizamul@gmail.com

ABSTRACT

Scrum is an agile method that has been proved as a successful one that handles the frequent requirements changing and quick progress. Researchers have been trying to enhance scrum productivity and validity. Increasing scrum team productivity is a vital element that enhances project productivity and stability. The main purpose of this paper is to find out the properties that improve members' productivity in a scrum team without adding people to the existing team. The paper includes findings from a course project done by a group of students in a graduate program. The project was run for four weeks, a planning phase of a week for customer meetings to get the system requirements, and three sprints for system development with one week each. Document analysis and observation were used as data collection methods in the study. At the first sprint, the team velocity for system production was 40%, and for the second sprint, it was 71%. The team had produced and delivered more tasks in the second sprint than the first one. The paper includes good practices that enhanced team productivity in a real-time scenario.

INTRODUCTION

Agile development is an iterative and incremental software building method, which is fast and rapid having the ability to take actions quickly and react to change, at the same time welcoming new change in any point of software development (Larman, 2003; Koch, 2004). There are several familiar agile development methods. Among them, Scrum, eXtreme Programming, adaptive software development, Feature-driven development, etc. are mostly used in software companies (Larman and Basili 2003). Scrum emphasizes different practices such as daily meetings, self-organizing teams, colocation, sprints of few weeks, demo to stakeholders regarding the features completed after each sprint (Larman, 2003).

Team productivity can be determined or measured in ways of timeliness and quantity (Melo et al., 2011). There are two possible ways to have better throughput in scrum projects: adding more people in the team and increased productivity of teams

(Maher, 2011). This paper will focus on the latter one: increasing team productivity. In this paper, the author will summarize some literature studies on this topic and will contribute by reflecting on a real-time scrum team. For increasing scrum team productivity, many companies are using a matrix model (Maher, 2011). According to this model, projects should be the central unit of an organization. Resources and people are moved to projects. For example, if a project requires two testers, the manager meets this request having two full-time testers or several part-time testers who are active in multiple projects. When the project ends, the people and resources are returned to their functional teams and keep them available for the next possible project. In an agile world of development, the team is the core unit in an organization. Teams are permanent, and projects are moved to the team. Team members are aware of their strengths and weaknesses and learn how to communicate, resolve conflict, and collaborate (Maher, 2011). Also, there is evidence that short-lived teams for a project are correlated with having

lower productivity (Katzenbach and Smith, 2015). Sutherland et al. (2008) discussed some complexities in agile teams, which can decrease members' productivity. These could be passed by to increase team productivity. One of those limitations is, for example, cultural differences. Three important things to overcome these differences are good personal relationships, an open team culture having direct communication, and an open company culture having an equal value system. Scrum principles have a vital role in scrum projects too. Two practices can increase the scrum team productivity: Pair programming, and collocation (Melo et al., 2011). However, increasing scrum team productivity is a vital element that enhances project productivity and stability. The study is consequently conducted to find out the properties that improve members' productivity in a scrum team without adding people to the existing team.

METHODOLOGY

An academic project was done by a scrum team consisting of a group of graduate students to create a web-based booking system for arranging conferences. The project was run for four weeks, a planning phase of a week for customer meetings to get the system requirements, and three sprints for system development with one week each. The scrum team was not familiar with the scrum method, more than half of the team had software development experience, and the others did not. The team was divided into different sub-teams or roles, for every sprint or phase. The author of the paper discussed the productivity factors for the first and second sprint only. Document analysis

and observation were used as data collection methods in this study. Empirical data from teamwork in the development project like reports, observation, team discussions, and reflections were collected.

RESULTS AND DISCUSSION

As shown in the team velocity chart (Figure 1), at the first sprint, the team velocity for system production was 40%, and for the second sprint, it was 71%.

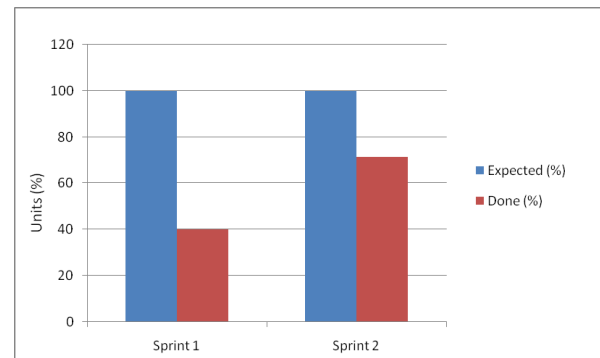


Figure 1: Velocity chart

In the first sprint, the scrum team delivered 0 out of 22 tasks, but in the second one, the team could deliver 7 tasks out of 29 (Table 1). As shown in table 1, team members were assigned to different roles in each sprint, for instance; the first sprint had five developers, three testers, and three designers. Since in the second sprint, some team's roles were changed according to the team sprint evaluation, the second sprint consisted of six developers, one tester, and four designers.

Table 1: Comparison between sprint 1 and 2

Project status	Sprint 1	Sprint 2
Incremental releases/ Closed tasks	0%	25%
Velocity	40%	71%
Developers	5	6
Testers	3	1
Designers	3	4
Daily Scrum meetings (%)	75%	100%
Face to face communication (%)	~50%	~85%-90%
Pair programming	no	yes

The team did not use pair programming practice in the first sprint while they used projector programming, where most of the developers worked together on the same task using a multimedia projector. Then in the second sprint, they stopped the projector programming style, they programmed separately in different tasks and used pair programming practice. Daily scrum meetings were conducted three times in the first sprint. In the second sprint, they were conducted on all four days.

As we can see in figure 1, the team had produced and delivered more tasks in the second sprint than the first one. The team had higher velocity compared to that in the first sprint; in other words, the team was more productive in the second sprint while both sprints were one-weeklong. It was four working days with a planning day in the week having no work at the weekend. Before starting sprint-1, the team had some customer meetings to decide system requirements. The team had many long internal discussions about user stories, scrum management tool to be used, code repository, and team communication tools. When sprint-1 started, developers and testers had not installed development environments in their computers yet, which took them two working days to successfully install the development tools. As the team had two remaining days to do the rest part, developers used the projector programming style. The development team was working on the same task which caused slow code production and decreased team productivity.

Accordingly, the testing team had not many tasks to do, because they were waiting for some tasks to be tested. Designers were involved in designing webpages. The team probably would underestimate the total production time for each user story. Also, some user stories in sprint-1 had many sub-tasks which caused the team to underestimate the production complexity. Scrum meetings were not conducted on all days. Delayed scrum meetings, in some days, with lower quality, caused bad information flows among the members. Also, groups were working mostly in different rooms; developers and/or testers in a common room, designers in a different room. The sprint had many “in development tasks”.

On the other hand, in sprint-2, the team had daily scrum meetings at 10:15 am, the whole team worked in the same room mostly. That caused better communication having a better information flow. While the production environment was already installed in the machines, developers could produce more tasks, testers had more tasks to test, designers were more involved and productive because they were designing the closed tasks. The user stories in this sprint consisted of single tasks, so time estimation was better and easier. The “in development” tasks per developer were less than sprint-1. Pair programming was used in the case of complex programming problems and complex bug fixing. The team used the simple design method and delivered more tasks than sprint-1, thus team cared about delivering functionality rather than a complex design.

Accordingly, the practices that enhanced team productivity in second sprints were:

Daily scrum meetings: Team members were more updated about the system development’s status and the next task for each member. Scrum meetings should be in the morning at a suitable time for the team.

Face to face communication: It gave a better performance and understanding of team members. Members could ask for help or inquiries instantly and got supports.

Working environment: Tools like a projector and whiteboards were very useful tools, they helped the team or group to discuss and explain issues.

Customer involvement: Especially in the first sprint, the team had some continuous internal and long discussions regarding system requirements; long discussions could be shortened by asking the customer directly whenever required.

User stories: Based on user stories, creating subtasks caused the team to have many “in development” tasks; which was inefficient in the project.

Pair programming: It was a good programming practice, though it is optional in scrum developments. Team members should decide how

much pair programming would be better to use, when, and where.

Projector programming: It was an inefficient programming method, time consuming, and in effective method.

Team roles: It is better to analyze the situation after each sprint and assign the suitable a role to the suitable person according to the sprint demands.

Simple design: It was very important to keep a simple design with working software, rather than a complex design without delivering something.

Respect: Respect to team members, time, and scrum principles are important to make scrum projects successful and to increase team productivity.

One project to engage with: Scrum team should have one project to work on at a time; the team can be assigned to another project after finishing the current one.

Development environments preparation: The team should decide and install development environments before starting the sprints so that team can start directly production phase in sprints. Also, the team needs to test and train using these development environments if they are not familiar with it. All team members have to install the development environments in their machines because roles can be changed.

CONCLUSION

The paper has mainly focused on scrum teams and projects. It has included reflections from a real-time sprint project and ended up with good practices for increasing members' productivity in a scrum team. A similarity check has been made between the literature and the project findings. Interviews and/or surveys can be conducted for medium or large scrum teams and to compare the results with the findings from literature studies can be interesting future research topics.

REFERENCES

- Katzenbach JR and Smith DK (2015). *The Wisdom of Teams: Creating the High-Performance Organization* (Reprint edition). Harvard Business Review Press.
- Koch AS (2004). *Agile Software Development: Evaluating the Methods for Your Organization*. Artech House.
- Larman C (2003). *Agile and Iterative Development: A Manager's Guide* (1 edition). Addison-Wesley Professional.
- Larman C and Basili VR (2003). Iterative and Incremental Development: A Brief History. 36, 47–56.
- Melo C, Cruzes DS, Kon F and Conradi R (2011). Agile Team Perceptions of Productivity Factors. 2011 Agile Conference, 57–66.
- Maher R (2011). Increasing Team Productivity: A project focus creates waste and leaves value on the table. 4.
- Sutherland J, Schoonheim G, Rustenburg E and Rijk M (2008). Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams. Agile 2008 Conference, 339–344.