Pros and Cons of Rotating Scrum Master Role – A Qualitative Study

Narasimha Bolloju
LNMIIT, JAIPUR
India
narsi.bolloju@lnmiit.ac.in

Rahul Chawla
LNMIIT, JAIPUR
India
y14uc214@lnmiit.ac.in

Rajat Ranjan
LNMIIT, JAIPUR
India
y14uc221@lnmiit.ac.in

ABSTRACT

Scrum has become a very popular software development methodology in recent years. In scrum teams, often the scrum master role is not a full time. Typically, a scrum master either contributes to multiple projects or shares the development effort along with other team members. In this latter situation, it is possible to fix a scrum master for the entire duration of the project or to rotate the responsibility among the team members. Though some practitioners and researchers suggested possible benefits from scrum master role rotation, there are no empirical studies reported in this regard. This paper summarizes perceptions of members of several teams who worked on projects following scrum methodology. As part of a quasi-experiment half of these teams of students of a software engineering course had their scrum master rotated after each sprint, and other half of these teams had a fixed scrum master for the entire project duration. This paper compares perceptions of various team members regarding the pros and cons of rotating the role of scrum master.

CCS CONCEPTS

• **Software and its engineering** → Software Development process management → Agile Software development

KEYWORDS

Agile project management, Scrum master, Qualitative study, Empirical Study

ISEC '18, February 9–11, 2018, Hyderabad, India

ACM ISBN 978-1-4503-6398-3/18/02□$15.00
https://doi.org/10.1145/3172871.3172883

ACM Reference format:
N. Bolloju, R. Chawla and R. Ranjan, 2018. In *Proceedings of Innovations in Software Engineering, Hyderabad, India, Feb 2018* (ISEC'18), 5 pages.

1 INTRODUCTION

Scrum is an iterative approach for software development in which the required system functionality evolves through collaborative effort of self-organizing cross-functional teams [1,2]. Scrum is a popular agile methodology [17] in which team members play three distinct roles named product owner, scrum master and developers. Whereas the product owner ensures that the developed product meets the client's requirements, the scrum master is responsible for ensuring that the team follows the methodologies and for removing any impediments as the team progresses. Thus, the scrum master is sometimes viewed as a protector of the team [6].

Although the scrum master role is played by a former project manager or a technical team leader, anyone knowledgeable about the scrum methodology should be able to handle it [5]. The scrum master – analysing the product backlog and sprint backlog of each sprint - ensures that the team does not over-commit themselves. In essence, a typical scrum master helps the organization to learn and adopt scrum and continually improves the software development process.

In scrum teams, often the scrum master role is not a full time job. Typically, a scrum master either contributes to multiple projects or shares the development effort with other team members. In this latter situation, it is possible to fix a scrum master for the entire duration of the project or to rotate the responsibility among the team members. Though some practitioners and researchers suggested possible benefits from scrum master role rotation, there are no empirical studies reported in this regard.

The main objective of this research is to understand differences between the perceptions of team members who worked on projects that rotate scrum master across different sprints and the perceptions of team members who worked on projects having a fixed scrum master for the entire project duration. Section 2 presents an overview of scrum methodology highlighting the different roles of scrum team members. Sections 3 and 4 describe the research hypothesis and methodology respectively. Section 5 summarizes the advantages and

disadvantages perceived by respondents who worked on teams that either had a fixed scrum master or had a different scrum master for each sprint. Finally, section 6 concludes the paper with a summary and limitations of this study

2 BACKGROUND

Software development has many challenges. Mainly two challenges are present in software development i.e. the technical and operational challenge. Here we would talk about the operational challenge. Scrum addresses these challenges in a way that teams are cross functional. They work at the same level of their fellow teammates. This is turn leads to better handling of the issues being faced during the project development phase at each level [9]. Scrum Master works on managing the project as well as working on the project at the same time. As all the team members are on the same level so there are no ego clashes and thus it does not hamper the overall project development speed[10].

In scrum methodology, there are mainly three roles. First role is that of the product owner who is responsible mainly for keeping a check whether the product which is being built is right or not. Product owner is mainly responsible to ensure that the team is moving in the right direction or not [11]. Scrum master acts as a coach and work towards helping the team work together in the best viable way [12]. Scrum master provides his/her services in the form of managing team meeting, facilitates effective communication between team members. Scrum master works towards removing all the obstacles in the progress of the project, performs the duty of project management as well [3]. The scrum master is like a coach for the team, helping the team do the best work it possibly can. Third role is that of the team members (or developers). They all work at the same level. Team members collaborate to work together on doing the task allocation. They determine realistic goals according to priority and work towards attaining the best of their goal towards project completion [13].

3 RESEARCH HYPOTHESIS

Small teams can become more productive as the team members work at the same level and the tasks allocation is done as per the specialty or skills of the individuals; better understanding between the team members will result in more work being done rather than just being planned [14]. In scrum teams, the rotation of scrum master is likely to affect the productivity, as every time rotation is done then team has to adjust with the new scrum master. This in turn could slow down the team's progress and thus affects the productivity of the team as well [16]. On the contrary, sometimes it is better to rotate the scrum master. By getting a new scrum master into this role we can anticipate continuous improvement and innovations. One scrum master may come up with a new structure for the retrospectives, and another may come up with better charts or metrics to present team performance improvement.

Following on the above, we hypothesize is that there should not be any major differences in the perceptions of team members working in teams that keep the role of scrum master as fixed versus teams that rotate the role of scrum master after each sprint.

4 METHODOLOGY

About half of these teams – belonging to one section of the course - had fixed one of their team members as the scrum master for all the three sprints of their project. The remaining the teams – from the other section - had their scrum master role assigned to a different member after each sprint. All these projects were completed in three sprints with each sprint of three-week duration. After the final sprint, the students who had their scrum master role rotated after each sprint were asked to provide their perceptions on three advantages and three disadvantages of changing their scrum master based on their project experience. The students who had worked in teams that had a fixed scrum master for all the sprints were also asked to provide possible advantages and disadvantages if they had their scrum master changed after every sprint. Although this set of students had worked on projects with a fixed scrum master we felt that their perceptions – due to issues they had with a fixed scrum master – on rotating the scrum master role will be useful for comparison purposes.

The textual responses (three advantages and three disadvantages; each of about 3 to 4 lines) were collected, and coded as part of the qualitative analysis. From the responses received, 20 responses were randomly selected for developing the coding scheme. Two of the authors of this paper have analysed the selected responses and coded the advantages and disadvantages independently. The identified codes were compared, and disagreements were discussed and resolved through discussion to arrive at the coding scheme.

After the coding scheme was finalized, a set of another 10 responses (i.e., three advantages and three disadvantages for each response) were randomly selected and independently coded for comparing effectiveness of the coding scheme. Inter-rater reliability was computed to verify the agreement between the two coders. The resulting value of Cohen's Kappa was 0.74, which indicates a satisfactory agreement between the coders.

The sets of responses used for development of coding scheme and for verifying inter-rater reliability were discarded, and the remaining 135 responses (see Table 1 for demographic details) were split between the two coders for coding the advantages and disadvantages using finalized the coding scheme.

	Female	Male	Total
Rotated Scrum Master Group (A)	15	57	72
Fixed Scrum Master Group (B)	13	50	63
Total	28	107	135

Table 1 – Demographic details

5 FINDINGS AND RESULTS

This section presents the findings from our qualitative study and highlights similarities and differences in advantages and disadvantages as perceived by team members from the two groups A and B. As expected, there are not many differences observed between the two groups of respondents. Table 2 depicts the most

frequently observed (top six) advantages and disadvantages – labeled with codes having prefix A and D respectively - across the groups that had their scrum master changed after every sprint (A) and the groups that had a fixed scrum master (B). The columns labeled n and % indicated the number of observations and the percentage among the respondent respectively.

Regarding the similarities in advantages, members from both groups perceived that rotating scrum master role among the team members a) provides equal opportunity for all members to develop as scrum masters, b) reduces the additional burden for the member who otherwise acts a scrum master for all sprints, and c) increases possibility for incorporating new ideas in handling scrum master responsibilities. The team members who had their scrum master rotated after each sprint stated advantages such as every member of their team realized responsibility, team gets a new scrum master when the current scrum master is not liked by the team, and team productivity/efficiency is likely to increase. On the contrary, the team members who worked with a fixed scrum master reported that the new scrum master may be more effective, possibly correct any mistakes by the previous scrum master, and the leadership skill is enhanced for all team members.

Regarding the similarities in disadvantages, interestingly members from both groups perceived that rotating scrum master role among the team members almost a similar set of problems. The disadvantages included the problem of getting an inexperience scrum master (not an issue once everyone gets the opportunity after sufficient sprints), and inefficiencies such as slowing down project development speed, possible disturbance in workflow, and increased opportunities for comparing different scrum masters. The team members who had their scrum master rotated after each sprint indicated that the team will have difficulty in adjusting to the new or different scrum master at the beginning of each sprint. A similar– though not identical - disadvantage is also indicated by the team members who worked with a fixed scrum master about the time required to learn the new scrum masters strengths and weaknesses.

Advantages after working with Rotated SM		**n**	**%**	**Advantages after working with Fixed SM**		**n**	**%**
A08	Equal opportunity to all for overall development	15	23.81	A08	Equal opportunity to all for overall development	23	31.94
A13	Burden is reduced for management	13	20.63	A12	New ideas get into project	21	29.17
A04	*Responsibility is realized by everyone*	13	20.63	A13	Burden is reduced for management	14	19.44
A12	New ideas get into project	12	19.05	A24	*Efficiency of new scrum master may be more*	10	13.89
A16	*Team gets new scrum master if they did not like previous one*	10	15.87	A03	*Correct previous master's mistakes*	10	13.89
A17	*Efficiency increases due to fresh ideas and enthusiasm*	10	15.87	A07	*Leadership skills is enhanced*	10	13.89
Disadvantages after working with Rotated SM		**n**	**%**	**Disadvantages after working with Fixed SM**			
D03	*Difficulty in adjusting to new scrum master*	19	30.16	D12	Leads to comparison of ideas and ways of work with previous SM	17	23.61
D10	Slows down project development speed	17	26.98	D02	Inexperienced person getting role	15	20.83
D02	Inexperienced person getting role	17	26.98	D06	Wastage of time in adapting to the new scrum master	12	16.67
D14	Disturbance in flow of working	16	25.40	D10	Slows down project development speed	11	15.28
D12	Leads to comparison of ideas and ways of work with previous SM	14	22.22	D14	Disturbance in flow of working	10	13.89
D06	Wastage of time in adapting to the new scrum master	14	22.22	D08	*Takes time to learn new scrum master's strength & weaknesses*	10	13.89

Table 2 - Comparison of perceived advantages and disadvantages of Scrum master role rotation

From the above, we can conclude that there is not much difference in perceptions – both with respect to advantages and disadvantages – between the two groups of respondents. There are, however, some common and specific perceptions of advantages and disadvantages. In general, we can summarize that by rotating the role of scrum master there are opportunities for improving the overall teamwork and for skill development, and at the same time it is likely that some inefficiencies in the teamwork are expected to result from the rotation.

6 CONCLUSION

This paper presents perceptions of members of several software development teams studied as part of a quasi-experiment. In this

experiment, half of the members were part of development teams that rotated the scrum master role among the team members, and half of the members worked were part of teams that had a fixed scrum master. Our preliminary results indicate that there is not much difference in perceptions regarding the pros and cons of rotation of scrum master role. We are planning to analyse the responses collected to further our investigation to understand any differences in perceptions as observed by specifically by product owners, scrum masters and developers.

This qualitative study has limitations with regard to analyzing responses collected from inexperienced students who are being introduced to scrum methodology for working on small projects spanning only three sprints. Despite these limitations, this study resulted in a set of specific pros and cons – from a large number of teams working on similar projects - that can be taken into account by scrum teams planning to rotate the role of scrum master. Thus,we obtain a more comprehensive view regarding scrum master role.

REFERENCES

[1] Dybå, Tore, and TorgeirDingsøyr. "Empirical studies of agile software development: A systematic review." *Information and software technology* 50.9 (2008): 833-859..

[2] Schwaber, Ken, and Mike Beedle. *Agile software development with Scrum*. Vol. 1. Upper Saddle River: Prentice Hall, 2002.

[3] Rising, Linda, and Norman S. Janoff. "The Scrum software development process for small teams." *IEEE software* 17.4 (2000): 26-32.

[4] Singh, Mona. "U-SCRUM: An agile methodology for promoting usability." *Agile, 2008. AGILE 08. Conference*. IEEE, 2008.

[5] Mahnic, Viljan. "A capstone course on agile software development using Scrum." *IEEE Transactions on Education* 55.1 (2012): 99-106.

[6] Malone, Michael W. *Process subversion in Agile Scrum software development: A phenomenological approach*. Diss. Capella University, 2014.

[7] Pinto, Jeffrey K. "Project management 2002." *Research-Technology Management* 45.2 (2002): 22-37.

[8] Boehm, Barry, and Richard Turner. "Management challenges to implementing agile processes in traditional development organizations." *IEEE software* 22.5 (2005): 30-39.

[9] Ceschi, Martina, et al. "Project management in plan-based and agile companies." *IEEE software* 22.3 (2005): 21-27.

[10] Paasivaara, Maria, Sandra Deresiewicz, and Casper Lassenius. "Using scrum in distributed agile development: A multiple case study." *Global Software Engineering, 2009. IGCSE 2009. Fourth IEEE International Conference on*. IEEE, 2009.

[11] Cervone, H. Frank. "Understanding agile project management methods using Scrum." *OCLC Systems & Services: International digital library perspectives* 27.1 (2011): 18-22.

[12] Schatz, Bob, and Ibrahim Abdelshafi. "Primavera gets agile: a successful transition to agile development." *IEEE software* 22.3 (2005): 36-42.

[13] Mundra, Ashish, Sanjay Misra, and Chitra A. Dhawale. "Practical scrum-scrum team: Way to produce successful and quality software." *Computational Science and Its Applications (ICCSA), 2013 13th International Conference on*. IEEE, 2013.

[14] Lindvall, Mikael, et al. "Empirical findings in agile methods." *Conference on Extreme Programming and Agile Methods*. Springer Berlin Heidelberg, 2002.

[15] Danait, Ajay. "Agile offshore techniques-a case study." *Agile Conference, 2005. Proceedings*. IEEE, 2005.

[16] Hammarin, Gabriella Uppsala University, Disciplinary Domain of Science and Technology, Mathematics and Computer Science, Department of Information Technology.

[17] "Why Scrum? | State of Scrum Report - Scrum Alliance." AccessedSeptember 24,2017. https://www.scrumalliance.org/why-scrum/state-of-scrum-report/2017-state-of-scrum.

Appendix A: List of Codes for Advantages and Disadvantages

Code – Description of Advantage

A01 - Responsibility of managing product backlog realized

A02 - Amalgamation of different ideas

A03 - Correct previous master's mistakes

A04 - Responsibility is realized by everyone

A05 - All do their best

A06 - New thinking brings innovative ideas

A07 - Leadership skills is enhanced

A08 - Equal opportunity to all for overall development

A09 - New roles are explored by everyone

A10 - Change in work leads to less monotony

A11 - All team members come to know each other

A12 - New ideas get into project

A13 - Burden is reduced for management

A14 - Realistic goals are given to coders

A15 -Previous scrum master can take charge if new one is not available

A16 -Every member gets new scrum master if they did not like previous one

A17 - Efficiency increases due to fresh ideas and enthusiasm

A18 - Skills of arranging and managing meetings developed

A19 - Equal opportunity is given to all

A20 - All work with full dedication and dependency gets reduced

A21 - Way of thinking is enhanced and developed

A22 - Each team member can induce their own ideas

A23 - Enjoy experience of both the developer and leader

A24 - Efficiency of new scrum master may be more

A25 - Extra focus on specific specialized area of SM

A26 - Equal work distribution is there

A27 - Proper understanding of project

A28 - Sprint completed in time efficient manner

A29 - Avoids cases of dominance and mismanagement

Code – Description of Disadvantage

D01 - Uncomfortable role allocation

D02 - Inexperienced person getting role

D03 - Difficulty in adjusting to new scrum master

D04 - Not so good scrum master for longer period

D05 - Less committed scrum master

D06 - Wastage of time in adapting to the new scrum master

D07 - Loss of faith in coming scrum masters

D08 -Takes time to learn new scrum master's strength & weaknesses

D09 - Unable to complete his part of development properly

D10 - Slows down project development speed

D11 - Takes time to decide about task allocation

D12 - Leads to comparison of ideas and ways of work with previous SM

D13 - Takes time to understand team quality

D14 - Disturbance in flow of working

D15 - Lack of coordination

D16 - Disturbance in communication between team members

D17 - Increases burden on team in changing already implemented features

D18 - Change in plans may or may not work

D19 - Can weaken SM and team bonding due to indifferent attitude

D20 - No individual with extensive knowledge about project

D21 - Not able to decide next scrum master

D22 - Overburden on an individual

D23 - Affects uniformity in development