Perspectives on the adherance to scrum rules in software project management

Leanah Alsaber, Ebtesam Al Elsheikh, Sarah Aljumah, Nor Shahida Mohd Jamail
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia

Article Info

Article history:

Received Apr 14, 2020
Revised Jun 7, 2020
Accepted Jul 20, 2020

Keywords:

Agile
Scrum
Scrum rules
Software engineering

ABSTRACT

Adapting users need to fulfill their requirements and delivering products to be on time within the planned cost, is critical matter that all software project managers (SPM) put the highest priority for it while considering the users satisfaction at the same time. Agile methodology is one of the solutions provided by software engineers (SE), to get the customers involved in the system development life cycle (SDLC) to avoid the risk nonconformance cost. Yet SPM's still facing the nonconformance costs and the dynamic changes, and the root cause of the issue is not pointed on to find a solution for it. This undertaking research aimed at determining whether software developers understand scrum rules. In addition, how does this knowledge gab affect the software projects success from the project management perspective. Furthermore, the engagement studied the impact of lack of enough knowledge on the topic to project delivery. The collected data from the qualitative and quantitative methods, which was conducted with scrum teams who worked in the health information system (HIS), Educational solutions, and Governmental solutions has showed deviations in organizational practices and team conflicting, competition, and pressure as well as declined product quality.

Corresponding Author:

Leanah Alsaber
College of Computer and Information Sciences
Riyadh, Prince Sultan University
66833 Rafha Street, Riyadh 11586 Saudi Arabia
Email: Leanah.Alsaber@gmail.com

1. INTRODUCTION

The idea of employing the SPM in software projects is meant to manage gaining a high-quality project [1]. SPM comprises of the steps taken to plan, implement, monitor, and deliver information technology related undertakings. However, these kinds of engagements tend to have higher levels of uncertainty and complexities as compared to traditional projects. This implies that project managers in the software industry must consistently adapt their approach throughout the exercise. As such, software project management plays an important role in facilitating successful execution of projects that have high risk of failure. Equally important is the SPM estimation of the effort of software projects is highly critical [2]. Not only but also focusing in not losing the control of it is not as easy as it sounds.

Agile methodology has presented the scrum model that it has more flexibility in accepting changes from the customer in the one of the SDLC phases. Scrum has its own clear defined roles to be followed to achieve it purpose and to have the software projects delivered on time and within budget [3]. Each member in the scrum team has responsibilities and roles to comply with; in spite of the listed roles, it has been observed that many software projects that was implemented under the scrum were delivered behind schedule and above budget. The research seeks to study the perspectives on the adherence to scrum rules in SPM in the software development industry perspectives, the appropriate use of scrum in implementing agile principles in projects, and wither it is followed correctly or no. In addition, it could be said that to avoid the software project failure, it is better to consider evaluating and estimating the software reliability at the beginning of any SDLC for a software [4].

Scrum emerges as the most widely used agile model in software development projects. Its aim to develop better softwares in timely manner within less budget [5]. Moreover, most companies vary the scrum methodology to some extent [6, 7]. The research will focus on proving the following hypothesis:
H1: Most developers who use scrum claim they know the rules but they do not.
H2: The lack of awareness of scrum rules leads to software project management issues such as delivery over budget and behind schedule as well as dissatisfaction within the team.

The document includes a literature review section presenting the state of literature on agile development techniques. To answer the research question, a study conducted observations, interviews, and administered an industry-wide questionnaire

2. LITERATURE REVIEW

In this section discussed 4 types of approach in software development process. They are Agile approach, Extreme Programming (XP) approach, Kanban and Scrum.

2.1. Agile

The agile approach to project execution relies on an iterative process through which requirements and solutions progress through concerted engagement between stakeholders. The methodology achieves success by facilitating autonomous and cross-functional teams. Moreover, it builds on the importance of change and continued fragmentation of projects for enhanced performance, and quality. As part of the 12 principles, the approach calls for simplicity, continuous collaboration, frequent delivery of working software and having a highly motivated team. Project managers can use different methods of implementing agile ideologies. The subsequent discussion will examine extreme programming (XP), Kanban, and scrum [8, 9].

2.2. Extreme Programming (XP)

The extreme programming approach to agile principles calls for the use of user stories drawn from customer requirements. Here, during project execution developers compose user stories on cards as a means of defining the system’s functional requirements. The method also supports practices such as test-driven development, which requires developers to pre-conceive ideas about a feature before actual coding. TDD also proposes small iterations that incrementally develop software while eliminating the need for detailed design at the start of the project [10, 11]. In doing so, the framework enhances flexibility and reiterates the importance of self-organizing teams that work at a sustainable pace [12].

2.3. Kanban

Kanban avails an easy methodology for enhancing team collaboration leading to the delivery of high-quality information technology projects to clients. Constant improvements to products and supporting infrastructure emerge as the focus of the approach. In terms of workflow, Kanban also proposes breaking down of complex team activities into smaller steps that ease validation. Kanban focuses on what should be done and when it should be done by prioritization of tasks, delivery timeline, and workflow [13]. Kanban eliminates waste of time, developers do not write more than they can test, they do not write specification more than they can code [14, 15]. The philosophy also relies extensively on the visualization of workflow as a tool for quick identification and resolution of issues. Kanban also limits the amount of work in progress to match the pace of the slowest step. This works to enhance teamwork while reducing the overall impact of changes [16].

2.4. Scrum

Lastly, scrum also provides a framework for effective information technology project execution relying on the time box philosophy. Here, as shown in Figure 1, the approach proposes timed iterations or sprints that dictate the features that a project team will complete. At the start of a project, the team creates a product backlog that lists all the required features in order of importance. During each sprint, the team selects features to complete from this list during a planning meeting before beginning on the work. The sprint backlog cannot change during execution. Core roles for scrum include the development team, stakeholders, scrum master, managers, and product owner [17-19]. Team members also meet daily in short 15-minute scrums where they update one another on current progress [20]. During the scrum, questions like what you have done yesterday, what are you doing today, and is there any blocks [21]. After finishing the sprint, the team presents the product and seeks feedback for improvement in the next cycle [22, 23].

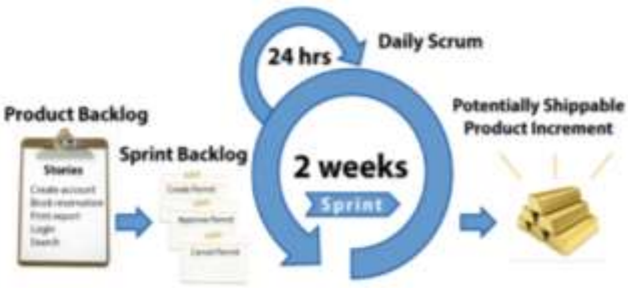

Figure 1. Graphical representation of the scrum framework

3. RESEARCH METHODOLOGY

As the customer opinion of a product is important to have in order to lead for their satisfaction [24]. This study employed mixed methods research to collect quantitative and qualitative data on the challenges arising from poor understanding of the basic scrum rules among scrum musters, product owners, and developers. The research also a deductive adopted because the researcher established the hypothesis beforehand. Firstly, starting with the observation methodology as the fundamental method for elicitation to collect data by observing the scrum team while working on the projects, and how the process is affected by their behavior. Secondly, after the collected data from the previous method, an interview is conducted with scrum teams who where involved in the observed working process. Lastly, an online questionnaire was distributed among all stakeholders who are directly or indirectly affected by scrum projects.

3.1. Observation

The researcher observed teams working at two established organizations working in the IT field. One of the organizations conducted projects for HIS solutions whereas the other provided solutions to the government. The observation happened for a period of two months with the researcher focusing on the work process, team engagement and level of communication at each establishment. Here, the study examined how the sprints worked throughout the life cycle. Through observation, the researcher also collected data on the organization-specific factors that affect the scrum process. Due to access, observation accords to research phenomena, the data collected is reliable. The names of the two companies will remain anonymous.

3.2. Interview

The conducted interviews with the team members involved in the project in which this process is focusing on pulling out the information and the team opinion of how they actually work on scrum [25]. The role for the team members was between developers, quality assurance engineers, project managers, team leaders, and more who could be part of the team who works in scrum. Moreover, the interview extended to reached scrum teams who do work in educational solutions; to see wither they have the same issues as the HIS and governmental solutions team who had the observation being conducted on. At this point, the study had already established some findings from the preceding stage. Based on these outcomes, the researcher used interviews to determine personal opinions on the process followed as well as prevailing issues. This method allowed the collection of a wide range of information from diverse perspectives that enhanced the quality of the observations.

3.3. Questionnaire

The study administered questionnaire that collected data from numerous companies - not just who placed under observation. The actual data collection tool comprised of 31 closed ended questions, which has been placed in Survey Monkey website and was distributed using social media platforms as LinkedIn, Twitter, and WhatsApp. To eliminate bias, the researcher carefully ordered the questions in a manner that hid its true intention from the participants. This hid the important questions leading to increased honesty in the responses received from the participants.

4. RESULTS AND DISCUSSION

The findings from the questionnaire provided excellent insight into the extent to which organizations follow scrum rules during projects. The findings reveal a variation in approach among organization based on the workflow processes reported. For instance, 33.33% of respondents indicated that planned sprints usually take 1 to 2 weeks to complete. In contrast, 66.67% of workplaces used 3 weeks or more. On the other hand, from the 66.67 there is 16.67% planned their sprint to be more that 6 weeks. This is considered as a huge release for one sprint that could be divided into two sprints to avoid big changes from the product owners, unless if they actually keep implementing the changes while the sprint is active. This implies there is a misunderstanding of changing rules for the sprint backlog as shown in Figure 2.

Moreover, 43.33% of participants work on one sprint at a time whereas 56.66% work on two or more at the same time. 30% of those interviewed also reported not checking user stories from previous sprints while working on subsequent sprints. This is referring to abusing the changing process without checking its effectiveness on the current active work. 73.33% of workplaces do accept changes when the sprint has started as shown in Figure 3. Here, 43.33% indicated that changes happen more than 3 times during each iteration with 50% of organizations implementing the change during the working sprint.

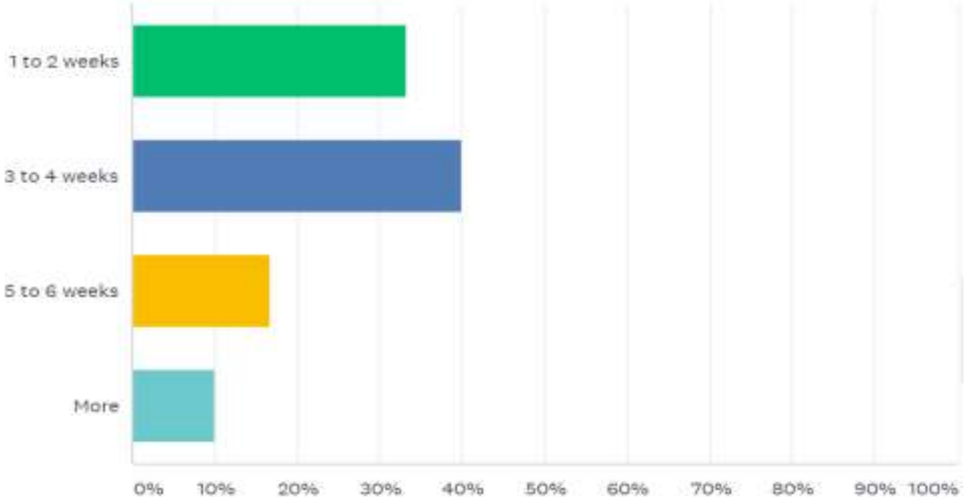

Figure 2. Distribution of responses on how many weeks sprints take

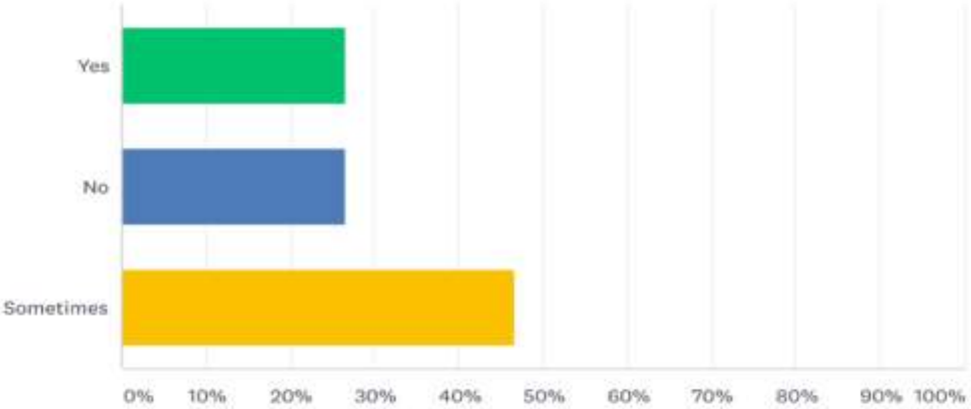

Figure 3. Frequency of change acceptance after the sprint has started

In terms of change control, 50% of firms did not have a board tasked with evaluating and approving proposed variations. This could explain the high number of product owners (83.33%) making changes to the current sprints. Furthermore, this happened by product owners without direction of wither these changes are critical to have or it should be postponed for later sprints. Moreover, 73.33% of firms also change the priority of requirements when the sprint starts. On the other side, 46.67% do not write a user story for each task as required under the scrum framework. Of the participants surveyed, 73.34% reported not differentiating between the product backlog and the sprint backlog in the project as shown in Figure 4. Again, this implies how scrum team does not follow scrum roles in order to lead software project toward success.

Another result, 46.6% of teams do not have daily meeting to discuss project progress as required under the framework. Here, teams reported varying meeting frequencies for discussing project progress. A significant proportion actually (26.67%) reported not knowing whether scrum teams should have daily meetings. To further bolster this point, 73.33% asserted they did not know whether they follow scrum rules as shown in Figure 5.

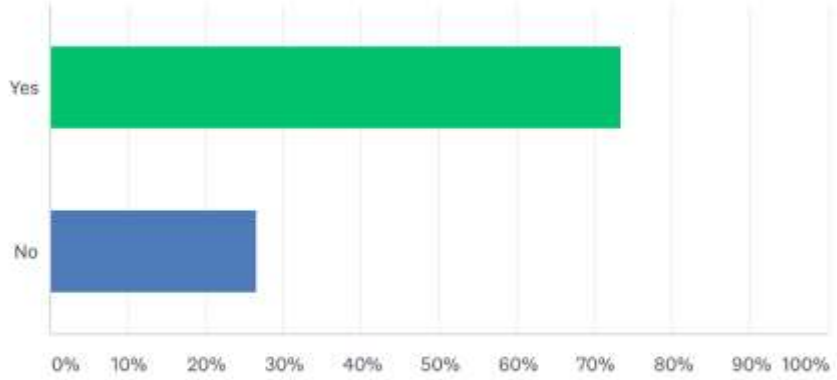

Figure 4. Frequency of responses on whether respondents differentiate the product and sprint backlog

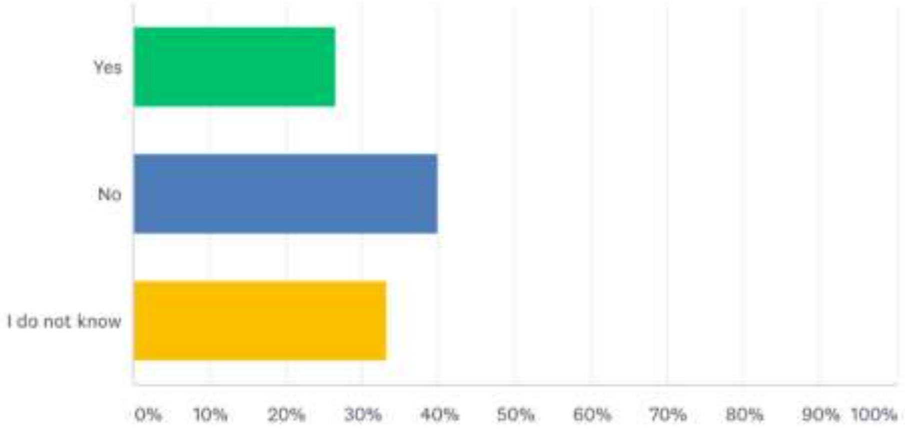

Figure 5. Response frequency on whether participants follow scrum rule

A further 60% indicated they did not have enough knowledge of scrum methodology to successfully apply to projects. Which is a critical matter to work in a process that a member do not understand what exactly should be done. Even besides not knowing, the cause of failure could be because the wrong followed process. Looking a team cohesion, a highest proportion of participants (55.17%) indicated they work in harmony with other team members. 46.67% and 60% also alluded to having clearly defined roles as well as sprint goals. 34.49% reported that the work environment at the organization does not support scrum.

Lastly, in relation to issues, 48.28% indicated that scrum created a high-pressure work environment. Other problems as shown in Figure 6 identified include:

a) Extra pressure – 62.07%
b) Conflict with the team – 31.03%
c) Competition with the team – 20.69%
d) Different results from product owners' expectations – 34.48%

Where scrum manifesto goes toward a happy ordered collaborative team.

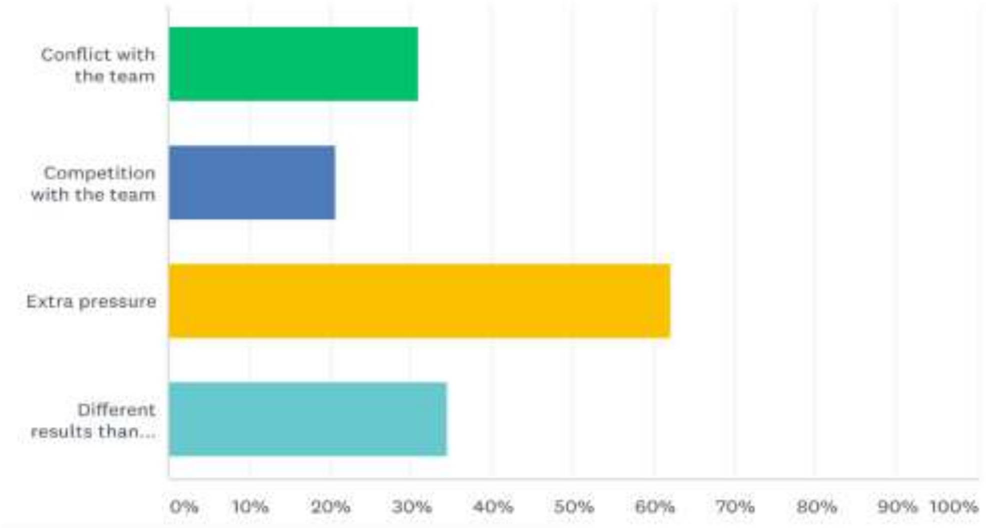

Figure 6. Response on issues faced in the scrum environment

The findings from the questionnaire answered the hypothesis posed in the introduction. First, the study had postulated that most scrum members who use scrum claim they know the rules but they do not. The results showed that only a minority of members claim to know scrum rules. Here, 26.67% reported knowing the rules as compared to 73.33% who indicated that they did not know. This lack of knowledge emerges in the lack of following standard procedure. For instance, the surveyed firms vary from best practices by:

a) Holding sprints that go beyond the 4-week recommendation
b) Working on more than one sprint at a time
c) Not checking previous sprints when working on current user stories
d) Accepting changes when a sprint has started
e) Applying changes during a sprint
f) Lack of a change control board during project execution
g) Changing the priority of requirements after the start of a sprint
h) Not differencing between the product backlog and print backlog
i) Incomplete progress reports
j) Lack of daily team meetings as recommended

In addition, the findings supported the second hypothesis stating that the lack of awareness of scrum rules leads to software project management issues such as delivery over budget and behind schedule as well as dissatisfaction within the team. Here, the study established that poor implementation led to conflict, competition, and pressure, continuous changing in requirements without any management, as well as unfulfilled product owner expectations. From a project management perspective these issues emerge due to poor communication among team members, poor leadership, changes in scope, and unplanned expenses due to poor change control. For instance, by not following scrum rules on weekly team meetings projects run the risk of communication breakdown. In turn, this could lead to mismatched team goals and objectives. Moreover, accepting changes after the start of a sprint introduces scope creep that could potentially increase the overall cost and duration of the project

5. CONCLUSION

In conclusion, the agile approach to agile project execution provides an effective approach for project managers in the information technology sector to manage uncertainties. This study of the perspectives on the adherance to scrum Rules in SPM successfully examined software development industry perspectives on the appropriate use of scrum in applying agile principles in projects. In doing so, the collected data from the conducted methodologies as the observation, interviwe, and qestionnair revealed that a significant proportion of developers did not know scrum rules. Moreover, it showes that the organizations implemented various forms of the approach based on the diversity of the data collected. Some of the problems resulting from this predisposition include team conflict, competition, and pressure as well as declined product quality. Future research could address the reasons for the observed variations in organization practices. Eventullay, the presneted hypotheires have been proven by scrum team members and SPM from the collected data.

ACKNOWLEDGEMENTS

This work was partially supported by the Artificial Intelligence Data Analytics Lab (AIDA), Prince Sultan University, Riyadh, Saudi Arabia.

REFERENCES

[1] F. Fachrurrazi, "The End User Requirement for Project Management Software Accuracy", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 2, p. 1112, 2018. Available: 10.11591/ijece.v8i2. pp. 1112-1121.

[2] ZAKRANI, M. HAIN and A. IDRI, "Improving Software Development effort estimating using Support Vector Regression and Feature Selection", *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 4, p. 399, 2019. Available: 10.11591/ijai.v8.i4.pp399-410.

[3] T. Budiman and J. Suroso, "Optimizing IT Infrastructure by Virtualization Approach", *IOP Conference Series: Materials Science and Engineering*, vol. 190, p. 012012, 2017. Available: 10.1088/1757-899x/190/1/012012.D. Hema Latha, & P. Premchand.

[4] "Estimating Software Reliability Using Ant Colony Optimization Technique with Salesman Problem for Software Process", *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 7, no. 2, pp. 20-29, 2018. Available: 10.30534/ijatcse/2018/04722018.

[5] R.Nishijima and J. Santos, "The Challenge of Implemenitng scrum agile methodology in a traditional development enviroment", *International Journal of Computers & Technology*, Vol.5, no. 2, ISSN 2277-3061, 2013.

[6] M. Luiz Monteiro Marinho, S. Cândido de Barros Sampaio, T. Lúcia de Andrade Lima and H. Perrelli de Moura, "Uncertainty Management in Software Projects", *Journal of Software*, vol. 10, no. 3, pp. 288-303, 2015. Available: 10.17706/jsw.10.3.288-303.

[7] J. Eckstein and H. Baumeister, "Extreme Programming and Agile Processes in Software Engineering", Berlin: Springer, 2005.

[8] P. Vanzant Stern, "Lean And Agile Project Management". [S.l.]: *CRC PRESS*, 2020.

[9] R.Kumar, P.Maheshwary, T.Malche, "Inside Agile Family: Software Development Methodologies", *International Journal of Computer Sciences and Engineering*, vol. 7, no. 6, June 2019, E-ISSN: 2347-2693, 2019, Available: doi.org/10.26438/ijcse/v7i6.650660.

[10] M. Lacey, "Scrum Field Guide", [Place of publication not identified]: *Addison-Wesley*, 2015.

[11] Ozkan, Necmettin, and Civan Kucuk. "A systematic approach to project related concepts of scrum." *Revista de Management Comparat International*, vol. 17, no. 4, pp. 320, 2016.

[12] J. Garbajosa, X. Wang, and A. Aguiar, "Agile processes in software engineering and extreme programming" in *19th International Conference, XP 2018*, Porto, Portugal, May 21-25, 2018, Proceedings. Cham: Springer International, 2018.

[13] Kirovska, Nevenka, and Saso Koceski. "Usage of Kanban methodology at software development teams." *Journal of Applied Economics and Business*, vol. 3, no. 3, pp. 25-34, 2015.

[14] H. Lei, F. Ganjeizadeh, P. K. Jayachandran, and P. Ozcan, "A statistical analysis of the effects of Scrum and Kanban on software development projects," *Robot. Comput. Integr. Manuf.*, 2017, pp. 59-67 doi: 10.1016/j.rcim.2015.12.001.

[15] K. Schwaber and J. Sutherland, "The Kanban Guide for Scrum Team", Scrum.org, pp. 3-6, 2019, Available: Scrum.org.

[16] E. Brechner, "Agile project management with Kanban. Redmond", Wash. D.c.: Microsoft, 2015.

[17] Mohammad, A. Khaled, A. Nidal, and T. Ahmed, "A Comparative Study of Agile Methods: XP versus SCRUM," *Int. J. Comput. Sci. Softw. Eng.*, vol. 4, no. 5, pp. 126-129, 2015.

[18] Y. Khmelevsky, X. Li, and S. Madnick, "Software development using agile and scrum in distributed teams," in *11th Annual IEEE International Systems Conference, SysCon 2017 - Proceedings*, pp. 1-4, 2017, doi: 10.1109/SYSCON.2017.7934766.

[19] Noll, M. A. Razzak, J. M. Bass, and S. Beecham, "A study of the scrum master's role," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 307-323, 2017, doi: 10.1007/978-3-319-69926-4_22.

[20] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, "Analysis of software development methodologies," *Int. J. Comput. Digit. Syst.*, pp. 449, 2019, doi: 10.12785/ijcds/080502.

[21] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press, 2004.

[22] E. Mendez and D. Pinto-Roa, "Optimization model for assigning tasks in Scrum Agile Development (work in progress)", *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, vol. 7, no. 1, p. 1, 2018.

[23] Craddock, B. Roberts, J. Stapleton, and J. Godwin, "Agile Project Management". Ashford: Agile Business Consortium Limited, 2017.

[24] O. Oueslati, A. I. S. Khalil, and H. Ounelli, "Sentiment analysis for helpful reviews prediction," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 7, no. 3, pp. 34-40, 2018, doi: 10.30534/ijatcse/2018/02732018.

[25] P. Thakur, R. Shrivastava. "A Review on Text Based Emotion Recognition System", *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 7, 2018, pp. 67-71, doi: 10.30534/ijatcse/2018/01752018.