The Proposed Critical Factors to Improve the Performance of Scrum Teams

Abdulaziz A Alzubaidi¹; Reda M Salama²; M. Rizwan Jameel Qureshi³

¹Department of Information Technology & King Abdul-Aziz University, Saudi Arabia

²Department of Information Technology & King Abdul-Aziz University, Saudi Arabia

³Department of Information Technology & King Abdul-Aziz University, Saudi Arabia

¹ aalzubaidi0176@stu.kau.edu.sa; ² rkhalifa@kau.edu.sa; ³ rmuhammd@kau.edu.sa

DOI: doi.org/10.47760/ijcsmc.2022.v11i02.008

Abstract— Scrum is one of the most popular and widely practiced agile software development methodologies. Scrum team consists of a Scrum Master, Product Owner, and developers. While using Scrum, several factors have been reported in the existing studies to improve or lower the performance of a Scrum team such as managing the priority of the product backlog items, role of an efficient product owner, cost and time to perform integration tests, accurate estimation of story points and code quality. This paper aims to determine critical factors those impact the most over the performance of the Scrum teams using a survey and systematic literature review. Furtherly, a model is proposed presenting the impact of critical factors to enhance the performance of Scrum teams to increase efficiency and work perfection, organizing and managing product backlogs, clarity in goals, execution of sprints in an estimated duration and increase productivity of product owner. The identified critical factors are validated using a survey and the results are found encouraging.

Keywords— Scrum teams, Improvement Factors, Product Backlog, PBI, Agile methodologies

I. INTRODUCTION

Scrum is a simplified approach that helps people, teams, and institutions. The objective is to increase their values by allowing them to adapt to changing circumstances. Scrum is implemented by Scrum Team that is composed of Scrum Master, Product Owner, and developers. In a Scrum Team, there is no sub-teams or divisions. It is a team of experienced members who collaborate to accomplish a clear purpose as a product in a stipulated time period. Each Scrum event is composed of necessary details to plan, design, develop and test. This information will help to manage the team progress in terms of product backlog management, Sprint planning and integration test management. However, there are factors that cause success or failure of these events, which affects the performance of Scrum method. This paper will determine the critical factors effecting the performance of Scrum teams. This research aims to identify critical factors effecting the performance of Scrum teams and propose a model to indicate the critical factors. The proposed model is based on systematic literature review and questionnaire based evaluations. The contributions of this paper are as follows:

- Create a model that displays the critical factors effecting the performance of the Scrum teams.
- Finding the impact of critical factors over the performance of Scrum teams.

This paper has arranged as follows: Section 2 provides the related work, and section 3 covers problem definition. The proposed solution is introduced and discussed in section 4, and the validation operation is addressed in section 5.

II. RELATED WORK

Canedo et al. [1] discussed the most used metrics and methodologies in agile software development, such as effort estimates, deadlines, and costs. It can be deduced that estimation metrics must constantly be tailored to the project context, as each project will have unique factors that influence the estimates' outcome. According to the findings, the most common method for agile teams in the planning process is Planning Poker. However, the most used metrics in agile development for evaluating the size, time, effort, productivity, and cost are Story Point and Point of Function. It is, nonetheless, required to perform.

From the standpoint of agile methodologies teams, Tanveer et al. [2] examined the understanding of the estimating process in terms of its correctness in the context of agile development. By utilizing case studies as a source of information. Three agile development teams were studied using two monitoring processes and eleven meetings. Estimating and accuracy are influenced by factors including the developer's skills, experience, and the complexity and impact of modifications on the underlying system. A dynamic prototype tool for visualizing data is being developed to promote transparency in the estimating method. However, this instrument should be regarded as a recommendation system rather than a decision-making system. It will allow for more excellent debate at estimating sessions, which will result in more precise estimates [2].

Two hybrid strategies are discussed for estimating effort in agile development (Functional points and Lines of Code) [3]. An architectural framework that transforms story points to function points has been proposed to aid in the reduction of issues in agile estimations and improve absoluteness from abstraction. Furthermore, HEETAD may be concluded to be a feasible technique for estimating Agile project effort. Diego et al. [4] analysed data from seventy-three new papers to find the most effective estimation models for project planning. They were able to identify six agile approaches ("Scrum, XP, TDD, Agile Unified Process, Kanban, and Distributed Agile Software Development") based on this data, each of which used expert-based estimates to estimate the effort. According to the findings, Planning Poker is still the most widely utilized estimating method in primary research. It's linked to the most common unit of measurement story points and how software components are distributed in agile approaches (user story). Furthermore, the paper's main flaw is that it fails to account for potential biases introduced during selection.

Usman et al. [5] studied the state-of-the-art empirical estimations that are used in agile software development. Expert judgment-based strategies, planning poker, and the use case points methodology were among the techniques examined in 25 primary research. According to the findings, these strategies are the most employed in agile effort estimating. However, most of the methods have failed to produce accurate forecast results. It was shown that there is a lack of consensus on appropriate cost drivers for ASD initiatives.

Afzal and Adnan [6] focused on estimating software work and managing knowledge when using the Scrum method, which are complex tasks in an agile environment. Using an ontology model in a multi-agent estimating system, proposed improving software effort estimation by concentrating on Scrum process and practices. Their experiment used MMRE and PRED (25) evaluation metrics on the collected estimated and actual efforts for 12 online projects. The results reveal that this method has a high level of prediction accuracy. Other types of agile approaches, on the other hand, must be included to anticipate the required effort for future projects accurately.

Rankovic et al. [7] used Taguchi's Orthogonal Arrays to reduce the relative magnitude error (MRE) in agile development estimation. Artificial Neural Networks (ANN) in two different topologies were offered as an efficient method for predicting and estimating software effort. When analyzing software effort, the two architectures performed better than traditional methods in terms of reaching the least MMRE. Furthermore. When comparing the ANN27 and ANN18 structures, the ANN27 structure produces better results, meaning a lower relative error value, and "therefore, they can be regarded as the best-proposed technique of nonparametric models for software effort estimation."

De Carvalho et al. [8] utilized the project management technique in SE to assist project teams in keeping projects underhand. One of the most important procedures in SW is project management, which provides for an accurate and realistic estimation of the needed effort to finish a project. The Extreme Learning Machine (ELM) was utilized to estimate system development efforts compared to earlier effort estimation algorithms. The ELM model generated excellent outcomes compared to other techniques for forecasting software design efforts in the literature. However, an optimization method known as Particle Swarm Optimization (PSO) must be employed to optimize the model's parameters to estimate the software development effort.

Fadhil et al. [9] defined a method for estimating Effort in terms of costs, which is required when building software products. The cost estimation models were optimized using the dolphin swarm method and the hybrid bat algorithm DolBat. The results show that the dolphin method outperforms earlier algorithms. However, the DolBat algorithm best calculates the COCOMO II model's coefficient value. Furthermore, the dolphin technique

can be hybridized with other algorithms, or a new approach can be used to assess the efforts of software programs.

Kula et al. [10] conducted a mixed-methods study for " Factors Affecting On-Time Delivery in Large-Scale Agile Software Development." [10] carried out a two methods investigation. A wide range of factors, including specifications refinement, effective coordination, and organizational factors, affect same-time delivery. Furthermore, the findings suggest that many environmental factors may significantly affect same-time deliveries than previously thought. The scaling of agile approaches, cross dependencies, security issues, the significance of corporate culture, team continuity, and risk monitoring, on the other hand, all want greater attention. The software delivery in agile environments becomes more predictable.

Silhavy et al. [11] proposed a new approach for estimating software development effort (Categorical Variable Segmentation Model). Relative Size is utilized as a segmentation parameter in this approach, based on data-set segmentation. This method uses a specific model that has been trained on a particular segment of data to predict the software development effort. The findings show that the new CVS model outperforms the baseline techniques in terms of accuracy. On the other hand, Linear Clustering was supposed to work well with SR Model Estimation, but it doesn't.

Khan et al. [12] investigated the factors that influence the cost estimation in global software development (GSD). The cost estimation methods used by GSD software companies were discussed and published. Most models are legal and scientific, and they rely on COCOMO or other scientific models to function. The most popular model was "analogy-based and expert judgment." Furthermore, the findings of this study can be used to create a model in terms of cost components observed.

Vera et al. [13] researched software development effort estimation (SDEE) to forecast software development efforts to lower project hazard. The primary factors utilized to characterize SDEE approaches are also discussed. The findings revealed no commonly acknowledged taxonomy for SDEE procedures since there are too many relevant and almost independent criteria to classify them. Furthermore, the hierarchical structure employed to display the taxonomies isn't particularly helpful in identifying clusters of techniques pertinent to a software company.

El Bajta et al. [14] conducted a systematic mapping study (SMS) to summarize product cost estimation inside the area of GSD study by answering nine mapping issues. Nine parameters were used to select and classify 16 papers for GSD projects: publishing site, publication date, research kind, research approach, contributions type, software cost estimate methodologies, cost estimation actions, cost objects, and cost estimation effectiveness. The findings show that in the latest years, there's been an increase in interest in assessing software costs for GSD initiatives, with discussions being the most frequently targeted publications. Models are the most prevalent earnings type for GSD study in software cost assessment, with software development cost being the most common activity. It's also challenging to develop practical answers regarding software cost estimation for GSD.

An Analogy-Based Estimate Model (ABE) was discussed by Shah et al, one of the promising estimation methods for estimating software development effort. An artificial Bee Colony (ABC) guided Analogy-Based Estimating (BABE) model is suggested in this paper, which blends ABC with ABE for reliable estimation. The BABE model, which integrates ABC with ABE for feature weight optimization and accurate effort assessment by comparing the targeted project to previous projects, was developed in this work. The suggested approach operates in a two-stage setting that includes testing and training. However, missing data imputation techniques should be combined with the model used in this work to improve further [15].

III. PROBLEM STATEMENT

Each Scrum event has necessary details to plan, design, develop and test product. The main philosophy is to manage the team progress as well such as product backlog management, sprint planning and integration tests. However, there are factors that cause success or failure of these events, which effect the performance of Scrum teams. As per the [16], "There are various technical and general concerns with the agile scrum approach. Assigning, sorting, and merging product backlog items are all technical difficulties." According to Usman et al. [5], "Changes throughout or at the end of any sprint may alter the total project schedules, project scope, and finished product.". Raza and Majeed [17] identified several issues related to performance of Scrum teams such as quality items pileup, module integration issues, code quality, disruption in teamwork, mature vs. immature scrum, sprint duration, lack of scrum training, release process, backlog management, no technical practices, multiple teams, metrics, risk management, documentation, too idealistic scrum and communication/scrums ceremonies. Therefore, a research question is narrated based on the papers of [5,16,17].

- What are the most critical and influential factors that can increase or decrease the performance of Scrum teams?

IV. THE PROPOSED SOLUTION

Fig. 1 presents the proposed factors to improve the performance of the scrum teams. The proposed factors help to increase efficiency and work perfection, organizing and managing product backlogs, clarity in goals, execution of sprints in an estimated duration, increasing product owner productivity, and reducing project risks. Each Scrum event has the necessary details to plan, design, develop and test. Product Backlog is a priority list of all items necessary to a single product in the scrum method like bugs, client enhancements, effective product activity, specific competitive features, and system upgrades are among the lists. The first challenge for a scrum team is to convert user stories into product backlog (PB). PB items from the system requirements specifications (SRS) is extremely challenging to manage for a scrum team. PB item are assigned to team members based on story points or program functionality. The project may lead to failure or delay in the delivery of increments if PB items are not organized correctly [16].

The scrum team will discuss the most important product backlog items with a product owner to include it in the current sprint. Integration testing is a systematic procedure for constructing the software architecture while at the same time conducting tests to uncover errors associated with interfacing. Scrum team will deliver the increment to a client after the completion of the development of PB items. More than 80% of the expense of testing is utilized just for relapse testing [16]. Shorter Sprints can be employed to generate more integration tests and limit risk of cost and effort to a smaller time frame [18]. A product owner function is one of the most crucial and challenging. He is responsible for the success, finance, goals and objectives of a project. A product owner makes sure the product backlog is clear, visible, and understandable and he develops an explicit product goal [18].

Fig. 1 The Proposed Factors to Improve the Performance of Scrum Teams

In Scrum, sprints are made up of working units that should be completed within a predetermined time frame to meet a requirement outlined in the backlog. During the cycle, no modifications (such as new backlog items) are performed. As a result, a sprint gives participants the chance to work quickly and sustainable pace. A key aspect of Scrum's success is to implement sprints within the predicted time frame [18]. The sprints must be fit into a predefined time-box that is typically 30 days and there is no change considered in the sprint backlog while implementing a sprint. Software quality is the outcome of sound project management and software engineering [18]. Because of the proposed factors, the scrum teams will be able to meet the deadlines accomplishing the agile projects successfully.

V. VALIDATION

The proposed solution has been validated using a survey approach in this paper. This validation aims to identify the factors that influence the scrum method's performance as well as the efficacy of possible solutions. Google Forms is used to display the results of the conducted survey. The sampled respondents are project leaders, directors, IT developers, IT managers, and programmers. We frequently utilize Google Forms for questionnaires, and we disseminate the results via various social media channels such as WhatsApp, Telegram, and Twitter. The participants responded to the Google survey using a five-numbers. 1 indicates strong disagreement, 2 indicates disagreement, 3 indicates neutrality, 4 defines agreement, and 5 indicates strong agreement. The survey consists of 20 questions broken into three goals.

- Goal 1 - Determine critical factors impacting the performance of Scrum teams.
- Goal 2 - Measure the impact of identified factors over the performance of Scrum teams.
- Goal 3 - Analyse the difference over the performance of Scrum teams before and after implementation of critical factors.

Fig. 2 illustrated the findings of Goal 1. Goal 1 was about determine factors impacting the performance of Scrum teams. According to fig. 2, 66% of the responses were in favour of Goal 1 of which 33% strongly agreed and 33% agreed to it. 11% of the respondents were not in favour of Goal 1. 22% of the respondents remained neutral.

Fig. 2 The cumulative analysis of goal 1

Cumulative responses of the Goal 2 (Measure the impact of identified factors on the performance of Scrum teams) are shown in fig. 3. Fig. 3 showed that 77% of the responses were in favour of Goal 2, of which 33% agreed while 44% strongly agreed to the effects of the identified factors over the performance of Scrum teams. 11% of the cumulative responses were neutral while 11% of the participants were not in favour of Goal 2.

Fig. 3 The cumulative analysis of goal 2

Cumulative responses of the Goal 3 were shown in fig. 4. Fig. 4 showed that 44% of the responses agreed with goal 3, of which 22% agreed while 22% strongly agreed that there is a significant difference over the performance of Scrum teams after applying critical factors. 33% of the responses remained neutral for Goal 3. 22% of the responses were not in favour of the Goal 3.

Fig. 4 The cumulative analysis of goal 3

Cumulative responses to the three goals were shown in Table 1. Table 1 showed that 63% of the responses agreed with 3 goals in which 33% of the software engineers strongly agreed and 30% of the professionals agreed to it. 22% of the responses remained neutral for 3 goals. A total of 14.7% responses disagreed with the three Goals as shown in fig. 5.

TABLE I
FINAL CUMULATIVE ANALYSIS OF THREE GOALS

Goal No.	Very Low	Low	Nominal	High	Very High
Goal 1	0	11	22	33	33
Goal 2	0	11	11	33	44
Goal 3	0	22	33	22	22
Total	0	44	66	88	99
Avg.	0%	14%	22%	30%	33%

Fig. 5 Final cumulative analysis of 3 Goals

VI. CONCLUSION

This research is conducted to identify critical factors effecting the performance of the Scrum teams using systematic literature review of existing studies and feedback of the Scrum professionals. The product backlog priority management, reduced cost of integration tests, accurate estimated duration of a sprint, and role of an effective product owner are identified as the most critical and influential factors to achieve high performance of a Scrum teams. Furtherly in this study, another survey is conducted to measure the impact of identified critical factors over the performance of Scrum teams. The results provide evidence that the proposed identified factors significantly improve the performance of agile scrum teams. The main limitation of this study is the limited sample size to generalize the results.

REFERENCES

- [1] E. Dias, D. P. Aranha, M. O. Cardoso, R. P. D. Costa, and L. L. Leite, "Methods for Estimating Agile Software Projects: A Systematic Review," in *Proc. Software Engineering and Knowledge Engineering*, 2018, p. 34-49.
- [2] T. Binish, L. Guzmán, and U. M. Engel, "Understanding and Improving Effort Estimation in Agile Software Development- an Industrial Case Study," in *Proc. Software and System Processes (ICSSP)*, 2016, p. 41-50.
- [3] N. A. Bhaskaran, and V. Jayaraj, "A Hybrid Effort Estimation Technique for Agile Software Development (HEETAD)," *Engineering and Advanced Technology*, vol. 9, pp. 1079-1087, Oct. 2019.
- [4] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão and E. Insfran, "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review," *IEEE Access*, vol. 8, pp. 166768-166800., 2020.
- [5] U. Muhammad, E. Mendes, F. Weidt, and R. Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review," in *Proc. Predictive Models in Software Engineering*, 2014, p. 82-91.
- [6] A. Muhammad, and M. Afzal, "Ontology Based Multiagent Effort Estimation System for Scrum Agile Method," *IEEE Access*, vol. 5, pp. 25993-26005, 2017.
- [7] R. Nevena, D. Rankovic, M. Ivanovic, and L. Lazic, "A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays," *IEEE Access*, vol. 9, pp. 26926-26936, 2021.
- [8] H. D. P. De Carvalho, R. Fagundes and W. Santos, "Extreme Learning Machine Applied to Software Development Effort Estimation," *IEEE Access*, vol. 9, pp. 92676-92687, 2021.
- [9] A. A. Fadhil, R. G. H. Alsarraj and A. M. Altaie, "Software Cost Estimation Based on Dolphin Algorithm," *IEEE Access*, vol. 8, pp. 75279-75287, 2020.
- [10] E. Kula, E. Greuter, A. Van Deursen and G. Georgios, "Factors Affecting On-Time Delivery in Large-Scale Agile Software Development," *IEEE Transactions on Software Engineering*, pp. 1-20, 2021.
- [11] P. Silhavy, R. Silhavy and Z. Prokopova, "Categorical Variable Segmentation Model for Software Development Effort Estimation," *IEEE Access*, vol. 7, pp. 9618-9626, 2019.
- [12] J. A. Khan, S. U. R. Khan, J. Iqbal and I. U. Rehman, "Empirical Investigation About the Factors Affecting the Cost Estimation in Global Software Development Context," *IEEE Access*, vol. 9, pp. 22274-22294, 2021.
- [13] Tomás Vera, Sergio F. Ochoa, Daniel Perovich, "Survey of Software Development Effort Estimation Taxonomies," http://www.dcc.uchile.cl/TR/2018/TR_DCC-20180320-002.pdf, 2018.
- [14] M. El Bajta et al., "Software project management approaches for global software development: a systematic mapping study," *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 690-714, December 2018.
- [15] M. A. Shah, D. N. A. Jawawi, M. A. Isa, M. Younas, A. Abdelmaboud and F. Sholichin, "Ensembling Artificial Bee Colony with Analogy-Based Estimation to Improve Software Development Effort Prediction," *IEEE Access*, vol. 8, pp. 58402-58415, 2020.
- [16] R. Vijay Anand and M. Dinakaran, "Issues in Scrum Agile Development Principles and Practices in Software Development," *Indian Journal of Science and Technology*, vol. 8, no. 35, pp. 1-5, 2015.
- [17] Akif Raza and Hammad Majeed, "Issues and Challenges in Scrum Implementation," *International Journal of Scientific and Engineering Research*, vol. 3, no. 8, pp. 1-4, 2012.
- [18] Ashraf, Muhammad, and Natash Ali, "Impact Agile Project Management: Identification and Analysis of Problems in Scrum Implementation," *VFAST Transactions on Computer Sciences*, vol. 1, pp. 60-67, 2013.