

The Importance of Robust Communication in Large - Scale Agile Development

M. Gangadharan

Assistant Professor of English, Kuppam Engineering College, Kuppam, Chittoor, Andhra Pradesh, India
Email: gangadharanmgs[at]gmail.com

Abstract: *The agile methodology stands out as a prevalent model for efficient software development, particularly favored for its adaptability and suitability in small - scale projects across various software industries. Nevertheless, its widespread adoption has brought to light certain communication challenges, particularly when applied to large - scale distributed teams. Agile, it appears, may not be the optimal choice for extensive teams engaged in global software development efforts. This study delves into the intricacies of issues faced by teams employing agile in the context of large - scale distributed development, particularly focusing on communication - related challenges and their repercussions. Our approach involved in - depth interviews with diverse developers and teams hailing from various sectors within the software industry. Moreover, we conducted an extensive quantitative analysis, surveying 50 developers representing different distributed teams. The outcomes of our investigation unearthed several communication - related deficiencies that significantly impact the development process. To arrive at these insights, we employed two robust statistical analysis methods: descriptive analysis and regression analysis. The implications of our findings have led us to propose innovative software solutions, bearing distinctive features engineered to mitigate the communication issues often encountered in large - scale software development. These solutions have the potential to enhance the efficiency and effectiveness of agile practices when applied in extensive and globally dispersed development endeavors.*

Keywords: agile methodology, communication challenges, large-scale development, distributed teams, software solutions

1. Introduction

Agile is the frequently used software development model in the software world. Agile software development gains popularity in recent decades due to its unique components to overcome deficiencies from efficient software Shariq Aziz Butt et al. / Procedia Computer Science 236 (2024) 224–232 225 2 Author name / Procedia Computer Science 00 (2019) 000–000 development. The most owing feature of the agile model is a change of requests from clients at any stage of a project. Agile was used in small - scale software development due to its small team size but now it is gaining popularity in large - scale software development. There are so many aspects that need to be considered for the successful software development and one of them is communication. Communication is the most significant aspect of software development because any project's success depends on effective lines of communication. However, as increasing numbers of teams move to agile methods, so must the way they interact. Agile development is based on simplification, adaptability, and continuous improvement. It's challenging to predict precisely where a project will be and what the specific responsibilities will be on any particular day because agile teams learn as they go. It makes conventional communication strategies that rely heavily on documentation difficult and ineffective. Currently, the open issue in large - scale agile development is the communication between the team members. Efficient communication means productive software development. Significant communication helps the team members to understand the all difficulties of developing software. This is the reason agile development arranged daily meeting sessions with team members. In this study, we are going to assess communication factors in the large scaled agile development. How do these factors affect the software development of large - scale teams and software companies which use the agile model for development? This study is also determining the critical

analysis of factors that cause project success or failure. For the assessment, we used hypothetical modeling and designed the hypotheses to reveal the outcomes by using the survey method from software developers. We have used the 3 statistical methods including Cronbach's alpha, descriptive, and regression testing. The main contribution of this study is to highlight the factors that impact communication in large - scale project development. This study has contributed to the following aspects of communication in agile development: • Reduce the communication issue • Enhance communication skills in large scaled software development. • Suggested a technological model for the development of software to reduce the issues of communication from agile.2. Background Agile methodology is a well - known model for efficient software development. It has fame more than any other SDLC model including waterfall, iteration, and so on. The main reason for the fame of agile methodology is its unique features that support efficient software development. Agile software development approaches were initially used by small teams and projects, where they were successful, and boosting these techniques is difficult [1, 2]. However, bigger organizations are also addressing the same difficulties that agile techniques address [3, 4, 5]. Since agile processes place a high emphasis on interactions with people, it stands to reason that communication and management approaches are significant potential success factors for agile development. Furthermore, interpersonal and managerial issues, a shortage of successful communication for the project's completion [6, 7, 8], and discomfort with the teams, rather than technological issues, are to blame for the majority of project failures. Therefore, it's crucial to comprehend which social aspects have a big impact on the success of agile projects. As longer projects appear to be failed increasingly frequently than shorter ones [5, 10, 11]. The main reason for this failure is the large - scale projects with large teams which required good communication among team members to succeed in the project. But agile

Volume 13 Issue 11, November 2024

Fully Refereed | Open Access | Double Blind Peer Reviewed Journal

www.ijsr.net

always support small team and small scaled projects. There are different studies have been done in agile large scaled project development. The coordination among several development processes, interactions across numerous distant teams, management of inter - team interdependence, and sharing of information among teams are the main issues resulting from increased complexity. To scale development groups that work in an agile environment and produce artifacts, this study compares potential agile frameworks [14, 15]. The key difficulties in the communication process were discovered to be resistance to change, an overaggressive rollout schedule, issues with quality assurance, and integration with pre - existing non - agile business operations [16, 19, 20]. The Critical Problems for scaling agile approaches were also presented in this study utilizing the condition of the difficulties having a prevalence of 50%. According to the results, six out of the eleven problems with expanding agile approaches are serious ones [21, 22]. A possible scenario was provided based on the difficulties mentioned, highlighting the connection between those issues and the adoption of agile approaches in the GSD context [23, 24, 27]. In comparison to the other divisions of the SFs, the study's findings showed that "technologies" represents the most 226 Shariq Aziz Butt et al. / Procedia Computer Science 236 (2024) 224–232 Author name / Procedia Computer Science 00 (2019) 000–000 3 essential component. Similar to this, a strong technological foundation is cited as a key element [25, 26]. Depending on the findings of the study, the researchers can conclude that its contribution goes beyond the creation of a classification system of communication factors to include the appropriate consideration of those excellence factors through the use of the Analytic hierarchy process approach, which helps software companies scale agile practices successfully in the Distributed environment [28, 29, 30].

3. Methodology To find the communication barriers across agile teams, we have surveyed the software developers and practitioners. In this survey main objective is to find the issues and proposed a solution to overcome such limitations from efficient software development at a large scale. The data specification, data gathering, and data collection are described in Table 1.1. We have collected the survey from 3 software industries as mentioned in Table.2. For the hypotheses identification, we have done a comprehensive study [12, 13]. Table.4 is illustrating the 3 software industries that were part of this study. These companies are using the agile model for development on large scale. They have adopted distributed software development models as well including SAFe, and LeSS. But still, these companies have issues. During the adopted model the software companies faced challenges in communication that still need to be addressed. In this study, we have arranged a discussion session with the distributed teams and tried to identify a possible solution. For detailed discussion we have used the questions that are mentioned in Table.3. We have designed the questionnaires to discuss the issues and find the parameters that should be adopted to overcome this issue from agile large - scale development [16, 17, 18]. For the responses, we have used the Likert scale of 5 points for measuring the responses obtained from respondents. We have arranged the interview sessions with the teams as mentioned in Table.2. We have arranged a session of about 25 - 30 minutes with each team to discuss agile large scaled

communication. Figure.1 is illustrating the research model with the factors that affecting the communication in agile software development. Fig.1. Research model (9). Table 1. Specification of data, survey objectives, data sourcing, and data collection. Objectives To recognize the main shortcoming in the large scaled projects development or at the GSD level. Subject Area Large - scale development. Main Research Question What are the main difficulties developers faced in agile development? Type of Data Questionnaires How Data Was Acquired Analysts develop Questionnaires for analysis Data Format Analyzed and statistical data Experimental Factors The data model consisted of software developers who mostly are software development persons in software industries. The researcher designed the questionnaire which contained data on communication issues associated with agile. Data Source Location IT Shariq Aziz Butt et al. / Procedia Computer Science 236 (2024) 224–232 2

participants regarding the designed hypotheses and gathered their opinions to resolve the limitations of large - scale agile software development [31, 32]. We have done 2 statistical analyses on the dataset collected after the survey session. We have applied descriptive analysis and regression analysis. Table.5 is describing the descriptive analysis of the outcomes got from participants after the survey session ended. In the descriptive analysis mean and mode of all variables in the descriptive are 1 and the estimations of SES and SEK are 0.72 and 0.14 individually for every one of the factors. The Descriptive Analysis of the data is mentioned in Erreur! Source du renvoi introuvable. The applied descriptive analysis on survey responses mentioned that 72% of the developers stated that processes that they need for the development of large scaled agile projects are not available and defined. Further, 70% of respondents stated that they faced communication issues during the collaboration with the team/s due not to being co - located. The majority of developers also stated that 80% of the developers located in other places do not reply properly. They have no proficient interest to reply on time due to time and location differences. Table 5. Descriptive Analysis of large - scale project development. Predictive. We have performed the regression analysis on the dataset as stated in Table.6 in the model summary the multiple R is the square of the R square and it collects the association within the factors/hypotheses. In Table.5 the correlation coefficient is between +1 and - 1. Regression Anova F - test illustrates how a predicted variable contributed to the predictions of the outcome variable and also shows the degree to which ambient factors and measurement items are correlated. Every F - value is statistically significant as Sig. F (p - value 0.05). To predict the needed variable based on the evaluation of the autonomous factor, B within the segmentation of the determination coefficient is used. It explains how one value or number will increase or decrease the variables and how many units will change the independent variable. The β regard is linked to the t - value and the importance of the t - regard to determine whether the coefficient is less than 0 or more than 0; hence, the β can either be + or - . Regression's β , t, and p sections are assessed, and because the p - values are greater than 0.05, none of the β values are exactly equal to 0. In the regression analysis, we have revealed that the highest score is for hypotheses 9 and 10. H10 has the 82% of

responses from participants, they have said that most of the issues while using the traditional communication medium are conventional and can't handle the updates of projects. Teams at different locations have a very poor motivation to timely communicate or respond to the other team member. While the H9 stated that 83% of co-located team members do not share the proper documentation and code of the project/unit on which they are working. It is due to insufficient behavior of teams regarding communication and collaboration with each other due to time differences. Respondents with 74% responded under the H1 that the processes that are mandatory to develop the software application are not available on the platform, particularly in the distributed teams in large-scale companies or global software development. Software bugs and defects are not reported timely, in H2 73% of respondents said that they have faced this issue. Due to this quality of the product suffers, time increase, and somehow the maintenance cost becomes increase. There is also another issue faced by the teams is requirements specification. In the H3 70% of respondents stated that the clarity of requirements for all team/s members is very difficult. Because time difference and different working environments and aptitude for work made them distinct to collaborate for requirements. In agile it's more Shariq Aziz Butt et al. / Procedia Computer Science 236 (2024) 224–232 229 6 Author name / Procedia Computer Science 00 (2019) 000–000 difficult to manage the scrum meetings because due to the time difference developers can't arrange the daily session to discuss the user story and its complexity. The hypothesis score is H4 67%, in this hypothesis respondents revealed that they did not find any technology or any kind of software that has been developed to support the large scaled distributed teams. The technology needs to be advanced to support distributed communication. In the H5 73% of respondents mentioned that they faced the inaccurate estimation of time, efforts, and budget of the project. It is due to less communication between the teams. Moreover, they stated that the no of modules has been assigned to a developer who has no expertise in work. H6 is supporting hypothesis H3 because due to less communication team can't assess the complexity of the user story. As 71% of respondents stated that they have observed that several user stories were complicated and predicted wrongly. This less communication in agile software development also affects the integration of modules of a project. 67% against H7 said that when they developed the software and then integrate the modules, some of the modules were as per requirements. Further, teams also faced issues with timely updating each other. Most of the developers 72% in H8 mentioned that late updates to each team or no mechanism to communicate with each other on time made the projects late to deliver and complete. Table 6. Regression analysis. Predictive variables Model Summary ANOVA Un-standardized 0.85 6.70. Discussion and Findings In this section, we have suggested some parameters that need to be used for the development of software for communication issues in large scaled agile software development. There are different software that facilitates communication such as zoom, but it does not support communication for teams with different time zones. It only supports the discussion but does not facilitate the sharing of codes, task updates, effort estimation, time management, and so on. Jira is also a tool used for user stories specifications

but does not support communication within the distributed teams. Therefore, we have suggested some guidelines to develop software to reduce communication issues from large-scale agile development. In Table.7 we have explained all such features that we have designed after the detailed survey with the developers. 230 Shariq Aziz Butt et al. / Procedia Computer Science 236 (2024) 224–232 Author name / Procedia Computer Science 00 (2019) 000–000 7 Table.7 Suggested features for the development of software for communication. Contents Research Findings Recommendation Impact of recommendation Scrum Daily meetings are not possible for the developers working from a distributed location. Developers because they feel their tasks are most important for them to complete rather than to discuss the other tasks of the same module. Add a feature in the software that can save the updates of the project and give a chance to look at the updates of the project. This will make both teams more comfortable locating the updates of each team's work. This will be an automated feature in the software i. e. any developer from any team does not need to put updates manually in the software. When there is any update in any module of the project then the software automatically will save it for all distributed teams to track the updates. Efforts and cost Requirements are not fixed and also not pertinent for distributed team/s in agile development. To estimate the accurate efforts and cost as per the developer's expertise in both teams. There should be a feature in the software developed to improve communication in agile development. The software automatically takes each developer's expertise against the user stories and predicts how much time and effort would be needed to develop the user story/s. This feature also will reduce the biased estimation from the agile software development as faced in planning poker. User story complexity. The main issue in distributed development is to find the accurate complexity of a user story. Establish a team effort estimation mechanism rather than support the time chief programmer or team lead. This feature will enhance the estimation process and will support the team to locate the accurate efforts of user stories. When the developers put their opinions against the user stories then the software will decide the accurate project's type. Moreover, the software will support the teams to locate the right developer for each user story. Modules integration Teams faced that when they integrate the project's modules then the project is not developed as per the requirements. A feature needs to be added to the software to integrate the project's modules. When the project starts then the software will divide the project into modules and whenever any module/task is finished by the developer then the system automatically will merge these to make a sequence of modules as per user stories. Lack of team motivation Teams have less motivation to participate in integrated software development. Software should be developed to reduce the interaction of the teams i. e. teams must put all the details of a product are all updates in the portal. It will reduce the daily meeting session of the agile software development. Moreover, if any developer from any team does not put updates then the software will generate a trigger to that developer as well as all the software developers. This will feature will enhance productivity by tracking modules and managing motivation.

2. Conclusion

The agile model is most useful in the software industries for the development of efficient software. Due to last few decades, agile has gotten more fame due to its distinctive features. On the other hand, agile is not pertinent for distributed or large - scale development. In this study, we have done a study on distributed development in agile regarding communication issues. We have done a survey study with the different software industries and also mentioned similar studies. This work has highlighted the challenges faced by the teams to communicate with each other.

References

- [1] Van Kelle, E., Visser, J., Plaat, A., & van der Wijst, P. (2015, May). An empirical study into social success factors for agile software development. In 2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering (pp.77 - 80). IEEE.
- [2] Wińska, E., & Dąbrowski, W. (2020). Software development artifacts in large agile organizations: a comparison of scaling agile methods. In Data - Centric Business and Applications (pp.101 - 116). Springer, Cham.
- [3] Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. International Journal on Advanced Science, Engineering and Information Technology, 6 (6), 828 - 837.
- [4] Reifer, D. J., Maurer, F., & Erdogmus, H. (2003). Scaling agile methods. IEEE software, 20 (4), 12 - 14.
- [5] Fitzgerald, B., & Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123, 176 - 189.
- [6] Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. Journal of Software: Evolution and Process, 30 (10), e1954.
- [7] Shameem, M., Chandra, B., Kumar, R. R., & Kumar, C. (2018, December). A systematic literature review to identify human related challenges in globally distributed agile software development: Towards a hypothetical model for scaling agile methodologies. In 2018 4th International Conference on Computing Communication and Automation (ICCCA) (pp.1 - 7). IEEE.
- [8] Shameem, M., Khan, A. A., Hasan, M. G., & Akbar, M. A. (2020). Analytic hierarchy process based prioritisation and taxonomy of success factors for scaling agile methods in global software development. IET software, 14 (4), 389 - 401.
- [9] Kaur, P., & Sharma, S. (2014). Agile software development in global software engineering. International Journal of Computer Applications, 97 (4).
- [10] Martini, A., Pareto, L., & Bosch, J. (2013, August). Communication factors for speed and reuse in large - scale agile software development. In Proceedings of the 17th international software product line conference (pp.42 - 51).
- [11] Garcia, A., Silva, T. S. D., & Silveira, M. S. (2019, May). Artifact - facilitated communication in agile user - centered design. In International Conference on Agile Software Development (pp.102 - 118). Springer, Cham.
- [12] Ghani, I., Lim, A., Hasnain, M., Ghani, I., & Babar, M. I. (2019). Challenges in distributed agile software development environment: A systematic literature review. KSII Transactions on Internet and Information Systems (TIIS), 13 (9), 4555 - 4571.
- [13] Ahmad, M. O., Lenarduzzi, V., Oivo, M., & Taibi, D. (2018, September). Lessons learned on communication channels and practices in agile software development. In 2018 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp.929 - 938). IEEE.
- [14] Ouriques, R. A. B., Wnuk, K., Gorschek, T., & Svensson, R. B. (2019). Knowledge management strategies and processes in agile software development: a systematic literature review. International journal of software engineering and knowledge engineering, 29 (03), 345 - 380.
- [15] Lampropoulos, G., & Siakas, K. (2018). Communication in Distributed Agile Software Development: Impact of Social Media-Social Networking. In SQM (Vol.26, p.43).
- [16] Khalil, C., & Khalil, S. (2020). Exploring knowledge management in agile software development organizations. International Entrepreneurship and Management Journal, 16 (2), 555 - 569.
- [17] Salman, A., Jaafar, M., Malik, S., Mohammad, D., & Muhammad, S. A. (2021). An empirical investigation of the impact of the communication and employee motivation on the project success using agile framework and its effect on the software development business. Business Perspectives and Research, 9 (1), 46 - 61.
- [18] Robinson, P. T. (2019, May). Communication network in an agile distributed software development team. In 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE) (pp.100 - 104). IEEE.
- [19] Butt, S. A., Ercan, T., Binsawad, M., Ariza - Colpas, P. P., Diaz - Martinez, J., Piñeres - Espitia, G., . . . & De - La - Hoz - Hernández, J. D. (2023). Prediction based cost estimation technique in agile development. Advances in Engineering Software, 175, 103329. .
- [20] Alzoubi, Y. I., & Gill, A. Q. (2020). An empirical investigation of geographically distributed agile development: The agile enterprise architecture is a communication enabler. IEEE Access, 8, 80269 - 80289.
- [21] Uludag, Ö., Kleehaus, M., Caprano, C., & Matthes, F. (2018, October). Identifying and structuring challenges in large - scale agile development based on a structured literature review. In 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC) (pp.191 - 197). IEEE.
- [22] Zia, A., Arshad, W., & Mahmood, W. (2018). Preference in using agile development with larger team size. International Journal of Advanced Computer Science and Applications, 9 (7), 116 - 123.

- [23] Kahya, M. D., & Seneler, Ç. (2018, September).
Geographical distance challenges in distributed agile
software development: