International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-10 Issue-2, December 2020

Enhancing Software Development Teams' Client Awareness: An Empirical Study of Its Impact on Productivity

Abdulrahman M. Qahtani

Abstract*: *In the software industry, a critical factor in a project's success is raising productivity, and software development teams must always consider its challenges. In today's competitive industry, the productivity of team members in software development is a serious issue that attracts considerable attention. Studies have been conducted on various aspects of team and individual productivity; however, the literature still refers to a lack of research into the impact of team awareness, observing that it is an essential element of knowledge management in the project's development life cycle. This study takes up this point and presents an actual software development case study to investigate the impact of increasing knowledge and producing adequate information on clients' domain and business model on both team productivity and that of each individual member. The study was undertaken with two development teams over one month, each receiving about 300 requirements. One of the teams was given sufficient information on the client's domain and background in terms of its business model, while the other was given nothing before it went to the client's workplace, without any knowledge of its domain. The results achieved were statistically significant, showing better productivity among the team with the information, with 261 of 300 requirements completed, whereas the other completed just 107. The findings of this study will help software research to focus both on the aspects of knowledge management that relate to software development and on the correlation between them. The study also supports software development project managers to enhance the value of knowledge when they are delivering training and to equate the time spent spreading knowledge to giving team members adequate information about the clients' domains and business models. This will be reflected in both the quality and productivity of the entire development process.

Keywords*: *About four key words or phrases in alphabetical order, separated by commas.

I. INTRODUCTION

The software engineering industry has seen major changes in recent decades, leading researchers to develop ways to deal with various aspects of software projects, such as the process, the people and the product itself [1]. In recent software history, information and data have become regarded as playing a vital role in software development projects at several levels and in many forms [2].

Revised Manuscript Received on December 30, 2020.
***** Correspondence Author
Abdulrahman M. Qahtani*, Computer Science, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia. Email: amqahtani@tu.edu.sa

One of the primary forms of such information is conveying knowledge on the project's artefacts, ranging from the client's business environment and business model to its requirements and how the development process is to proceed. This study reviews the literature on knowledge management in software engineering and finds that this topic has an important impact on software development in general; moreover, it becomes more significant in the distributed development domain, when clients are on distant sites [3]. Several studies indicate that both global and distributed development projects, especially those dealing with several domains and clients at the same time, may not practise adequate knowledge management [4]. Consequently, this study focuses on knowledge management in software development projects of a multi-site nature, as this domain suffers most from the absence of essential knowledge and almost fails to understand clients' requirements enough to meet their expectations [4].

According to [5], training and learning refer to information-sharing and continuous learning capabilities that increase the probability of success in the practices involved in agile software development. Training is a critical factor of success behind large-scale development transformations, and teams that are not adequately trained often struggle to implement agile practices correctly [6]. These practices do not rely on formal training for knowledge sharing; instead, there is a focus on mentoring and professionally guided discussions, as these yield better results [5]. Since agile practices and techniques do not follow a strict 'how to' guide, learning is achieved through continuous experimentation [7]. This factor refers to the utilization of knowledge, along with the conditions that allow teams to accomplish their tasks [8].

Authors of [5] maintain that a highly capable team enables the rapid and frequent delivery of working software that meets a customer's requirements. Besides technical competence and expertise, other attributes are cited by [9]: team members' motivation and commitment; agile, knowledgeable managers with an adaptive management style; and appropriate provision of technical training to the project team. Aspects such as commitment and technical expertise are, according to [10], the drivers that enable teams to deal better with risk, thus improving the likelihood of the project's success. Bearing these factors in mind, this study was designed to investigate the impact of knowledge management on development productivity at the team level, as well as the level of the individual.

A case study was conducted to experiment in a real-world situation. An anonymous software development company was chosen, and it nominated eight developers to be involved in the experiment. Intensive training was delivered to one team of four members, whose productivity was compared with the other team.

The experiment's output was next analysed statistically and discussed. The results indicate a significant difference in the productivity of the team with sufficient knowledge of the client's domain and business model, relative to the team that worked on the same project without knowledge of the client's business model or environment. These findings support the managers of software development projects to secure better quality and productivity by increasing the time allocated to training and knowledge dissemination before starting the development.

The rest of this article has the following structure. Section 2 gives the background to this research: knowledge management; knowledge management in software engineering; and awareness in distributed development projects. Section 3 presents the empirical study's procedures and setting. Section 4 presents the results of the evaluation process and a discussion. Finally, section 5 concludes this study concerning its findings and meeting the study goals.

II. LITERATURE REVIEW

This section discusses works related to this study, organized into three sub-sections. The first concerns knowledge management and studies related to this topic. The second reports on studies on knowledge management in software engineering. The third discusses the awareness of distributed working and software engineering applications in this area, especially in the distributed domain.

A. Knowledge management

Knowledge, in the world of administrative organizations, is a resource that it essential to the success of both firms and their projects [11][12]. Its importance increases with a firm's size and its number of employees. Part of its critical role in an organization's success lies in the value that it adds, enhancing both continuity and competitiveness. The need for knowledge and its management is not limited to a single level of an administrative and organizational structure but relates to the entire organizational hierarchy, from top management to the bottom rungs of the employment ladder. Knowledge management has attracted attention in several broad domains, primarily software engineering [13]. By its nature, this field involves a high degree of knowledge work in various forms, such as on requirements, documents, codes and policy and procedures.

Consequently, software companies have a keen interest in knowledge management and value knowledge and information in their projects and plans. A review of the literature reveals many studies that discuss knowledge management for both software development and administration in various contexts. These range from research on how software engineers deal with challenges that arise in projects to achieve success in terms of accurate software, and how to learn from failure, both their and others', as discussed by [14]. Moreover, some investigate knowledge management in software engineering in terms of mechanisms to choose appropriate approaches and methods for development, and like using the Agile or waterfall models to manage projects [15][16]. According to [17], in recent years the software industry has come to rely on several innovative tools and technologies to make this knowledge accessible and manageable throughout a software project.

B. Awareness in software development projects

There are various definitions of awareness, depending on the context. The first is a consciousness gained by collaborators' perceptions of knowledge and information about their environment. According to [18], it is 'an understanding of the activities of others, which provides a context for your activities'. Gutwin and Greenberg [19] state that awareness is a technique for enhancing coordination when people work together. In the software engineering field, awareness plays a vital role in several areas. Software development involves many developers and practitioners working together on the same project, using the same resources, and this demands cooperative and coordinated efforts if high-quality systems are to be produced[20]. Awareness in a software project impacts on the processes being undertaken by enhancing the sharing and understanding of vital knowledge concerning the activities and expertise of the developers, the stage of project tasks and the state of various artefacts. The knowledge exchange is known as vital *awareness information*, and involves managers, developers and clients in its communication, either directly or tangentially, typically using formal regular meetings, informal personal interruptions and electronic messages[21].

C. Vital information awareness in the distributed domain

Several studies have been conducted on vital information awareness during software engineering projects. For example, [22] investigated this aspect of workspace awareness in real-time distributed groupware, proposing a framework to clarify its various types in development projects as shown in Fig.1. Another study investigates it in the software engineering workplace, examining the impact of various factors on awareness of knowledge at a specific workplace [23]. In recent years, several studies have been conducted on the level of this awareness in the distributed domain in various aspects of software development projects. For instance, [24], [25]discuss security in a software project and how to improve its awareness among those working on it. Other research discusses awareness and its impact on agile development projects [26].

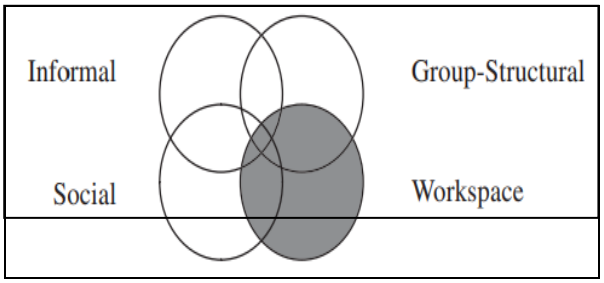

Fig. 1. Types of awareness [22]

In summary, the literature review demonstrates that knowledge management is central to all organizations and that knowledge-sharing in software engineering is essential to create among project team members an awareness of the client's business. Moreover, it reveals a lack of empirical studies on the phenomenon of teams' awareness of their clients' environment.

III. RESEARCH METHODOLOGY

In this section, the empirical experiment on an actual case is discussed in detail to examine the impact of increased awareness in the development workplace at both team and individual level. It assesses team productivity in terms of the requirements completed in a specific time. In short, the study follows the guidelines laid down by [27].

A. Problem definition

Success in a software development project comes with incredible difficulty. One of the most significant challenges in this type of project is to understand the client's business model and spread awareness among the members of the development team to prevent failure and increase productivity [9]. Therefore, this study discusses the impact on process productivity of practitioners' awareness of a client's business model and domain during a development project.

B. Hypotheses and research questions

Quality and speed are crucial to all software development projects, representing a tough challenge for a distributed developers working in multi-client teams. This study aims to discuss the impact on the development process of increasing awareness of the client's business model and domain through improving development team productivity. The hypothesis of this study is as follows:

H1: Increases in the development team's awareness from training its members on the client's domain will increase the productivity of the development project.

H2: The productivity of individual development team members will increase after intensive training on the client's environment.

The objective of this study is to discuss the impact of awareness-raising on both the software development project and individual team members. It aims to measure in development projects for multi-clients the number of requirements completed in a specific time by trained members and untrained members, respectively, to ascertain whether there is an increase in productivity.

To meet the research objectives and accomplish these aims, the following research questions were formulated:

RQ1: What is the impact of increasing the awareness of software development team members by giving them training on clients' domain on the productivity of the development process in the multi-client domain?

RQ2: What is the impact of delivering intensive training for development team members on individual productivity?

C. Experimental Setting and Procedure

In software engineering research, there are several ways to conduct experiments and collect data to either build or confirm a theory. One is to apply an experiment on a real software engineering project as a case study. Experimentation in a case study explores wide-ranging information in depth, with a focus on a range of people, organizations or contexts [28]. According to [29], case studies aid practical understanding and extend our experience of a subject. This methodology is appropriate for software engineering research as it studies various phenomena in their natural context [30]. Therefore, this research has used a real case study and collected data from a software development company whose project involvement and dealings with the client's requirements took place in the distributed domain. The company, based in Saudi Arabia, has developed a variety of software solutions for sectors such as academia, healthcare and business intelligence and has significant experience in developing products for distributed clients.

This study aims to investigate the impact on a development project and its team members from possessing adequate knowledge. Thus, the study recruited eight developers with the same experience and skills for two development projects. They were split into two teams of four developers. The first underwent an intensive training programme for one month to learn about the client organization's business model and domain. By contrast, the second team went straight to the clients' site with no training. Next, the requirements were distributed equally among both teams' members to ensure that the experiment setting was consistent.

D. Participants

Software development projects rely heavily on developers' experience and skills. This experiment aimed to investigate the productivity of both the developer and the entire process. It was therefore necessary to recruit the participants in an extremely careful and accurate manner. Two projects that had a similar business model and business environment were identified, and each was allocated four developers, all with two to five years of experience in developing educational systems. To control the variables accurately, each developer was assigned an equal number and type of requirements for one month.

E. Data analysis

The selected company in this study uses a workflow system known as JIRA [31], which is project management software to manage clients' requirements and other processes throughout the development cycle. This made observing the developers and counting their resolved requirements easier and more accurate. The data were collected from JIRA in the form of the requirements completed by each developer over the one-month period, as shown in Figure 1.

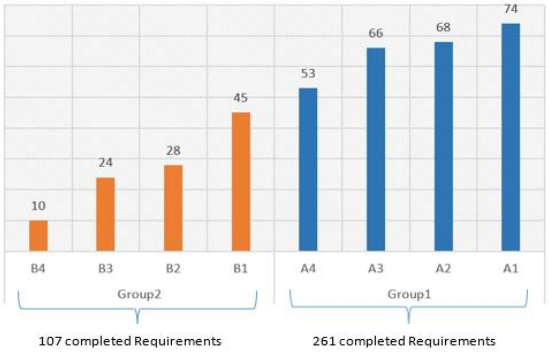

Fig. 1. Nos. of requirements completed by each developer in Teams 1 and 2 (in one month)

Productivity is the usual measure of quality in the software development process. Therefore, this study defined as its primary goal the examination of the impact on a software development project of increasing awareness of both individual and team productivity. This was undertaken by comparing the productivity of two teams.

Each team received 300 requirements, distributed equally among its members. Figure 1 shows the number of completed by each team: 261 for Team 1, which had intensive training and broad knowledge regarding the clients' domain; and 107 for Team 2, which received no training and had little knowledge of the client's domain and business model (only from information available in documents). Also shown is the number of requirements completed by individual developers in each team.

Table I: Results of statistical comparison between Team 1 (one month of training) and Team 2 (no training)

Client team	N[1]	Mean[2]	Median	P-value
Team 1	4	4.50	67	0.03 (P <0.05)
Team 2	4	2.50	26	

[1] Number of developers in each team. [2] Mean of rank for each team.

To conduct a statistical analysis of achieved results in Figure 1, a statistical test was undertaken as the data do not show a normal distribution. A ranked test was necessary, and the appropriate test was the Mann-Whitney U test. The results in Table 1 show the output used to compare the number of requirements completed by the two teams.

The median of Team 1 displays a variance of 26 to 67 requirements. At the same time, the P-value was 0.03 less than 0.05, indicating a significant difference in the productivity of the two teams. This led to the rejection of the null hypothesis and acceptance of the alternative hypothesis, which states that there was a significant increase in development team productivity in Team 1 relative to Team 2.

IV. RESULTS AND DISCUSSION

The main goal of the experiment was to investigate the importance of knowledge and developers' awareness to a project's productivity. Productivity was measured by comparing the requirements completed by the developers assigned to the trained and non-trained developer teams, respectively. Therefore, the experiment aimed to examine the following hypothesis:

H1: Increases in the development team's awareness from training its members on the client's domain will increase the productivity of the development project.

The results presented in Table 1 show a statistically significant increase in the productivity of the development. Completed requirements for Team 1, which had a high level of background knowledge of the client's domain, numbered 261, whereas Team 2 completed only 107 of the 300 requirements in the same period (one month). This significant result led to the rejection of the null hypothesis and acceptance of the alternative hypothesis (H1), confirming that there is a discrepancy in the results of the two teams and answering the research question RQ1.

The second hypothesis defined in this study was H2:

H2: The productivity of individual development team members will increase after intensive training on the client's environment.

The results of the experiment indicate a variance in the number of requirements completed by each developer who received sufficient training and background knowledge regarding the client's domain. They completed between 54 and 74 requirements, an average of 65.25 requirements during the month, while the untrained team's members completed between 10 and 45 requirements, an average 26.75 in that time, as shown in Figure 1. Therefore, the results confirm hypothesis H2 and demonstrate to project managers that spending time on training to improve team members' knowledge of the client's requirements is a worthwhile investment that will repay the project in terms of both individual productivity and quality.

V. CONCLUSION

This study conducted an empirical case study on the productivity of both the team and the individual members during a software development project. This article investigates the impact on productivity of enhanced knowledge management and awareness among team members. The study meets the goals defined by this research, confirming that there were increases in both total team productivity and that of the individual team members when they have adequate training and full awareness of the customer's environment and business model.

All the study's goals were confirmed by the experiment on an actual software development project. The case study compared two teams, one trained and with adequate information and the other that went immediately into the workplace to deal with the client's requirements. This research will help software engineering researchers to focus on this aspect and investigate further the correlation of knowledge management and software development productivity. The study reveals the importance to a project's success of spending time in training to provide team members with adequate knowledge.

REFERENCES

1. V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri, "Software startup engineering: A systematic mapping study," *Journal of Systems and Software*, vol. 144, pp. 255–274, 2018, doi: https://doi.org/10.1016/j.jss.2018.06.043.
2. V. Garousi and B. Küçük, "Smells in software test code: A survey of knowledge in industry and academia," *Journal of Systems and Software*, vol. 138, pp. 52–81, 2018, doi: https://doi.org/10.1016/j.jss.2017.12.013.
3. J. R. Frota Dos Santos, A. B. Albuquerque, and P. R. Pinheiro, "Requirements Prioritization in Market-Driven Software: A Survey Based on Large Numbers of Stakeholders and Requirements," in *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, 2016, pp. 67–72, doi: 10.1109/QUATIC.2016.020.
4. S. S. Bharadwaj and K. B. C. Saxena, "Knowledge Management in Global Software Teams," *Vikalpa*, vol. 30, no. 4, pp. 65–76, 2005, doi: 10.1177/0256090920050406.
5. S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices," *Journal of Systems and Software*, vol. 82, no. 11, pp. 1869–1890, 2009, doi: https://doi.org/10.1016/j.jss.2009.05.052.
6. T. Dingsøyr, N. B. Moe, T. E. Fægri, and E. A. Seim, "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation," *Empirical Software Engineering*, vol. 23, no. 1, pp. 490–520, 2018, doi: 10.1007/s10664-017-9524-2.

7. K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016, doi: https://doi.org/10.1016/j.jss.2016.06.013.
8. M. R. Haas, "Knowledge gathering, team capabilities, and project performance in challenging work environments," *Management Science*, 2006, doi: 10.1287/mnsc.1060.0530.
9. T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, no. 6, pp. 961–971, 2008, doi: https://doi.org/10.1016/j.jss.2007.08.020.
10. A. Arthur, C. R. Y, and D. Urs, "A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies," *Journal of Enterprise Information Management*, vol. 28, no. 1, pp. 7–33, Jan. 2015, doi: 10.1108/JEIM-08-2013-0060.
11. F. J. López-Arceiz, A. J. Bellostas-Pérezgrueso, J. M. Moneva-Abadía, and M. P. Rivera-Torres, "The role of corporate governance and transparency in the generation of financial performance in socially responsible companies," *Spanish Journal of Finance and Accounting / Revista Española de Financiación y Contabilidad*, vol. 47, no. 1, pp. 44–80, 2018, doi: 10.1080/02102412.2017.1379798.
12. A. H. Gold, A. Malhotra, and A. H. Segars, "Knowledge Management: An Organizational Capabilities Perspective," *Journal of Management Information Systems*, vol. 18, no. 1, pp. 185–214, 2001, doi: 10.1080/07421222.2001.11045669.
13. R. Hislop, D., Bousa, R., Helms, *Knowledge Management in Organizations: A critical introduction*, 3rd ed. Oxford: Oxford University Press, 2013.
14. K. Lyytinen and D. Robey, "Learning failure in information systems development," *Information Systems Journal*, 1999, doi: 10.1046/j.1365-2575.1999.00051.x.
15. T. Dybå, "Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues," *Empirical Software Engineering*, vol. 7, no. 4, pp. 387–390, 2002, doi: 10.1023/A:1020535725648.
16. V. R. Basili, G. Caldiera, and H. D. Rombach, "Experience Factory," in *Encyclopedia of Software Engineering*, 2002.
17. P. Quezada-Sarmiento, L. Enciso, and J. Garbajosa, "Use of body knowledge and cloud computing tools to develop software projects based in innovation," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 2016, pp. 267–272, doi: 10.1109/EDUCON.2016.7474564.
18. A. B. Pelegrina, C. Rodríguez-Domínguez, M. L. Rodríguez, K. Benghazi, and J. L. Garrido, "Integrating groupware applications into shared workspaces," 2010, doi: 10.1109/rcis.2010.5507305.
19. C. Gutwin and S. Greenberg, "The importance of awareness for team cognition in distributed collaboration.," in *Team cognition: Understanding the factors that drive process and performance.*, Washington, DC, US: American Psychological Association, 2004, pp. 177–201.
20. F. Lanubile, F. Calefato, and C. Ebert, "Group awareness in global software engineering," *IEEE Software*, 2013, doi: 10.1109/MS.2013.30.
21. F. Shull *et al.*, "Knowledge-Sharing Issues in Experimental Software Engineering," 2004, doi: 10.1023/b:emse.0000013516.80487.33.
22. C. Gutwin, S. Greenberg, and M. Roseman, "Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation," in *People and Computers XI*, 1996, pp. 281–298.
23. A. Sarma, G. Bortis, and A. Van Der Hoek, "Towards supporting awareness of indirect conflicts across software configuration management workspaces," 2007, doi: 10.1145/1321631.1321647.
24. M. Niazi *et al.*, "Challenges of project management in global software development: A client-vendor analysis," *Information and Software Technology*, vol. 80, pp. 1–19, 2016, doi: https://doi.org/10.1016/j.infsof.2016.08.002.
25. P. Zhang, X. Han, D. Zhang, K. Qian, and S. Xiong, "A Security Situation Awareness System based on Wide Deep," in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2018, pp. 107–111, doi: 10.1109/CCIS.2018.8691189.
26. M. Levy and O. Hazzan, "Knowledge management in practice: The case of agile software development," in *2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, 2009, pp. 60–65, doi: 10.1109/CHASE.2009.5071412.
27. D. E. Perry, A. A. Porter, and L. G. Votta, "Empirical studies of software engineering: A roadmap," 2000, doi: 10.1145/336512.336586.
28. D. E. Gray, *Theoretical perspectives and research methodologies*. 2014.
29. K. E. Johnson and R. E. Stake, "The Art of Case Study Research," *The Modern Language Journal*, 1996, doi: 10.2307/329758.
30. P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009, doi: 10.1007/s10664-008-9102-8.
31. Atlassian, "JIRA Software," *La herramienta de desarrollo de software líder de los equipos ágiles*, 2017.

AUTHORS PROFILE

Dr Abdulrahman Qahtani is currently Assistant professor in computer science department at Taif Univeristy. Dr. Qahtani received his PhD from Southampton University in 2015. He has extensive experience in software engineering and development process in a distributed domain. His research focused on customization process across organizational boundaries. Recently, he applied machine learning algorithms on software engineering data to predict time and cost estimation for multi-clients projects.