New York Science Journal

Websites:
http://www.sciencepub.net/newyork
http://www.sciencepub.net

Emails:
newyorksci@gmail.com
editor@sciencepub.net

MARSLAND PRESS
Multidisciplinary Academic Journal Publisher

Novel Framework to Improve Communication and Coordination among Distributed Agile Teams

M. Rizwan Jameel Qureshi, Mohammed Basheri, Ahmed AlZahrani

Faculty of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia
rmuhammd@kau.edu.sa, mbasheri@kau.edu.sa, Aalzahrani8@kau.edu.sa

Abstract: This paper discusses the roles of communication and coordination (C & C) in the agile team. C & C are important activities that a project manager has to deal with tactically during the development of software projects to avoid the consequences. Their importance further increases especially in case of distributed software development (DSD). C & C are considered as project drivers to accomplish a project successfully within budget and schedule. There are several issues associated to poor C & C that can lead to fail a software project such as budget deficit, delay in delivery, conflicts among team members, unclear goals of project and architectural, technical and integration dependencies. C & C issues are critical and vital for collocated teams but their presences in distributed teams are disastrous. Scrum is one of the most widely practiced agile models and it is gaining further popularity in agile community. Therefore, a novel framework is proposed to address the issues that are associated to C & C using Scrum methodology. The proposed framework is validated through a questionnaire. The results are found supportive to reflect that it will help to resolve the C & C issues effectively and efficiently.

[Qureshi M. R. J., Basheri M., AlZahrani A. **Novel Framework to Improve Communication and Coordination among Distributed Agile Teams.** *N Y Sci J* 2020;13(10):32-38]. ISSN 1554-0200 (print); ISSN 2375-723X (online). http://www.sciencepub.net/newyork. 7. doi:10.7537/marsnys131020.07.

Keywords: Scrum Master; communication; coordination; distributed agile Teams; productivity

1. Introduction

Scrum methodology clearly describes human roles. The success of Scrum methodology mainly relies on effective and efficient contribution of Scrum teams. Strong communication between team members will increase coordination, knowledge sharing and satisfaction of team members. The effective participation of a member effects to entire Scrum team especially in distributed software development (DSD) environment. Scrum also defines communication and coordination (C & C) cycles i.e., collaborate, deliver, reflect and improve.

Scrum works well for single collocated teams but it is affected when it comes to distributed teams. The most common problems of distributed teams are poor C & C due to different geographical locations, social and cultural differences (Pressman, 2010). The main objective of Scrum is to deliver a product in increments within 2 to 4 weeks. The main phases of Scrum methodology are Plan. Build, Test and Review (Cockburn, 2007). The phases are implemented through core Scrum activities i.e., product backlog, sprint planning, sprint backlog, sprint, shippable product, sprint review and retrospective (Cockburn, 2007). Product backlog is a set of prioritized user stories by product owner. Sprint planning meeting is conduct among team members to decide the sprint backlog to tackle the number of user stories in a sprint. Sprint is workable software that is completed within 2 to 4 weeks. It is up to the team that how many sprints are completed to deliver a shippable product to customer. Sprint is reviewed by team and retrospective is used to judge that Scrum is successfully implanted by the team or it needs further improvement.

C & C activities help agile teams to achieve common goals such as problem solving, decision making, agree on a sprint backlog and review a sprint and Scrum methodology. These activities are also helpful for multiple teams. CC activities are also supportive to deal with the social, technical, integration and architectural dependencies. Scrum is one of the popular agile models and many successful stories are reported for collocated teams. Therefore, there is a strong need to propose a novel CC framework to improve the performance of agile teams in DSS environment. The same is accomplished in this paper to solve the problems of distributed teams.

The paper is further organized as: section 2 covers the related work. The problem selected in this paper is covered in section 3. Section 4 describes the details of the proposed solution. The validation of the proposed solution is illustrated in section 5.

2. Related Work

Scrum is found extremely useful for improving the performance of software projects in several domains (Cockburn, 2007). Another popular area of research is to amalgamate Scrum with other methodologies. A framework is proposed to amalgamate Scrum and V to achieve agility in developing embedded software for automobile industry (Cockburn, 2007). Freitas et al. (2014) to amalgamate Scrum and Model Driven Development (MDD) to develop an educational system. Similarly, another research has been conducted to improve Scrum by developing educational system to manage the documentation that is required to take accreditation of ABET quality standard (Mokhtar et al., 2014). The main advantages of the document management system are to avoid redundancy of data and increase the speed of searching documents during the audit process (Mokhtar et al., 2014).

The key of success to successfully adopt Scrum mainly depends on attitude and expertise of team members (Wangenheim et al., 2014, Scott et al., 2014). Many researchers are of the opinion that Scrum is a suitable methodology to teach educational courses. This is due to the fact that human factor is vital in order to successfully execute the management practices of Scrum. The use of Scrum in a course brings into several advantages such as high productivity, cohesiveness, communication and coordination (Wangenheim et al., 2014, Scott et al., 2014). Wangenheim et al. (2014) use a simulation game to train Scrum in order to speed up the knowledge acquisition process. A model is also proposed to distinguish associations between Scrum and learning mechanism. The proposed model observes performance of the students following the Felder–Silverman model using a tool naming Virtual Scrum (Scott et al., 2014).

The success of team members to transform from traditional methodology to agile Scrum mainly relies on their attitudes (Overhage et al., 2011). A framework is proposed to increase the speed of delivery without compromising quality in the changing business environments. The Technology Acceptance Model (TAM) is used to propose the framework (Overhage et al., 2011). A hybrid fuzzy-ontological system is proposed to engineer a software process simulation modeling (SPSM) system (Orłowski et al., 2014). SPSM simulates management processes and project roles of Scrum (Orłowski et al., 2014). Sprint review and retrospective meetings are important activities of Scrum (Parvez, 2012). It is an important area of research to identify the best testing strategy using Scrum. A study is conducted to propose a model that is used to analyze the performance of regression test by developing mobile application using Scrum (Parvez, 2012).

A study is conducted to catalogue the risk factors that can decrease the efficiency of agile teams while working on distributed development projects (Shrivastava and Rathod, 2014). The objective is to encourage the agile community to conduct further research to control risks that are associated to distributed software development projects (Shrivastava and Rathod, 2014).

Scrum is enriched with management practices and it lacks in engineering practices (Khalane and Tanner, 2013). This limitation of Scrum attracts agile teams to mingle Scrum with other methodologies. A theory building case study is illustrated to report software quality assurance (SQA) using Scrum. It is reported that it is vital for the management of a company to plan the adoption of Scrum to get its fruitful benefits (Khalane and Tanner, 2013). A study is presented to show the adoption of Scrum to develop high quality software with increased customers' satisfaction (Vlietland and Vliet, 2014). The main recommendation is to use dedicated teams to get the desired results (Vlietland and Vliet, 2014).

Scrum works on sprints and each sprint covers clearly defined goals to deal with project's constraints such as cost, time, resources, and technical dependencies (Heikkilä et al., 2014). The main strengths of Scrum are that it can manage severe development risks using strong management practices such as product backlog, sprint planning, sprint backlog, sprint review and retrospective through C & C among stakeholders. It is recommended to use multiple dedicated teams on a large project by dividing into several small projects (Heikkilä et al., 2014).

A novel solution is presented in the area of Agile Scrum to manage large teams (Qurashi and Qureshi, 2014). Qurashi and Qureshi (2014) report several limitations of Scrum using large teams such as repetition of work, poor communication, architectural and integration constraints. Scrum of Scrum methodology is proposed to address the main limitations of using large size teams. The main limitation of the study is that it is kept silent in the cases of more than one product owner and Scrum master (Qurashi and Qureshi, 2014). Two case studies are reported to test the implementations of Scrum of Scrum on two large projects with distributed teams (Paasivaara et al., 2012).

Scott et al. (2014) report a survey indicating that many software development organizations face problems to manage the distributed agile teams. It is described that these problems can be managed by organizing workshops throughout the development of software. Scott et al. (2014) emphasis that the software companies must consider the geographical distances and time zones while selecting the distributed teams and plan strategies to scale agile models for DSD projects.

Dorairaj and Noble (2013) describe different means that can be used to improve trust and communication among DSD teams such as social media applications, collaboration, knowledge dissemination and culture of learning. It is also suggested to arrange training sessions to meet the strategic learning requirements of employees and organizations (Dorairaj and Noble, 2013).

Scrum works extremely well for collocated teams due to its strong management practices but there are few studies available to check its performance in a distributed environment (Pries-Heje, 2011). A case study is conducted using Scrum to develop DSD project to provide evidence. Three frameworks are applied to analyze data. It is concluded that Scrum also works well in a distributed environment because of its better management practices as compared to other agile models. It is also required to introduce state of the art project management computer aided software engineering (CASE) tools to plan; monitor and control distributed teams and improve communication and coordination among distributed teams.

Requirements are changed during the development of software projects due to variety of factors like on customer demands and business needs (Alyahya et al., 2011). The project management is extremely tactical with distributed teams and changing requirements. This needs strong communication and coordination among distributed teams to accomplish the DSD projects successfully. It would help other researchers to identify the issues that the distributed teams could face in DSD projects (Kamaruddin et al., 2011). A survey is reported to gather the issues that are encountered in DSD projects such as social, communication and coordination. It is recommended to use CASE tools to resolve social, communication and coordination issues in a DSD environment.

Scrum methodology needs be customized to match the requirements of distributed projects (Bass, 2010). Bass (2010) classifies nine different roles of the product owner such as tutor, councilor, capable to prioritize user stories, expert and foresee risks. Bass (2010) generally emphasizes about the different roles of product owner in DSD projects. It is further required to investigate the functions of Scrum Master and Scrum team in DSD projects.

Jalali and Wohlin (2010) report a survey on to describe the strengths and consequences of practicing agile methodologies in DSD projects. It is also advised that how to tailor agile methodologies (and their practices) to increase their effectiveness as per the nature of software projects in a distributed environment. It is concluded that the success of agile methodologies mainly rely on their adoption as per the requirements and situations. The role of distributed teams is also promising while adopting agile methodologies to achieve success in DSD projects.

Estler et al. (2012) examine sixty six (66) software projects to compare traditional and modern methodologies. The objective of study is in two folds. The first idea of the study is to set criteria that can be used to select an appropriate model as per the requirements. The second idea is to establish a relation between methodology and degree of achievement, value, inspiration and interaction. It is inferred that it does not matter that the software company select traditional (structured) or modern (agile) methodology to develop a DSD project successfully. Estler et al. (2012) mention that there are other factors that affect to accomplish a successful DSD project such as selection of distributed team structure, communication, coordination, expertise of team members, attitude of team members, management support and domain knowledge.

Al-Zaidi and Qureshi (2017) report the main communication issues that are encountered during DSD projects such as lack of cohesion, unclear goals, poor coordination and lack of knowledge sharing. A novel solution is proposed using Scrum practices to mitigate the communication risks of distributed teams. This is an attempt to address the communication issues of distributed teams but the study lacks in covering the coordination issues of distributed teams. The proposed solution needs to be further validated using multiple case studies to generalize the results.

3. Problem Definition

Agile Scrum model is one of the most widely adopted agile models (Pressman, 2010). It is used to report many successful stories those are available in the existing literature not only in software development domains but also in other domains like manufacturing and education (Cockburn, 2007). Many software companies face problems while adopting agile models. One of the major problems is to keep the pace of software development following agile Scrum values to get its full benefits. This is due to the strong management practices of Scrum like daily meetings, sprint planning meetings, sprint review meetings and retrospective meetings. These all meetings need high C & C.

Companies are somehow able to manage successful execution of these meetings if the Scrum teams are collocated but it becomes highly difficult if the Scrum teams are located at different geographical regions. Majority of the software companies are dealing with the distributed teams and poor C & C fail the whole spirit of strong management practices of Scrum (Ghosh, 2012). The consequences of poor C &

C are delay in sprint delivery dates, increase cost, frustration among team members, conflicts and decrease the support of top management to deploy Scrum. Therefore, a research question is narrated to address the limitations of poor C & C (Ghosh, 2012).

- How to improve the communication and coordination among distributed teams to successfully deploy Scrum?

4. The Proposed Solution

The proposed solution is using Scrum methodology to address the research question. Scrum provides a strong management framework to deliver increments of software in the form of sprints. Scrum is an incremental methodology to deliver software as increments. The increments in Scrum are called as sprints. The sprint release cycle is shown in figure 1.

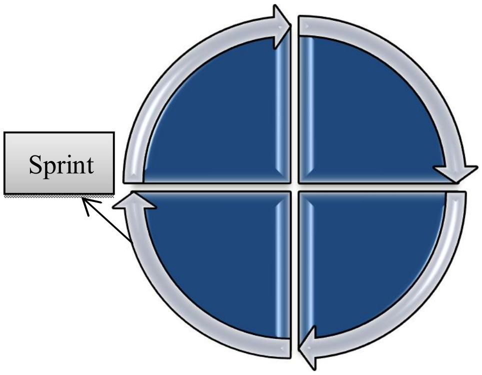

Figure 1. The Release Cycle of a Sprint

The role of people is vital for the success of agile software development especially using Scrum methodology. It is a well-known methodology due to its strong management practices. Scrum development pays more attention to people due to several reasons like people help each other to identify problems, solve problems and take smart decisions to accomplish the project as per the expectations of stakeholders. This paper proposes a novel framework with an aim to help Scrum Master in achieving strong communication and coordination among distributed agile teams.

Figure 2 shows the proposed framework to improve C & C among distributed agile teams using Scrum methodology. Each distributed team has one Scrum Master who acts as communication coordinator and technical support person. The distributed structure of Scrum teams with three locations is shown in figure 3 but it is scalable as per the number of locations of a software company. In figure 3, the Scrum Master of each location is communicating with the Scrum Masters of other locations daily using online meeting tools to discuss the progress and dependencies of project. It is the responsibility of Scrum Master to disseminate knowledge to the team after conducting meetings with the Scrum Masters of other locations. It will help the Scrum Masters to accelerate the progress the project to accomplish it successfully. There are several telecommunication tools available to meet; share knowledge and message among distributed teams such as CISCO WebEx and TeamViewer. CISCO WebEx is introduced to arrange video conferencing. TeamViewer tool controls remote screens to provide technical support. The telecommunication tools will help the distributed teams to improve communication through online meetings, desktop sharing and group messaging.

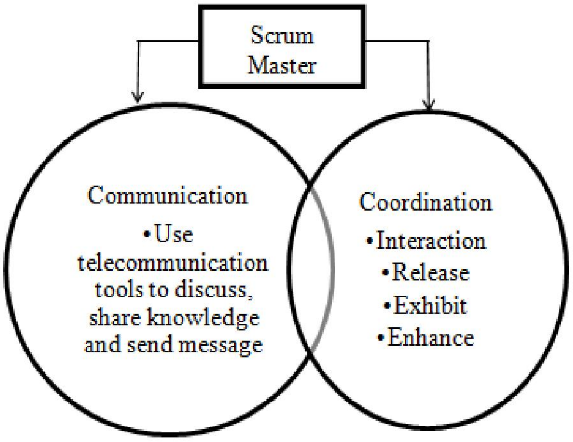
Scrum Master
Communication
•Use telecommunication tools to discuss, share knowledge and send message
Coordination
•Interaction
•Release
•Exhibit
•Enhance

Figure 2. The Proposed C & C Framework

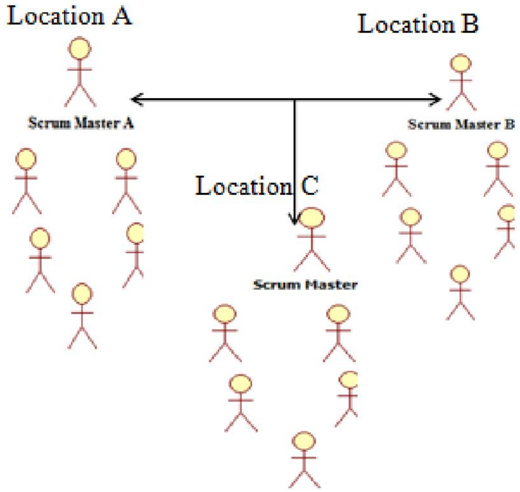

Figure 3. The Distributed Structure of Scrum Teams

Figure 4 shows that the Scrum Master will run the coordination life cycle through 'Interaction', 'Release', 'Exhibit' and 'Enhance' phases. The Scrum Master and team will interact with each other to disseminate knowledge to complete a sprint. Scum Masters of multiple locations will also interact with each other. It will be accomplished through daily meetings. It will help the distributed teams to cope with the adaptability in DSD environment.

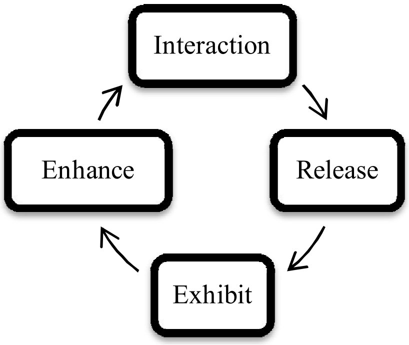

Figure 4. The Proposed Phases of Coordination Life Cycle

The DSD environment needs pipeline development. The pipeline development is hard in its nature to implement in a DSD project because of changing requirements, solutions and languages at each single location. The distributed teams are making decisions to solve the problems. These decisions have consequences affecting the DSD project such as conflicts, ambiguities and unclear goals. Therefore, it is vital to have strong interaction mechanisms to mitigate the risks of poor coordination to complete the DSD project within budget and schedule and as per the values of agile manifesto. Scrum methodology emphasizes that Scrum Master should organize daily meetings with the team to remove impediments.

As a sprint is released, it must exhibits what is committed during the sprint planning meeting. A sprint must be enhanced after sprint review meeting. The authors are proposing to divide the coordination cycle into two sub cycles i.e., 'interaction' and 'release'. The objective of this division is to run two sub cycles in parallel to enhance the coordination among distributed teams. The Scrum Master ensures to exhibit the enhancement in 'Interaction' and 'Release' phases to implement the proposed framework. The enhancement of interaction, between Scrum Masters and distributed teams, is exhibited by decreasing the time, cost, effort and resources to develop a sprint. The enhancement in a release is exhibited by its quality that should be further improved with each upcoming sprint. It is recommended that the Scrum Masters should run the 'Interaction' and 'Release' sub cycles throughout the software development to deal with the C & C issues of the distributed teams.

5. Validation of the Proposed Solution

A questionnaire is used to validate the proposed framework and it is composed of twenty questions. Seven goals are designed to validate the proposed framework.

- Goal 1. Online meeting tools effect to improve communication.
- Goal 2. Remote desktop sharing effects to improve communication.
- Goal 3. Group messaging tools effect to improve communication.
- Goal 4. Emphasis on 'Interaction' phase effects to improve coordination among distributed teams.
- Goal 5. Emphasis on 'Release' phase effects to improve coordination.
- Goal 6. Emphasis on 'Exhibit' phase effects to improve coordination.
- Goal 7. Emphasis on enhancement in 'Interaction' and 'Release' phases effect to improve coordination.

A descriptive analysis is performed to show the results using a frequency table and bar chart. The questions are evaluated using the likert scale i.e., 1 to 5. The cumulative analysis of each goal is performed and Table 1 shows the final cumulative analyses of goals 1 through 7.

Table 1- Final Cumulative Analysis of Seven Goals

Goals	Str. Disagree	Dis-agree	Neutral	Agree	Str. Agree
Goal 1	2.6	0.0	31.6	50.0	15.8
Goal 2	2.6	7.9	21.1	50.0	18.4
Goal 3	0.0	2.6	5.3	34.2	57.9
Goal 4	2.6	5.3	21.1	47.4	23.7
Goal 5	0.0	7.9	21.1	50.0	21.1
Goal 6	0.0	13.2	31.6	34.2	26.3
Goal 7	0.0	7.9	31.6	34.2	26.3
Total	7.8	44.8	163.4	311	173.7
Avg.	1.1	6.4	23.3	44.4	24.8

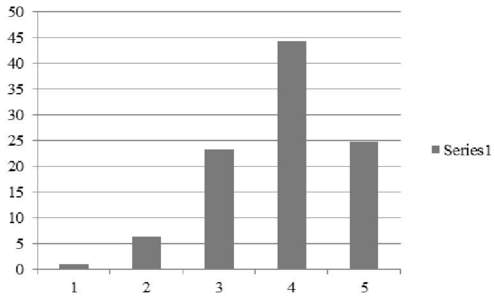

Figure 5. The Final Cumulative Analysis of Seven Goals

According to Table 1, 69.2% of the respondents are agreed with the seven goals in which 24.8% of the software engineers are strongly agreed and 44.4% of the respondents are agreed. 23.3% of the participants are remained neutral for 7 goals. A total of 7.5% responses are disagreed with the seven goals, of which, 6.4% of the respondents are disagreed while 1.1% of the software engineers are strongly disagreed with the 7 Goals as shown in figure 5.

6. Conclusion

Scrum is a well-known agile methodology and it has become one of the choices of project managers from last few years to develop commercial software projects due to its strong management practices. There are many successful stories to apply Scrum using collocated teams. Scrum methodology mainly maintains communication and coordination (C & C) among the Scrum Master and team using sprint planning, daily, sprint review and retrospective meetings. Scrum works well in an environment where team is collocated but it is hard to implement in case of distributed environment. Poor communication and coordination affect DSD projects by incorporating major risks such as delay in shipments of sprints, over budgeted shipments, lack of motivation, unclear goals and friction among team members. There is a firm need to propose a novel framework to address the issues associated to poor C & C. The same is accomplished in this paper by proposing a novel C & C framework. The proposed framework recommends improving communication using telecommunication tools to conduct daily meetings among Scrum Masters of multiple locations. A novel coordination life cycle is also proposed to improve coordination. ‘Interaction’, ‘Release’, ‘Exhibit’ and ‘Enhance’ are the main phases of coordination life cycle. A survey is conducted to check the validity of the proposed C & C framework and it is supported by 69.2% of the respondents. It is concluded from the results that the proposed framework will solve the problem in hand. The future work is to test the proposed framework using case study research method to generalize the results.

Corresponding Author:
Dr. M. Rizwan Jameel Qureshi
Department of Information Technology
Faculty of Computing & Information Technology,
King Abdulaziz University
80221 Jeddah 21589, Saudi Arabia
E-mail: rmuhammd@kau.edu.sa

References

1. Pressman RS. Software Engineering. McGraw Hill Press, USA, 2010.
2. Cockburn A. Agile Software Development: The Cooperative Game, USA, 2007.
3. de Freitas C. R., de Oliveira Junior J. A., Hino F. Y. A., Neto J. S., Vicente L. M., da Cunha A. M. and Dias L. A. V. Applying Interdisciplinarity and Agile Methods in the Development of an Embedded System. Proc. 11th Int. Conf. Information Technology: New Generations, Las Vegas, USA, 2014.
4. Mokhtar R., Jaafar N. H., Tahar N. F., Sukiman S. A., Aris A., and Abu Bakar N. F. An integrated document management system for managing self programme accreditation using scrum approach. Proc. Int. Symposium on Technology Management and Emerging Technologies (ISTMET), Bandung, Indonesia, 2014.
5. von Wangenheim C. G., Savi R., and Borgatto A. F. SCRUMIA—An educational game for teaching SCRUM in computing courses. Journal of Systems and Software 2013; 86:2675-2687.
6. Scott E., Rodríguez G., Soria Á., and Campo M. Are learning styles useful indicators to discover how students use Scrum for the first time?. Computers in Human Behavior 2014:36:56-64.
7. Overhage S., Schlauderer S., Birkmeier D., and Miller J. What makes IT personnel adopt scrum? A framework of drivers and inhibitors to developer acceptance. Proc. 44th Hawaii Int. Conf. System Sciences (HICSS), USA, 2011.
8. Parvez M. A Managed Approach to Interact between Agile Scrum and Software Configuration Management System. International Journal of System Computer 2012; 2:87-91.
9. Shrivastava S. V. and Rathod U. Categorization of risk factors for distributed agile projects. Information and Software Technology 2014; 58:373-387.

10. Khalane T. and Tanner M. Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. Proc. Int. Conf. Adaptive Science and Technology (ICAST), South Africa, 2013.
11. Vlietland J. and van Vliet H. Towards a governance framework for chains of Scrum teams. Information and Software Technology 2014; 57:52-65.
12. Heikkilä V. T., Paasivaara M., Rautiainen K., Lassenius C., Toivola T., and Järvinen J. Operational release planning in large-scale scrum with multiple stakeholders—A longitudinal case study at F-Secure corporation. Information and Software Technology 2014; 57:116-140.
13. Qurashi S. A. and Qureshi M. R. J. Scrum of scrums solution for large size teams using scrum methodology. Life Science Journal 2014; 11:443-449.
14. Paasivaara M., Lassenius C., and Heikkila V. T. Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrums really work?. Proc. Int. Conf. Empirical Software Engineering and Measurement (ESEM), Sweden, 2012.
15. Dorairaj S. and Noble J. Agile Software Development with Distributed Teams: Agility, Distribution and Trust. Proc. Int. Conf. Agile Conference (AGILE), Australia, 2013.
16. Pries-Heje L. P.-H. J. Why Scrum Works: A Case Study from an Agile Distributed Project in Denmark and India. Proc. Int. Conf. Agile Conference (AGILE), Salt Lake City, UT, 2011.
17. Alyahya S., Ivins W., and Gray W. Co-ordination Support for Managing Progress of Distributed Agile Projects. Proc. Global Software Engineering Workshop (ICGSEW), Helsinki, 2011.
18. Kamaruddin, N. K., Arshad N. H., and Mohamed A. Chaos issues on communication in Agile Global Software Development. Proc. Int. Conf. Business Engineering and Industrial Applications Colloquium (BEIAC), Kuala Lumpur, 2012.
19. Bass J. Agile Method Tailoring in Distributed Enterprises: Product Owner Teams. Proc. Int. Conf. Global Software Engineering (ICGSE), Bari, 2013.
20. Jalali S. and Wohlin C. Agile Practices in Global Software Engineering - A Systematic Map. Proc. Int. Conf. Global Software Engineering (ICGSE), Princeton, NJ, 2010.
21. Estler H.-C., Nordio M., Furia C., Meyer B. and Schneider J. Agile vs. Structured Distributed Software Development: A Case Study. Proc. Int. Conf. Global Software Engineering (ICGSE), Porto Alegre, 2012.
22. Al-Zaidi A., and Qureshi M. R. J. Global Software Development Geographical Distance Communication Challenges. International Arab Journal of Information Technology 2017; 14:215-222.

10/24/2020