

Agile Development Overcomes GSD Challenges: A Systematic Literature Review

Abdulaziz Alsahli¹, Hameed Khan² and Sultan Alyahya³

^{1,2,3} Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia

¹abdulazizalsahli9@gmail.com, ²hkhan@ksu.edu.sa, ³sualyahya@ksu.edu.sa

ABSTRACT

This paper aims to discover several Global Software Development (GSD) challenges existing in the literature and review several agile practices available to mitigate those challenges with special focus on Scrum and its practices since it is the most widely used in GSD setting. A total of 24 papers are identified as relevant papers and data from those papers are used to reveal how agile practices can be used to mitigate known GSD challenges. Thus, this paper can be a reference to GSD developers and agile practitioners to recognize latest agile remedies available to reduce GSD challenges.

Keywords: *Agile, Agile practices, GSD, Scrum, Systematic literature reviews.*

1. INTRODUCTION

In recent years, a new approach has emerged to develop software when software developers are located in different parts of the world. This type of development is now known to researchers and practitioners as global software development (GSD), and many firms are moving toward this type of development. GSD is currently a favorite approach, and it is the most recent trend in software development that does not recognize borders [1]. It has contributed tremendously to overall cost reduction and increased performance by accessing unavailable local expertise. GSD does have barriers both to developing organizations and even to individual developers. Challenges of GSD include communication, coordination and control, and they are caused by temporal, socio-cultural, and geographical distances [2]. The failure rate in software development using this environment is high due to inappropriate requirements management [3].

The agile development can be used to overcome many challenges that are faced in global software development [4, 5]. Certain agile practices when applied in the GSD environment can minimize

many GSD issues, e.g., challenges of big bang integration that frequently lead to failure can be minimized by incremental integration of work. Appropriate agile practices can also overcome communication and coordination challenges, e.g., challenges of communication can be overcome by daily scrum meetings [5]. The iterative nature of agile development can also play a supportive role in overcoming challenges of unclear and changing requirements during GSD.

This paper attempts to recognize several GSD challenges and systematically reviews the literature of using agile methods and practices to overcome these known GSD challenges with more focus on using Scrum practices. It can be used as a reference for researchers and practitioners willing to understand the how the agile development is used to overcome distributed development challenges. The aim is to provide them with a useful reference that they can use once they need to understand more about how agile practices can be used for mitigating GSD challenges as they will be find several known GSD challenges and several suggested mitigating agile practices that they can use. Also this research put more focus in Scrum as the most widespread agile method in GSD and mention several Scrum practices that can be used effectively in GSD challenge and identify several GSD challenges that they can mitigate. The search was on three popular databases IEEE Xplore, Springer Link and ACM Digital Library. A total of 24 papers were identified with publication date range from 2006 to 2016. Section 2 of the paper provides a background of GSD and agile development and the research motivation .Section 3 explains the research method used in this research. Section 4 presents the research results. Finally a conclusion and future work is presented in section 5.

2. BACKGROUND AND MOTIVATION

In recent years and due to globalization, which hugely benefited from ease of communication via the Internet, a new approach to software development now known as GSD has emerged that makes the best use of software development resources available elsewhere in world with geographically separated locations for development. The approach is a departure from developing software in-house; the new approach makes use of external expertise [6]. GSD is currently a favorite approach and many firms are moving toward this type of development and it is the most recent trend in software development that does not recognize borders. Herbsleb explains, “Globally-distributed projects are rapidly becoming the norm for large software systems” [7]. Björndal et al. explains that “GSD is a growing phenomenon in industry” [8]. Ågerfalk et al. concluded that “Organizations are increasingly moving to the global software development” [1]. Ågerfalk et al. also described “Global software development (GSD) is a phenomenon of increasing importance, given the perennial pressures of the need to remain profitable and competitive in the global landscape” [1].

GSD has contributed tremendously to overall cost reduction and increasing performance by accessing unavailable local expertise. Ågerfalk et al. and Conchúir et al. explain many benefits from development using this new approach. A project may benefit from human expertise not found locally. Moreover, the salaries of software development teams and operations expenses can be considerably lower elsewhere. For example, the cost of hiring a programmer in India cannot be compared with the cost of hiring one with the same qualifications and experience in the U.S.A. or Europe [1, 9]. Other benefits include gaining knowledge of specific market which result in a better competence, working with partners open the door for new opportunities, access to new resource might not available in-house [10].

It is also clear to many who have worked in or studied this setting that GSD does have barriers both to developing organizations and even to individual developers. Challenges of GSD include communication, coordination and control, and they are caused by temporal, socio-cultural, and geographical distances [2]. Khan et al. concluded that communication is the major challenge in GSD [11]. After researching different risks and the causes of each risk, Khan et al. explained, “GSD is not a simple task, and the organizations face

different challenges. However, Communication is a major issue, and it becomes more complicated during the process of requirements” [11]. Coordination is also a major challenge during GSD. Herbsleb explains, “it becomes clear that global distribution of a project seriously impairs critical coordination” [12]. Other challenges include initial team building and control breakdown [10].

Incremental development of software began as early as the 1950s but did not become popular until the late 1990s when Scrum and Extreme Programming were introduced, followed by the publication of the Agile Manifesto in 2001 [13]. Agile software development has emerged as a response to the need to minimize planning and heavy documentation as much as possible, which was not possible in the plan-driven software development. Serena suggests, “Agile software development isn’t a set of tools or a single methodology, but a philosophy” [14]. The main difference between agile and plan-driven methods is that plan-driven methods concentrate on sequential development (predictive), whereas agile is based in iterations (adaptive) and is flexible in response to requirements change [15]. Erdogmus conducted a survey to examine to what level the agile software development had come of age or not [16]. The respondents were 150 agile development specialists and about 65% of them believe that agile software development become mature.

There are various reasons why development organizations prefer agile software development, including the fact that its iterative style is good for unclear requirements and for providing flexibility in response to change. Moreover, agility keeps a development team motivated and productive because of milestones tests at the end of iterations every few weeks. Agile software development has proven its ability to produce quality software with minimum defects because of early detection of problems and more chances to deliver the software within time and budget if proper agile development is adopted [17]. During agile development, customers usually participate not only during requirements gathering and testing phases but in almost all phases which result in better customer satisfaction and it help to keep the customer expectations realistic [17]. Refining and improving the process continuously is also expected during agile development as teams usually hold a meeting at the end of each iteration every few weeks and the result from those meeting is an improved process as developer usually refine the process during those meeting [17].



Sriram and Mathew states that both agile methods and GSD were considered a huge improvement to software development in recent years and that, once combined, the two methods produce amazing results[18]. Jalali and Wohlin performed a systematic literature review to reveal how agile development and GSD are integrated successfully, what agile practices are used during GSD, and what modifications are needed to fit the distributed setting.[19]. Their aim is not to review how using agile development mitigates GSD challenges but to emphasize under which circumstances they can be successfully integrated. Hossain et al. reveal how Scrum can be tailored for effective use in GSD to minimize distributed challenges [20]. Sriram and Mathew claim that Scrum is the most widely used and most preferable agile method that suits GSD [21].

There are many agile methods and many agile practices in each method. Not all agile methods and practices are suitable to be used in GSD [22]. However, the literature reveals many agile practices can be used to mitigate some well known GSD challenges. The literature also lacks a study that comprehensively reviews current agile practices available and suitable to mitigate those GSD challenges. For the above reason this research is motivated and meant to investigate several challenges in GSD and explore the literature to find if any agile practices are recommended by researchers with special attention to most widely used Scrum practices.

3. RESEARCH METHOD

This paper uses a systematic literature reviewing to review the literature of using agile development to mitigate GSD challenges. It is a systematic way of extracting data from databases and reporting the results founded in relevant extracted papers [23]. Kitchenham proposed a comprehensive guidelines to conduct a systematic literature reviewing (SLR) [23]. Several activities in Kitchenham SLR like identification of review protocol, how the relevant studies are selected among other studies, how data is extracted from those relevant studies and how the result is reported are followed in this research [23]. Answering and more clarification of the following questions is the objective of this research: RQ1) is agile development useful in GSD? Which project size is suitable for it? Any modification needed?

RQ2) how can agile practices used to mitigate GSD challenges?

RQ2) is Scrum a useful agile method in GSD? What are the expected benefits from its use?

Only research available online and written in English is considered. Three well known databases are used to search for journal papers and conference proceeding. Those databases are IEEE explorer, Springer Link and ACM digital library. A total of 24 papers were identified as relevant and selected and analysis revealed their publication date range from 2006 to 2016. Keywords used for this research is divided into four categories and listed in figure 1 and a combination of each category is used in each search phrase. An example of a research phrase is as follows: *Agile practices mitigate global software engineering challenges.*

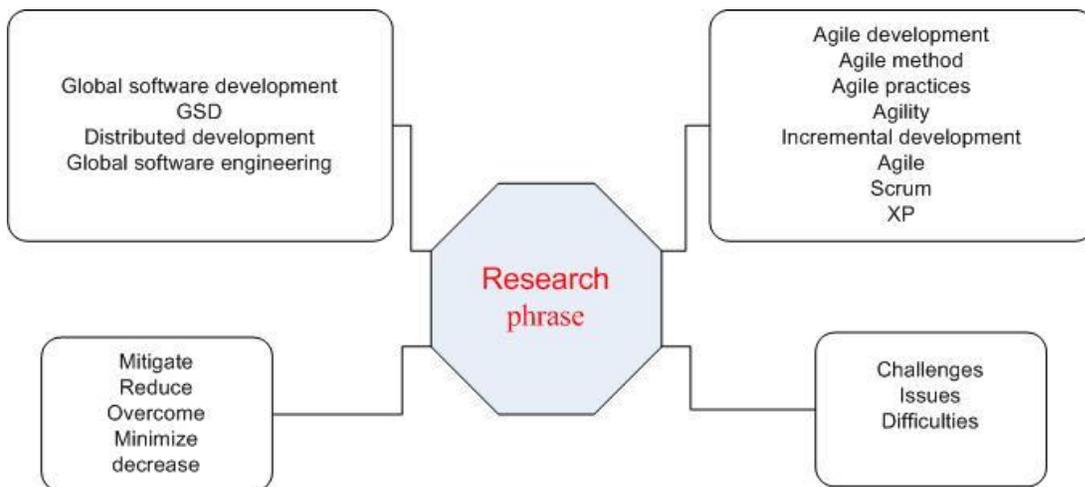


Fig. 1. Categories used in the research phrase

Four stages were used for selecting those 24 relevant papers. Stage one a total of 128 papers were retrieved as possibly related using their titles. However since titles of those papers not necessary representing a relevant content therefore in stage two 50 papers were excluded and remaining was 78 papers as possibly related. Stage three 30 papers were excluded based on irrelevant abstracts and remaining was 48 papers as possibly related. Final

stage 24 papers were excluded because they did not pass the quality assessment checklist as in Table 1. All the papers must pass the criteria 1,2,4 in order to be qualified as relevant papers. Criteria 3 indicate more paper's quality but passing it is not a condition in order to be qualified as relevant paper. Therefore finally, Twenty Four relevant papers remained and reviewed in this research.

Table 1: Quality Assessment checklist

Criteria#	Questions
1	Is a positive impact of using a certain agile method to overcome GSD challenges mentioned? Is a positive impact of using a certain agile practice to overcome GSD challenges mentioned? Is a positive impact of using Scrum to overcome GSD challenges mentioned and how?
2	Are the challenges of GSD clearly stated?
3	Did the introduction clearly define the research problem? Did it mention research questions?
4	Was the result clearly stated and described?

The types of the selected papers are as follows .One paper was a book chapter P1 [24]. Seven papers were journal articles P2 [4], P3 [25], P4 [26], P5 [22], P6 [27], P7 [28] and P8 [29]. Remaining papers were either conference papers P9 [5], P10 [19], P11 [20], P12 [21], P13 [30], P14 [31] , P15 [32], P16 [33], P17 [34], P18 [35], P19 [36], P20 [37], P21 [38], P22 [39], P23 [40] and P24 [41]. Table 2 shows the publication date of

relevant papers. Table 3 shows the focus of relevant papers. For each paper selected as relevant, data were extracted in systematic way using an identifier and extraction data for each paper. Each paper was classified according to their type either chapters in book, journal papers and conference papers. The aim and summery of each paper was extracted from each paper by the authors as well.

Table 2: Publication Date of Relevant Papers.

Year	2006	2008	2009	2010	2011	2012	2014	2016
Paper	P2 [4] P13 [30] P19 [36]	P15 [32] P16 [33] P22 [39]	P4 [26] P11 [20] P14 [31] P18 [35] P23 [40] P24 [41]	P3 [25] P5 [22] P10 [19]	P1 [24] P6 [27] P20 [37] P21 [38]	P12 [21] P17 [34]	P9 [5]	P7 [28] P8 [29]

Table 3: Focus of Relevant Papers.

Focus	Paper
Using Scrum to mitigate GSD Challenges	P5 [22], P12 [21], P6 [27], P21 [38], P20 [37], P22 [39], P23 [40], P14 [31], P11 [20], P17 [34], P18 [35], P24 [41], P8 [29]
Using of other agile methods and practices to Mitigate GSD challenges	P2 [4] , P9 [5], P13 [30], P15 [32], P16 [33], P3 [25], P10 [19], P19 [36], P4 [26], P1 [24], P7 [28]



4. RESULTS

As already explained, twenty four papers were selected as relevant papers that will be used for data extraction and analysis. This section presents the finding of the analysis of the full text of relevant papers. It is divided into three sub-sections based on the research questions presented earlier presented in section 3.

4.1 is agile development useful in GSD? Any modification or consideration needed? Which project size is suitable for it?

The agile development can be used to overcome many challenges that are faced in GSD [4, 5]. However agile software development by its nature depends heavily on communication, whereas GSD's largest challenge is communication [15, 42]. Paasivaara and Lassenius explained how challenges could be turned into benefits with successful application of agile GSD [30]. Young and Terashima claim that agile development will work incredibly well during GSD only if enough concentration is

dedicated to provide effective communication between remote team and explain their way of communication which resulted in successful agile development during GSD [32]. They further explain that agile development is proper and useful in GSD if enough emphasizing on software architecture that is suitable to all teams [32]. Jalali and Wohlin found that a modification and customization to agile practices in GSD is common in most of the studies he reviewed in order to effectively apply them reduce GSD challenges [19]. Hillegersberg et al. studied how agile methods were used for large projects during GSD in a company named Cordys over seven years [24]. The authors reported several GSD challenges and for each challenge how Cordys managed to mitigate it using suitable agile practices. The conclusion of their study is that agile development can be used to minimize GSD challenges not only for small and medium size projects but even for large projects [24]. Table 4 summarized the contribution of P1, P10 and P15 on the agile development suitability in GSD and whether any modification is needed and for which project size.

Table 4: Suitability of Agile Development in GSD

<i>Claim</i>	<i>Evidence</i>	<i>Based on results of</i>
<p>Agile will work in GSD only if enough emphasis on effective communication is considered.</p> <p>Agile will work in GSD only if software architecture is suitable to all teams from the beginning.</p>	P15 [32]	Studying how three teams in three different organizations adapt agile to overcome GSD challenges.
In most cases agile practices needed to be modified in order to be used effectively in GSD.	P10[19]	Conducting a systematic review on the use of agile practices in GSD.
Agile is useful not only in small and medium but also in large size projects in GSD.	P 1[24]	A case study conducted at Cordys which is a company that adopt agile method in large GSD projects.

4.2 how can agile practices used to mitigate GSD challenges?

Beecham et al. conducted a case study in a company using plan-driven GSD and found 17 challenges. The authors claimed that 13 of them could be solved by agile practices [3]. The authors concluded that many of the issues faced

during GSD can be solved using agile practices e.g. temporal distance during GSD can be reduced using XP pair programming [5]. The authors further explain that many challenges in GSD can be solved by related agile practices. For example, challenges of communication in GSD can be solved by daily stand-ups, and the challenge of requirements changes can be solved by the iterative nature of agile development [5]. Shrivastava and Date mentioned several agile practices used during

GSD and how agile practices can be used to solve known GSD challenges [25]. Therrien suggest that the use of a detailed written requirements as in agile development will improve the communication and provide a better understanding of product owner in GSD [33]. Alsahli et al. proposed an agile case-based repository to overcome the challenge of synchronizing the requirements change in software architecture in GSD [28]. They claim the use of agile practices made their approach iterative and more effective to solve the synchronizing challenge.

There are also many other reports of successful implementation of agile practices in global software development. Certain agile practices when applied in the GSD environment can minimize many GSD issues, e.g., challenges of big bang integration that frequently lead to failure can be minimized by incremental integration of work [30]. Paasivaara and Lassenius explain many benefits of using agile methods in GSD including those when using both XP and Scrum to solve GSD challenges [30]. The authors further explained that agile development provides many practices that overcome communication challenge of GSD [30]. Hazzan and Dubinsky argue that agile development can be used to deal with diversity which is inherited in any project developed in GSD [36]. Diversity is faced by distributed teams and includes cultural, management and gender diversity and therefore a suitable solution is needed and it is a key to success during GSD. Diversity can be enhanced by several agile practices according to the authors. Agile development creates an environment in which both male and female can communicate in a similar way. Management diversity is enhanced by agile development

particularly extreme programming where a mixture of people work together to develop software in an efficient way as the roles and responsibilities in agile teams are well defined. The identified different roles in agile development allow entire team to view the process of development in different perspectives during GSD. Opinion diversity is also foster by agile short cycles as those short cycles provide a continuous opportunity to developers to improve share their understating during planning game, daily stand-ups and previews meeting [36].

According to Hossain et al. agile practices can play an important role in minimizing coordination risks and improve coordination process during GSD [26]. The authors considered Mintzberg's coordination framework which contain three mechanisms standardization, direct supervision and mutual adjustment to identify the impact of GSD challenges on those coordination mechanisms. Standardization was hugely impacted by GSD distances but agile practices like sprint planning meeting, retrospective meeting and refactoring helped distributed developers to preserve standards during GSD. Direct supervision was also effected by GSD distances but the impact of those distances in direct supervision can be minimized by using agile practices like daily stand up meeting and sprint review meeting. Finally Mutual Adjustment was also affected by GSD distances ,however, agile practices like daily stand up, sprint planning, sprint review meetings were very helpful in reducing challenges caused by distances on mutual adjustment [26]. Table 5 summarized several GSD challenges and suggested agile practices to mitigate them.

Table 5: Agile Practices Mitigate GSD Challenges.

<i>GSD Challenge</i>	<i>Suitable agile practices</i>	<i>Evidence</i>	<i>Based on results of</i>
Communication	Daily stand ups Sprint reviews Requirements in Backlog Short iterations Frequent builds	P9 [5] P3 [25] P16 [33] P13[30] P13[30]	A conducted case study Reviewing the literature A conducted case study Reviewing the literature Reviewing the literature
Managing requirements change	Iterations	P9 [5]	A conducted case study
Configuration management issues	Continuous integration	P3 [25]	Reviewing the literature

Synchronizing requirements change in software architecture	Agile CBR	P7 [28]	Expert reviews and a conducted case study
Trust	Frequent meetings	P13[30]	Reviewing the literature
Big bang integration	incremental integration	P13[30]	Reviewing the literature
Misunderstandings	Continuous integration	P13[30]	Reviewing the literature
Management diversity	Pair programming	P19[36]	Studying distributed computer science students playing distributed developers roles and using agile to overcome associate GSD diversity challenges
Opinion diversity	Short Cycles	P19[36]	Studying distributed computer science students playing distributed developers roles and using agile to overcome associate GSD diversity challenges
Coordination includes A) Standardization	sprint planning meeting, retrospective meeting and refactoring reduces standardization issues	P4 [26]	conducting an industry based case study
B) Direct supervision	daily stand up meeting and sprint review meeting reduces direction supervision challenges	P4 [26]	conducting an industry based case study
C) Mutual Adjustment	daily stand up, sprint planning, sprint review meetings were very helpful in reducing mutual adjustment challenges	P4 [26]	conducting an industry based case study

4.3 is Scrum a useful agile method in GSD? What are the expected benefits from its use?

We previously discussed how several agile practices can be used in GSD to overcome its challenges. In this section we will explore more of using of Scrum in GSD since it is the most suitable and widespread agile method to be used in distributed setting according to [22]. Paasivaara and Lassenius argue that Scrum is a very suitable agile method for application in GSD since it offers a variety of practices that are proper to use in the context of globally distributed software development [22]. Sriram and Mathew also claim that Scrum is the most widely used and most preferable agile method that suits GSD [21].

Hossain et al. revealed that among several agile practices available, Scrum practices are the most adopted agile method in GSD [27]. Hossain et al. reported successful application and an increase interest in applying Scrum in GSD due to its many advantages it offers to distributed software development [38]. Hossain et al. explain that Scrum is a flexible agile method that offers an opportunity to project stakeholders to tailor Scrum and modify certain Scrum practices according to their needs. They further explained how seven Scrum practices are used in GSD context in several projects to overcome GSD challenges. They finally concluded that tailored Scrum allowed several companies to better control distributed software

development and improve communication between distributed teams [38]. Paasivaara et al. claim that Scrum can be useful in GSD not only in small and medium size projects but also can be applied to large projects and they explained how Scrum practices can be used in larger distributed software development projects [37, 39]. Khan et al. studied several Malaysian companies struggling to cope with GSD challenges and found that majority of them abandoned their distributed waterfall model and start to use agile development specially Scrum in distributed development setting [29]. The authors revealed that 70% of Malaysian companies are now using Scrum to mitigate GSD challenges. One of the senior managers in one of those companies explains “We use SCRUM, because it was the first alternative to an informal process, and because we know SCRUM is better than other models” [29]. Paasivaara et al. claim that with Scrum trust between team members can be improved, motivation of the teams’ members can be much better and communication will be much easier [39]. Sutherland et al. conducted an empirical study and concluded that Scrum mitigate several GSD challenges as communication and coordination is notably enhanced by using Scrum and frequent and early of delivery of software with Scrum can solve several issues including misunderstanding [40]. Hossain et al. reviewed the literature on using

Scrum practices in global software development and suggested strategies to address associated challenges [31]. Hossain et al. developed a framework to deals with known GSD challenges by using Scrum practices. Their framework is a reference to researchers interesting in understanding how Scrum can be optimally used in GSD [27]. They reveal how Scrum can be tailored for effective use in GSD to minimize distributed challenges [20].

Bannerman et al. examined how Scrum offer unique practices that can be used to reduce coordination challenge caused by the temporal, geographical and cultural distances during GSD [34]. The authors further explained how seven Scrum practices used in different case studies to mitigate GSD coordination challenges [34]. Paasivaara et al. mentioned several Scrum practices and how can they best used in GSD setting and their potential benefits to minimize GSD challenges [35]. According to the authors, practices to support best application of Scrum in GSD such as frequent visit to other sites and an ambassador to other sites are recommended [35]. Table 6 list several Scrum practices and their suitability to minimize several GSD challenges. Figure 2 shows the frequency of Scrum practices in 13 relevant papers.

Table 6: Scrum practices mitigate GSD challenges.

Scrum Practices	Mitigate Which Challenge	Evidence	Base on results of
daily Scrums	Coordination, communication and integration.	P5 [22]	Multiple-case study of four distributed project using Scrum.
	Productivity	P12[21]	Reviewing the literature
	Trust and transparency	P6 [27]	A developed framework based on literature
	Participation	P21 [38]	A multiple case study
	Communication, coordination and project management	P22 [39]	A case study, expert reviews
	Communication , coordination and integration	P8 [29]	Interviews with experts in 10 Malaysian companies
	Focus on customer priorities	P23 [40]	Conducting a case study
	Coordination	P17[34]	Examination of four case studies
Coordination and communication	P18[35]	A multiple case study	
	Coordination, time distance and	P5 [22]	Multiple-case study of four distributed project using Scrum

Scrum-of-Scrums	cultural distance.	P6 [27]	A developed framework based on literature
	Trust		
	Coordination	P17[34]	Examination of four case studies
Sprints	Transparency, misunderstanding and feedback	P5 [22]	Multiple-case study of four distributed project using Scrum
	Quality issues	P12[21]	Reviewing the literature
	Trust	P6 [27]	A developed framework based on literature
	Sense of teamness	P21 [38]	A multiple case study
	Coordination	P17[34]	Examination of four case studies
Sprint Planning Meetings	Visibility and cohesion	P5 [22]	Multiple-case study of four distributed project using Scrum
	Sense of teamness	P6 [27]	A developed framework based on literature
	Coordination	P17[34]	Examination of four case studies
Sprint Demos	Visibility and feedback	P5 [22]	Multiple-case study of four distributed project using Scrum
	Transparency	P6 [27]	A developed framework based on literature
Retrospective Meetings	Reflection and commitment.	P5 [22]	Multiple-case study of four distributed project using Scrum
	Trust	P6 [27]	A developed framework based on literature
	Maintain standards	P14[31]	Reviewing the literature
	Coordination	P17[34]	Examination of four case studies
Backlogs	Task management and progress monitoring	P5 [22]	Multiple-case study of four distributed project using Scrum.
	Informal contact	P6 [27]	A developed framework based on literature
	Control	P21 [38]	A multiple case study
	Progress visibility	P22 [39]	A case study, expert reviews
	Coordination	P17[34]	Examination of four case studies
Frequent Deliveries	Trust, Coordination and cultural difference	P12[21]	Reviewing the literature
Sprint Review	Coordination	P17[34]	Examination of four case studies

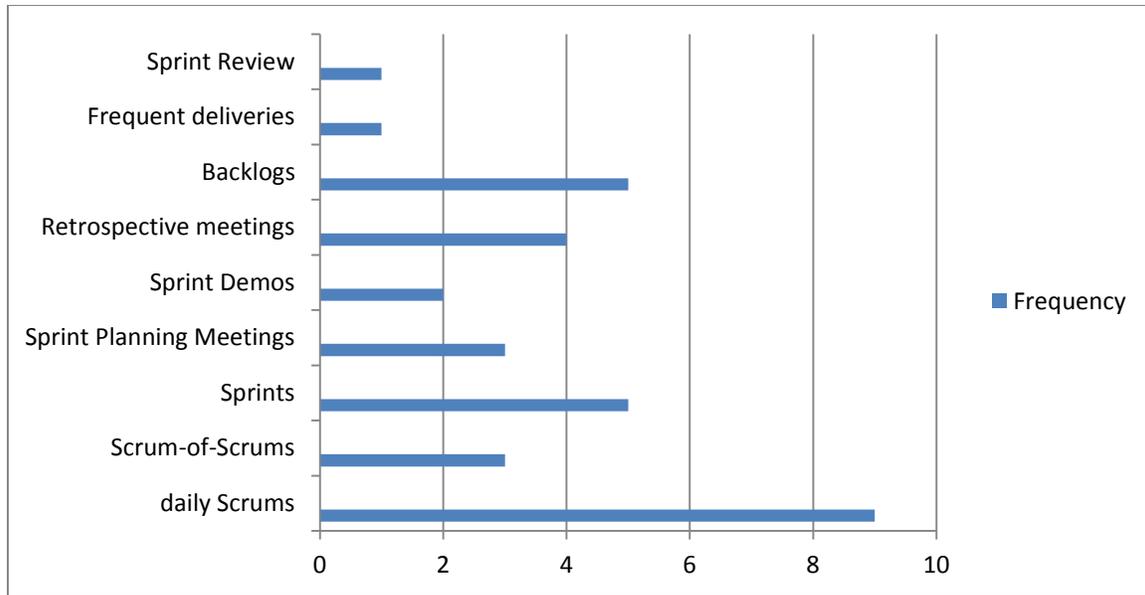


Fig. 2. Frequencies of Scrum Practices.

It is clear from the literature that scrum practices such as daily scrums, sprints and backlogs are the most important and favorable Scrum practice in GSD [41]. One of the most useful Scrum practice is daily Scrum meeting which usually last around 15 minutes where developer answers what have been done and explaining any obstacles faced and what they will do next [22]. Daily Scrum meeting can be arranged using video conferencing while the teams are geographically distributed in some overlapping time and using web camera will make this meeting more natural and give it a collocated feel. Daily Scrum meeting clearly minimize coordination challenge which is a big challenge in GSD and it also support team member to communicate and it allows teams to understand what is happening in other sites. When the number of distributed teams is more than three teams then it is more practical to allow each team to have their own Scrum meeting and then use Scrum-of-Scrums to allow the information to be share between teams regarding of what each team is doing [22]. Only one member from each Scrum team is involved in Scrum-of-Scrums meeting which is usually a weekly meeting that lasts about half an hour and the three scrum questions are answered. Scrum-of-Scrums in general is useful to coordinate several teams [22].

Other useful Scrum practices in GSD include Sprints, Sprint planning meetings and backlogs. Software in Scrum is developed in several iterations named Sprints and each sprint last usually between two weeks and one month. If there are several scrum teams then it is recommended that teams start and end sprint synchronously. Sprints allow teams to work in iterations with expected outcome and fixed deadline for each iteration. Sprints allow teams to feel that they are in a

single team and work for a collective goal. Sprint planning meeting is a meeting that is held before each sprint where several items from product backlog is selected divided into tasks and then teams collectively plan the sprint. Sprint planning meeting allows distributed members to know each other well and participate before actual start of sprint allowing a better understanding and also it assigns tasks to distributed members so that everyone knows their responsibilities. List of requirements and features to be developed are stored in product backlog. It is responsibility of product owner for priorities items in product backlog. Features in product backlog is broken into task and divided into several sprint backlogs. Product and sprint backlogs are useful to keep an eye on progress in GSD [22].

5. CONCLUSION AND FUTURE WORK

GSD has emerged to make the best use of resources available elsewhere in the world with geographically separated locations for development. It is currently a favorite approach and many firms are moving toward this type of development. It also has contributed tremendously to overall cost reduction and increasing performance by accessing unavailable local expertise. However it is also clear to many who have worked in or studied this setting that GSD does have barriers both to developing organizations and even to individual developers. Challenges of GSD include communication, coordination and control, and they are caused by temporal, socio-cultural, and geographical distances. This research presents a systematic literature reviewing on how agile development and practices can contribute

to mitigating those known GSD challenges by identifying several GSD challenges and exploring several agile practices that can help to minimizing them. Special attention was paid to Scrum and its practices as it is the most widely used method in GSD setting. The result of this systematic literature reviewing provides a useful reference to researchers and practitioners to recognize latest agile remedies available to reduce GSD challenges by identifying several known GSD challenges from the literature and then listing several suitable agile practices that they can use. Also this research offers researchers and practitioners willing to understand how Scrum, the most adopted agile method in GSD, a reference by allowing them to find several Scrum practices and list of GSD challenges that can be minimized by using them.

Although the literature reveals a good progress is made in understanding many benefits of using agile development in GSD setting to minimize its associated challenges but still exists many research questions that need further research. Further examination of agile practices in GSD by trying to classify them according to their effectiveness in minimizing GSD challenges either very effective, effective, partially effective or not effective is needed. This will allow a better understanding of their quality and effectiveness in mitigating GSD challenges. Other research area that requires further investigation in the future is to investigate how each agile practice effectively applied in GSD setting to minimize its associate challenges.

REFERENCES

- [1] Ågerfalk, P.J., et al., Benefits of Global Software Development: The Known and Unknown. Springer, 2008.
- [2] Holmstrom, H., et al., Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance in International Conference on Global Software Engineering. 2006, IEEE.
- [3] Lai, R. and N. Ali, A Requirements Management Method for Global Software Development. AIS: Advances in Information Sciences, 2013.
- [4] Holmström, H., et al., Agile Practices Reduce Distance in Global Software Development. Information System Management, 2006. 23(3).
- [5] Beecham, S., J. Noll, and I. Richardson, Using Agile Practices to Solve Global Software Development Problems -- A Case Study in International Conference on Global Software Engineering Workshops 2014, IEEE
- [6] Vanzin, M., et al., Global Software Processes Definition in a Distributed Environment 2005.
- [7] Herbsleb, J.D., Global Software Engineering: The Future of Socio-technical Coordination, in 2007 Future of Software Engineering. 2007, IEEE Computer Society. p. 188-198.
- [8] Björndal, P., K. Smiley, and P. Mohapatra, Global Software Project Management: A Case Study. Springer, 2010.
- [9] Conchúir, E.Ó., et al., Global Software Development: Never Mind the Problems – Are There Really Any Benefits?, in 29th Information Systems Research Seminar in Scandinavia. 2006.
- [10] Parviainen, P., Global Software Engineering Challenges and Solutions Framework. 2012.
- [11] Khan, A.A., S. Basri, and P.D.D. Dominic, Communication Risks in GSD During RCM: Results from SLR, in International Conference on Computer and Information Sciences 2014, IEEE.
- [12] Herbsleb, J.D., Global Software Engineering: The Future of Socio-technical Coordination, in Future of Software Engineering. 2007, IEEE Computer Society.
- [13] Wikipedia. Agile Software Development. 2015 26/07/2015]; Available from: https://en.wikipedia.org/wiki/Agile_software_development.
- [14] Serena, An Introduction to Agile Software Development. 2007.
- [15] Jammalamadaka, K. and V.R. Krishna, Agile Software Development And Challenges. International Journal of Research in Engineering and Technology, 2013.
- [16] Erdogmus, H., Agile's Coming of Age ...or Not. IEEE Software, 2007.
- [17] Knowledge, G., 12 Advantages of Agile Software Development. 1-800-Courses, 2011.
- [18] Sriram, R. and S.K. Mathew, Global Software Development Using Agile Methodologies: A Review of Literature in International Conference on Management of Innovation and Technology 2012 IEEE
- [19] Jalali, S. and C. Wohlin, Agile Practices in Global Software Engineering - A Systematic Map in 5th IEEE International Conference on Global Software Engineering 2010.
- [20] Hossain, E., P.L. Bannerman, and R. Jeffery, Towards an Understanding of Tailoring Scrum in Global Software Development: A Multi-case Study. Proceedings of the 2011 International Conference on Software and Systems Process 2011.
- [21] R.Sriram and S.K. Mathew, Global Software Development using Agile Methodologies: A Review of Literature, in International Conference on Management of Innovation and Technology 2012, IEEE.
- [22] Paasivaara, M. and C. Lassenius, Using Scrum Practices in GSD Projects, in Agility Across Time and Space. 2010, Springer-Verlag.
- [23] Kitchenham, B., Procedures for Performing Systematic Reviews. 2004.
- [24] Hillegersberg, J.v., G. Ligtenberg, and M.N. Aydin, Getting Agile Methods to Work for Cordys Global Software Product Development, in New Studies in Global IT and Business Service Outsourcing. 2011, Springer Berlin Heidelberg. p. 133-152.
- [25] Shrivastava, S.V. and H. Date, Distributed Agile Software Development: A Review. Journal of Computer Science and Engineering, 2010.



- [26] Hossain, E., M.A. Babar, and J. Verner, How Can Agile Practices Minimize Global Software Development Co-ordination Risks? Springer-Verlag Berlin Heidelberg 2009.
- [27] Hossain, E., P.L. Bannerman, and D.R. Jeffery, Scrum Practices in Global Software Development: A Research Framework. Springer-Verlag, 2011.
- [28] Alsahli, A., H. Khan, and S. Alyahya, Toward an Agile Approach to Managing the Effect of Requirements on Software Architecture during Global Software Development. Scientific Programming, 2016.
- [29] Khan, A.R., R. Akbar, and D.W.H. Ten, A Study on Global Software Development (GSD) and Software Development Processes in Malaysian Software Companies. Journal of Telecommunication, Electronic and Computer Engineering, 2016. 8(2).
- [30] Paasivaara, M. and C. Lassenius, Could Global Software Development Benefit from Agile Methods?, in IEEE International Conference on Global Software Engineering 2006.
- [31] Hossain, E., M.A. Babar, and H.-y. Paik, Using Scrum in Global Software Development: A Systematic Literature Review in Fourth IEEE International Conference on Global Software Engineering. 2009.
- [32] Young, C. and H. Terashima, How Did We Adapt Agile Processes to Our Distributed Development?, in Agile Conference. 2008
- [33] Therrien, E., Overcoming the Challenges of Building a Distributed Agile Organization, in Agile Conference. 2008.
- [34] Bannerman, P.L., E. Hossain, and R. Jeffery, Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?, in International Conference on System Sciences. 2012.
- [35] Paasivaara, M., S. Durasiewicz, and C. Lassenius, Using Scrum in Distributed Agile Development: A Multiple Case Study, in Fourth International Conference on Global Software Engineering. 2009, IEEE
- [36] Hazzan, O. and Y. Dubinsky. Can Diversity in Global Software Development be Enhanced by Agile Software Development. in Proceedings of the international workshop on Global software development for the practitioner. 2006.
- [37] Paasivaara, M. and C. Lassenius, Scaling Scrum in a Large Distributed Project, in International Symposium on Empirical Software Engineering and Measurement. 2011.
- [38] Hossain, E., P.L. Bannerman, and R. Jeffery, Towards an Understanding of Tailoring Scrum in Global Software Development: A Multi-case Study. Proceedings of the 2011 International Conference on Software and Systems Process 2011.
- [39] Paasivaara, M., S. Durasiewicz, and C. Lassenius, Distributed Agile Development: Using Scrum in a Large Project, in IEEE International Conference on Global Software Engineering. 2008.
- [40] Sutherland, J., G. Schoonheim, and M. Rijk. Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams. in Proceedings of the 42nd Hawaii International Conference on System Sciences. 2009.
- [41] Hossain, E., et al., Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework. 16th Asia-Pacific Software Engineering Conference, 2009.
- [42] Kamaruddin, N.K., N.H. Arshad, and A. Mohamed, Chaos Issues on Communication in Agile Global Software Development in Engineering & Industrial Applications Colloquium 2012.