GLOBAL SOFTWARE DEVELOPMENT: A REVIEW OF ITS PRACTICES

Naveed Ali[1] and Richard Lai[2*]

[1,2]Department of Computer Science and Information Technology, La Trobe University
Melbourne, VIC 3083, Australia

Email: naveedz.ali@gmail.com[1], r.lai@latrobe.edu.au[2*] (corresponding author)

DOI: https://doi.org/10.22452/mjcs.vol34no1.5

ABSTRACT

Global Software Development (GSD) is multi-site software development with software teams scattered across different places around the world. To gain the benefits of the lower cost of software development and access to international talents, many organizations are using GSD. However, earlier studies reported that achieving these benefits can be difficult and GSD involves risks. This paper presents a systematic review of GSD and aim to provide the findings about the current practices, benefits, associated risks, and difficulties associated with it.The review was conducted in accordance with the systematic review procedures and processes defined by Kitchenham. By following these guidelines, six digital libraries to gather information on the prevalent trends and practices in GSD were searched. The outcomes presented in this study are based on 204 studies, published in peer-reviewed conferences and journals. Our findings will enable readers to understand the current practices, benefits, risks and difficulties associated with GSD. As a result, they can form realistic expectations before making a decision to engage in GSD or not, and formulate better pre-contract and post-contract planning in order to increase the chances of project success. A merit of this paper as compared to previous survey papers on GSD is that it reports the issues related to cultural diversity, requirements engineering and software architecture.

Keywords: ***Global software development, distributed teams, systematic review***

1.0 INTRODUCTION

To improve profits and to respond promptly to customers' requirements, Information Technology (IT) organizations started looking for solutions across the world. As a result, the practice of software development became global in the 90s. Global Software Development (GSD) is multi-site software development with software teams scattered across different places around the world [208, 220, 221]. The number of organizations engaged in GSD is increasing and a considerably large amount of funds has been poured into it [235, 239].

Despite the potential benefits of GSD, it has introduced additional challenges for stakeholders which are either not present in the collocated development of software projects [218, 217, 226, 264]. In the distributed environment, lingual, social, geographical, cultural and temporal differences between software development teams create challenges for stakeholders, and greatly impact the software development process. As a result, the development and maintenance of software systems through collaboration with partners in different geographical locations is a challenge for organizations [220, 248, 259].

Apart from expanding trends in the IT sector, GSD has increasingly become a subject of research [206]. Despite the popularity of GSD, this area is also attracting attention due to the difficulties and challenges associated with it. Current state-of-the-art practices are still evolving, hence findings reported in empirical studies remain controversial [258]. The majority of existing research is investigative in nature and focuses more on the identification of issues, rather than defining solutions for them. Consequently, the amount of empirical evidence reporting success stories in relation to GSD practices is not significant [209, 211]. To date, no formula has been proposed for determining the success of distributed software development projects. Moreover, due to the diverse nature and scope of these projects, it is difficult to apply the lessons learned from one project to another [258]. It is therefore necessary for researchers to present their research findings in a clear way, which can be beneficial for practitioners in improving GSD projects [213, 258].

As globalization is becoming a prominent aspect of everyday life, organizations are expanding their practices and approaches in globally distributed environments. Due to a lack of empirical evidence and expertise required to work in distributed environments, organizations are experimenting with different ways to realise the benefits of GSD. There are 26 review papers published; 8 on Agile practices, 5 on communication and coordination challenges, 4 on process models, 2 on management related issues, 1 on effort estimation, 2 on knowledge transfer and management, 1 on state-of-the-art practices, and 3 on general advices on GSD. The findings of these papers are being relatively older (refer to Figure 1); and also, the challenges related to cultural diversity, requirements engineering and software architecture have not been covered so far in the existing studies. To conduct a systematic review on the current state-of-the-art practices and the challenges in GSD is thus useful; including cultural diversity, requirements engineering and software architecture.

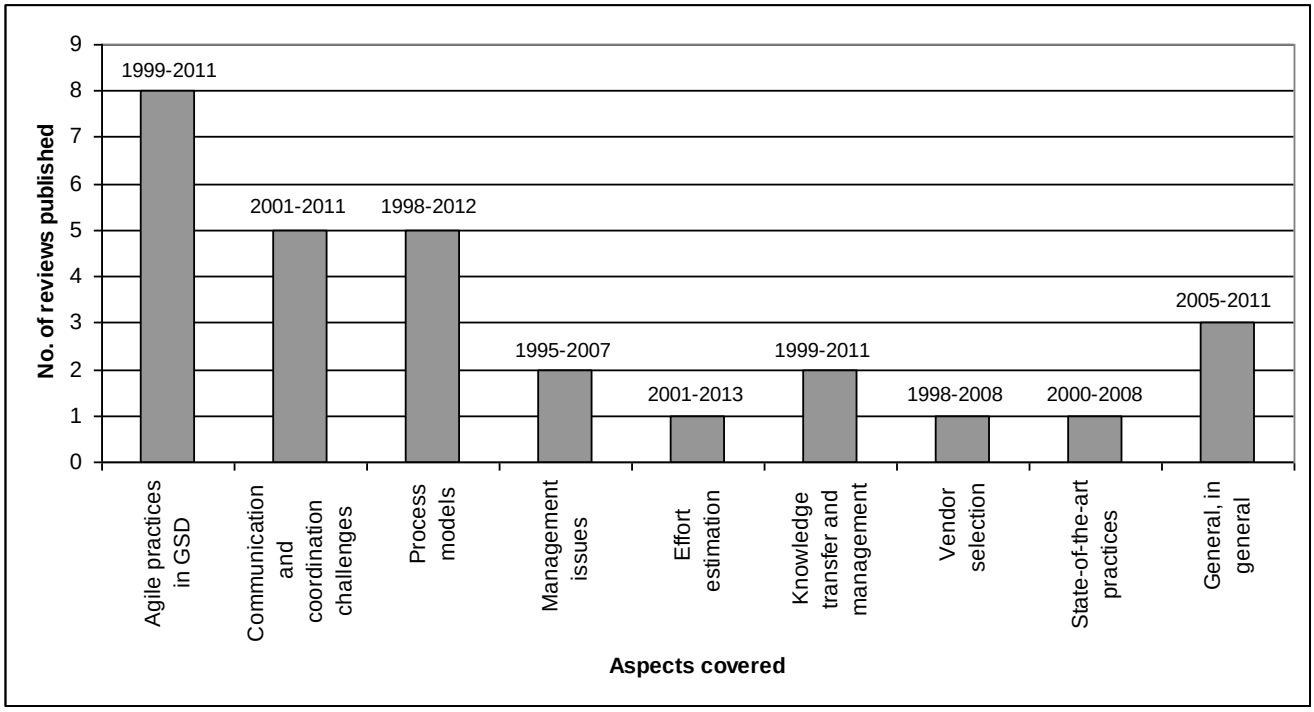

Fig. 1: List of reviews published on GSD

It is anticipated that the results of our review will enable readers to be better informed about the benefits, risks, problems, and difficulties associated with GSD, so that they can better manage GSD projects and make appropriate decisions in a timely fashion.

2.0 SYSTEMATIC REVIEW PROCEDURE

The systematic review is divided into seven stages: defining the research questions; data searching; data selection, data extraction; data synthesis; first draft of outcomes; and final draft of outcomes. In the first five stages, the preliminary stages of systematic review were conducted, and reported the outcomes in the form of a paper in the last two stages. The review was conducted in accordance with the systematic review procedures and processes defined by Kitchenham [238]. In the following, the first five stages are described below in detail.

2.1 Defining research questions

To gain an understanding of the trends, practices, benefits and challenges in GSD, two research questions were formulated.

RQ 1: How GSD is practised by organisations? To answer this question, the following aspects of GSD describing their relation to the organisations which practise it were investigated (please refer to Table 1).

Table 1: List of GSD aspects to be investigated in relation to organisations practising it

Aspects	**Section #**	**Title**
Why do organizations want to engage in GSD?	3.1.1	Motivation behind GSD
Which types of organizational setups are used in GSD?	3.1.2	Types of organizational setups involved
What are the geographical locations of the development sites?	3.1.3	Geographical locations of development sites
What percentage of reported GSD cases are successful and what percentage are failures?	3.1.4	Successful cases of GSD

RQ 2: What are some of the challenges faced by GSD practitioners?
The following aspects of software development which present challenges to practitioners when they develop a GSD project were investigated (please refer to Table 2).

Table 2: List of challenges to be investigated faced by GSD practitioners

Challenges	**Section #**	**Title**
What are the issues which can create difficulties in communication for stakeholders?	3.2.1	Communication
What are the issues in cross-cultural setups?	3.2.2	Culture
What are the factors which can create challenges for stakeholders while implementing software processes?	3.2.3	Software processes
What are the factors which can create challenges for stakeholders while performing requirements engineering?	3.2.4	Requirements engineering
What are the factors which can impact the process of effort estimation in GSD?	3.2.5	Effort estimation
What are the factors which can hamper the process of knowledge sharing and management between remote teams?	3.2.6	knowledge sharing and knowledge management
Which factors are important in devising a software architecture for GSD projects?	3.2.7	Software architecture

2.2 Data searching

To answer the research questions, a detailed search in four steps was undertaken. In step 1, the fundamental areas to finalize the scope of the systematic review were finalised. In step 2, for each of the defined areas, a set of keywords was selected. Although there are many keywords available which can be used for GSD, the search was limited to six keywords (refer to Table 3). In step 3, based on the information from the first two steps, the search expression is: {Expression = (A_1 OR A_2 OR A_3 OR A_4 OR A_5 OR A_6) AND (B_1 OR B_2 OR B_3 OR B_4 OR B_5)}. Finally, in step 4, the search expression in five different digital libraries for data searching purposes was used.

Table 3: Categories and keywords

Category	Area	Keywords / Strings
A	Global Software Development	A_1 - Global Software Development
		A_2 - Global Software Engineering
		A_3 - Distributed Software Development
		A_4 - Distributed Software Engineering
		A_5 - Multi-site Software Development
		A_6 - Multi-site Software Engineering
B	Nature of study	B_1 – Case study
		B_2 – Experiments
		B_3 – Surveys
		B_4 – Industrial
		B_5 – Literature reviews

The concept of GSD was introduced in the late 90s, but gained momentum after 2000 as a result of globalization, hence GSD is viewed as a trend of the 21st century [222]. Therefore, the research work and contributions to GSD over the last 15 years only were covered . Details on the selected digital libraries are as follows:

Research repositories: IEEE, ACM, Science Direct (Elsevier), Wiley Inter Science, ISI (Web of Knowledge), Springer
Paper types: Research papers and journal articles
Publication date: Published between 2000 – 2014
Language: English

2.3 Study selection

Once the research questions and data searching mechanism were defined, the process of study selection, selecting studies, which fell under the defined scope and keywords of the review process, was started. As shown in category A of Table 3, the area of GSD is still evolving and incorporates a wide array of terminology. As a result, our search retrieved hundreds of research papers and studies. After screening these papers, it is concluded that approximately 27% of papers were relevant to this study, thus, adopting a high precision searching mechanism, as described by [219]. The overall selection process comprised five steps, as shown in Table 4.

Table 4: Study selection process

Analysis Phase	Inclusion Criteria	Number of Papers
1. initial search	• Papers written in English • Published after 2000 (i.e. b/w 2000-2014) • Available online • Contain search keywords and strings	751
2. Scrutinizing titles	• Only published in journals, conferences, workshops and books • Not an editorial, seminar, tutorial or discussion	590
3. Scrutinizing abstract	• Experiments, case studies, literature reviews, industrial and surveys	380
4. Analyzing introduction and conclusion	• Main contribution in the areas of search strings	300
5. Analyzing main contribution	• Reported significant contribution • Originality of work • Sole focus related to the theme of this review study	204[1]

[1] Refer to Appendix 1

At this stage, it is important to note that papers were not selected on the basis of their impact factor, because it was assumed that the quality of the selected material had already been evaluated by the peer review process of the journals and conferences. Moreover, several papers appeared in more than one research repository. The repetitions were eliminated and only one instance of a paper was considered. There were 50-55 cases of such repetitions of a paper, all of which were eliminated. Details about repetition do not provide any significant piece of information, except the names of articles which are published by more than one publishing authority (e.g. IEEE, ACM). As a result, the names of repetitive articles, which were found during the study selection process, were not mentioned.

2.4 Data extraction

After completing the study selection process, the basic information on each paper in data extraction form (refer to Table 5) to gather information on the motivation behind GSD, the geographical locations of development sites, the mode of collaboration between sites, success and failure stories, the focus of the study (i.e. experiment, case studies, literature reviews, and simulation and survey), the challenges associated with GSD practices, the consequences of the identified challenges, the proposed solutions to minimize the effect of the challenges, and future recommendations were recorded. The non-experimental models which presented a proposal without conducting experiments were also applied.

Table 5: Data extraction process

Aspects	Details
Study ID	Paper id
Title	Title of paper
Authors	Names of authors
Publishers	Name of publishing authority
Publishing date	Date of publication
Motivating factors	Motivation behind GSD
Geographical locations	Locations of development sites
Mode of collaboration	Type of organizational setup
Success and failure stories	Evidence of success and failure outcomes
Study focus	Focus and perspective of paper
Reported challenges and consequences	Challenges and consequences associated with GSD practices
Study findings	Lessons learned from paper
Future recommendations	Implications for future work

2.5 Data synthesis

The extracted data were analysed using both qualitative and quantitative methods to answer our research questions. To perform quantitative analysis, the Descriptive Statistics was used to develop bar and line charts. For qualitative analysis, the Constant comparison method [242] was used to present the findings on the focus of the studies, the proposed methods, shared best practices, the categorization of the reported evidence into success and failure stories, and the lessons learned from the studies [205].

3.0 RESULTS AND DISCUSSIONS

This section presents the findings of our systematic review for our two research questions

3.1 How GSD is practised by organisations? (RQ1)

3.1.1 Motivation behind GSD

To gain the benefits of lower-priced software development and access to international talent, GSD is a feasible solution for many organizations [214, 248]. Earlier studies reported that realising the benefits of GSD is often a difficult task

[226, 218, 214, 264]. Despite the challenges involved in globally distributed software development, the popularity of GSD is increasing. It is therefore necessary to identify the set of underlying factors which contribute to the growth of GSD, hence, the literature identified nine factors (refer to Figure 2). The results in Figure 2 show that 61 of the 204 studies reported details on motivating factors; 40 reporting cost as the primary reason for organizations becoming involved in GSD. Cost reduction strategies were also mentioned by [258] as the main motive behind commencing globally distributed software development. Following cost, the other factors which were identified as a result of the literature search are as follows: delivery time; merger, acquisition and alliance; broader knowledge, quality and flexibility; business advantages; low turnover and reduced cycle time; competition; and innovation and shared best practices.

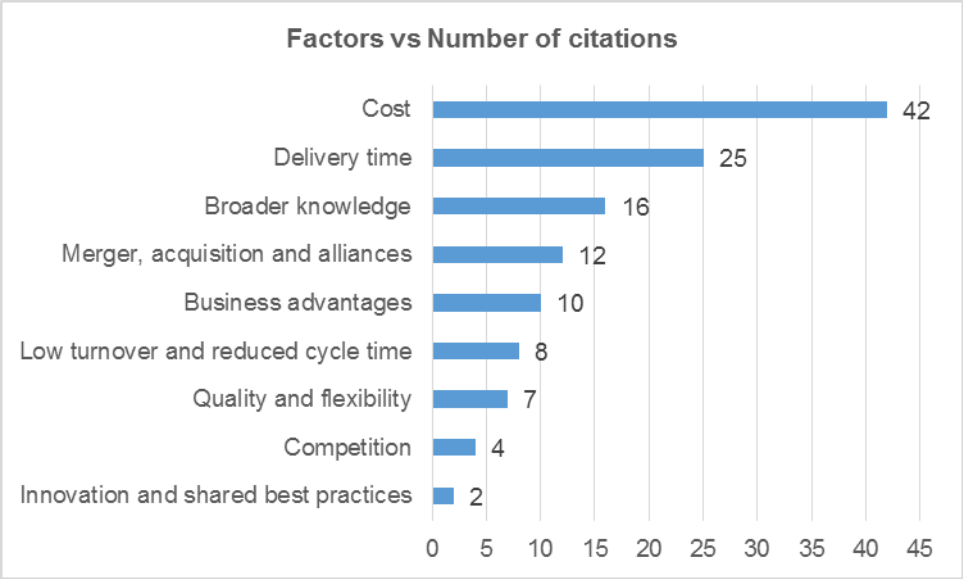

Fig. 2: Motivating factors behind GSD
(Factors identified from 61 out of 204 studies)

The results shown in Figure 2 provide insight into the identified motivating factors which has resulted in GSD being implemented in many organizations. However, it is not possible to extract data on these on a year-by-year basis. Therefore, another search was undertaken to classify the information in Figure 2 with respect to the years in which they are mentioned in the selected pool of studies. As shown in Figure 3, between 2000 and 2005, the number of studies which reported the different motivations for implementing GSD is quite low. However, with the increased awareness of GSD, there was a corresponding increase in the frequency of cited motivations in the literature. Even though the number of citations is not high, they are sufficient to show that the considerations of cost, delivery time and broader knowledge are no longer the only reasons for organizations to go global. In addition to these factors, organizations have started implementing GSD as a way to make new mergers, acquisition and alliances at an international level, to strive for better quality software development and services, to become a part of competitive world and gain innovative ideas to improve their businesses.

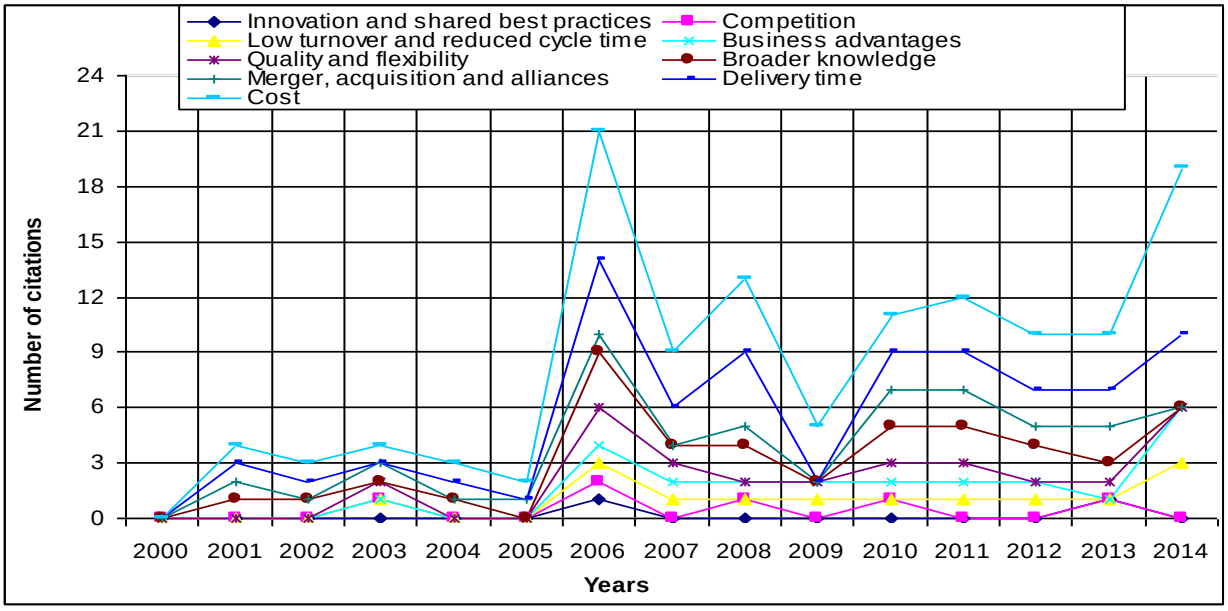

Fig. 3: Year-wise classification of motivating factors

3.1.2 Types of organizational setups involved in GSD

Going global offers many benefits, such as cost reduction strategies, access to broader knowledge and reduced development time. With the increased awareness of GSD, organizations have started embracing GSD as a way by which they can make new mergers and alliances, obtain business advantages, strive for better quality controls, and have a future ability for the enhancement and utilization of new technologies. Hence, organizations have embarked on collaborations with software teams around the world. These collaborations can be inter-organizational, intra-organizational and of a mixed type (inter+intra) in nature. Each of these has distinct theories and practices which cannot be directly applied to other types of project settings [247]. During GSD, usually development work is carried out at different geographical locations under different organizational settings and collaborations. It is therefore necessary to investigate what proportion of GSD studies was conducted in which type of setting. To identify the different GSD arrangements, the studies selected were examined. As shown in Figure 4, 63 studies reported information on organizational setups as follows: 41 referred to intra-organizational collaboration; 15 referred to inter-organizational linkages; and 7 referred to a mixed type of setup. This indicates that there is a significant difference between the number of intra and inter-organizational setups, which is typically due to the management and communication processes. In inter-organizational setups, there are often dissimilarities in the processes used. Consequently, the number of studies reporting details about inter-organizational setups is considerably lower than intra-organizational arrangements. To minimize the effects, organizations started to implement a mixed type of collaboration when commencing new mergers and acquisitions, and to strengthen relationships with international organizations.

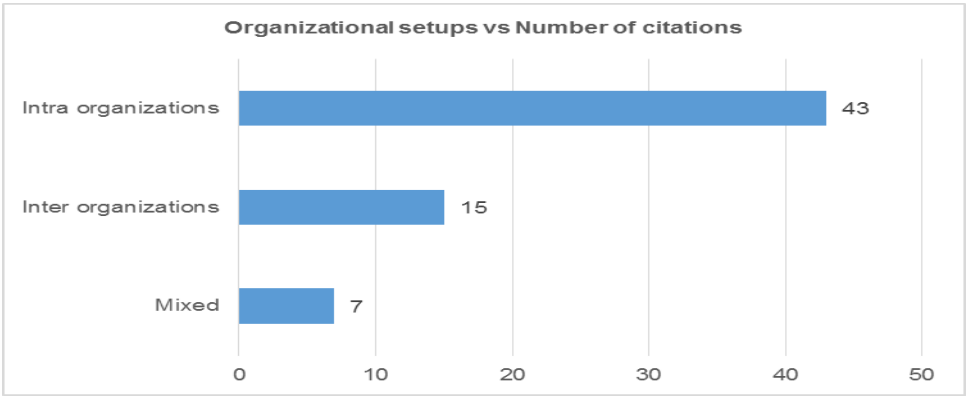

Fig. 4: Organizational setups involved in GSD
(Organizational setups identified from 63 out of 204 studies)

Figure 4 provides information on the different types of existing organizational setups, but it is difficult to identify trends in each of these setups. Therefore, the data in Figure 4 on a year-by-year basis were classified (refer to Figure 5). The results in Figure 5 show the current trends in organizational setups used in GSD. Of the three types, an intra-organizational setup is preferred by the majority of companies because of its perceived advantages. Organizations have therefore started establishing development sites around the globe, or engaging in collaborations to build joint business enterprises and foreign linkages. These findings are in line with the findings of [252, 258], and also support our results, shown in Figure 3.

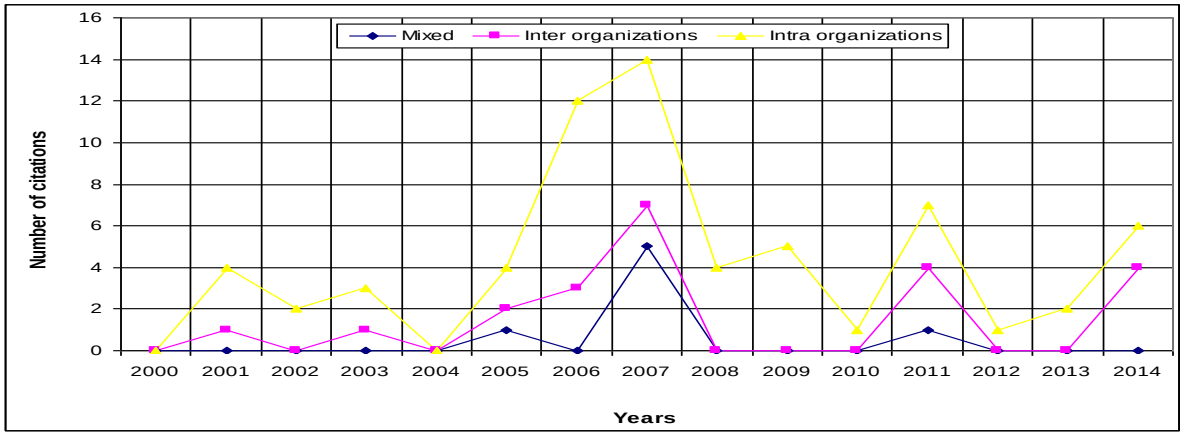

Fig. 5: Current trends in organizational setups

3.1.3 Geographical locations of development sites

The factors mentioned in section 3.1.1 drive organizations to establish business ventures and foreign collaborations. As a result, a large number of organizations have either established their development sites or made linkages with existing development sites around the world. To investigate the geographical locations of the various development sites involved in GSD, the studies selected identified that 48 reported details on the locations (refer to Figure 6), finding that the continents with the most globally distributed development sites are America (27 locations), Asia (37 locations) and Europe (19 locations). In the Americas, the most frequently listed sites are in North America (24) and South America (3). The most frequently listed locations in Asia are India (21), Cambodia (4), Japan (4), China (3), Thailand (2), Singapore (1), Malaysia (1), and Russia (1). In Europe, the most frequently listed development sites are in Ireland (6), England (3), Germany (2), France (1), Italy (1), Romania (1), Norway (1), Kenya (1), Denmark (1), Netherlands (1), and Sweden (1). Development sites are also located in the Asia-Pacific region; however the numbers are much lower, with

only 7 sites in Australia and 2 in New Zealand. Based on the selected GSD reported studies and Figure 7, it is likely that development sites are mainly located in Asian countries.

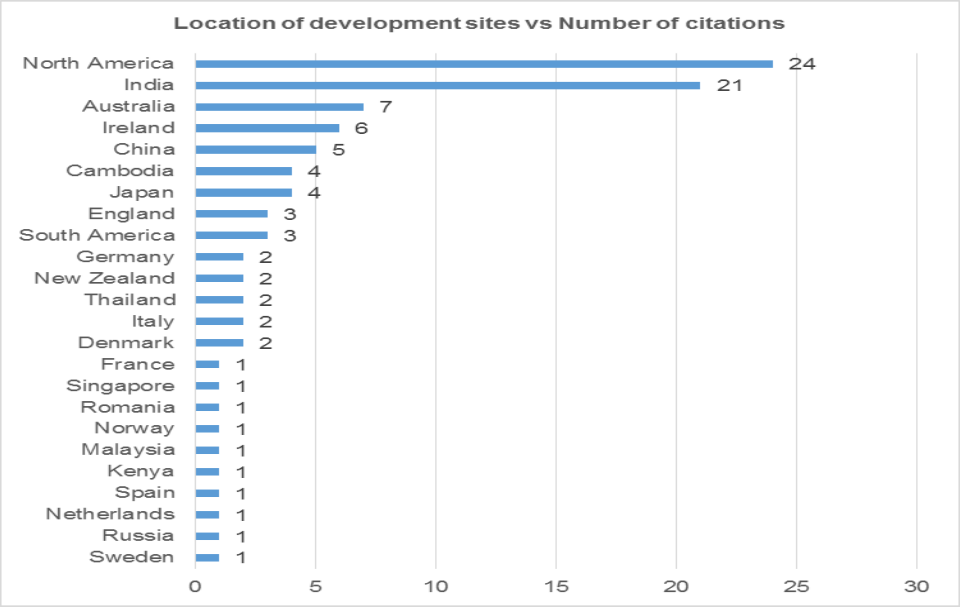

Fig. 6: Geographical location of development sites
(Geographical locations identified from 48 out of 204 studies)

Figure 6 identifies the regions which have the most development sites. However, information on a year-by-year basis cannot be extracted; hence it is difficult to know when the practice of GSD began in a particular country. Therefore, the 48 studies were categorized into geographical locations on a year-by-year basis (refer to Figure 7). The results in Figure 8 show that before 2005, only 8 of the 24 identified locations (i.e. Singapore, New Zealand, China, North and South America, Japan, Australia and India) were used for GSD collaborations. Over time, organizations started establishing development sites in other countries also. As a result, enterprises started sending software work to low-cost sites in Asia. In addition, Asian organizations also established their own development sites in different parts of America and Europe. During 2006-2014, 16 new locations were used for GSD arrangements (i.e. Ireland, Cambodia, England, Germany, Thailand, France, Italy, Romania, Norway, Malaysia, Kenya, Denmark, Spain, Netherlands, Russia, and Sweden). Hence, a large number of organizations from different continents are now involved in GSD.

Fig. 7: Start of GSD practice in different geographical locations

Section 3.1.2 discussed the three types of collaborations which are used in globally distributed development sites, these being inter-organizational collaboration, intra-organizational collaboration and a mixed type of collaboration. The collaboration patterns in the geographically distributed locations were identified in Figure 6, finding that of the 43 studies, 34 provided information about GSD collaborations but 9 did not report these details in a clear way. Of these 34 studies, 24 reported intra-organizational, 5 inter-organizational, and 5 mixed types of collaboration setups. Thus, a large number of studies indicated that intra-organizational collaborations were implemented in GSD, which supports our findings presented in section 3.1.2. From these findings, it can be observed that to minimize the risks and challenges associated with GSD, organizations prefer to engage in intra-organizational collaborations rather than inter-organizational collaborations. Although dissimilarities in the management and communication processes create difficulties for organizations to work in inter-organizational collaborations, organizations have started combining inter- and intra-organizational collaborations to be used in scenarios when it is difficult to meet project requirements within one setup.

3.1.4 Successful cases of GSD

The underlying reasons for the majority of organizations becoming involved in GSD are more or less similar, as mentioned in section 3.1.1; but due to the lack of a pre-defined set of frameworks and techniques, the benefits of GSD have not been achieved substantially. To gain an in-depth knowledge of GSD, it is necessary to investigate the successful and unsuccessful cases of GSD reported in the literature. An example of GSD is considered successful if the practices used and/or implemented were successful. Conversely, if an example of GSD describes practices which failed, this is viewed as a problematic example of GSD. With this definition, a search for the selected studies was undertaken, identifying 81 studies which discussed the outcomes of the case studies. Of these 81, 52 reported success stories and 40 reported problematic cases. As shown in Figure 8, some papers appear in both successful and problematic categories, as they discussed several case studies in a single paper.

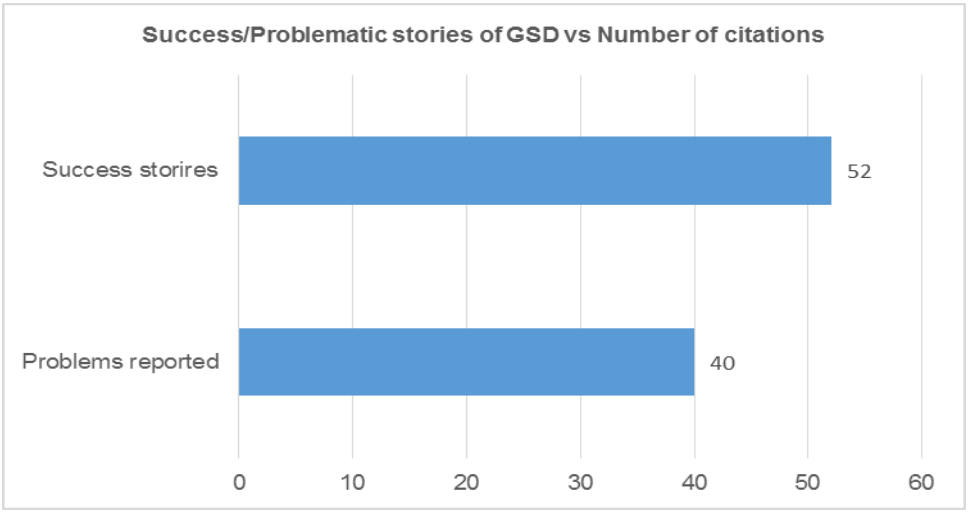

Fig. 8: Success and problematic stories
(Success and problematic stories identified from 81 out of 204 studies)

To further analyze the successful and problematic GSD stories identified in Figure 8, the available information were categorised on a year-by-year basis, as shown in Figure 9, finding that after 2005, there was an increase in GSD practices with the number of success stories increasing each year. Over the last 14 years, solutions have been proposed to minimize the effects of these issues, but most of these issues require further contributions from researchers and practitioners. Therefore, the number of success stories available in this regard is quite low.

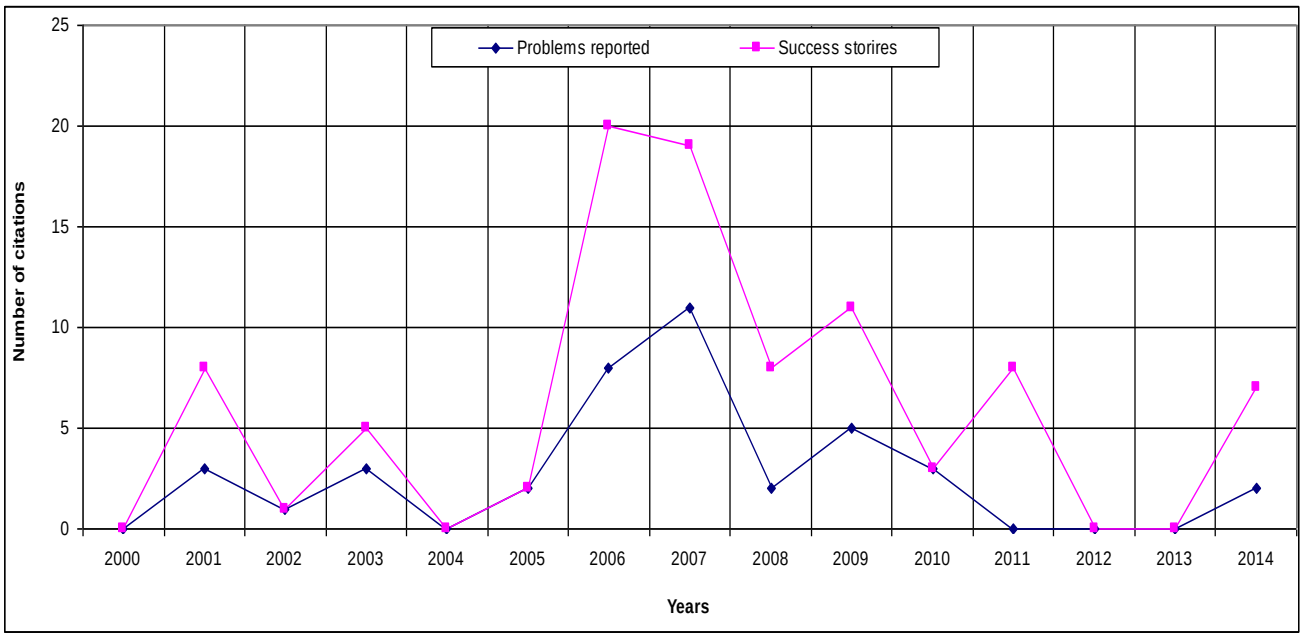

Fig. 9: Number of Success and failure stories reported

The results were analysed and shown in Figure 8 to depict the success and failure rate in inter-, intra- and mixed organizational collaborations, finding that of the 41 intra-organizational collaborations, 22 reported successes and 19 problems; of the 15 inter-organizational collaborations, 6 reported successes and 9 problems; and of the 7 mixed collaborations, 2 reported successes and 4 failures. Due to the challenges involved in inter-organizational collaborations,

the percentage of success stories is comparatively lower in magnitude than intra-organizational collaborations, which supports our findings presented in section 3.1.2.

3.2 What are some of the challenges faced by GSD practitioners? (RQ1)

3.2.1 Communication

Communication is a basic building block in the software engineering process from requirements collection to product delivery [250]. In addition to formal communication, informal communication also plays a vital role in the success of a GSD project. To successfully engage in communication in a GSD environment, different tools may be used. Although these tools facilitate team communication, the lack of face-to-face interaction and dissimilar communication standards often hamper the communication and coordination process. Consequently, the frequency of communication between GSD teams tends to decrease, affecting the software development process. To minimize this problem, it is necessary to identify the issues which create difficulties for stakeholders attempting to communicate in a GSD environment. As shown in Figure 10, 60 of the 204 studies report the following issues as being key barriers to effective communication in GSD: cultural diversity; a large distance between stakeholders; limited trust and confidence; differing organizational cultures; an inadequate amount of time spent on communication; poorly defined requirements; difficulties in coordination among teams; dissimilar roles in organizational development; distinct preferences; improper distribution of workload; different terminologies used during the communication process; knowledge sharing; language barriers; and interpersonal conflicts.

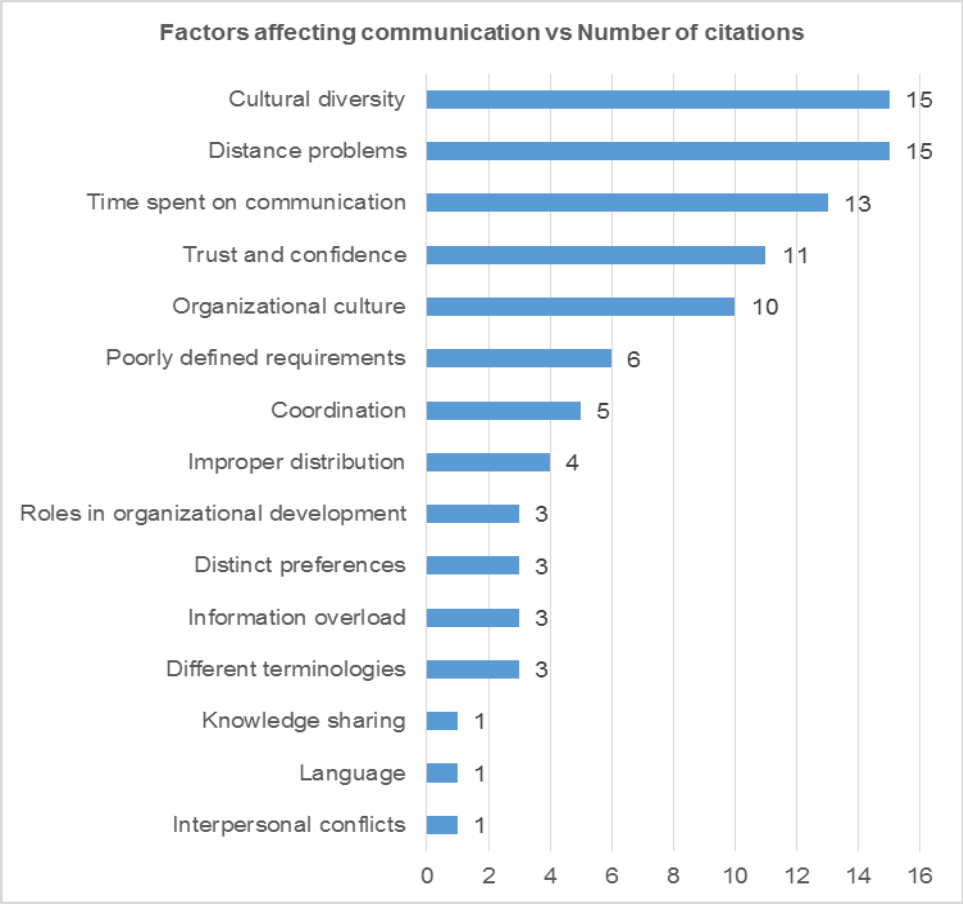

Fig. 10: Factors affecting communication
(Factors identified from 60 out of 204 studies)

To minimize the effects of the issues identified in Figure 10, different types of synchronous and asynchronous communication mechanisms are commonly used. Of the 60 studies in our analysis, 20 reported the following communication mechanisms: emails (13 of 20), video conferencing (11 of 20), instant messaging (9 of 20), audio conferencing (6 of 20), desktop/application sharing (5 of 20), and message boards and forums (4 of 20). In addition to these, the following communication tools, which are primarily designed to support specific software development activities, were reported in 16 of 60 studies: requirements tools - used for requirements elicitation, analysis, specification and validation (7 out of 16); design tools - used to design software (2 out of 16); testing tools - used to test software designs and coding (1 out of 16); management tools - used for project planning and risk management (3 out of 16); process tools - used for software process modelling (2 out of 16); and virtual meetings- used for communication between geographically distributed stakeholders (4 out of 16). The communication types and mechanisms identified from the different studies helped GSD practitioners to overcome the issues in communication, not to identify the issues.

3.2.2 Culture

Culture plays an important role in shaping the personality of humans. A person's culture identifies several things about them, such as education, religion, past experiences, personality, individual ways of thinking and working, social involvement and affection. Every nation has its own culture, therefore, moving from one part of the globe to another necessitates interaction with diverse cultures. People from different cultures react in different ways to the same situation, hence their styles of working and solving problems are different. Misunderstandings in relation to cultural issues often cause difficulties for stakeholders who need to interact with each other to achieve a common goal [254]. In a globally distributed environment, software teams are distributed across different parts of the world and often belong to dissimilar cultural backgrounds. As a result, there are differences in personal aptitudes, working styles, ways of reacting to different situations, and adherence to traditional values, making it difficult for remotely distributed team members to maintain effective communication and coordination. To examine the issues arising from teams in different multicultural contexts, the selected studies identified the factors which act as a hurdle to organisations involved in cross-cultural collaborations. As shown in Figure 11, 40 of the 204 studies reported the following factors which affect cross-cultural collaborations: power distance, a low level of trust and respect, personal aptitude to do the work, different organizational setups and background, a lack of cross-cultural training, a lack of skills and abilities, a lack of communication and coordination, the caste system, load balancing, difficulties in forming relationships with others, religion and other related festivities, inadequate communication, knowledge sharing, language barriers, interpersonal conflicts, gender bias, long term vs. short term working habits, and low morale of team mates.

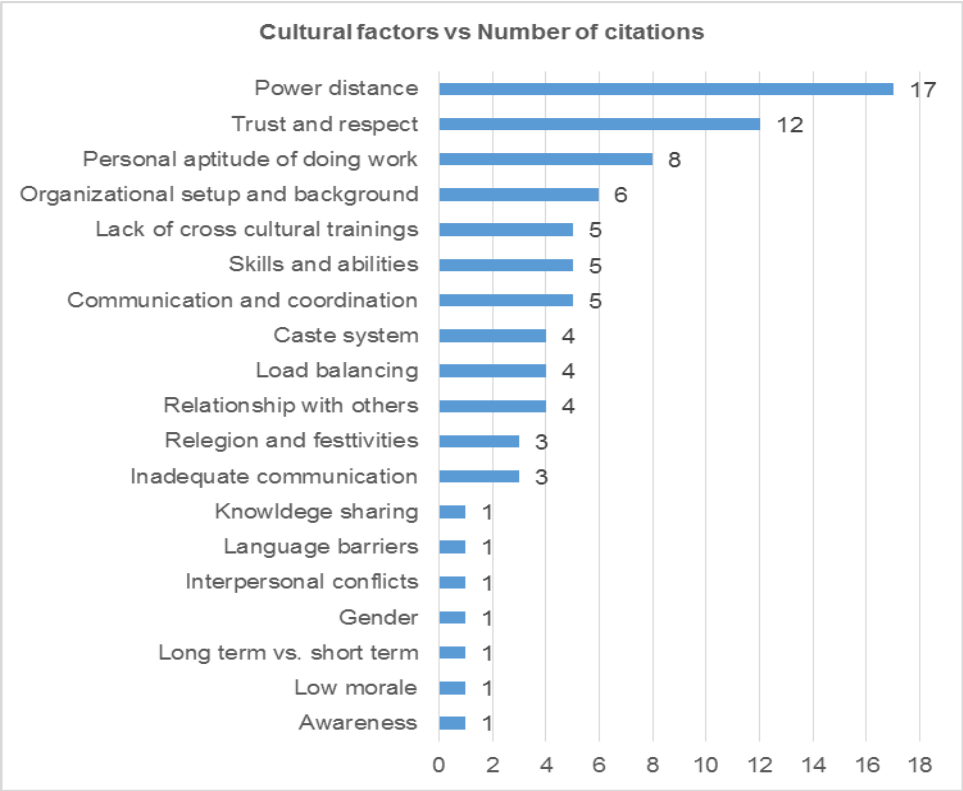

Fig. 11: Factors affecting cross-cultural collaboration
(Factors identified from 40 out of 204 studies)

The selected studies identified variations in different cultural settings, finding the following five variations: (1) high context versus low context – low context cultures focus more on written statements or what is actually written whereas in high context cultures, communication plays an important role in distributing related knowledge and views and thus prefer spoken communication; (2) monochronic vs. polychronic - monochronic cultures prefer to do one job/task at a time, while polychromic cultures prefer to do multiple jobs at a time; (3) past vs. present vs. future – past-oriented cultures prefer to do things in their traditional ways and do not want to explore new directions for solving the problem; present- oriented cultures go for interim benefits and do not want to use traditional ways or past experience; and future-oriented cultures are the most adventurous and want to explore new ways for solving and managing problems; (4) quantity of time - some cultures consider time as a limited or predetermined element, whereas other cultures take it as an imaginative activity. Societies with a limited time culture view time as a necessary and compulsory thing and usually do not waste it, particularly in creating and maintaining trust with other people. However, countries with an unlimited time culture focus more on establishing trust between parties and they respect it; and (5) individualism vs. collectivism - in individualist cultures, people want to do things on their own, relying more on themselves compared to collectivist cultures, whereas, people from collectivist cultures want to engage in joint work and enjoyed this. In short, working in different cultural contexts may result in chaos and sometimes deadlock.

The locations of development teams, identified in Figure 6, were classified into the following cultural types: Latin American, Anglo sphere, Indo sphere and Sino sphere (refer to Table 6). Germans and Americans place more emphasis on written communication compared to the Chinese, Arabs and Japanese who place greater emphasis on spoken communication. Generally, Americans and French prefer to do a single job/task at a time in contrast to Germans who undertake multiple jobs at a time. Those in China, Japan, Britain, and some parts of Latin America prefer to do things in a traditional way, and do not explore new ways of problem solving; however, the majority of those in Spanish countries do not want to use traditional ways to accomplish tasks; while those in the USA and Brazil prefer to explore new ways to solve and manage problems. Thus, from the results shown in Table 4, it can be observed that the most cited cultural setups are the Anglo sphere, Indo sphere and Sino sphere.

Table 6: Classification of geographical locations in different cultural setups

	Latin American	Anglo sphere	Indo sphere	Sino sphere
Malaysia			x	x
Norway		x		
Romania		x		
Singapore			x	x
Italy		x		
France		x		
Thailand			x	
New Zealand		x		
Germany		x		
China				x
South America	x			
Canada		x		
England		x		
Japan				x
Cambodia		x		
Ireland		x		
Australia		x		
India			x	
Denmark		x		
Spain		x		
Netherlands		x		
Sweden		x		
Russia		x		
North America		x		

To minimize the barriers of communication, problems associated with multiculturalism and low trust, several strategies and approaches have been proposed [227]. The salient features of these contributions are: (i) an emphasis on communication in small teams, rather than large teams; (ii) sending work to closely-related cultural teams; () training stakeholders about cultural differences, time zones, communication via groupware tools, and language barriers; (iv) the use of collaboration platforms and communication mechanisms; and finally (v) the use of agile methods to reduce communication barriers. Even though these solutions and strategies are available, the widespread applicability of these approaches is yet to be verified. It is therefore necessary to identify to what extent these approaches can overcome the previously stated problems, and whether it is practically and economically feasible to implement them or not.

3.2.3 Software processes

In the first half of the 1990s, the literature reported on enhancements in software processes for collocated projects; while the latter half of the 1990s deals with the learning process of developing distributed software processes [216]. The era of software process improvement has ended, and it is time to focus on new challenges such as integration, coordination and acceleration in software processes [216]. In distributed software development, process models should be carefully chosen, because there is a complex relationship between work allocation and the type of development process used. The nature of software development varies from project to project, and as a result, a combination of software processes can be used. It is therefore necessary to consider task allocation, the capabilities and cultural traditions of the development teams, and the applicability of the process model itself, before selecting process models for globally distributed environments. Thus, the successful implementation of software processes usually depends on several factors, which if not addressed appropriately, can hamper the development process.

To investigate this, the selected studies identified the factors which can create challenges for stakeholders while implementing software processes in a globally distributed environment (refer to Figure 12), identifying 28 of 204 studies which discussed this issue and reported the following as being problematic factors: different organizational setup and politics, a lack of communication and collaboration, a lack of involvement of team leaders, diverging standards and

procedures, unjustified distribution of workload within and between teams, multicultural background of team members, a lack of process validation, deficient shared repository and tools, multiple quality mechanisms, insufficient training of team members or staff, ownership issues of processes, inconsistencies between processes, a lack of awareness of software process improvement, time pressure from stakeholders, a low level of involvement of staff members, and finally problems related to integrating different sets of processes.

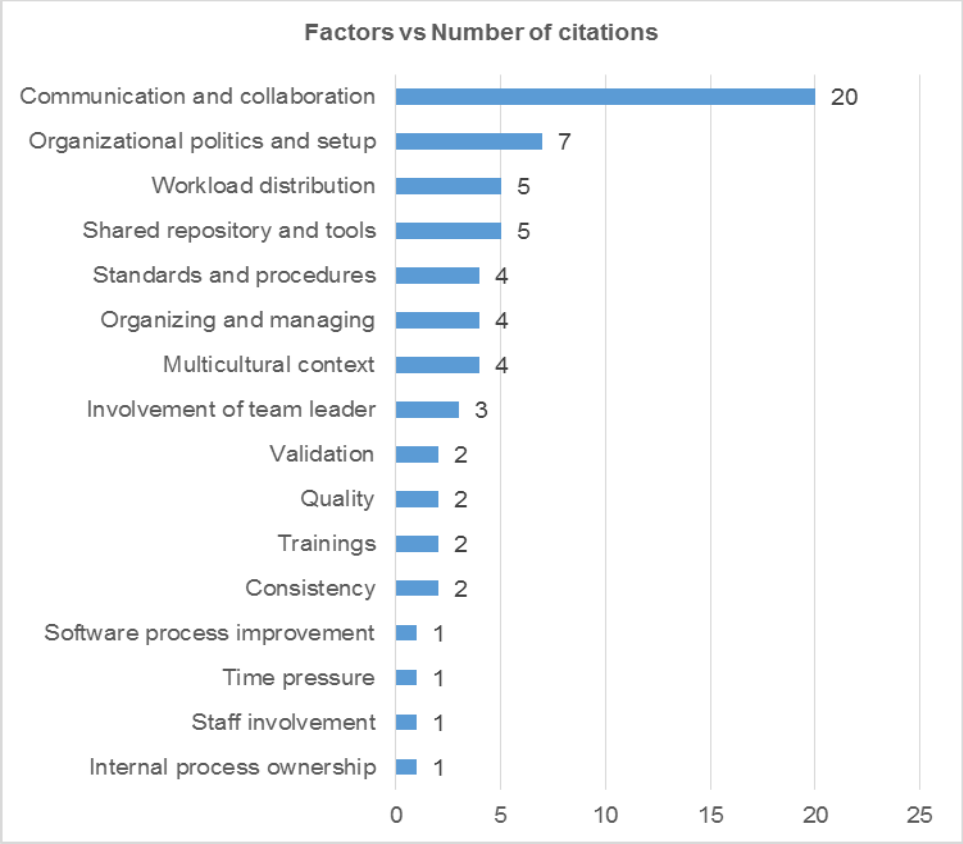

Fig. 12: Factors affecting software processes
(Factors identified from 28 out of 204 studies)

After identifying the factors affecting software processes, the studies examined the software processes being used by stakeholders in a globally distributed environment, finding 7 different types of processes, as detailed in Table 7 with the number of citations. The results show an increasing trend in the use of agile processes compared to traditional software development processes. Organizations started adopting agile processes due to their adaptable nature and to overcome the issues of communication and coordination between remotely distributed teams.

Table 7: Used software processes in GSD projects

Software Process	Number of citations
Agile (Scrum, XP)	3
Iterative	1
Incremental	2
Model-based testing	1
Rational unified process (RUP)	2
Spiral	1
Waterfall	2

The results shown in Table 7 provide insight into the most frequently used software processes, however, the success and failure rates cannot be determined. For this reason, the studies analysed which software processes were successful and which were unsuccessful. The results detailed in Table 8 show that the success vs. failure (problematic) rate for most software processes is either the same or very similar. For iterative and model-based testing, it was unable to identify evidence relating to success or failure rates in our selected studies. Based on these findings, it is clear that the use of agile processes gained momentum in the last few years over other software processes. To tackle the issues of miscommunication and the low level of trust between remote team members, researchers have introduced the concept of agile methodologies (e.g. Extreme Programming (XP), SCRUM…) in GSD projects. The basic motive behind this is to provide flexible modes of informal communication and promote trust between remote team members. However, achieving the desired level of informal communication and other benefits through agile methods is not an easy task, and requires considerable effort from the research community. Thus, it can be said that neither the traditional methods of software development (e.g. waterfall, spiral…) nor the new approaches (XP, agent-based, prototype methodologies) have resolved the issues of global software processes as yet.

Table 8: Success vs. failure rate in software processes

Software Process	Success reported	Failure / problems reported
Agile (Scrum, XP)	20	13
Iterative	-	-
Incremental	1	1
Model-based testing	-	-
Rational unified process (RUP)	1	1
Spiral	1	1
Waterfall	1	1

3.2.4 Requirements engineering

Requirements engineering (RE) is a systematic process of gathering, analyzing, validating and managing requirements [223]. The success of software projects mainly depends on the practice and performance of these RE activities. To complete these activities, an extensive amount of communication and collaboration between stakeholders is usually required. Due to the distance between stakeholders, it is often difficult for them to engage in frequent communication. Consequently, they often make assumptions about the requirements on behalf of the other stakeholders, resulting in noticeable differences between the expected and completed software outcomes. Therefore, globally distributed collaborations are a significant challenge for RE [212]. In addition to communication difficulties, there are several other factors which can be problematic for stakeholders performing different RE activities. To investigate these, it was found that 39 of the 204 studies highlighted the following factors which can impede the overall process of RE in a globally distributed environment: a lack of communication among stakeholders, changes in requirements, misunderstood requirements, conflicting requirements and diverging goals of stakeholders, multicultural background of stakeholders, a lack of collaborative tools support, low quality of requirements, inconsistencies in requirements specification, a lack of management and traceability, a lack of clarity of requirements, a lack of a shared repository, the language barrier, low-level of client interest, uncertainty, different organizational structures and aligning RE processes with communication tools (refer to Figure 13).

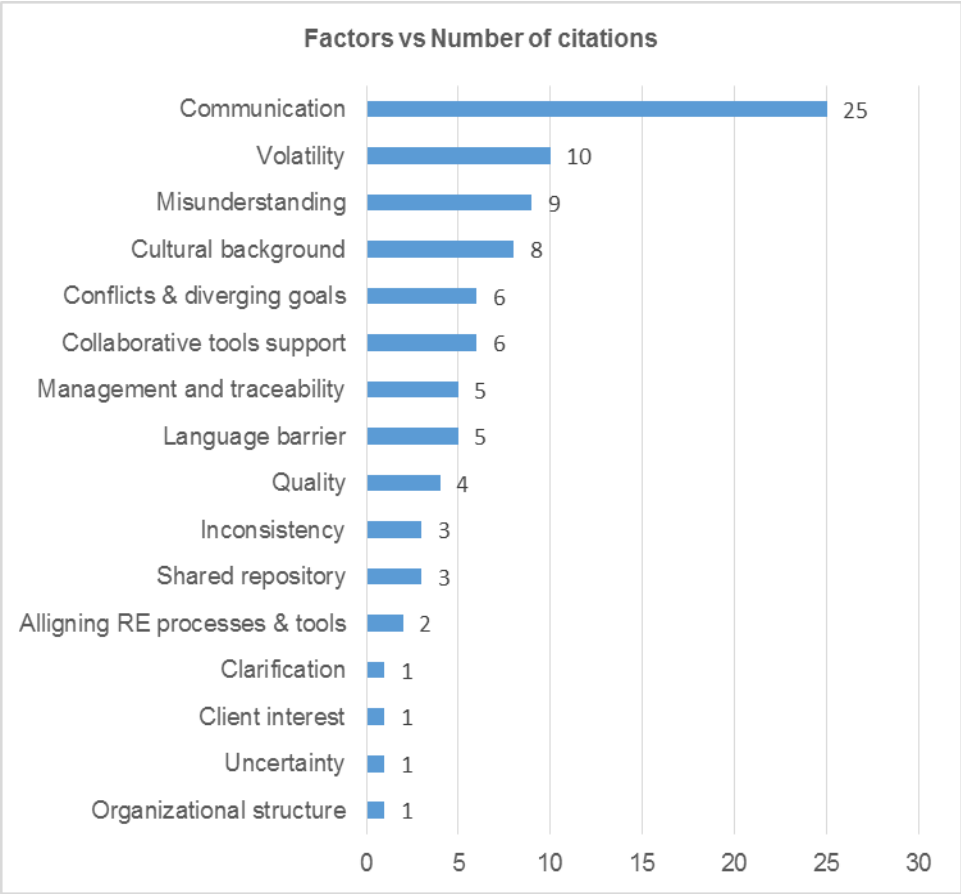

Figure 13: Factors affecting distributed requirements engineering
(Factors identified from 39 out of 204 studies)

The selected studies identified the methods and approaches used for different RE processes, finding that the approaches used for requirements elicitation, analysis, specification, and validation in a collocated environment are also used in globally distributed environments (refer to Table 9). Even though the basis of RE remains similar for both collocated and distributed projects, the non-involvement of culture, time zones, knowledge management and communication across geographical boundaries in GSD makes the existing RE methods less suitable to be used in GSD environment. Therefore, methods are required which can help stakeholders during the identification and minimization of conflicts regarding cultural differences, time zones, communication and coordination, and media selection. To undertake the different activities and processes in RE, various types of communication mechanisms are used (refer to section 3.2.1) by the stakeholders which can be either synchronous or asynchronous in nature. Asynchronous communication is used to eliminate uncertainties; whereas, synchronous discussions are preferably used to minimize ambiguities in requirements.

Table 9: List of used RE approaches during GSD

RE activity	Used approaches
Elicitation	Questions and answers, interviews, focus groups, scenarios and use cases, joint-application development, user stories, and questionnaires
Analysis	Brainstorming, story boards, prototyping, card sorting, affinity diagrams, and function allocation
Specification	Software requirements specification (SRS) documents
Validation	Visual animations, theorem checking
Management	Collaborative platforms, shared requirements repositories

Following the identification of the RE approaches, the selected studies gathered information on the collaborative tools or platforms used by stakeholders during distributed RE, identifying 7 tools, as described in Table 10. The first 4 tools are for research purposes and the rest are for commercial purposes.

Table 10: Available tools for RE in GSD

Tool	Description
EGRET	*Eclipse-based Global Requirements Tool* Facilitate stakeholders in developing awareness of other stakeholders, to enhance informal communication, knowledge and requirements change management.
IBIS	*Internet-Based Inspection System* Support informal communication during the evaluation of requirements specification. Also, to help stakeholders in generating a shared understanding of requirements
BJORN	Allow remotely distributed stakeholders to take part during the requirements prioritization process
DCPT	*Distributed Collaboration and Prioritization Tool* Help stakeholders during requirements prioritization by means of win-win requirements and negotiation system
Rational Requisite Pro	Aid development teams in requirements management and traceability, communication and collaboration, minimizing rework, and increasing quality
Telelogic Doors	Provide solutions for requirements identification and management, enhancing quality by supporting communication and coordination among stakeholders, and verification of gathered requirements
Stellation	Facilitate stakeholders in communication and change management

The studies examined the reported number of success and failure stories after using the various RE approaches and collaborative tools, finding that of the 39 studies, 8 reported success, 10 failure and the remaining 21 did not discuss information about success and failure stories. The reasons behind these findings are primarily related to the non-involvement of social, cultural, lingual, temporal and geographical aspects of GSD in the used RE approaches.

3.2.5 Efforts estimation

Software effort estimation is the process of estimating the optimal set of resources and effort required by stakeholders to develop software systems. The estimates could be used as input to requests for proposals (RFPs), budgeting, schedules and plans, human effort, and so on. The high level of communication and coordination which needs to take place between stakeholders makes this process difficult in favourable circumstances of software development, but even more challenging in the GSD environment. Several factors can cause barriers during this process, and a preliminary underestimation of these can often result in rework, over budgeting and other potential losses. Of the 204 studies, it was identified that 34 reported the factors causing these problems: a low level of communication and coordination between stakeholders, a low level of trust between stakeholders, the need to travel between development sites, a lack of knowledge transfer between sites, inappropriate task and workload distribution, multicultural background of stakeholders, requirements evolution and modification, difficulties in understanding government rules and regulations, inadequate training of employees, a lack of collaborative tool support, integration issues between software processes and components, a lack of experience, overtime work duties, the unavailability of data required for effort estimation, and finally the size of data (refer to Figure 14).

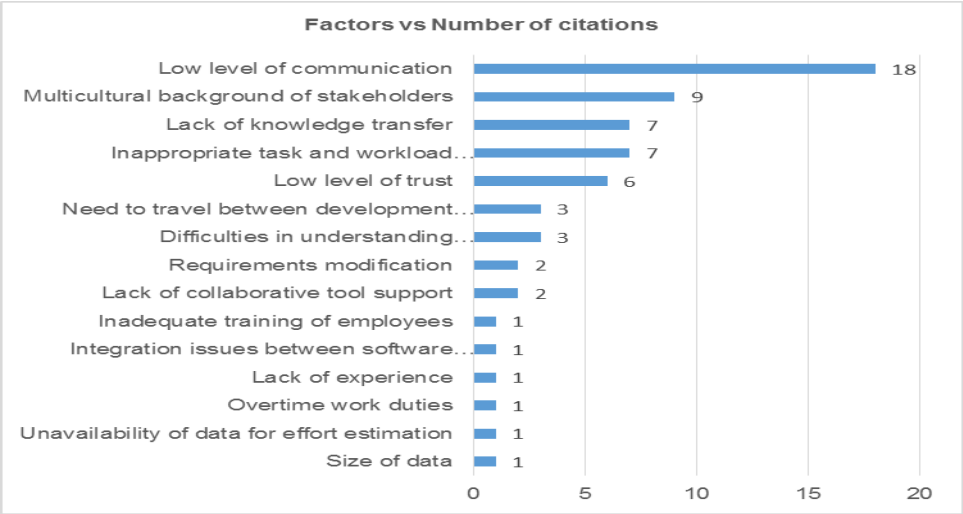

Figure 14: Factors affecting effort estimation
(Hindering factors identified from 34 out of 204 studies)

After identifying the factors which can affect effort estimation, it was found that of the 34 studies, only 14 offered potential solutions, as shown in Table 11. From the findings in Table 11, it can be seen that researchers and practitioners started considering integration, travelling, government rules and regulations, requirements volatility and modifications, trust among stakeholders, task distribution between sites, knowledge transfer, cultural diversity, and communication aspects, which are usually not considered in collocated software development.

Table 11: List of possible solutions to reduce challenges in effort estimation

Factors	**Possible solutions**
Integration issues between software processes and components	Division of work into clusters on a functionality basis, so that development work will be carried out independently at different sites. Later on, integrate different clusters in an incremental way
Traveling between development sites	Visit sites, if physical presence is required
Difficulties in understanding government rules	Obtain earlier awareness of government rules and regulations before starting the project
Requirements volatility and modifications	Manage changes in requirements during the specified time
Low level of trust	Encourage stakeholders to become familiarized with other stakeholders, which can reduce the issues of a lack of trust
Inappropriate task and workload distribution	Allocate a sufficient amount of time on making decisions about what to send and where to send it
Lack of knowledge transfer	Use appropriate synchronous and asynchronous communication mechanisms to transfer knowledge, and ensure that development sites have received data beforehand
Multicultural background of stakeholders	Train stakeholders about different cultural setups
Low level of communication	Use of communication mechanisms, as mentioned in section 3.2.1

To evaluate the effectiveness of the proposed solutions, the existing solutions that help practitioners to minimize the issues associated with effort estimation were examined. As a result, it was identified that out of 34 studies, only 6 reported success stories, 11 failure stories and 17 do not discuss success/failure stories. Due to the low frequency of success stories, it can be said that the methods developed for collocated environments are still in use.

3.2.6 Knowledge sharing and management

Knowledge management is the process of gathering, representing, sorting, distributing, utilizing, and maintaining knowledge. Several types of knowledge can be shared between stakeholders, such as: requirements communication, design and architectural information, software processes and components, testing artefacts, and documentation. Due to the distance between globally distributed sites, knowledge between development teams cannot be shared easily due to differences in organizational structures, project goals, cultural setups, and language barriers [245]. Heeks et al. described difficulties in sharing tacit knowledge; differing cultural contexts and a lack of informal information sharing [225]. Knowledge about software processes and technology can be transferred easily compared to tacit and informal knowledge. The lack of informal communication and information sharing between development sites creates difficulties and challenges in understanding tacit knowledge. Several factors contribute to the success or failure of knowledge sharing and knowledge management. To identify these, the selected studies found that 34 of the 204 studies discussed knowledge sharing and knowledge management and listed the following factors as being the basis for most problems: unavailability of centralized repositories to record and use project knowledge; differences in software processes and organizational structures; inadequate support for communication and coordination tools; a lack of awareness about stakeholders and limited experience working in cross-cultural collaborations; cultural differences between stakeholders, and low quality documentation (refer to Figure 15). As a result, development teams found it difficult to share and use explicit and tacit knowledge.

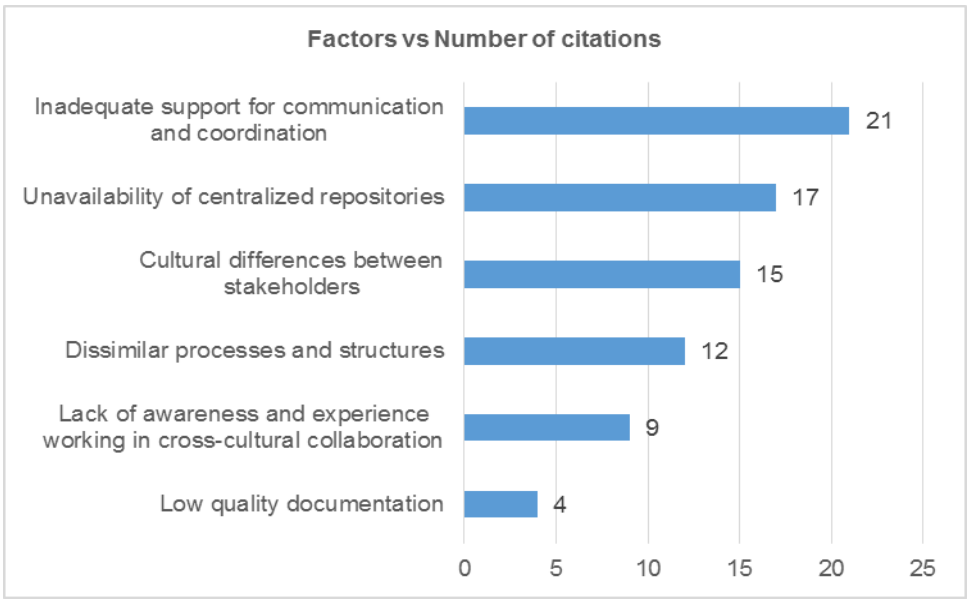

Figure 15: Factors affecting knowledge sharing and management
(Factors identified from 34 out of 204 studies)

In the selected studies, information about possible solutions to lessen the effects of these factors were extracted, unfolding that 16 out of 34 studies emphasized the following: (1) establishing and maintaining a joint repository to record and retrieve project data; (2) communication and coordination by using groupware tools; (3) similar software processes to provide greater ease in understanding project knowledge recorded by other teams; (4) clearly defined organizational and communication structure; and (5) a fair distribution of workload (refer to Table 12). These strategies are based on the philosophy of agile methods.

Table 12: List of possible solutions to reduce challenges in knowledge sharing and management

Factors	Possible solutions
Communication and coordination	Frequent interactions across development sites, face-to-face meetings, visiting other sites when physical presence is required, and use of appropriate communication and collaborations mechanisms
Software processes	Use of similar software processes to avoid issues at later stages
Organizational structure	Well-defined organizational structure with communication responsibilities, clearly defined roles and responsibilities, and consistency in goals and objectives
Centralized repository	Establishing a joint repository
Cultural differences	Treat all cultures equally
Workload distribution	Appropriate number and size of team, fair workload distribution
Estimation techniques	In addition to other factors, consideration of cultural, temporal, distance and social aspects of GSD during effort estimation

To evaluate the effectiveness of the proposed solutions, the literature identified five cases of success and five cases of failure. Although there are mechanisms and collaborative tools available to support knowledge sharing and knowledge management, the practicality and usability of these methods and tools has yet to be determined.

3.2.7 Software Architecture

Software architecture is defined as the combination of plans that make up a software system [260]. It provides the blue print for a software system and covers both the technical and non-technical (social, business and organizational) aspects of software development. In a globally distributed environment, most software systems are developed in the form of inter-relating software components, which in the end, integrates to create overall systems. Due to the reduced amount of interaction between remote team members and inconsistencies in development styles, development teams often face difficulties combining different pieces of software. To identify the factors which can affect the architectural process, it was found that only 23 of the 204 studies listed the following which affect software architecture: a lack of communication and coordination between remote sites; difficulty analysing and evaluating the architecture; poor quality of underlying architectural components; different interpretations of the same systems; differing organizational philosophies behind architectural selection; poor decisions relating to buying and/or developing; and finally, validation and scalability of architecture (refer to Figure 16). In relation to these factors, often teams show their concern over the trustworthiness and quality of components developed by other teams, due to the lack of common criteria for ensuring the quality of software components. The main reason for using component development is its reusability feature. To develop a software component in a timely manner, developers usually compromise on some of the desired features which results in an end product that does not entirely meet the requirements. Thus, the management of these components becomes very difficult.

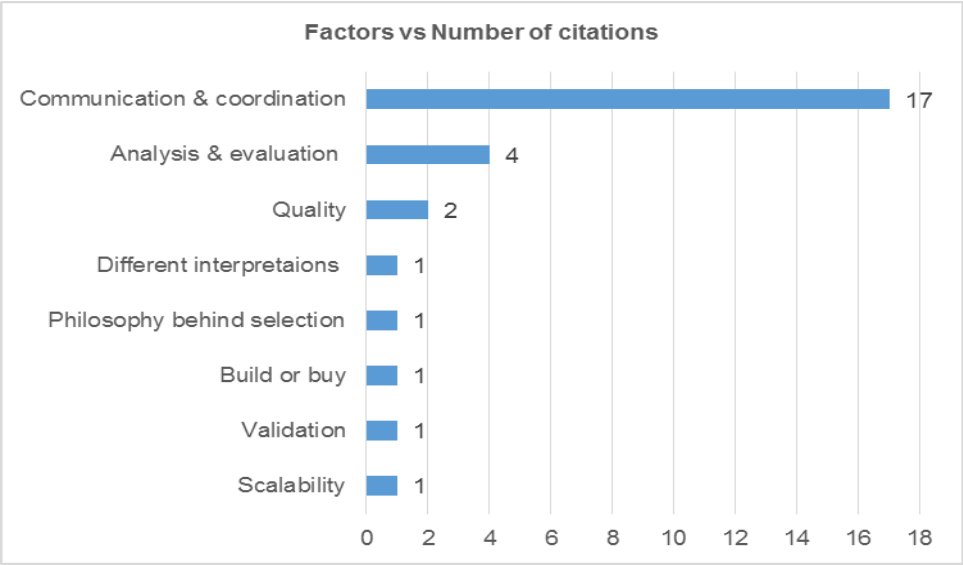

Fig. 16: Factors affecting software architecture
(Factors identified from 23 out of 204 studies)

After gaining insight into the factors which can affect the architectural process in a distributed environment, the selected studies identified the possible solutions or strategies to address this problem. As a result, information on the existing solutions were gathered and presented in Table 13. Since a lack of communication and coordination is the basis of most of the problems, instances of groupware tools, a collaboration matrix, and knowledge sharing at regular intervals can be seen. In addition to these findings, evidence is also available when the preferences of all stakeholders are taken into account while designing software architectures. In view of these findings, it can be seen that efforts are being made to address the earlier stated issues. Still, it is necessary to develop software architectures which are flexible (scalable), trustworthy, fulfil the mutually agreed quality criteria, can be interpreted unambiguously, and ensure the development of software systems in a short time.

Table 13: List of possible solutions to minimize challenges in software architecture

Factors	Possible solutions
Analysis and evaluations	Groupware tools can be used to evaluate software architectures
Communication and coordination	"Design Structure Matrix" (DSM) can be used to represent the component structure of software architecture. DSM is consistent with "Social Network Matrix" (SNM) , which is best suited for communication and coordination Distributed meetings can be used effectively if the tools used are of high quality
Architectural viewpoints	Used to design architecture in accordance with the perspectives of different stakeholders
Validation	Scalability and validation of software architecture can be examined by using tool support or theorem checking
Quality	Equal load-balancing / task distribution to avoid compromise on quality and time
Integration	Performed on an incremental level
Different interpretations	Knowledge sharing on a regular basis to minimize different interpretations

4.0 KEY FINDINGS OF OUR REVIEW

The findings of this systematic review will give readers to gain an understanding of the current state-of-the-art practices and challenges in GSD. Our findings will be beneficial for practitioners when making decisions regarding GSD.

4.1 How GSD is practised by organisations?

4.1.1 Motivation behind GSD

From 2001 to 2005, reduced cost, shorter development time and merger with other organizations were considered to be the main motivations of GSD. With passage of time and more awareness about GSD, organizations have started considering other factors too, such as: innovation and shared best practices; low turnover and reduced cycle time; quality and flexibility; competition; business advantages and access to broader knowledge.

4.1.2 Types of organizational setups involved in GSD

To reduce the challenges associated with management and communication processes in inter-organizational setups in a globally distributed environment, it is recommended that a mixed type of collaboration (i.e. inter+intra organization) be implemented when commencing new mergers and acquisitions, and to strengthen relationships with international organizations.

4.1.3 Geographical locations of development sites

Before 2005, the development sites were mostly located in Singapore, New Zealand, China, North and South America, Japan, Australia and India. With passage of time and easy access to technology, 16 new development centres were established in 16 other countries, which include: Ireland; Cambodia; England; Germany; Thailand; France; Italy; Romania; Norway; Malaysia; Kenya; Denmark; Spain; Netherlands; Russia; and Sweden.

4.1.4 Successful cases of GSD

Based on the selected GSD reported studies, some organizations are experimenting different ways and procedures to gain the actual benefits of GSD. It is therefore advisable to think about intellectual policies, government rules and regulations, road and rail networks, the future ability for enhancement, pre/post contract planning, and copyright issues before going global. Consider the potential benefits and risks associated with the GSD project beforehand, evaluate the success vs. failure rate, and only opt for GSD if the potential benefits are greater than the risks and challenges.

4.2 What are some of the challenges faced by GSD practitioners? (RQ2)

4.2.1 Communication

The lingual, social, geographical, cultural and temporal differences between software development teams often hamper the communication and coordination process. It is therefore recommended that details about the different types of communication modes, mechanisms and tools which could be used for communication purposes between teams should be mutually agreed and documented.

4.2.2 Culture

Cultural differences are present in GSD and will remain there. To overcome these differences, it is highly recommended to educate stakeholders or development teams about these differences before initiating the project. By developing an awareness of the list of factors which usually create challenges in cross-cultural collaborations, a significant amount of time, effort and revenue could be saved, which was wasted previously, and the rate of success stories of GSD could be increased.

4.2.3 Software processes

The existing research in GSD is mostly investigative in nature and primarily focuses on the identification of issues, rather than defining solutions for them. The software processes and methods which could be suitably used for globally distributed environment are not available. As a result, the processes and techniques which were developed for collocated software development are now being used for GSD projects. New approaches to software development are therefore required which can tackle the cultural, social and geographical aspects of GSD.

4.2.4 Requirements engineering

Traditional approaches to requirements elicitation and analysis do not consider factors such as distance, time zones, cultural diversity, communication and language barriers. As a result, the ways by which requirements are elicited, analyzed, documented, validated and managed in collocated software development projects cannot be used effectively in globally distributed environment. Different tools (e.g. EGRET, IBIS, BJORN…) and techniques (e.g. user stories, joint application development, shared repositories…) were proposed to facilitate stakeholders during distributed RE, but due to the non-involvement of GSD factors, their applicability is still under discussion.

4.2.5 Effort estimation

The existing techniques of effort estimation do not consider cultural, lingual, temporal, social and geographical aspects; thus, there is a difference between the anticipated and obtained outcomes. It is therefore necessary to spend some time deciding what to send and where to send it, and how much difference will be encountered in cultural setups, time zones, language and government rules. If it is necessary for travel to take place between sites, issues regarding who will pay for the visit and how long the visit will last will need to be settled. Furthermore, a reasonable estimate about changes in requirements should be made and incorporated with other software estimates.

4.2.6 Knowledge sharing and management

Beside the lack of techniques, the low level of trust between distributed teams is considered to be one of the biggest challenges in knowledge sharing and management across global boundaries. Differences also occur between the team members of different departments. Hence, there is a rising need to develop such type of techniques and practices which can be used to lessen the gap between the knowledge of distributed teams and establish a greater level of trust between them.

4.2.7 Software architecture

Organizational setups have a great impact on the architectures developed by them. This situation is clearly visible in GSD where the software architecture developed by one team cannot be effectively used by other distributed teams. It is therefore required to develop such type of architectures which are scalable, trustworthy, fulfil the mutually agreed quality criteria, can be interpreted unambiguously, fits to the organizational structure, and support the business needs, should be used.

5.0 THREATS TO THE VALIDITY

The findings presented in this review study have the following validity threats.

(i) Construct validity: this is primarily related to obtaining the right information by defining the right scope. At this stage, the biggest challenge is to decide what should be included in the review. To address this issue, all those studies which provided empirical, case study, experimental, industrial and survey-related information about GSD were considered.

(ii) External validity: the findings of this review could not be generalized because the results are based on a specific set of keywords, publication years and the research repositories that have been used for the data collection. Thus, our results could be limited and cannot be applied in every organizational setup.

(iii) Results validity: the initial concept of GSD was introduced in the early 1990s, but interest in this area accelerated after 2000 as a result of globalization. The area is still evolving and a large set of keywords are available which can be used to represent the concept of GSD. In this review, six different keywords which are mostly used in the context of globally distributed software development were considered, and six research repositories (i.e. IEEE, ACM, Science Direct (Elsevier), Wiley Inter Science, ISI (Web of Knowledge), and Springer) were used for conducting an initial search in the study selection process (refer to Table 2). Due to the fact that Google scholar was not used, it could be possible that some of the existing studies were not included in the review process. Thus, our findings are only based on the selected set of keywords and from six research repositories. Also, the analysis uses the *number* of studies as its basis. However, not all papers are necessarily equal; and weights based on the complexity and/or number of projects they discuss could be assigned to them to improve the analysis.

(iv) Internal validity: this is mainly related to the capability of replicating similar findings. This aspect was addressed adhering to the systematic review procedure, described in section 2. Two researchers were involved in the review process, who after a period of time, worked together to avoid duplications in the identified studies. However, it could be possible that if this study is replicated by other researchers, minor variations in the identified studies will be observed due to differences in personal aptitude and thinking. Regardless of this fact, the findings presented in this review will enable readers to obtain a clear picture of GSD.

(v) Conclusion validity: The number of research articles presented in this study does not indicate the actual number of GSD practices being undertaken in reality. Thus, the number could only be used to make inference of how successful is GSD.

6.0 COMPARISON WITH RELATED WORK

A total of 26 literature reviews have been published on GSD. All of them addressed particular aspects of GSD, namely, agile practices in GSD, communication and coordination challenges, distributed process models, management-related issues in globally distributed projects, effort estimation, knowledge transfer and management, vendor selection, state-of-the-art practices, and potential solutions for increasing the likelihood of GSD success (refer to Tables 14-22).

Table 14: Agile practices in GSD

Paper #	Title	Studies selected	Focus	Reference
1	Global software engineering and agile practices	1999-2009	Agile practices	[232]
2	Systematic literature reviews in global software development	Prior 2011	Analyze published systematic reviews focusing on Agile practices	[263]
3	Using Scrum in Global Software Development	2003-2009	SCRUM practices	[228]
4	Scrum Practices in Global Software Development	Not mentioned	SCRUM practices	[229]
5	Distributed Agile Software Development	Not mentioned	Agile practices	[255]
6	Global Prospect of Distributed Agile Software Development	Not mentioned	Agile practices	[261]
7	Agile Practices in Global Software Engineering	1999-2009	Agile practices	[231]
8	Directions and Advancements in Global Software Development	Not mentioned	Analyze Agile practices	[207]

Table 14, provides details on the systematic literature reviews (SLRs) which primarily focus on agile practices in GSD. Jalali et al. conducted a survey on papers published during the period 1999 to 2009 on GSD agile practices and reported

that the use of agile methods had increased and communication remained an issue [232]. Verner et al. reviewed the SLRs on GSD to identify the areas covered, the researchers involved and the quality of published material on GSD [263]. Hossain et al. presented the issues and difficulties involved in using SCRUM for globally distributed projects and the tips available to address these [228]. Furthermore, they have developed a framework to demonstrate how SCRUM practices could be used to address the issues identified in [229]. Shrivastava and Date discussed the challenges faced by distributed teams when using SCRUM and strategies to mitigate the issues [255]. Tiwari et al. discussed the issues faced by distributed teams when using agile practices in GSD [261]. Jalali and Wohlin examined the literature on agile practices in GSD to identify under what circumstances these practices could be used effectively [231]. Akbar et al. examined the existing literature on the use of agile practices in GSD and made recommendations to improve software development in a distributed environment [207].

Table 15: Communication and coordination challenges in GSD

Paper #	Title	Studies selected	Focus	Reference
9	Challenges and Improvements in Distributed Software Development	2001-2008	Communication and coordination challenges	[234]
10	Tools used in Global Software Engineering	2003-2010	Communication and coordination tools	[251]
11	Risk Identification and Risk Mitigation Instruments for Global Software Development	2007-2009	Communication and coordination issues in GSD	[246]
12	Communication Risks and Practices during Requirements Change Management	Not mentioned	Communication issues and practices in change management	[236]
13	Issues in Global Software Development	2006-2011	Communication, trust and coordination issues	[224]

Table 15 presents details on the SLRs which primarily focus on communication and coordination challenges. Jimenez et al. analysed the papers published between 2001-2008 on the communication and coordination challenges in distributed software development and studied improvements in the source code, collaboration and the integration of existing tools [234]. Based on the papers published between 2003 and 2010, Rodrigues et al. reported on the communication, coordination and control tools used for GSD purposes, most of which were research based [251]. Nurdiani et al. examined the papers published between 2007 and 2009 to identify the risks and challenges related to communication, coordination and control issues in GSD. Based on the review analysis, the authors presented a list of mitigation strategies to minimize the effect of these challenges [246]. Khan et al. investigated the communication issues and their effects which occur during requirements change management in GSD [236]. Haq et al. discussed the communication and coordination issues between client and vendor in GSD, and presented strategies to overcome these issues [224].

Table 16: Process models in GSD

Paper #	Title	Studies selected	Focus	Reference
14	Process models in the practice of distributed software development	1998-2007	Process models for distributed software development	[249]
15	Solutions in global software engineering	Prior 2009	Process areas of GSD	[253]
16	Global Software Development and Quality Management	2000-2011	Quality and process management	[243]
17	Software Integration Challenges in Global Software Development Environment	Prior 2012	Integration issues in GSD	[230]

Table 16 presents details on the SLRs which principally focus on process models. Based on the papers published between 1998 and 2007, Prikladnicki and Audy focussed on process models for distributed software development and defined the need for distributed models at different levels [249]. Schneider et al. analysed and empirically validated the solutions that were proposed by different researchers in relation to GSD. These solutions were then subdivided into different process areas (i.e. management, engineering and support) to identify the research contributions and gaps [253]. Mishra et al. presented an SLR on quality management of GSD projects, highlighting the existing issues and suggesting future directions to overcome these issues [243]. Ilyas and Khan identified the challenges related to software integration with respect to size and type in a globally distributed environment from a software vendor's perspective. They recognized the factors which need to be addressed before, during and after the integration process [230].

Table 17: Management issues in GSD

Paper #	Title	Studies selected	Focus	Reference
18	Challenges and Solutions in Distributed Software Development Project Management	1998-2009	Challenges in requirements management	[256]
19	Problems and Solutions in Distributed Software Development	1995-2007	Challenges in GSD	[233]

Table 17 presents details on the SLRs which mostly focus on management issues. Silva et al. reviewed the papers published between 1998 and 2009 on distributed project management, highlighting the issues faced by software teams in managing requirements in a GSD environment and presented a model to minimize the effect of the identified issues [256]. Jimenez and Piattini analyzed the papers published between 1995 and 2007 on communication, collaboration and coordination, source control, knowledge and project management, process support, quality and measurement, and the challenges of defect detection in distributed software development. Furthermore, the authors presented potential solutions to address these challenges [233].

Table 18: Effort estimation in GSD

Paper #	Title	Studies selected	Focus	Reference
20	Effort estimation in global software development	2001-2013	Effort estimation in GSD	[210]

Table 18 presents details on an SLR which primarily focuses on effort estimation. Britto et al. presented an SLR on effort estimation in GSD, and discussed the state-of-the-art practices and the solutions that exist in the literature [210].

Table 19: Knowledge transfer and management in GSD

Paper #	Title	Studies selected	Focus	Reference
21	Models and Tools for Managing Distributed Software Development	2000-2010	Tools for supporting GSD management	[215]
22	Knowledge transfer challenges and mitigation strategies in global software development	1999-2011	Knowledge transfer practices in GSD	[244]

Table 19 presents details on the SLRs which mainly focus on knowledge transfer and management. Costa et al. conducted a systematic review to identify the tools and techniques that can be used to manage knowledge in distributed software development. Based on their findings, it is clear that the tool support which exists is not sufficient to address the emerging needs of GSD projects [215]. Nidhra et al. reviewed the literature to reveal the challenges in knowledge management and presented mitigation strategies for them. The authors grouped the challenges and strategies into technology, projects and personnel factors. A lot of support exists for personnel and project factors, but less has been

done for technology. Without proper technological support, successful transfer of knowledge cannot be guaranteed [244].

Table 20: Vendor selection in GSD

Paper #	Title	Studies selected	Focus	Reference
23	Factors influencing clients in the selection of offshore software outsourcing vendors	1998-2008	Barriers in selecting vendors	[237]

Table 20 presents details on an SLR which primarily focuses on vendor selection. Khan et al. surveyed the papers published during the period 1998 to 2008, identifying the barriers that affected the selection of outsourcing vendors and found that language and cultural barriers were the main factors affecting the process [237].

Table 21: State-of-the-art practices in GSD

Paper #	Title	Studies selected	Focus	Reference
24	Empirical evidence in global software engineering	2000-2008	State-of-the-art practices	[258]

Table 21 presents details on an SLR which mainly focuses on state-of the-art practices. [258] analysed the literature to identify: the factors which motivate practitioners to develop software projects in a globally distributed environment; the geographical location of development centres; the types of organizations (i.e. inter, intra and inter+intra organizations) involved in GSD; the successful and unsuccessful GSD projects GSD; and the areas studied in GSD.

Table 22: GSD, in general

Paper #	Title	Studies selected	Focus	Reference
25	Risk Mitigation Advice for Global Software Development	2005-2011	Suggestions to increase the likelihood of GSD success	[262]
26	Systematic Literature Reviews in Distributed Software Development	2008-2011	Summarize the findings of existing review articles	[240]

Table 22 presents details on the SLRs on GSD, in general. Verner et al. presented risk mitigation advice to improve GSD projects. Most of this advice related to agile methods and the communication aspects of GSD; however less support is available for the other areas [262]. Marques et al. reviewed 14 SLRs on distributed software development and categorised them into different areas. Of the 14 reviews, 7 shed light on management aspects, 4 on engineering processes, and 1 on requirements, design and software engineering education, respectively. These topics are limited and primarily focus on problems rather than solutions [240].

Table 23: Comparison of the review papers
AP = agile practices, CC = communication and coordination, PMD = process models, MA = management aspects, EE = effort estimation, KM = knowledge transfer and management, SP = state-of-the-art practices, CD = cultural diversity, RE = requirements engineering, and SA = software architecture.

References	Year	AP	CC	PMD	MA	EE	KM	SP	CD	RE	SA
[232]	1999 - 2010	x									
[262]		x									
[228]		x									
[229]		x									
[255]		x									
[261]		x									
[231]		x									
[207]		x									
[234]	2001 - 2011		x								
[251]			x								
[246]			x								
[236]			x								
[224]			x								
[249]	1998 - 2011			x							
[253]				x							
[243]				x							
[230]				x							
[256]	1998 - 2009				x						
[233]					x						
[210]	2001 - 2013					x					
[215]	1999 - 2011						x				
[244]							x				
[224]	1998 - 2008							x			
[258]	2000 - 2008								x		
[263]	2005 - 2011									x	
[240]										x	
Naveed and Lai	2000 - 2014	x	x	x	x	x	x	x	x	x	x

Table 23 compares all the review papers considered in this paper. The merits of our systematic review include: (i) the use of a larger set of keywords in the data searching process broadened the scope of our study; (ii) the review was conducted on papers published between 2000– 2014, and as a result, up-to-date information on the current state-of-the-art practices and the challenges which appeared between 2008 to 2014 is available; (3) in addition to communication and coordination challenges, our review also provides an analysis of the challenges which arise due to cultural diversity, which has not been covered to date in the existing studies; and (4) in addition to the other technical challenges, details were provided on the challenges in requirements engineering and software architecture, which have not been reported in any of the previous reviews.

7.0 CONCLUSION AND FUTURE WORK

Over the last decade, the practice of developing and managing software projects in a global software development environment has gained momentum. This paper has presented a review on the practices and challenges in GSD. The results of this systematic review helped the reader to obtain a comprehensive view of a comparatively new area which should be addressed in a detailed manner. Due to the diverse configurations of GSD projects and the challenges associated with them, the area of GSD has become wide and requires addressing in relation to both soft and technical aspects. The studies thus examined were more investigative in nature and focused on the identification of issues, rather than providing useful solutions for them. As a result, the number of studies reporting success stories is relatively low and opens room for further research and development.

To provide answers to the research questions, our findings in both quantitative and qualitative ways have been provided. Based on our analysis, the following conclusions can be drawn: (i) the practice of GSD is here to stay and will evolve and become more mature in upcoming years; (ii) the advantages of GSD can be beneficial if the conflicts in culture, time zones, language, communication and coordination, group awareness and technical aspects can be identified and minimized beforehand; (iii) a lack of awareness about working in distributed environments creates difficulties for stakeholders, and may eventually result in failure or major rework; (iv) considering the lack of empirical evidence, it is hard to identify which type of software and communication processes will be suited to which type of distributed arrangement; and (v) the existing software processes and models are used in distributed environments, but the lack of cultural, social and geographical aspects makes these processes less useful.

7.1 Implications for future work

Based on the outcomes of this study, the aspects which require future work are listed below. The main benefit of GSD is cost reduction. However, what looks attractive at first glance could involve technical, communication and cultural issues. Therefore, decisions about GSD should be taken after evaluating the feasibility of going global. How to determine whether a GSD project is a success or a failure is still not well understood. Therefore, new methods are required to gauge whether GSD practices will succeed or fail. Over the past few years, the use of agile practices can be found in globally distributed environments, but still the practicability of them for larger GSD projects remains unaddressed. An examination of how the efficient assignment of workloads to development teams which can lead to the correct estimation of GSD projects needs to be explored. There is a need for collaborative tool support which can be accessible to the development teams located in different countries, so that they can retrieve and share their work and discuss project-related issues. The impact of multicultural and multilingual development teams on GSD needs to be investigated. Mechanisms and approaches are required for evaluating the breadth of existing software development methods in the GSD environment, so that their actual benefits can be identified and understood by stakeholders. To further improve the practice of GSD, additional amount of empirical evidence is required, in which studies are performed at several sites, rather than only at few sites.

Appendix 1: Selected Studies

[1] P. S. Brockmann, T. Thaumuller, Cultural Aspects of Global Requirements Engineering: An Empirical Chinese-German Case Study, Fourth IEEE International Conference on Global Software Engineering (ICGSE ‘09), 2009, pp. 353-357.

[2] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, B. Fitzgerald, Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance, International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 3-11.

[3] D. C. Gumm, Mutual Dependency of Distribution, Benefits and Causes: An Empirical Study, Second IEEE International Conference on Global Software Engineering (ICGSE ’07), 2007, pp. 113-124.

[4] M. Korkala, P. Abrahamsson, Communication in Distributed Agile Development: A Case Study, 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, pp. 203-210.

[5] R. Prikladnicki, J. L. N. Audy, D. Damian, T. C. Oliveira, Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring, Second IEEE International Conference on Global Software Engineering (ICGSE ’07), 2007, pp. 262-274.

[6] C. Ebert, B. K. Murthy, N. N. Jha, Managing Risks in Global Software Engineering: Principles and Practices, IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 131-140.

[7] D. E. Damian, D. Zowghi, The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, proceedings of IEEE Joint International Conference on Requirements Engineering, 2002, pp. 319- 328.

[8] M. Paasivaara, S. Durasiewicz, C. Lassenius, Using Scrum in Distributed Agile Development: A Multiple Case Study, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 195-204.

[9] D. Smite, C. Wohlin, R. Feldt, T. Gorschek, Reporting Empirical Research in Global Software Engineering: A Classification Scheme, IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 173-181.

[10] L. Pilatti, J. L. N. Audy, Global Software Development Offshore Insourcing Organizations Characteristics: Lessons Learned from a Case Study, International Conference on Global Software Engineering (ICGSE ’06), 2006, pp. 249-250.

[11] J. D. Herbsleb, A. Mockus, An empirical study of speed and communication in globally distributed software development, IEEE Transactions on Software Engineering, Vol. 29 No. 6, 2003, pp. 481- 494.

[12] J. D. Herbsleb, A. Mockus, T. A., Finholt, R. E. Grinter, An empirical study of global software development: distance and speed, Proceedings of the 23rd International Conference on Software Engineering (ICSE ’01), 2001, pp. 81- 90.

[13] C. Ebert, C. H. Parro, R. Suttels, H. Kolarczyk, Improving validation activities in a global software development, Proceedings of the 23rd International Conference on Software Engineering (ICSE ’01), 2001, pp. 545- 554.

[14] R. Prikladnicki, J. L. N. Audy, R. Evaristo, A Reference Model for Global Software Development: Findings from a Case Study, International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 18-28.

[15] A. Begel, N. Nagappan, Global Software Development: Who Does It? IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 195-199.

[16] I. Richardson, G. Avram, S. Deshpande, V. Casey, Having a Foot on Each Shore - Bridging Global Software Development in the Case of SMEs, IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 13-22.

[17] V. Mikulovic, M. Heiss, J. D. Herbsleb, Practices and Supporting Structures for Mature Inquiry Culture in Distributed Software Development Projects, International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 245-246.

[18] J. Makio, S. Betz, On educating globally distributed software development — A case study, 24th International Symposium on Computer and Information Sciences (ISCIS ’09), 2009, pp. 480-485.

[19] A. Taweel, B. Delaney, T. N. Arvanitis, Z. Lei, Communication, Knowledge and Co-ordination Management in Globally Distributed Software Development: Informed by a scientific Software Engineering Case Study, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 370-375.

[20] E. O. Conchui, H. Holmstrom, P. J. Agerfalk, B. Fitzgerald, Exploring the Assumed Benefits of Global Software Development, International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 159-168.

[21] A. E. Milewski, M. Tremaine, R. Egan, S. Zhang, F. Kobler, P. O'Sullivan, Guidelines for Effective Bridging in Global Software Engineering, IEEE International Conference on Global Software Engineering (ICGSE '08), 2008, pp. 23-32.

[22] D. Petkovic, R. Todtenhoefer, G. Thompson, Teaching Practical Software Engineering and Global Software Engineering: Case Study and Recommendations, 36th Annual Conference on Frontiers in Education, 2006, pp. 19-24.

[23] A. Avritzer, D. Paulish, C. Yuanfang, Coordination Implications of Software Architecture in a Global Software Development Project, Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA '08), 2008, pp. 107-116.

[24] J. J. Treinen, S. L. Miller-Frost, Following the sun: Case studies in global software development, IBM Systems Journal Vol. 45 No. 4, 2006, pp. 773-783.

[25] T. Niinimaki, A. Piri, C. Lassenius, Factors Affecting Audio and Text-Based Communication Media Choice in Global Software Development Projects, Fourth IEEE International Conference on Global Software Engineering (ICGSE '09), 2009, pp. 153-162.

[26] T. Niinimaki, C. Lassenius, Experiences of Instant Messaging in Global Software Development Projects: A Multiple Case Study, IEEE International Conference on Global Software Engineering (ICGSE '08), 2008, pp. 55-64.

[27] M. Cataldo, M. Bass, J. D. Herbsleb, L. Bass, On Coordination Mechanisms in Global Software Development, Second IEEE International Conference on Global Software Engineering (ICGSE '07), 2007, pp. 71-80.

[28] M. Paasivaara, S. Durasiewicz, C. Lassenius, Distributed Agile Development: Using Scrum in a Large Project, IEEE International Conference on Global Software Engineering (ICGSE '08), 2008, pp. 87-95.

[29] J. M. Bhat, M. Gupta, S. N. Murthy, Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing, IEEE Software Vol. 23 No. 5, 2006, pp. 38-44.

[30] A. Piri, T. Niinimaki, C. Lassenius, Descriptive Analysis of Fear and Distrust in Early Phases of GSD Projects, Fourth IEEE International Conference on Global Software Engineering (ICGSE '09), 2009, pp. 105-114.

[31] R. Urdangarin, P. Fernandes, A. Avritzer, D. Paulish, Experiences with Agile Practices in the Global Studio Project, IEEE International Conference on Global Software Engineering (ICGSE '08), 2008, pp. 77-86.

[32] D. Damian, Stakeholders in Global Requirements Engineering: Lessons Learned from Practice, IEEE Software Vol. 24 No. 2, 2007, pp. 21-27.

[33] D. Damian, F. Lanubile, T. Mallardo, On the Need for Mixed Media in Distributed Requirements Negotiations, IEEE Transactions on Software Engineering Vol. 34 No. 1, 2008, pp. 116-132.

[34] N. G. Lester, F. G. Wilkie, Evaluating UML tool support for effective coordination and communication across geographically disparate sites, The 12th International Workshop on Software Technology and Engineering Practice (STEP), 2004.

[35] W. J. Lloyd, M. B. Rosson, J. D. Arthur, Effectiveness of elicitation techniques in distributed requirements engineering, IEEE Joint International Conference on Requirements Engineering Proceedings, 2002, pp. 311-318.

[36] R. Zanoni, J. L. N. Audy, Project management model for a physically distributed software development environment, Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003.

[37] B. Al-Ani, D. Redmiles, In Strangers We Trust? Findings of an Empirical Study of Distributed Teams, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 121-130.

[38] B. Al-Ani, E. H. Keith, A Comparative Empirical Study of Communication in Distributed and Collocated Development Teams, IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 35-44.

[39] B. Al-Ani, D. Redmiles, Investigating Decision Making Processes in Distributed Development Teams: Findings of a comparative Empirical Study, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 51-60.

[40] R. G. Gowda, D. J. Polzella, Comparison of Selected Survey Instruments for Software Team Communication Research, International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 43-54.

[41] A. Lamersdorf, J. Munch, D. Rombach, A Survey on the State of the Practice in Distributed Software Development: Criteria for Task Allocation, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 41-50.

[42] A. H. Dutoit, J. Johnstone, B. Bruegge, Knowledge scouts: Reducing communication barriers in a distributed software development project, Eighth Asia-Pacific Software Engineering Conference (APSEC ‘01), 2001, pp. 427- 430.

[43] A. Lamersdorf, Empirically-Based Decision Support for Task Allocation in Global Software Development, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 281-284.

[44] C. Ebert, P. Neve, Surviving global software development, IEEE Software, Vol. 18 No. 2, 2001, pp. 62-69.

[45] V. Vitale, IBM Software Development Leveraging Geographically Distributed Teams - The Interactive Solution Marketplace 2.0 (ISM) Case Study, International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 259-260.

[46] T. Nguyen, T. Wolf, D. Damian, Global Software Development and Delay: Does Distance Still Matter? IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 45-54.

[47] A. Boden, G. Avram, L. Bannon, V. Wulf, Knowledge Management in Distributed Software Development Teams - Does Culture Matter? Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 18-27.

[48] A. Boden, G. Avram, Bridging knowledge distribution - The role of knowledge brokers in distributed software development teams, ICSE Workshop on Cooperative and Human Aspects on Software Engineering (CHASE '09), 2009, pp. 8-11.

[49] V. Casey, Leveraging or Exploiting Cultural Difference? Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 8-17.

[50] R. Tang, A. E. Hassan, Y. Zou, Techniques for Identifying the Country Origin of Mailing List Participants, 16th Working Conference on Reverse Engineering (WCRE '09), 2009, pp. 36-40.

[51] B. Lings, B. Lundell, P. J. Agerfalk, B. Fitzgerald, A reference model for successful Distributed Development of Software Systems, Second IEEE International Conference on Global Software Engineering (ICGSE ’07), 2007, pp. 130-139.

[52] M. Paasivaara, C. Lassenius, Could Global Software Development Benefit from Agile Methods? International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 109-113.

[53] P. Layzell, O. P. Brereton, A. French, Supporting collaboration in distributed software engineering teams, Seventh Asia-Pacific Software Engineering Conference Proceedings (APSEC ’00), 2000, pp. 38-45.

[54] G. Gaughan, B. Fitzgerald, M. Shaikh, An Examination of the Use of Open Source Software Processes as a Global Software Development Solution for Commercial Software Engineering, 35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA '09), 2009, pp. 20-27.

[55] D. Damian, Stakeholders in Global Requirements Engineering: Lessons Learned from Practice, IEEE Software Vol. 24 No. 2, 2007, pp. 21-27.

[56] J. Rudzki, I. Hammouda, T. Mikkonen, Ensuring Architecture Conventions in Multi-site Development, 32nd Annual IEEE International Computer Software and Applications (COMPSAC '08), 2008, pp. 339-346.

[57] A. Avritzer, A. Lima, An Empirical Approach for the Assessment of Scheduling Risk in a Large Globally Distributed Industrial Software Project, Fourth IEEE International Conference on Global Software Engineering (ICGSE 2009), 2009, pp. 341-346.

[58] M. Paasivaara, S. Durasiewicz, C. Lassenius, Using Scrum in Distributed Agile Development: A Multiple Case Study, Fourth IEEE International Conference on Global Software Engineering (ICGSE ’09), 2009, pp. 195-204.

[59] P. Wongthongtham, E. Chang, T. S. Dillon, Towards 'ontology'-based software engineering for multi-site software development, 3rd IEEE International Conference on Industrial Informatics (INDIN '05), 2005, pp. 362- 365.

[60] M. A. Babar, D. Winkler, S. Biffi, Evaluating the Usefulness and Ease of Use of a Groupware Tool for the Software Architecture Evaluation Process, First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), 2007, pp. 430-439.

[61] R. Bartholomew, Globally distributed software development using an immersive virtual environment, IEEE International Conference on Electro/Information Technology (EIT ’08), 2008, pp. 355-360.

[62] M. Cristal, D. Wildt, R. Prikladnicki, Usage of SCRUM Practices within a Global Company, IEEE International Conference on Global Software Engineering (ICGSE ’08), 2008, pp. 222-226.

[63] T. A. Finholt, Keynote: The Coming Age of Virtual Organizations: Applying Lessons from the Early History of Geographically-Dispersed Collaboration, IEEE International Conference on Global Software Engineering (ICGSE ‘08), 2008.

[64] N. S. Shami, N. Bos, Z. Wright, S. Hoch, K. Y. Kuan, J. Olson, G. Olson, An experimental simulation of multi-site software development, Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, 2004, pp. 255 – 266.

[65] D. Woit, K. Bell, Student Communication Challenges in Distributed Software Engineering Environments, ACM SIGCSE Vol. 37 No. 3, 2005, pp. 286 – 290.

[66] M. Cataldo, J. D. Herbsleb, Communication patterns in geographically distributed software development and engineers' contributions to the development effort, International Conference on Software Engineering, 2008, pp. 25-28.

[67] N. Ramasubbu, R. K. Balan, Globally distributed software development project performance: an empirical analysis, Foundations of Software Engineering, 2007, pp. 125–134.

[68] M. R. Thissen, J. M. Page, M. C. Bharathi, T. L. Austin, Communication tools for distributed software development teams, Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, 2007, pp. 28–35.

[69] M. Cataldo, J. D. Herbsleb, Communication networks in geographically distributed software development, Proceedings of the 2008 ACM conference on Computer supported cooperative work, 2008, pp. 579-588.

[70] A. Meneely, L. Williams, On preparing students for distributed software development with a synchronous, collaborative development platform, Proceedings of the 40th ACM technical symposium on Computer science education, 2009, pp. 529-533.

[71] D. Petkovic, G. Thompson, R. Todtenhoefer, Teaching practical software engineering and global software engineering: evaluation and comparison, ACM SIGCSE Vol. 38 No. 3, 2006, pp. 294–298.

[72] A. J. Birkin, R. W. Collins, Intercultural work assignments in global software development preparation for entry and re-entry, Proceedings of the ACM SIGCPR conference on Computer personnel research, 2002, pp. 72–75.

[73] P. Keil, D. J. Paulish, R. S. Sangwan, Cost estimation for global software development, Proceedings of the international workshop on Economics driven software engineering research, 2006, pp. 7–10.

[74] A. Gotel, V. Kulkarni, D. Phal, M. Say, C. Scharff, T. Sunetnanta, Evolving an infrastructure for student global software development projects: lessons for industry, Proceedings of the 2nd India software engineering conference, 2009, pp. 117-126.

[75] L. Pilatti, J. N. Audy, R. Prikladnicki, Software configuration management over a global software development environment: lessons learned from a case study, Proceedings of the international workshop on Global software development for the practitioner, 2006, pp. 45 – 50.

[76] V. Clerc, Towards Architectural Knowledge Management Practices for Global Software Development, Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge, 2008, pp. 23-28.

[77] M. A. Babar, J. M. Verner, P. T. Nguyen, Establishing and maintaining trust in software outsourcing relationships: An empirical investigation, Journal of Systems and Software Vol. 80 No. 9, 2007, pp. 1438-1449.

[78] A. Avritzer, D. Paulish, Y. Cai, K. Sethi, Coordination implications of software architecture in a global software development project, Journal of Systems and Software, In Press, Corrected Proof, 2010.

[79] M. A. Babar, B. Kitchenham, L. Zhu, I. Gorton, R. Jeffery, An empirical study of groupware support for distributed software architecture evaluation process, Journal of Systems and Software Vol. 79 No. 7, 2006, pp. 912-925.

[80] A. A. Bush, A. Tiwana, H. Tsuji, An empirical investigation of the drivers of software outsourcing decisions in Japanese organizations, Information and Software Technology Vol. 50 No. 6, 2008, pp. 499-510.

[81] M. D'Mello, T. Y. Eriksen, Software, sports day and sheera: Culture and identity processes within a global software organization in India, Information and Organization Vol. 20 No. 2, 2010, pp. 81-110.

[82] T. Galinac, Empirical evaluation of selected best practices in implementation of software process improvement, Information and Software Technology Vol. 51 No. 9, 2009, pp. 1351-1364.

[83] L. Hossain, D. Zhu, Social networks and coordination performance of distributed software development teams, The Journal of High Technology Management Research Vol. 20 No. 1, 2009, pp. 52-61.

[84] L. Layman, L. Williams, D. Damian, H. Bures, Essential communication practices for Extreme Programming in a global software development team, Information and Software Technology Vol. 48 No. 9, 2006, pp. 781-794.

[85] M. Niazi, M. A. Babar, J. M. Verner, Software Process Improvement barriers: A cross-cultural comparison, Information and Software Technology, Vol. 52 No. 11, 2010, pp. 1204 – 1216.

[86] R. Prikladnicki, J. N. Audy, Process models in the practice of distributed software development: A systematic review of the literature, Information and Software Technology Vol. 52, No. 8, 2010, pp. 779-791.

[87] A. Taweel, P. Brereton, Modelling software development across time zones, Information and Software Technology Vol. 48 No. 1, 2006, p. 1-11.

[88] M. Cataldo, S. Nambiar, The impact of geographic distribution and the nature of technical coupling on the quality of global software development projects, Journal of Software Maintenance and Evolution: Research and Practice, 2010.

[89] V. Casey, I. Richardson, Implementation of Global Software Development: A Structured Approach, Software Process Improvement and Practice Vol. 14, 2009, pp. 247–262.

[90] E. O. Conchuir, H. H. Olsson, P. J. Agerfalk, B. Fitzgerald, Benefits of Global Software Development: Exploring the Unexplored, Software Process Improvement and Practice Vol. 14 No. 5, 2009, pp. 201–212.

[91] O. Gotel, V., Kulkarni, M. Say, Scharff, T. Sunetnanta, Quality indicators on global software development projects: does 'getting to know you' really matter? Journal of Software Maintenance and Evolution: Research and Practice, 2010.

[92] M. A. Babar, A framework for groupware-supported software architecture evaluation process in global software development, Journal of Software Maintenance and Evolution: Research and Practice, 2010.

[93] A. E. Milewski, M. Tremanie, F. Kobler, R. Egan, S. Zhang, P. O. Sullivan, Guidelines for Effective Bridging in Global Software Engineering, IEEE International Conference on Global Software Engineering (ICGSE '08), 2008.

[94] B. S. Andaloussi, A. Braun, A Test Specification Method for Software Interoperability Tests in Offshore Scenarios: A Case Study, In the Proceedings of the 1st International Conference on Global Software Engineering (ICGSE '06), 2006, pp. 169–178.

[95] M. A. Babar, B. Kitchenham, R. Jeffery, Comparing distributed and face-to-face meetings for software architecture evaluation: A controlled experiment, Empirical Software Engineering Vol. 13 No. 1, 2008, pp. 39–62.

[96] M. A. Babar, B. Kitchenham, R. Jeffery, Distributed versus face-to-face meetings for architecture evolution: a controlled experiment, In the Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering, 2006, pp. 252–261.

[97] R. D. Battin, R. Crocker, J. Kreidler, K. Subramanian, Leveraging resources in global software development, IEEE Software Vol. 18 No. 2, 2001, pp. 70–77.

[98] A. Boden, B. Nett, V. Wulf, Coordination Practices in Distributed Software Development of Small Enterprises, In the Proceedings of the 2nd International Conference on Global Software Engineering (ICGSE '07), 2007, pp. 235–246.

[99] F. Calefato, D. Damian, F. Lanubile, An Empirical Investigation on Text-Based Communication, in Distributed Requirements Workshops of the 2nd International Conference on Global Software Engineering, 2007, pp. 3–11.

[100] V. Dalberg, E. Angelvik, D. R. Elvekrok, A. K. Fossberg, Cross-cultural collaboration in ICT procurement, In the International Conference on Software Engineering, Proceedings of the 2006 international workshop on Global software development for the practitioner, 2006, pp. 51–57.

[101] D. Damian, L. Izquierdo, J. Singer, I. Kwan, Awareness in the Wild: Why Communication Breakdowns Occur, In the Proceedings of the 2nd International Conference on Global Software Engineering (ICGSE '07), 2007, pp. 81–90.

[102] D. Damian, D. Zowghi, The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, In the Proceedings of the IEEE Joint International Conference on Requirements Engineering, 2002, pp. 319–328.

[103] C. R. B. De Souza, T. Hildenbrand, D. Redmiles, Toward Visualization and Analysis of Traceability Relationships in Distributed and Offshore Software Development Projects, Lecture Notes in Computer Science Vol. 4716, 2007, pp. 182–199.

[104] L. Harjumaa, Distributed software inspections—An experiment with Adobe Acrobat, In the Proceedings of the IASTED International Conference on Computer Science and Technology, 2003, pp. 26–31.

[105] J. D. Herbsleb, D. J. Paulish, M. Bass, Global software development at Siemens: Experience from nine project, In the proceedings of the 27th International Conference on Software Engineering, 2005, pp. 524–533.

[106] H. Holmström, B. Fitzgerald, P. J. Ågerfalk, E. O. Conchúir, Agile practices reduce distance in global software development, Information Systems Management Vol. 23 No. 3, 2006, pp. 7–18.

[107] R. Kommeren, P. Parviainen, Philips experiences in global distributed software development, Empirical Software Engineering Vol. 12 No. 6, 2007, p. 647–660.

[108] J. Kotlarsky, I. Oshri, K. Kumar, J. V. Hillegersberg, Globally distributed component-based software development: an exploratory study of knowledge management and work division, Journal of Information Technology Vol. 22 No. 2, 2007, pp. 161–173.

[109] S. Lasser, M. Heiss, Collaboration maturity and the offshoring cost barrier: The trade-off between flexibility in team composition and cross-site communication effort in geographically distributed development projects, In the Proceedings of the IEEE International Professional Communication Conference, 2005, pp. 718–728.

[110] J. Ma, J. Li, W. Chen, R. Conradi, J. Ji, C. Liu, An Industrial Survey of Software Outsourcing in China, published by LNCS Vol. 4589, 2007, pp. 5–19.

[111] N. B. Moe, D. Smite, Understanding lacking trust in global software teams: A multi-case study, in the proceedings of the 8th international PROFES conference, Lecture Notes in Computer Science, 2007, pp. 20–34.

[112] B. Nicholson, S. Sahay, Some political and cultural issues in the globalisation of software development: case experience from Britain and India, Information and Organization Vol. 11 No. 1, 2001, pp. 25–43.

[113] I. Oshri, J. Kotlarsky, L. P. Willcocks, Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects, The Journal of Strategic Information Systems Vol. 16 No. 1, 2007, pp. 25–49.

[114] P. Ovaska, M. Rossi, P. Marttiin, Architecture as a coordination tool in multi-site software development, Software Process Improvement and Practice Vol. 8 No. 4, 2003, pp. 233–247.

[115] R. Prikladnicki, J. L. N. Audy, R. Evaristo, Global software development in practice lessons learned, Software Process Improvement and Practice Vol. 8 No. 4, 2003, pp. 267–281.

[116] B. Ramesh, L. Cao, K. Mohan, P. Xu, Flexible and distributed software processes: old petunias in new bowls? Can distributed software development be agile, Communications of the ACM Vol. 49 No. 10, 2006, pp. 41–46.

[117] D. Smite, Project Outcome Predictions: Risk Barometer Based on Historical, in the Proceedings of the 2nd International Conference on Global Software Engineering (ICGSE '07), 2007, pp. 103–112.

[118] D. Smite, Global software development projects in one of the biggest companies in Latvia: Is geographical distribution a problem? Software Process Improvement and Practice Vol. 11 No. 1, 2006, pp. 61–76.

[119] D. Smite, A case study: Coordination practices in global software development, Lecture Notes in Computer Science Vol. 3547, 2005, pp. 234–244.

[120] D. Spinellis, Case studies: Global software development in the freeBSD project, in the International Conference on Software Engineering Proceedings of the 2006 international workshop on Global software development for the practitioner, 2006, pp. 73–79.

[121] K. J. Stewart, G. Sanjay, The moderating role of development stage in free/open source software project performance, Software Process Improvement and Practice Vol. 11 No. 2, 2006, pp. 177–191.

[122] A. Taweel, P. Brereton, Developing software across time zones: An exploratory empirical study, Informatica (Ljubljana) Vol. 26 No. 3, 2002, pp. 333–344.

[123] L. Taxen, An integration centric approach for the coordination of distributed software development projects, Information and Software Technology Vol. 48 No. 9, 2006, pp. 767–780.

[124] A. Välimäki, J. Kääriäinen, Requirements Management Practices as Patterns for Distributed Product Management, published by LNCS Vol. 4589, 2007, pp. 188–200.

[125] V. D. Walle, C. Campbell, F. P. Deek, The Impact of Task Structure and Negotiation Sequence on Distributed Requirements Negotiation Activity, Conflict, and Satisfaction, published by LNCS Vol. 4495, 2007, pp. 381–394.

[126] M. Paasivaara, Coaching Global Software Development Projects, Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering ICGSE'11, 2011.

[127] Y. Dittrich, R. Giuffrida, Exploring the Role of Instant Messaging in a Global Software Development Project, Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering ICGSE'11, 2011.

[128] C. Scharff, Guiding global software development projects using Scrum and Agile with quality assurance, Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering ICGSE'11, 2011.

[129] E. Hossain, P. L. Bannerman, 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 ICSSP'11, 2011.

[130] A. Gupta, N. T. Denny, K. O. Toole, R. Bondade, D. Halder, Global software development using the 24-Hour Knowledge Factory, International Journal of Computer Applications in Technology Vol. 40 No. 3, 2011.

[131] J. Munch, Risk Management in Global Software Development Projects: Challenges, Solutions, and Experience, Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering WorkshopICGSE-W '11, 2011.

[132] B. M. Yildiz, B. Tekinerdogan, Architectural Viewpoints for Global Software Development, Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering WorkshopICGSE-W '11, 2011.

[133] F. C. Serce, K. M. Swigger, F. N. Alpaslan, R. Brazile, G. Dafoulas, V. Lopez-Cabrera, Exploring the communication behaviour among global software development learners, International Journal of Computer Applications in Technology Vol. 40 No. 3, 2011.

[134] D. A. Tamburri, P. Lago, Supporting communication and cooperation in global software development with agile service networks, Proceedings of the 5th European conference on Software architecture ECSA'11, 2011.

[135] S. I. Hashmi, V. Clerc, M. Razavian, C. Manteli, D. A. Tamburri, P. Lago, E. D. Nitto, I. Richardson, Using the Cloud to Facilitate Global Software Development Challenges, Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering Workshop ICGSE-W '11, 2011.

[136] D. Mishra, A. Mishra, A review of non-technical issues in global software development, International Journal of Computer Applications in Technology Vol. 40 No. 3, 2011.

[137] C. Deiters, C. Herrmann, R. Hildebrandt, E. Knauss, M. Kuhrmann, A. Rausch, B. Rumpe, K. Schneider, GloSE-Lab: Teaching Global Software Engineering, 6th IEEE Conference on Global Software Engineering ICGSE, 2011, pp. 156-160.

[138] G. K. Hanssen, D. Smite, N. B. Moe, Signs of Agile Trends in Global Software Engineering Research: A Tertiary Study, 6th IEEE Conference on Global Software Engineering Workshop ICGSEW, 2011, pp. 17-23.

[139] V. Clerc, P. Lago, H. Vliet, Architectural Knowledge Management Practices in Agile Global Software Development, 6th IEEE Conference on Global Software Engineering Workshop ICGSEW, 2011, pp. 1-8.

[140] P. Tell, M. A. Babar, Requirements for an Infrastructure to Support Activity-Based Computing in Global Software Development, 6th IEEE Conference on Global Software Engineering Workshop ICGSEW, 2011, pp. 62-69.

[141] R. Bavani, Governance Patterns in Global Software Engineering: Best Practices and Lessons Learned, 6th IEEE Conference on Global Software Engineering ICGSE, 2011, pp. 50-54.

[142] M. Helen, N. Nahar, Key barriers of globally distributed software products development, Proceedings of Technology Management in the Energy Smart World (PICMET), 2011, pp. 1-13.

[143] P. Tell, M. A. Babar, Supporting activity based computing paradigm in global software development, IEEE/ACM International Conference on Automated Software Engineering (ASE), 2011, pp. 508-511.

[144] B. Wilson, K. Ceuppens, Reverse Offshore Outsourcing Experiences in Global Software Engineering Projects, 6th IEEE Conference on Global Software Engineering ICGSE, 2011, pp. 55-59.

[145] D. Smite, C. Wohlin, A Whisper of Evidence in Global Software Engineering, IEEE Software Vol. 28 No. 4, 2011, pp. 15-18.

[146] K. Swigger, M. Hoyt, F. C. Serçe, V. Lopez, F. N. Alpaslan, The temporal communication behaviors of global software development student teams, Computers in Human Behavior Vol. 28 No. 2, 2011, pp. 384-392.

[147] J. S. Persson, L. Mathiassen, I. Aaen, Agile distributed software development: enacting control through media and context, Journal of Software Maintenance and Evolution: Research and Practice, 2011.

[148] F. Q. B. Da Silva, R. Prikladnicki, A. C. C. França, C.V. F. Monteiro, C. Costa, R. Rocha, An evidence-based model of distributed software development project management: results from a systematic mapping study, Journal of Software Maintenance and Evolution: Research and Practice, 2011.

[149] S. Jalali, C. Wohlin, Global software engineering and agile practices: a systematic review, Journal of Software Maintenance and Evolution: Research and Practice, 2011.

[150] P. Lago, H. Muccini, M. A. Babar, An empirical study of learning by osmosis in global software engineering, Journal of Software Maintenance and Evolution: Research and Practice, 2011.

[151] M. Jimenez, M. Piattini, A. Vizcaino, Challenges and Improvements in Distributed Software Development: A systematic Review, Advances in Software Engineering No. 9, 2009.

[152] D. Smite, D. Wohlin, R. Feldt, T. Gorschek, Empirical evidence in global software engineering: a systematic review, Empirical Software Engineering Vol. 15 No. 1, 2010, pp. 91-118.

[153] A. Braun, A. H. Dutoit, A. G. Harrer, B. Brüge, iBistro: a learning environment for knowledge construction in distributed software engineering courses, in Proceedings of the Ninth Asia-Pacific Software Engineering Conference, IEEE Computer Society, 2002, pp. 197–203.

[154] D. Damian, A. Hadwin, B. Al-Ani, Instructional design and assessment strategies for teaching global software development: a framework, in International Conference on Software Engineering, ACM Press, New York, NY, USA, 2006.

[155] C. Ebert, Global Software Engineering: Distributed Development, Outsourcing, and Supplier Management, Wiley, IEEE Computer Society Books, Los Alamitos, USA, 2010.

[156] B. Bruegge, A. H. Dutoit, T. Wolf, Sysiphus: enabling informal collaboration in global software development, in International Conference on Global Software Engineering, Florianopolis, Brazil, 2006, pp. 139–148.

[157] R. Bartholomew, Evaluating a networked virtual environment for globally distributed avionics software development, in Proceedings of the IEEE International Conference on Global Software Engineering, IEEE Computer Society, 2008, pp. 227–231.

[158] B. Sengupta, S. Chandra, V. Sinha, A research agenda for distributed software development, Proceedings of the 28th international conference on Software engineering, 2006, pp. 731-740.

[159] R. Thissen, M. C. Bharathi, T. L. Austin, Communication tools for distributed software development teams, in Proceedings of the 2007 ACM SIGMIS CPR Conference on Computer Personnel Research: The Global Information Technology Workforce, ACM, St. Louis, Missouri, USA, 2007, pp. 28–35.

[160] A. Lamersdorf, J. Münch, A. Fernández-del Viso Torre, C. Sánchez, D. Rombach, Estimating the Effort Overhead in Global Software Development, In Proceedings of the IEEE International Conference on Global Software Engineering, Princeton, USA, 2010.

[161] L. Bendix, J. Magnusson, M. Pendleton, Configuration Management Stories from the Distributed Software Development Trenches, IEEE Seventh International Conference on Global Software Engineering, 2012.

[162] F. Calefato, D. Damian, F. Lanubile, Computer-mediated communication to support distributed requirements elicitations and negotiations tasks, Empirical Software Engineering Vol. 17, 2012, pp. 640–674

[163] D. Smite, C. Wohlin, Z. Galvina, R. Pirkladnicki, An empirically based terminology and taxonomy for global software engineering, Empirical Software Engineering, 2012.

[164] J. Verner, O. Brereton, B. Kitchenham, M. Turner, M. Niazi, Risk Mitigation Advice for Global Software Development from Systematic Literature Reviews, Journal of Information and Software Technology Vol. 56 No. 1, 2014, pp. 54-78.

[165] M. Passivaara, C. Lassenius, D. Damian, P. Raty, A. Schroter, Teaching students global software engineering skills using distributed scrum, in Proceedings of the 2013 International Conference on Software Engineering, 2013, pp 1128-1137.

[166] J. Schenk, Evaluating awareness information in distributed collaborative editing by software-engineers, International Workshop on User Evaluation for Software Engineering Researchers (USER), 2012.

[167] R. El-Bahey, A. Zeid, Establishing a globally distributed software development system in academic settings: An ergonomic perspective, In IEEE International on Professional Communication, 2013, pp. 1-8.

[168] J. Baldwin, D. Damian, Tool usage within a globally distributed software development course and implications for teaching, 3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development, 2013, pp. 15-19.

[169] M. Qahtani, B. Wills, M. Gravell, Customising software products in distributed software development A model for allocating customisation requirements across organisational boundaries, 2013 International Conference on Information Society (i-Society), 2013, pp. 92-98.

[170] D. Wickramaarachchi, R. Lai, A method for work distribution in Global Software Development, IEEE 3rd International on Advance Computing Conference, 2013, pp. 1443-1448.

[171] A. Dubey, J. Hudepohl, Towards Global Deployment of Software Engineering Tools, IEEE 8th International Conference on Global Software Engineering (ICGSE), 2013, pp. 129-133

[172] L. Fortaleza, R. Costa Vieira, O. Olimpio Matos Junior, R. Prikladnicki, T. Conte, Using Distributed Software Development in the improvement of communication and collaboration skills in SE courses: An observational study, IEEE 26th Conference on Software Engineering Education and Training (CSEE&T), 2013, pp. 139-148.

[173] E. Milewski, C. Serce, K. Swigger, Quantitative and qualitative analysis of globally distributed team collaboration in design and development of software, The 6th International Conference on Human System Interaction (HSI), 2013, pp. 443-450

[174] M. Azzeh, Software cost estimation based on use case points for global software development, 5th International Conference on Computer Science and Information Technology (CSIT), 2013, pp. 214-218.

[175] G. Lee, A. Espinosa, H. DeLone, Task Environment Complexity, Global Team Dispersion, Process Capabilities, and Coordination in Software Development, IEEE Transactions on Software Engineering Vol. 39 No. 12, 2013, pp. 1753-1771

[176] S. Dorairaj, J. Noble, Agile Software Development with Distributed Teams: Agility, Distribution and Trust, Agile Conference (AGILE), 2013, pp. 1-10.

[177] M. Niazi, S. Mahmood, M. Alshayeb, R. Rehan, K. Faisal, N. Cerpa, Challenges of project management in Global Software Development: Initial results, Science and Information Conference (SAI), 2013, pp. 202-206.

[178] J. Monasor, A. Vizcaino, M. Piattini, J. Noll, S. Beecham, Towards a Global Software Development Community Web: Identifying Patterns and Scenarios, IEEE 8th International Conference on Global Software Engineering Workshops (ICGSEW), 2013, pp. 41-43.

[179] S. Beecham, P. OLeary, I. Richardson, S. Baker, J. Noll, Who Are We Doing Global Software Engineering Research For? IEEE 8th International Conference on Global Software Engineering, 2013, pp. 41-50.

[180] A. Kumar, A., Thangavelu, K., (2013) Factors affecting the outcome of Global Software Development projects: An empirical study, International Conference on Computer Communication and Informatics (ICCCI) pp. 1-10

[181] B. Marques, R. Carvalho, R. Rodrigues, T. Conte, R. Prikladnicki, S. Marczak, An Ontology for Task Allocation to Teams in Distributed Software Development, IEEE 8th International Conference on Global Software Engineering, 2013, pp. 21-30.

[182] A. Babar, M. Zahedi, Understanding Structures and Affordances of Extended Teams in Global Software Development, IEEE 8th International Conference on Global Software Engineering, 2013, pp. 226-235.

[183] Y. Kharitonov, A Software Architecture for High-Level Development of Component-Based Distributed Virtual Reality Systems, IEEE 37th Annual Computer Software and Applications Conference (COMPSAC), 2013, pp. 696 – 705.

[184] M. Zahedi, M. A. Babar, Knowledge Sharing for Common Understanding of Technical Specifications through Artifactual Culture, EASE '14: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ACM, 2014.

[185] M.A. Chauhan, A Reference Architecture for Providing Tools as a Service to Support Global Software Development, Proceedings of the WICSA 2014 Companion Volume, ACM, 2014.

[186] S. A. Licorishm, S. G. MacDonell, Personality Profiles of Global Software Developers, EASE '14: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ACM, 2014.

[187] M. Zahedi, M. A. Babar, Towards an Understanding of Enabling Process Knowing in Global Software Development: A Case Study, ICSSP 2014: Proceedings of the 2014 International Conference on Software and System Process, ACM, 2014.

[188] P. Bjorn, J. Bardram, G. Avram, L. Bannon, A. Boden, D. Redmiles, C. R. B. De Souza, V. Wulf, Global Software Development in a CSCW Perspective, CSCW Companion '14: Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing, ACM, 2014.

[189] S. Matthiesen, P. Bjorn, L. M. Petersen, Figure Out How to Code with the Hands of Others: Recognizing Cultural Blind Spots in Global Software Development, CSCW '14: Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing, ACM, 2014.

[190] M. J. Monasor, A. Vizcaino, M. Piattini, J. Noll, S. Beecham, Assessment Process for a Simulation-based Training Environment in Global Software Development, ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education, ACM, 2014.

[191] R. Giuffrida, Y. Dittrich, How Social Software Supports Cooperative Practices in a Globally Distributed Software Project, CHASE 2014: Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, ACM, 2014.

[192] S. Beecham, J. Noll, I. Richardson, Using Agile practices to solve Global Software Development problems – A Case Study, 2014 IEEE International Conference on Global Software Engineering Workshops, 2014, pp. 5-10.

[193] M. A. Chauhan, M. A. Babar, Towards a Reference Architecture to Provision Tools as a Service for Global Software Development, IEEE/IFIP Conference on Software Architecture, 2014, pp. 167-170.

[194] C. Manteli, B. Hooff, H. Vliet, W. Duinkerken, Overcoming challenges in Global Software Development: The role of Brokers, IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), 2014, pp. 1-9.

[195] A. Alsri, S. Almuhammadi, S. Mahmood, A Model for Work Distribution in Global Software Development Based on Machine Learning Techniques, Science and Information Conference, 2014, pp. 399-403.

[196] M. J. Monasor, A. Vizcaino, M. Piattini, J. Noll, S. Beecham, J. Parkes, Global Software Development Education: A commercial perspective from a case study, IEEE 9th International Conference on Global Software Engineering (ICGSE), 2014, pp. 173 – 182.

[197] H. H. Khan, M. N. Mahrin, S. Chuprat, Situational Requirement Engineering Framework for Global Software Development, International Conference on Computer, Communications, and Control Technology (I4CT), 2014, pp. 224 – 229.

[198] J. Noll, I. Richardson, S. Beecham, Patternizing GSD Research: Maintainable Decision Support for Global Software Development, IEEE 9th International Conference on Global Software Engineering, 2014, pp. 110-115.

[199] J. Noll, GSD Sim: A Global Software Development Game, IEEE International Conference on Global Software Engineering Workshops, 2014, pp. 15-20.

[200] R. Britto, V. Freitas, E. Mendes, M. Usman, Effort Estimation in Global Software Development: A Systematic Literature Review, IEEE 9th International Conference on Global Software Engineering, 2014, pp. 135-144.

[201] T. Dingsoyr, D. Smite, Managing Knowledge in Global Software Development Projects, IT Professional Vol. 16 No. 1, 2014, pp. 22-29.

[202] M. Usman, F. Azam, N. Hashmi, Analysing and Reducing Risk Factor in 3-C's Model Communication Phase used in Global Software Development, International Conference on Information Science and Applications (ICISA), 2014, pp. 1 – 4.

[203] G. Tjørnehøj, M. B. Balogh, C. Iversen, S. Sørensen, Designing Project Management for Global Software Development Informality through Formality, International Working Conference on Transfer and Diffusion of IT, 2014, pp. 113–132.

[204] I. Smirnova, J. M¨unch, M. Stupperich, A Canvas for Establishing Global Software Development Collaborations, International Conference on Information and Software Technologies, 2014, pp. 73–93.

REFERENCES

[205] A. Abran, J. W. Moore, P. Bourque, R. Dupuis, Guide to the software engineering body of knowledge, *IEEE Computer Society*, 2004.

[206] P. J. Ågerfalk, B. Fitzgerald, H. Holmström, B. Lings, B. Lundell, E. O. Conchúir, A Framework for Considering Opportunities and Threats in Distributed Software Development, *in Proceedings of the International Workshop on Distributed Software Development, Austrian Computer Society, 2005, pp. 47-61*.

[207] R. Akbar, M. F. Hassan, A. Abdullah, S. Safdar, M. A. Qureshi, Directions and Advancements in Global Software Development: A Summarized Review of GSD and Agile Methods, *Research Journal of Information Technology*, 2011.

[208] D. Atkins, M. Handel, J. Hersleb, A. Mockus, D. Perry, G. Wills, Global Software Development, The Bell Labs Co-laboratory in International Conference on Software Engineering, Toronto, 2001.

[209] R. D. Battin, R. Crocker, J. Kreidler, K. Subramanian, Leveraging resources in global software development, *IEEE Software*, Vol. 8 No. 2, 2001, pp. 70–77.

[210] R. Britto, V. Freitas, E. Mendes, M. Usman, Effort estimation in global software development: A systematic literature review, *International Conference on Global Software Engineering (ICGSE), 2014, pp. 135-144.*

[211] E. Carmel, P. Abbott, Why 'Near shore' means that distance matters, *Communication of the ACM*, Vol. 50 No. 10, 2007, pp. 40–46.

[212] B. Cheng, J. Atlee, Research Directions in Requirements Engineering, *in the International Conference on Software Engineering, Future of Software Engineering, 2007, pp. 285-303.*

[213] T. Clear, S. G. MacDonell, Understanding technology use in global virtual teams: Research methodologies and methods, *special issue on Studying Work Practices in Global Software Engineering in Information and Software Technology*, Vol. 53 No, 9, 2011, pp. 994-1011.

[214] E. O. Conchuir, H. Holmström, P. J. Ågerfalk, B. Fitzgerald, Exploring the assumed benefits of global software development, *in Proceedings of the 1st International Conference on Global Software Engineering, 2006, 159–168.*

[215] C. Costa, C. Cunha, R. Rocha, A. C. C. França, F. Q. Silva, R. Prikladnick, Models and Tools for Managing Distributed Software Development: A Systematic Literature Review, 14th International Conference on Evaluation and Assessment in Software Engineering (EASE), 2010.

[216] B. Curtis, The Global Pursuit of Process maturity, *IEEE Software*, Vol. 17 No. 4, 2000, pp. 76–78.

[217] D. Damian, D. Moitra, Global software development: how far have we come? *IEEE Software*, Vol. 23 No. 5, 2006, pp. 7–19.

[218] D. Damian, D. Zowghi, Requirements Engineering Challenges in Multi-site Software Development Organizations, *Requirements Engineering Journal*, Vol. 8, 2003, pp. 149-160.

[219] O. Dieste, A. G. Padua, Developing search strategies for detecting relevant experiments for systematic reviews, *in First International Symposium on Empirical Software Engineering and Measurement (ESEM), 2007, pp. 215– 224.*

[220] C. Ebert, *Global Software Engineering: Distributed Development, Outsourcing, and Supplier Management, Wiley*, IEEE Computer Society Books, Los Alamitos, USA, 2010.

[221] C. Ebert, P. D. Neve, Surviving Global Software Development, *IEEE Software*, Vol. 18 No. 2, 2001, pp. 62–69.

[222] T. L. Friedman, *The world is flat: brief history of the 21st century*, New York, 2005.

[223] N. Gronau, J. Fröming, S. Schmid, U. Rüssbüldt, Approach for requirement oriented team building in industrial processes, *Computers in Industry* Vol. 58, 2007, pp. 179–187.

[224] S. Haq, M. Raza, A. Zia, M. N. A., Khan, Issues in Global Software Development:A Critical Review, Software *Engineering & Applications*, Vol. 4, 2011, pp. 590-595.

[225] R. Heeks, S. Krishna, B. Nicholson, S. Sahay, Synching or sinking: global software outsourcing relationships, *IEEE Software*, Vol. 18, 2001, pp. 54-60

[226] J. D. Herbsleb, D. Moitra, Global software development, *IEEE Software*, Vol. 18 No. 2, 2001, pp. 16-20.

[227] A. Hernández-López, R. Colomo-Palacios, A. García-Crespo, P. Soto-Acosta, Trust Building Process for Global Software Development Teams: A review from the Literature, *International Journal of Knowledge Society Research (IJKSR)*, 2010.

[228] E. Hossain, M. A. Babar, H. Paik, Using Scrum in Global Software Development: A Systematic Literature Review, *Fourth IEEE International Conference on Global Software Engineering, 2009.*

[229] E. Hossain, P. L. Bannerman, D. R. Jeffery, Scrum Practices in Global Software Development: A Research Framework, LNCS 6759, 2011, pp. 88–102.

[230] M. Ilyas, S. U. Khan, Software Integration Challenges in Global Software Development Environment: A Systematic Literature Review Protocol, *IOSR Journal of Computer Engineering (IOSRJCE)* ISSN: 2278-0661 Vol. 1 No. 6, 2012.

[231] S. Jalalai, C. Wohlin, Agile Practices in Global Software Engineering – A Systematic Map, *5th IEEE International Conference on Global Software Engineering (ICGSE), 2010, pp. 45–54.*

[232] S. Jalali, C. Wohlin, Global software engineering and agile practices: a systematic review, *Journal of Software Maintenance and Evolution: Research and Practice*, 2011.

[233] M. Jimenez, M. Piattini, Problems and Solutions in Distributed Software Development: A Systematic review, Software Engineering Approaches for Offshore and Outsourced Development Lecture Notes in Business information Processing, Vol. 16, 2009, pp. 107-125.

[234] M. Jimenez, M. Piattini, A. Vizcaino, Challenges and Improvements in Distributed Software Development: A Systematic Review, *Advances in Software Engineering*, No. 9, 2009.

[235] A. Johri, Sociomaterial bricolage: The creation of location-spanning work practices by global software developers, *special issue on Studying Work Practices in Global Software Engineering, in Information and Software Technology* Vol. 53 No. 9, 2011, pp- 955-968.

[236] A. A. Khan, B. Basri, P. D. D. Dominic, F. E. Amin, Communication Risks and Best Practices in Global Software Development during Requirements Change Management: A Systematic Literature Review Protocol, *Research Journal of Applied Sciences, Engineering and Technology* Vol. 6 No. 19, 2012, pp. 3514-3519.

[237] S. U. Khan, M. Niazi, R. Ahmad, Factors influencing clients in the selection of offshore software outsourcing vendors: An exploratory study using a systematic literature review, *The Journal of Systems and Software* Vol.84, 2011, pp. 686–699

[238] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering – A systematic literature review, *Journal of Information and Software Technology* Vol. 51 No. 1, 2009.

[239] J. Lipnack, J. Stamps, Virtual *Teams: People Working Across Boundaries with Technology*, New York: John Wiley & Sons, 2000.

[240] B. Marques, R. Rodrigues, T. Conte, Systematic Literature Reviews in Distributed Software Development: A Tertiary Study, IEEE *Seventh International Conference on Global Software Engineering, 2012.*

[241] B. Martin, *An Introduction to Medical Statistics*, 3rd edition, Oxford Medical Publications, 2000.

[242] M. B. Miles, A. M. Huberman, *Qualitative Data Analysis: an Expanded Sourcebook*, Second edition, Thousand Oaks, 1994.

[243] D. Mishra, A. Mishra, R. Colomo-Palacios, Global Software Development and Quality Management: A Systematic Review, LNCS 8186, 2013, pp. 302–311.

[244] S. Nidhra, M. Yanamadala, W. Afzal, R. Torkar, Knowledge transfer challenges and mitigation strategies in global software development—A systematic literature review and industrial validation, *International Journal of Information Management*, Vol. 33 No. 2, 2013.

[245] I. Nonaka, H. Takeuchi, *The Knowledge-Creating Company*, Oxford University Press, New York, 1995.

[246] I. Nurdiani, R. Jabangwe, D. Smite, D. Damian, Risk Identification and Risk Mitigation Instruments for Global Software Development: Systematic Review and Survey Results, *Sixth IEEE International Conference on Global Software Engineering Workshop, 2011, pp. 36-41.*

[247] T. Poikolainen, J. Paananen, Performance criteria in inter-organizational global software development projects, *in Proceedings of International Conference on Global Software Engineering ICGSE, 2007, pp. 60–70.*

[248] R. Prikladnicki, J. L. N. Audy, D. Damian, T. C. Oliveira, Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring, *Second IEEE International Conference on Global Software Engineering, 2007, pp. 262-274.*

[249] R. Prikladnicki, J. N. Audy, Process models in the practice of distributed software development: A systematic review of the literature, *Information and Software Technology*, Vol. 52 No. 8, 2010, pp. 779-791.

[250] P. N. Robillard, P. Kruchten, P. D'Astous, *Software Engineering Process with the UPEDU*, Addison Wesley: Pearson Education, 2002.

[251] J. P. Rodríguez, A. Vizcaíno, M. Piattini, S. Beecham, Tools used in Global Software Engineering: A systematic mapping review, *Information and Software Technology* Vol. 54, 2012, pp. 663–685

[252] S. Sahay, B. Nicholson, S. Krishna, *Global IT Outsourcing: Software Development across Borders*, Cambridge University Press, 2003.

[253] S. Schneider, R. Torkar, T. Gorschek, Solutions in global software engineering: A systematic literature review, *International Journal of Information Management*, 2012.

[254] C. Shewell, *Good Business Communicates Across Cultures: A Practical Guide to Communication*, Bristol: Mastek Publications, 2000.

[255] S. V. Shrivastava, H. Date, Distributed Agile Software Development: A Review, *Journal Of Computer Science And Engineering*, Vol. 1 No. 1, 2010.

[256] F. Q. B. Silva, C. Costa, A. C. C. Franca, R. Prikladinicki, Challenges and Solutions in Distributed Software Development Project Management: a Systematic Literature Review, *International Conference on Global Software Engineering, 2010, pp. 87-96.*

[257] D. Smite, D. Wohlin, R. Feldt, T. Gorschek, Reporting empirical research in global software engineering: a Classification Scheme, *in Proceedings of International Conference on Global Software Engineering ICG-SE, 2008, pp. 173–181.*

[258] D. Smite, D. Wohlin, R. Feldt, T. Gorschek, Empirical evidence in global software engineering: a systematic review, *Empirical Software Engineering*, Vol. 15 No. 1, 2010, pp. 91-118.

[259] H. Spanjers, M. T. Huurneç, B. Graaf, M. Lormans, D. Bendas, R. Van, Tool support for distributed software engineering, *in International Conference on Global Software Engineering*, Florianopolis, Brazil, 2006, pp. 187–198.

[260] R. N. Taylor, N. Medvidovic, E. M. Dashofy, *Software Architecture: Foundations, Theory, and Practice*, John Wiley & Sons, 2008.

[261] A. Tiwari, P. K. Patel, M. Darbari, Global Prospect of Distributed Agile Software Development: A Review, *International Conference on Recent Trends In Computing and Communication Engineering, 2013.*

[262] J. M. Verner, O. P. Brereton, B. Kitchenham, M. Turner, M. K. Niazi, Risk Mitigation Advice for Global Software Development from Systematic Literature Reviews ISSN 1353-7776., 2012

[263] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, M. K. Niazi, Systematic literature reviews in global software development: A tertiary study, Proceedings of the EASE - Published by the IET, 2012.

[264] P. Wongthongtham, E. Chang, T. Dillon, Multi-site Distributed Software Development: Issues, Solutions, and Challenges, *proceedings of the 2007 International Conference on Computational Science and its Applications, 2007, pp. 346-359.*