Information & Management xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Information & Management

journal homepage: www.elsevier.com/locate/im



# Empirical studies of geographically distributed agile development communication challenges: A systematic review

Yehia Ibrahim Alzoubi [a,*], Asif Qumer Gill [a], Ahmed Al-Ani [b]

[a] School of Software, University of Technology Sydney, Sydney, Australia
[b] School of Electrical, Mechanical and Mechatronic Systems, University of Technology Sydney, Sydney, Australia

ARTICLE INFO

*Article history:*
Received 21 December 2014
Received in revised form 18 July 2015
Accepted 13 August 2015
Available online xxx

*Keywords:*
Geographically distributed agile development
Communication challenges
Communication techniques
Agile approaches
Systematic literature review

ABSTRACT

There is increasing interest in studying and applying geographically distributed agile development (GDAD). Much has been published on GDAD communication. There is a need to systematically review and synthesize the literature on GDAD communication challenges. Using the SLR approach and applying customized search criteria derived from the research questions, 21 relevant empirical studies were identified and reviewed in this paper. The data from these papers were extracted to identify communication challenges and the techniques used to overcome these challenges. The findings of this research serve as a resource for GDAD practitioners and researchers when setting future research priorities and directions.



## 1. Introduction

The combination of geographically distributed development and agile practices [5], known as "geographically distributed agile development" (GDAD), seems to offer many benefits, such as low production cost, the opportunity to involve the most talented developers around the world and faster time to market [2,25,35]. Specifically, GDAD refers to agile development that involves teams or/and team members working together to accomplish project goals from different geographic locations [29,50]. GDAD teams or team members may be "locally distributed" in different physical locations within the same country or "globally distributed" around the world in different time-zones or countries [42,50]. Despite the abovementioned lucrative benefits, GDAD also involves many challenges [2,12,27,29,45]. Among these challenges, communication between distributed teams and customers is considered to be the most important [4,18,24,26]. According to Herbsleb and Moitra [27], poor communication (e.g., delivering an incomplete, inaccurate or inadequate message) is a major risk to GDAD. Similarly, human communication and knowledge sharing are highlighted as important concerns for GDAD [4,18,24].

Malone and Crowston [38] defined communication as the management of relationships between different concerned parties. Communication also refers to the process of exchanging information between senders and receivers [41]. These definitions draw our attention to the importance of the effectiveness of communication (i.e., delivering clear and understandable message [13,14,36]) between the parties included in agile development. Clark and Brennan [14] defined communication as a collective activity that "requires the coordinated action of all the participants. Grounding is crucial for keeping that coordination on track." Communication grounding facilitates efficient communication (i.e., rapid communication with minimum effort [14,36]) and effective communication [43].

Agility, the core of agile development, identifies how the agile team should communicate and respond to requirement's changes. Lee and Xia [36, p. 90] defined software development agility "as the software team's capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle." Qumer and Henderson-Sellers [49, p. 281] define agility as "a persistent behaviour or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal

* Corresponding author. Tel.: +61 404759757.
*E-mail addresses:* yehia.i.alzoubi@student.uts.edu.au (Y.I. Alzoubi), asif.gill@uts.edu.au (A.Q. Gill), Ahmed.Al-Ani@uts.edu.au (A. Al-Ani).

http://dx.doi.org/10.1016/j.im.2015.08.003
0378-7206/

and external environment." Conboy [15] systematically examined various agility definitions and facets from related disciplines and provided by far the most comprehensive definition of software development agility. He defined software development agility as a continued readiness "to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" [15, p. 340].

It is clear from the above agility definitions that agile team members need to communicate efficiently and effectively. Therefore, agile methods require efficient and effective communication among team members and customers to achieve the highest software quality and customer satisfaction [5,47,S10]. To achieve efficient and effective communication among agile team, agile approaches depend heavily on face-to-face communication and coordination among co-located team members and customers [1,2,37,48], which is difficult to implement in GDAD environments due to communication constraints (challenges) [1,21,52]. Communication challenges refer to the characteristics of each medium that decrease communication efficiency and effectiveness [14,16]. The fewer challenges incurred by a medium, the better it is for communication process [14,16].

The extant literature reports a number of concepts, such as physical distance, time-zone differences, cultural diversity and language differences, which contribute to the complexity GDAD communication (e.g., [3,29,52]). The literature also recommends some techniques for mitigating the impact of GDAD communication challenges, which range from using available communication tools to following certain communication practices (e.g., [3,S11]). Despite the growing interest in adopting GDAD, little is known about how efficient and effective GDAD communication is achieved in practice and what techniques can be used to enhance GDAD communication [17,23,25,S9]. Many of the suggestions for the improvement of GDAD communication tools, techniques, and practices have come from experienced practitioners [2,4,18,17,19]. Hence, this research paper aims to fill this literature gap by systematically reviewing the empirical studies of GDAD to identify, synthesize and present the GDAD communication challenges and techniques that address these challenges from existing studies published in the public domain.

To the best of the authors' knowledge, there are no recent studies published in the public domain (at least, at the time that this study was initiated) that systematically review the empirical studies in the context of GDAD communication challenges. The most recent study systematically reviewing traditional process and documentation driven global software development [45] focused on generic global software development communication challenges and social computing tools. Traditional distributed software development is different from interpersonal collaboration and communication driven GDAD. Further, we assessed the quality of the literature sources in our study, which seems to have been overlooked by previous studies of GDAD communication. GDAD has attracted more interest from the software industry community in recent years. The previous papers stated the need for empirical evidence of how agile practices enhance GDAD communication and how GDAD communication challenges can be mitigated (e.g., [4,31]), which is the main aim of this study. Therefore, this paper attempts to shed more light on the empirical studies conducted in the field of GDAD communication and thereby identify the practical GDAD communication challenges and relevant mitigation techniques. Non-empirical (e.g., theoretical and conceptual) studies are beyond the scope of this paper. Hence, this paper focuses on the following main research question:

**RQ:** What is empirically known about GDAD communication? (main research question).

This study also tries to answer the following two sub-questions related to the main research question:

**RQ1.** What are the challenges or factors that limit GDAD communication?

**RQ2.** Which techniques have been used to overcome these challenges and enhance GDAD communication?

The main contributions of the paper are as follows. First, this study provides a granular understanding and yields pragmatic guidance for project leaders about GDAD communication. It helps researchers and practitioners understand GDAD communication challenges and adopt techniques to address these challenges. This study represents an initiative for developing and testing theories for guiding communication in a GDAD environment so that organizations can effectively build and sustain communication, which will ultimately improve their GDAD projects.

Second, unlike prior agile development communication reviews that have not explicitly distinguished between the different dimensions of communication, we investigate the multidimensional communication concept, which is comprised of different capabilities. We identify two key agile development communication dimensions, namely, communication efficiency and communication effectiveness, by applying the *Common Ground* communication concept [14]. Indeed, agile development approaches promote communication between all stakeholders in an efficient and effective manner [36,37,57].

Third, this study extends the previous findings in the context of agile communication (e.g., [31]) by distinguishing new challenge categories for GDAD communication (i.e., organizational factors and human factors). These two categories distinguish "locally distributed" GDAD from "globally distributed" GDAD. This identification has been achieved using the guides and concepts of the *Unified Model of Information Software Development Success* [53].

Finally, this research uncovers the relationship between the two dimensions of agile development communication and software development success in a GDAD environment. Although the efficiency and effectiveness of communication will decrease in GDAD, a positive effect of these dimensions on GDAD success has been found in the literature [18,54].

This paper is organized as follows. First, the research background and related work are presented in Section 2. The research method is discussed in Section 3. Section 4 discusses the research results. Section 5 discusses the research implications and limitations. Finally, conclusions are presented in Section 6.

## 2. Background and related work

Agile development practices focus on informal communication among team members. Informal communication can be defined as personal, interactive and peer-oriented communication [10,S9]. Additionally, it can be defined as the communication that takes place outside the official structure and without the knowledge of management [10,27], which seems helpful for quickly identifying and auctioning issues and risks [22,58]. While agile development prefers informal communication to formal communication in co-located teams, formal communication could be of great importance in GDAD environments [24]. Formal communication refers to explicit, clear communication, such as the agile requirements backlog, plans and card walls [10,26].

Because agile approaches depend heavily on face-to-face communication among co-located team members and customers, physical proximity is essential for participants to engage in informal communication [42,44,48,50]. This type of communication, in the co-located and local context, saves time and effort and



Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

reduces documentation, which increases the benefits for business and enhances customer satisfaction [9,49]. The success of agile development in co-located small and medium-sized teams motivates large software organizations to scale and adopt agile approaches in the GDAD environment. However, there are a number of challenges in GDAD, especially concerning communication [35]. It has been argued that delivering a GDAD project takes 2.5 times more resources than delivering the same project in the co-located local agile environment [26]. Korkala et al. [S9] argued that communication related issues are the root of all other GDAD challenges. Despite these challenges, it is not practical to assume that every project can or should be delivered in a co-located local agile environment [11,52].

Many studies have discussed agile communication in large agile organizations within the same time-zone and geographical context (e.g., [8,34]). Kuusinen et al. [34] mentioned that process management and communication are the main problems facing large teams. They argued that communication problems could be due to the reduced opportunity for team members to meet face-to-face, a lack of synchronization among team members and a lack of frequent feedback between users, designers and architecture teams [34]. Ineffective communication may lead to many issues, such as a lack of collaboration and coordination between different teams and a lack of understanding of customer requirements [34]. Ali Babar et al. [8] mentioned that cross-team communication is the greatest problem facing large organization teams. They argued that the ideal way to overcome this issue is by reducing cross-team communications [8].

Martini et al. [39] surveyed agile developers in three large organizations. The survey covered five challenging communication categories: (1) architecture (i.e., unnecessary flow or misunderstanding of communication due to the definition of the system and software structures), (2) technology (i.e., the differences in the tools and programming languages used by GDAD teams), (3) processes (i.e., the identification of who is responsible for delivering what), (4) organization (i.e., the structure of task allocation, coordination and supervision), and (5) people (i.e., personal or group attitude, mindset or knowledge). The authors found that the architecture category was very significant for system agility or reuse. The authors suggested using a reference architecture, achieving agreement on the high-level system requirements at the beginning of a project, and including reusability as an important quality. Process represents a boundary for development in general. The authors suggested providing developers with frequent interfaces for strategic aspects, coordinating these aspects with the overall development strategy, the local process, and adjusting the team attitude to match the overall strategy. Team distribution was found to slow development. The authors suggested promoting frequent onsite meetings and using social media tools to support and increase the frequency of GDAD communication. Organizational differences between GDAD teams were found to negatively affect communication. The authors suggested avoiding organizational dependencies caused by the architecture. They also suggested relocating team members working on different projects to promote knowledge sharing and build relationships among members of different teams.

Sutherland et al. [55,56] reported on their experiences with distributed *Scrum* in multi-team configurations. They reported that GDAD faces many challenges, such as issues related to team culture and company culture. They also claimed that these challenges have a negative impact on communication between GDAD teams. The authors argued that using *Scrum* practices (e.g., *Scrum team*, *Product Owner* and *Scrum Master*) decreases the effect of GDAD challenges and increases productivity [55,56]. In addition, using scalable *Scrum* (e.g., *Scrum* of *Scrums*) implementations with minimal tooling has been suggested to support real-time information sharing [55,56]. Some strategies have been suggested to enhance communication between distributed *Scrum* teams [56], such as *Scrum* meetings that facilitate all necessary communication between distributed teams, separate meeting rooms that are equipped with communication tools such as "digital burn down charts", video conferencing with audio equipment, and local meetings in addition to the cross-team meetings. The aim of these strategies is to ensure that requirements are clear to developers.

Other studies have discussed GDAD communication issues (e.g., [S4,S11]). Dorairaj et al. [S4] interviewed distributed agile practitioners in the USA and India to investigate GDAD communication challenges. They identified a number of GDAD communication challenges, such as different time-zones, communication tools, language barriers, and teamwork tasks. First, having team members located in different time-zones limit the communication opportunities and make it difficult for GDAD teams to organize group meetings outside the local standard working hours. To address the different time-zone challenge, the authors suggested minimizing time-zone differences between GDAD teams such that members with only small time-zone differences are included in a given GDAD team [S4]. Second, communication tools can be used to support communication among GDAD teams. However, communication tools may negatively impact team communication if these tools are not tailored and systematically deployed to fit the purpose of GDAD [S5]. Third, the use of a variety of languages may limit communication between GDAD teams, even if developers are proficient in common foreign business communication languages, such as English. It has been noted that speaking slowly, informing team members about the topics of discussion in the meeting and addressing the language issue as early as possible in the project are strategies that may reduce the impact of language barriers [S4]. Finally, teamwork requires all members to communicate and understand each other, even if some members do not wish to communicate [S4]. Dorairaj et al. [S4] recommended enhancing trust and facilitating effective formal communication (e.g., via an inception meeting at the beginning of the project, weekly or daily meetings with other distributed members, meetings with customers) and informal communication among team members.

Layman et al. [S11] studied distributed *XP* teams' communication in the USA and Czech Republic. They studied the effect of temporal distance (time-zone), geographical distance, customer communication, technology (communication tools), and different languages on the communication among GDAD teams. The authors recommended some guidelines using *XP* practices in a GDAD environment, such as

- Encouraging developers to work closely with project management teams on a daily basis;
- Assigning an individual to play the role of the customer up front;
- Using the available asynchronous communication tools as a substitution for face-to-face meetings and
- Using the available project management tools to record and track the daily development progress.

Other studies reported that informal communication can be problematic in complex GDAD projects relative to simple co-located agile projects [S8]. Depending on the use of excessive informal communication represents a challenge, especially if the project members have weak communication skills or if inadequate technology hinders communication with external parties or teams [31].

## 3. Research method

This paper adopts a systematic literature review (SLR) approach [33,51] to identify and synthesize GDAD communication challenges. The SLR is a structured and systematic approach to

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

identifying, selecting and synthesizing recent literature relevant to the research question [33]. Additionally, to ensure the quality of this study, we also use guidelines for citation and evaluation procedures to complement the original SRL approach [19,51]. SLR has been applied in distinct stages: (1) selection of inclusion and exclusion criteria, (2) selection of data sources and search strategies, (3) citation and inclusion decision management, (4) final selection and quality assessment, and (5) data extraction and synthesis.

### 3.1. Selection of inclusion and exclusion criteria

In this review, we included only those studies that presented empirical data on GDAD and passed the minimum quality criteria discussed in Section 3.4 (Table 3). Therefore, studies were excluded if their focus was not on GDAD or if they did not present empirical data. This review included studies up to July 2014; qualitative, quantitative or mixed measurement studies; small, medium and large GDAD organizations; and both professional and academic experimental software projects. Only papers written in English were included. Furthermore, as our research questions are concerned with GDAD communication (irrespective of any specific agile method), studies that focused on single methods, mixed methods (i.e., *Scrum* and *XP*) or agile methods in general were included.

### 3.2. Selection of data sources and search strategies

The following five well-known electronic databases were used in this review.

- IEEE Xplore (www.ieeexplore.ieee.org/Xplore/)
- ACM Digital Library (www.portal.acm.org/dl.cfm)
- Elsevier ScienceDirect (www.sceincedirect.com/)
- SpringerLink (www.springerlink.com/)
- Google Scholar (http://scholar.google.com.au/)

These databases are assumed to provide sufficient literature coverage for this study. In addition, the following relevant and important agile conference proceedings discussing the use of GDAD were manually searched and included in this study: *XP*, *XP*/Agile Universe, and Agile Development Conference.

The reviewed papers come from industrial, qualitative/quantitative and experimental academic studies. Fig. 1 shows the review process and the number of papers identified at each stage. Table 1 shows the terms and keywords used to run the first stage of the search. We included the terms *XP* and *Scrum* in the search terms because these two are the most used agile methods [17]. In this stage, each item from the first category (i.e., "Communication Practice") was combined with each item from the second category (i.e., "Distributed Software Development") and each item from the third category (i.e., "Use of Agile Practice") using the Boolean "AND" operator, which finds all articles on "Communication Practices", "Distributed Software Development" and "Use of Agile Practice". That is, we searched every possible combination of one item each from the first category AND the second category AND the third category. The search excluded articles that addressed discussion comments, prefaces, editorials, news, article summaries, reviews, correspondences, discussions, reader's letters, summaries of tutorials, workshops, panels, and poster sessions. This search strategy resulted in a total of 1587 "hits"; however, after excluding the duplicate papers, the number of hits dropped to 799.

### 3.3. Citation and inclusion decision management

This study followed the citation procedure reported by Dybå and Dingsøyr [19]. *EndNote* was used to store the relevant citations

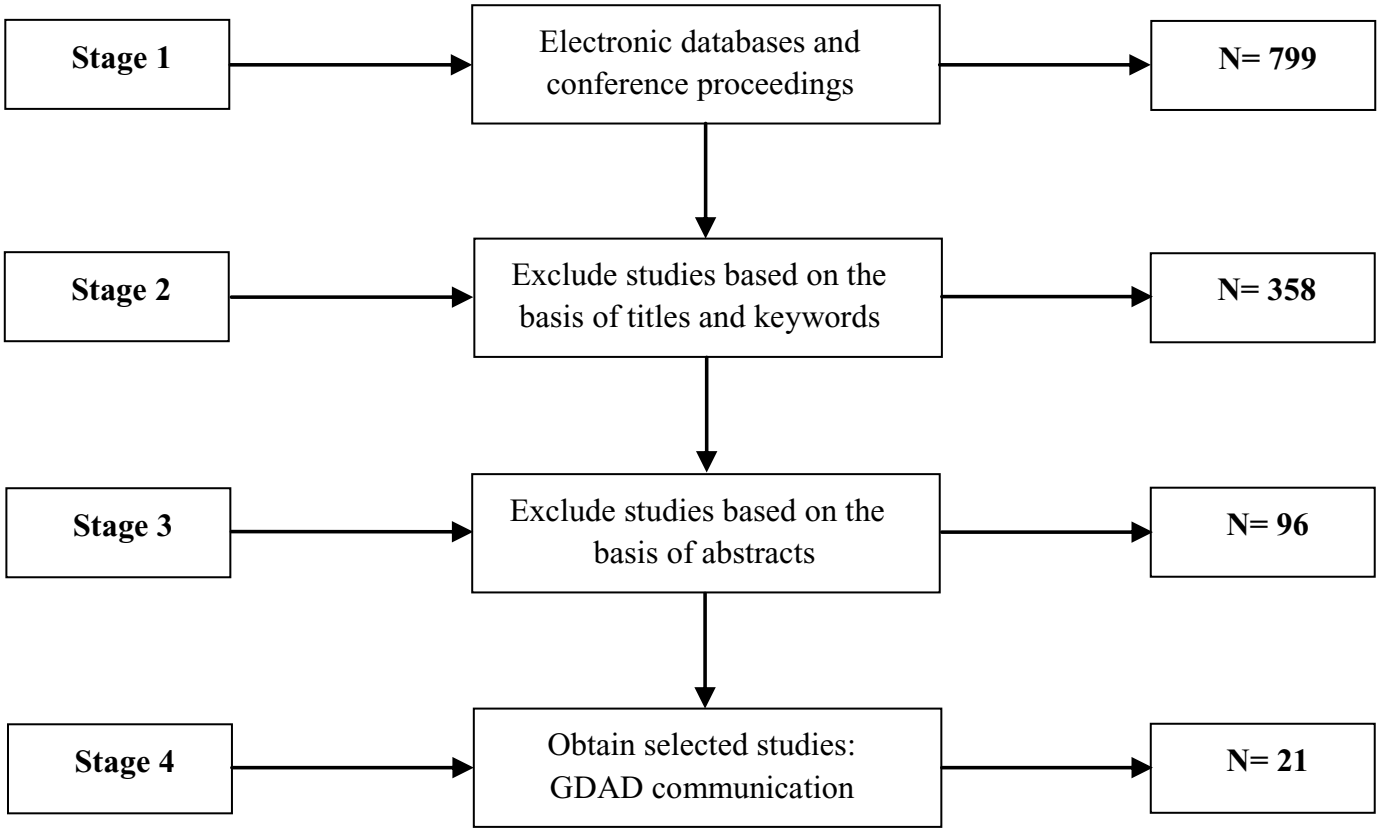


**Fig. 1.** Study selection process.

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

**Table 1**
Search terms.

| Search category | Keywords |
|---|---|
| Communication Practice | Communication, team communication, cross-team communication, offshore communication, outsourcing communication, customer communication, social media communication, communication tool, communication technology |
| Distributed Software Development | Distributed agile, multi-sites agile, global agile development, multi-team agile, global software development, distributed development, distributed development teams, global software development, global development, global software engineering, offshore development, off shoring, outsourcing development, open source, near shore, near shoring, multi-sites development, global software engineering, dispersed teams |
| Use of Agile Practices | Agile, agile methods, agile practice, *Scrum*, *Scrum* practice, *Scrum* method, *XP*, extreme programming, *XP* practice, *XP* method |

from stage 1 ($N = 799$). The citations were then imported into *Excel* sheets, where the source of each citation and subsequent inclusion/exclusion decision were recorded. *EndNote* databases and *Excel* sheets were separately established for each stage. In this stage, we reviewed the titles of all 799 studies and excluded studies that were clearly not about GDAD communication. In some cases, the title failed to clearly identify whether the study was within the scope of this review. In this case, the articles were included in the next review stage. In this stage, we identified 358 relevant studies. Table 2 summarizes the assessment method and criteria for each stage.

At stage 3, we excluded the studies that did not focus on GDAD communication and did not present empirical data by scanning the abstracts. Some abstracts were misleading, gave little indication of the content of the full paper or did not clearly indicate whether the study was indeed empirical. Therefore, at this stage, we included all studies that indicated some form of GDAD experience. In this stage, 262 articles were excluded, leaving 96 articles for the final selection stage.

### 3.4. Final selection and quality assessment

If the study did not clearly indicate GDAD communication in the title, abstract, and keywords, we included it in a detailed final quality assessment. At the final stage, screening criteria were used to ensure the relevance and quality of the selected study. We used the following screening criteria to ensure that the studies address the research topic, as shown in Table 3. The criteria were adopted from Dybå and Dingsøyr [19]. These 11 criteria ensure the quality, relevance, credibility and rigour of the studies used in this research [19]:

- Three screening criteria were used to ensure the quality of the reviewed study by identifying the study's rationale (i.e., reported empirical research), aims (i.e., clearly reported aims and objectives), and context (i.e., the context in which the study was carried out), as shown in the first three questions of Table 3. These criteria were used to ensure that only empirical studies were included and represent the minimum quality threshold of the selected studies. Only criterion 1 was used as a basis for including or excluding a study in the final stage.
- Five criteria were used to ensure that the study is rigorous (i.e., used thorough and appropriate approaches for collecting and analyzing data). The following criteria were used to assess the rigour of the selected studies (questions 4–8 in Table 3):

1. The study used an appropriate research design to address its aims.
2. The study adequately described the sample used and the methods for identifying the sample.
3. The study used control groups to compare treatments.
4. The study used appropriate data collection methods.
5. The study adequately described the data analysis methods.

- Two criteria were used to ensure the credibility (i.e., the validity and meaningfulness of the findings) of the selected studies (questions 9 and 10 in Table 3).

1. Relationship between the researcher and participants was considered to an adequate degree.
2. The study provided clear findings with credible results and justified conclusions.

- Finally, one criterion was used to ensure the relevance (usefulness for GDAD industry and research community) of the selected studies (question 11 in Table 3) by identifying whether the study provided value for practice and/or research.

After applying the first criterion, we selected only 21 studies out of the 96 studies that passed stage 3. We excluded some papers that were published in different years but used the same empirical data. For instance, we found that studies [46] and [S15] used the same empirical data, as did studies [30] and [S1]. In this case, only the study most related to our research (i.e., [S1,S15]) was included in the extraction and synthesis stage. Table 4 summarizes the number of selected studies in each stage. Most studies were found in the "IEEE Xplore" and "SpringerLink" databases, comprising 33% of the selected studies for each. Three studies were found in the "Google Scholar" database, representing14% of the selected

**Table 2**
Assessment method.

| Filtration stage | Method | Assessment criteria |
|---|---|---|
| 1st filtration | Identify relevant studies form searched databases | Keywords (all items) |
| 2nd filtration | Exclude studies on the basis of titles | Title = search term(s)<br>Yes = accepted<br>No = rejected |
| 3rd filtration | Exclude studies on the basis of abstracts | Abstract = communication<br>Yes = accepted<br>No = rejected |
| Final filtration | Obtain selected papers and critically appraise studies | Address GDAD communication<br>Discuss empirical research<br>(Yes = accepted, No = rejected) |

**Table 3**
Quality criteria [19].

| Quality criteria |
|---|
| 1. Is the paper based on research? |
| 2. Is there a clear statement of the aims of the research? |
| 3. Is there an adequate description of the context in which the research was carried out? |
| 4. Was the research design appropriate for addressing the aims of the research? |
| 5. Was the recruitment strategy appropriate for the aims of the research? |
| 6. Was there a control group with which to compare treatments? |
| 7. Were the data collected in a way that addressed the research issue? |
| 8. Was the data analysis sufficiently rigorous? |
| 9. Has the relationship between the researcher and the participants been considered to an adequate degree? |
| 10. Is there a clear statement of findings? |
| 11. Is the study of value for research or practice? |

**Table 4**
Search results.

| Database | 1st filtration | 2nd filtration | 3rd filtration | Studies selected | Percent selected (%) |
|---|---|---|---|---|---|
| IEEE Xplore | 328 | 119 | 39 | 7 | 33 |
| ACM Digital Library | 167 | 97 | 22 | 2 | 10 |
| SpringerLink | 125 | 83 | 20 | 7 | 33 |
| Google Scholar | 104 | 37 | 8 | 3 | 14 |
| ScienceDirect | 75 | 22 | 7 | 2 | 10 |
| Total | 799 | 358 | 96 | 21 | 100 |

studies. Only 2 studies each were found in the "ACM Digital Library" and "ScienceDirect" databases.

In each stage of this study, two researchers searched independently, and a third researcher always accompanied each researcher to review the findings separately from the other researcher. At the end of each stage, all authors sat together and discussed the agreement on the number of studies to be included in the next stage. All disagreements were solved by the three researchers' meeting (all hands in meetings). For example, in the final stage (i.e., obtain selected study) we found 82 (85%) agreements among 96 assessments. All disagreements were resolved by three researchers through rigorous discussions. As a result of the discussions, another set of 61 articles was excluded at this stage, leaving a final set of 21 articles for data extraction and synthesis.

### 3.5. Data extraction and synthesis

We used the *Unified Model of Information Software Development Success* [53] to derive and code the categories used for structuring and analyzing the results of the selected studies. Based on a comprehensive literature review, this model synthesizes past research efforts in the software development field and identifies important factors that influence the software development process (e.g., team factors, organizational factors and human factors). This model is suitable for coding our findings as it provides sufficient coverage of the software development process as well as its input and output factors (Fig. 2).

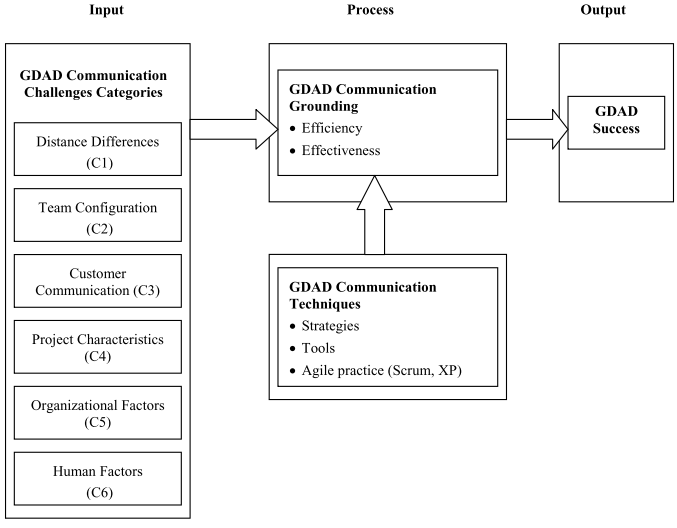


**Fig. 2.** Theoretical framework and selected categories.

## 4. Results

We identified and reviewed 21 [S1–S21] empirical studies on GDAD using the overarching SLR approach described in the previous section; see Appendix A. First, we provide an overview of the selected studies. Next, we discuss the characteristics of the selected studies, the research methods, and the quality of these studies. Finally, we analyze the selected studies to answer the identified research questions (i.e., to identify GDAD communication challenges and strategies and recommendations to overcome these challenges).

### 4.1. Overview of studies

This SLR study found that all of the selected studies were conducted in or after 2006. With respect to the types of agile methods studied, it can be observed from Table 5 that 8 (37%) of the studies in this review did not mention any specific agile method (i.e., the agile method used was not identified). Meanwhile, 33% of the selected studies were relevant to the *Scrum* method. The mixed *Scrum* and *XP* method was studied in four empirical studies (20%). Finally, only two studies (10%) applied the *XP* method, representing the smallest category.

Analysis of the selected studies reveals a strong focus on using the *Scrum* method in the GDAD environment. The reason may be due to the project management and coordination roles, artefacts and practices offered by *Scrum* [55]. For instance, *Scrum* includes a *Product Owner* role, which can facilitate communication requirements among GDAD teams. *Scrum* artefacts include the *Product Backlog* and *Scrum Backlog*, which can also be scaled to support GDAD [56]. On the other hand, the *XP* method is more preferred by and used for development by co-located teams (e.g., pair programming). In many cases, the mixed *Scrum* and *XP* method was used by GDAD teams to support project communication and management as well as project development.

Table 6 provides an overview of the selected studies according to the publication channel. The single largest publication channel was found to be the "ICGSE" and "Agile Processes in Software Engineering and Extreme Programming", which published three studies each. Ten studies were published in conferences. Six studies were published as book chapters, and only five empirical studies (relevant to the research questions in hand) appeared in scientific journals.

Regarding the year of publication, we found no empirical studies on GDAD prior to 2006 that matched our selection criteria. We found that two empirical research studies were published in 2006, three in 2007, two in 2009, four in 2010, two in 2011, two in 2012, four in 2013, and only one study each in 2008 and 2014. These numbers indicate that a focus on GDAD study was more prevalent during 2010 and 2013. This review searched databases up to July 2014, which may explain the small number of papers published in 2014 (i.e., only one study).

**Table 5**
Agile method used.

| Agile method | Number | Year | Percentage | Study |
|---|---|---|---|---|
| *Scrum* | 7 | 2008–2013 | 33 | S1, S14, S15, S17–S19, S21 |
| *XP* | 2 | 2006–2010 | 10 | S11, S13 |
| Mix (*Scrum*, *XP*) | 4 | 2006–2012 | 20 | S7, S9, S12, S16 |
| Not-identified (general) | 8 | 2007–2014 | 37 | S2–S6, S8, S10, S20 |

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

**Table 6**
Publication channel.

| Publication channel | Type | Study | Number |
|---|---|---|---|
| Information and software technology | Journal | S11 | 1 |
| Information systems management | Journal | S7 | 1 |
| JSS | Journal | S8 | 1 |
| VINE | Journal | S5 | 1 |
| Empirical software engineering | Journal | S12 | 1 |
| EUROMICRO SEAA | Conference | S10 | 1 |
| HICSS | Conference | S1 | 1 |
| ICGSE | Conference | S2, S15, S21 | 3 |
| ICSE | Conference | S14 | 1 |
| PACIS | Conference | S20 | 1 |
| International conference on product focused software | Conference | S9 | 1 |
| SPLASH | Conference | S17 | 1 |
| WASET | Conference | S6 | 1 |
| Agility across time and space | Book section | S3, S13 | 2 |
| Agile processes in software engineering and extreme programming | Book section | S4, S16, S19 | 3 |
| Enterprise interoperability III | Book section | S18 | 1 |

### 4.2. Methodological quality

As mentioned in Section 3, we assessed the quality of each of the selected studies according to 11 quality criteria (Table 3) adopted from Dybå and Dingsøyr [19]. These 11 criteria provided a measure of our confidence that a selected study could make a valuable contribution to our review. Each of the 11 criteria was graded on a dichotomous (“yes” or “no”) scale. As mentioned above, the inclusion of the studies was based on the first assessment criterion (i.e., the study should report empirical research). The results of the quality assessment are shown in Table 7.

Because we only included research papers in our review, all selected studies were graded as “1” on the first screening criterion (first column). With regards to the study’s aim, only one study was found not to have a clear statement of its aims. The third criterion (context of the study) showed that all studies included some form of description of the context in which they were carried out. For the chosen research design, seven studies did not adequately report the research design used to achieve the aims of the research. As many as fifteen out of the twenty-one studies did not clearly report the recruitment strategy (i.e., sampling) used to achieve their stated aims. Only three studies reported using other groups or baselines with which to compare their findings. Only one study did not describe its data collection method. Seven studies did not report their data analysis procedures. Only three studies reported how they avoided biasing the findings. It was found that research bias issues, recruitment strategy and control group assessment were often not reported in the selected studies. This may be related to the fact that most of the selected studies were conference papers (Table 6), as conference papers generally report fewer details about the research methods. The overall quality of the selected studies score was greater than 7.7, which is satisfactory. One study was graded 11 (full score), and none obtained negative scores in the quality assessment. The highest number of negative scorings was five. Thus, overall, all selected studies were of acceptable quality for this study and added value to the body of knowledge. However, these findings also evidence the need for more quality empirical studies in the area of GDAD communication.

### 4.3. Scope of the selected studies

The selected studies varied in their scope, which can be categorized as follows: (i) the challenges of adopting GDAD (several studies), (ii) the scale-up of local agile development to a GDAD environment, (iii) the successful implementation of agile methods in GDAD, (iv) the challenges of GDAD communication, and (v) solutions to GDAD challenges in general and GDAD communication challenges in particular (see Appendix C). As our focus in this paper is on GDAD communication challenges and solutions (RQ1 and RQ2), we derived our findings from those scopes that included GDAD communication. To this end, we categorized our findings into two broad categories: GDAD communication challenges and techniques to overcome GDAD communication challenges. By investigating these two research

**Table 7**
Quality assessment.

| Study | 1 Research | 2 Aim | 3 Context | 4 R design | 5 Sampling | 6 Control group | 7 Data collection | 8 Data analysis | 9 Reflexivity | 10 Findings | 11 Value | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 8 |
| S2 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 9 |
| S3 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| S4 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 9 |
| S5 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 10 |
| S6 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 7 |
| S7 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 9 |
| S8 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 7 |
| S9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 |
| S10 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 7 |
| S11 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 7 |
| S12 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 7 |
| S13 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| S14 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 8 |
| S15 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 8 |
| S16 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| S17 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 6 |
| S18 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 9 |
| S19 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 9 |
| S20 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 7 |
| S21 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 8 |
| Total | 21 | 20 | 21 | 14 | 6 | 3 | 20 | 14 | 3 | 21 | 21 | |

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

questions, we aim to provide a synthesis of the literature of GDAD communication challenges and techniques for both researchers and practitioners.

### 4.4. Theoretical framework and categories

We used communication *Common Ground* [14] and the *Unified Model of Information Software Development Success* [53] to develop our theoretical framework (Fig. 2). Communication *Common Ground* refers to the mutual, common or joint knowledge and the beliefs and suppositions held by the communication parties. Communication *Common Ground* enables successful coordination, as it allows interdependent actors to adjust their actions in a manner appropriate for the other actors. *Common Ground* facilitates the delivery of an understandable message with minimum effort (i.e., efficient and effective communication). Communication *Common Ground* can be enhanced by using different communication techniques based on the communication medium. It also identifies constraints (i.e., challenges) that decrease communication efficiency and effectiveness. Thus, *Common Ground* offers a comprehensive view of the communication process. As discussed in Section 3.5 (data extraction and synthesis), we used the *Unified Model of Information Software Development Success* [53] to derive and code the categories. The combination of *Common Ground* and the *Unified Model* provides us with a comprehensive framework to evaluate GDAD communication.

As discussed in the introduction, GDAD refer to "locally distributed" or "globally distributed" (i.e., affected by global factors, such as culture and language) agile development. [4]. Some studies and recommendations refer to three main communication challenges of agile development in general: *Team Distribution*, *Team Size*, and *Project Characteristics* (e.g., [31]). However, because we are investigating the GDAD environment, the *Team Distribution* category will be considered through four separate categories (i.e., *Distance Differences, Customer Communication, Organizational Factors* and *Human Factors*). This helps us distinguish between the two types of GDAD (i.e., "locally distributed" and "globally distributed") and better identify the challenges of these categories in terms of the big issues for GDAD communication, such as national culture and organizational culture [4,18,32]. We also refer to the *Team Size* category as *Team Configuration* [4] as a more comprehensive definition. While both "locally distributed" and "globally distributed" agile development share the first 4 categories of the communication challenge categories (i.e., *Distance Differences*, *Team Configuration*, *Customer Communication*, and *Project Characteristics*), categories 5 and 6 (i.e., *Organizational Factors* and *Human Factors*) may only be experienced in "globally distributed" agile development communication (Fig. 2 and Table 8). The following section discusses the categories of GDAD communication challenges in detail.

The input of the GDAD communication theoretical framework is represented by the GDAD communication challenge categories (C1–C6). These categories will negatively affect the GDAD communication grounding process, which negatively reflects on communication efficiency and communication effectiveness [14]. The GDAD communication grounding process is enhanced by applying certain techniques that improve communication efficiency and effectiveness [14]. These techniques include strategies, tools, and agile practices. The *Scrum* and *XP* methods were given special focus in the analysis because, as mentioned previously in this paper, they are considered the most widely used agile methods in industry [17]. The output of the GDAD communication grounding process is the GDAD success, which can be represented by such factors as project success and user satisfaction [6,7,53].

**Table 8**
GDAD communication challenges and categories.

| Ref. | Category | GDAD communication challenges |
|---|---|---|
| C1 | Distance Differences | • Time-zone differences<br>• Geographic differences |
| C2 | Team Configuration | • Team size<br>• Number of teams<br>• Coordination |
| C3 | Project Characteristics | • Project domain<br>• Project architecture |
| C4 | Customer Communication | • Customer involvement<br>• Customer representative involvement |
| C5 | Organizational Factors | • Project management process<br>• Communication tools<br>• Communication infrastructure<br>• Organizational culture |
| C6 | Human Factors | • Language<br>• National culture<br>• Trust in team or team members<br>• Personal practice |

### 4.5. GDAD communication challenges

#### 4.5.1. RQ1 – challenges that limit GDAD communication

To answer the first research question, we analyzed and interpreted the data from the selected empirical studies and identified the categories of GDAD communication challenges as discussed in Section 4.4 (Fig. 2). Each category has one or more challenges. A total of 17 GDAD communication challenges were identified (Table 8).

Table 9 shows the challenge categories and the statistics related to each category. These categories and statistics are discussed in the following paragraphs. Moreover, the identified challenges may become issues (i.e., negative impacts) and limit GDAD communication. These impacts are also discussed in the following paragraphs and are summarized in Table 10. These impacts were summarized under each concept category, as for GDAD communication challenges (Table 8), and extracted from the same studies for each category.

**Distance differences:** This category refers to differences in the time-zones and geographical contexts of the GDAD teams. Two common characteristics of distance differences have been defined in the context of GDAD: time (temporal) and geographical distances [3]. Temporal distance is "a directional measure of the dislocation in time experienced by two actors wishing to interact. Temporal distance can be caused by time zone difference or time shifting work patterns" [3, p. 3]. Geographical distance is "a directional measure of the effort required for one actor to visit another at the latter's home site. Geographical distance is best measured in ease of relocating rather than in kilometres" [3, p. 3]. This category has been heavily

**Table 9**
Statistics for GDAD communication challenge categories.

| Ref. | Category | Frequency | Percentage | Selected studies |
|---|---|---|---|---|
| C1 | Distance Differences | 16 | 76 | S1–S4, S6–S8, S10–S17, S21 |
| C2 | Team Configuration | 10 | 48 | S2–S4, S6, S13, S14, S17–S20 |
| C3 | Customer Communication | 4 | 19 | S9–S11, S14 |
| C4 | Project Characteristics | 2 | 10 | S1, S3 |
| C5 | Organizational Factors | 11 | 52 | S3–S9, S11, S14, S19, S21 |
| C6 | Human Factors | 10 | 48 | S1–S4, S7, S11, S12, S14, S15, S21 |

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

**Table 10**
Impact of challenges on GDAD.

| Ref. | Category | Impacts |
|---|---|---|
| C1 | Distance Differences | Reduced communication opportunities, lack of face-to-face and informal communication, communication delay, long-time communication using technology, longer meetings, difficulty in holding group meetings outside working hours (overlaps), tendency to lose track of the overall work process, synchronization problems due to differences in religious holidays, developer availability difficulties, increased integration, coordination and communication costs for face-to-face meetings, lack of task awareness, lack of trust |
| C2 | Team Configuration | Early communication difficulties, difficulties with the formation of team member configurations, difficulties with ensuring that all members communicate with each other, unwillingness to communicate with other team members, less understanding of teamwork, slowing of the communication speed and pair programming |
| C3 | Customer Communication | Less frequent communication with customers, weak relationships with customers, customer un-involvement, hiding information from customers, miscommunication of the requirements leading to developers basing decisions on their experience or guessing at the requirements |
| C4 | Project Characteristics | Misunderstanding or unnecessary flow of communication due to the definition of systems and software structures, unnecessary communication, lack of matching among the processes of different sites and teams, lack of communication opportunities |
| C5 | Organizational Factors | Lower team efficiency, less business value and product quality, more constraints and risks, more organizational modes, less access control, more semantic interoperability, more contingency and disaster recovery plans, more communication channels, extra cost incurred to train teams on different tools, technical and organizational culture incompatibilities between sites, lack of trust |
| C6 | Human Factors | Misunderstanding and miscommunication difficulties, different holidays among distributed team members, inconsistency in work place, longer time for sharing, communication difficulties due to language, limited communication among GDAD teams, interpretation problem, silence of some participants |

referenced in the literature and is represented in the highest proportion (i.e., 76%) of the selected studies. Distance differences have been reported as the main challenge of GDAD communication [S9,S10]. The selected studies reported that distance differences adversely impact communication common ground between GDAD teams or team members (e.g., [S4,S7,S15]). Distance differences minimize communication efficiency and effectiveness between GDAD teams and decrease the opportunity for groups to organize meetings outside standard local working hours [S4,S15]. In addition, distance differences increase the cost and logistics of holding face-to-face meetings for GDAD teams [S3]. Distance differences also reduce informal interaction, which may lead to a lack of task awareness and reduce trust among GDAD teams and team members [S7,S12,S16]. This will limit the efficiency and effectiveness of communication between teams or team members [S3].

**Team configuration:** This category refers to the team size, number of teams, and coordination (e.g., early communication between teams, shared understanding, and cross-team communication) between distributed teams [4]. This category has received a fair amount of attention in the selected studies and is mentioned in ten studies, corresponding to 48% of the selected studies. Paasivaara et al. [S14] and Green et al. [S6] argued that it is important to pay attention to early communication (i.e., at the beginning of the new project) among GDAD teams when implementing GDAD. Maruping [S13] argued that team member configuration (i.e., the management of team members across physical locations, such as 3-3-3 or 1-2-6) plays an important role in teamwork efficiency. The selected studies mention that GDAD teams or team members suffer from decreased communication speed and pair programming when a greater number of team members and teams are included in a GDAD project [S14]. It has been reported that some members do not wish to communicate and have less understanding of team coordination [S4].

**Customer communication:** This category refers to communication with customers or with customer representatives (i.e., those who set the requirements or change the requirements of a project). This category has not received enough attention, comprising only 19% of the selected studies. In agile development, customers or customer representatives have to be involved in the development process, and the project information must not be hidden from them [S8,S10]. However, in a GDAD environment, customer communication may be difficult. The selected studies have reported that a lack of customer communication may result in weak relationships and a misunderstanding of requirements, which may lead developers to either base decisions on their experience or guess at the customer's requirements [S10,S11].

**Project characteristics:** This category refers to such project characteristics as the project architecture (definition of a system and software structure), project domain, and type of project, such as single customer or commercial use [4,18]. This category has rarely been mentioned in the literature; it is mentioned in only two studies, representing 10% of the selected studies. Most of the findings have indirectly identified project characteristics as a challenge to GDAD communication. The challenges related to project characteristics are related to misunderstandings or an unnecessary communication flow due to the definition of a system and software structure [S3]. A lack of predefined project characteristics decreases the knowledge sharing and communication capability of GDAD teams and team members, increases unnecessary communication, reduces integration with other processes, and decreases communication opportunities [S3].

**Organizational factors:** This category refers to the tools and infrastructure capabilities that support GDAD communication, project management processes, and organizational culture. Project management processes refer to the processes (i.e., who is responsible for delivering what to whom and when) used to establish and maintain communication between GDAD teams and team members. Organizational culture is defined as the values, attitudes, and behaviours that represent an organisation's working environment, vision, and subjective [28]. This category has attracted significant attention in the selected studies as a challenge for GDAD communication and collaboration. This category makes up the second highest percentage (i.e., 52%) of the selected studies. The tools used for GDAD communication include synchronous tools (e.g., phone, instant messaging) and asynchronous tools (e.g., *email*) [S7]. The following issues related to organizational factors have been identified and reported in this study: the use of unsuitable tools by an agile team, technical incompatibilities between different sites [S14], the overhead of inadequate quality of video conference communication and coordination, and a lack of electronic tool support [S3,S19]. Gill and Bunker [S5] identified 14 categories (issues) that need to be taken into account when developing a comprehensive tool for GDAD communication. These categories include communication channel, business value, technology use case, quality, type, interface management, mode, access control, semantic interoperability, constraint, risk, contingency and



Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

disaster recovery, dependency and recommendation. The most appropriate tool can be chosen by using these 14 categories as a guideline for a particular GDAD communication context. Moreover, the bureaucratic organization culture is considered a challenge in GDAD because it decreases the efficiency and effectiveness of communication among team members [S10].

**Human factors:** This category refers to differences in the language, national culture, trust, and personal practices between GDAD teams and team members. Language refers to the use of different languages among GDAD teams and team members. National culture refers to differences in national or local culture among GDAD teams and team members. National culture is defined as the collective programming of the mind that distinguishes the members of one group or category of people from another. It encompasses norms, values, spoken language and styles of communication [28]. Trust refers to the difference in trust levels expressed by GDAD teams and developers depending on their location or country [S2]. Personal practice refers to the differences in the personal attitudes and skills of GDAD developers [S7]. This category is mentioned in 48% of the selected studies. A number of issues related to human factors have been identified and reported in this study: language misunderstandings and poor mutual understanding [S7,S15], longer teleconference meetings, longer daily *Scrum*, the silence of some participants [S2,S4,S15], confusion among team members [S7], reduced coordination due to the observance of different holidays among the GDAD teams and personal (or group) attitudes [S1,S3], and different interpretations of the negative and sensitive issues of the project [S4,S7].

### 4.6. Solutions to GDAD communication challenges

#### 4.6.1. RQ2 – techniques used to address GDAD communication challenges

Here, we have organized the techniques (strategies, tools, agile practices) used to address the GDAD communication challenges into the same concept categories as for the GDAD communication challenges (Table 8). These recommendations were extracted from the same studies for each category. Table 11 summarizes these recommendations.

**Distance differences:** This study identified a number of techniques to address issues related to distance differences: synchronize the work hours between the GDAD team members, synchronize communication between different teams or between the local *Product Owner* and distributed *Scrum Master*, scale *Scrum* with the use of *Scrum* of *Scrums*, divide the meeting into several parts (distributed and onsite parts), create local teams for each geographical or same time-zone area, hold strict all-hands meetings in which all members are required to attend or share, reduce the number of whole distributed project meetings, use asynchronous tools (e.g., *Wiki*, *email*, and so on), minimize dependencies among teams, combine the flexibility of agile methods with the rigidity of traditional methods, enhance regular visits and face-to-face communication, split the project into small parts, and centralize the experts in the home country [S1,S2,S6,S11,S12,S15–S17,S21].

**Team configuration:** A number of techniques to address the team configuration issues were identified in this study: frequently demonstrating products, promoting face-to-face meetings at the

**Table 11**
Techniques to overcome GDAD communication challenges.

| Ref. | Category | Techniques |
|---|---|---|
| C1 | Distance Differences | • Strategies: ensure small differences in time-zone countries chosen for distribution, divide work between no more than 2 sites, divide meetings into several parts (distributed and onsite parts), provide more flexibility in the working environment, combine the flexibility of agile methods with the rigidity of traditional methods, encourage regular visits and face-to-face communication, create a structure of trust, minimize dependencies among teams, encourage individuals to work closely with both developers and project management teams, enforce meetings and commitment, localize component ownership, enhance coordination by promoting social skills<br>• Tools: switch to the most appropriate tools, use synchronous and asynchronous tools, use an architectural roadmap, use secure shared information repositories<br>• Agile practices: use an information hub, such as *Product Owner*; synchronize communication between local *Product Owner* and distributed *Scrum Master*; scale *Scrum* with the use of *Scrum* of *Scrums* |
| C2 | Team Configuration | • Strategies: encourage face-to-face meetings at the beginning of each project, promote product demonstration at the end of each iteration to ensure communication synchronization, encourage periodic meetings, use a simple design, form local teams, promote clear roles and responsibilities for each site, systematically check for refactoring, involve customers in development, encourage trust between distributed teams and team members, increase effective formal communication<br>• Tools: use synchronous communication tools<br>• Agile practices: avoid distributed pair programming, encourage small releases |
| C3 | Customer communication | • Strategies: ensure regular agile meetings involving the customer, promote the customer representative as playing the role of the customer up front<br>• Agile practices: promote frequent interaction with *Product Owner* (*Scrum* roles) |
| C4 | Project Characteristics | • Strategies: achieve agreement on the requirements by all teams and relevant members at the beginning of the project, promote coordination with the overall strategy and local process, encourage participation of offshore representative in onshore daily meetings to help resolve any misunderstanding<br>• Tools: use explanation of reference architecture (overall architecture) by all teams and relevant members |
| C5 | Organizational Factors | • Strategies: promote a corporate organizational culture that supports rapid communication, trust between agile stakeholders, and quick customer feedback<br>• Tools: use different communication technologies, use available project management tools<br>• Agile practices: employ *Scrum Master* and *Product Owner* for each site, encourage frequent visits of *Scrum* roles to other sites, encourage joint daily *Scrum*/2-week review and monthly retrospective meetings, form single-site *Scrum* teams only |
| C6 | Human Factors | • Strategies: conduct exchange visits between distributed sites, treat all developers as equal, encourage connection team leader, speak slowly and clearly, ensure members are aware of a meeting's topics before the meeting, address the language issue as early as possible in the project, encourage individual to work closely with both developers and project management teams, use less detailed communication, use easily editable *Wiki* pages, keep shared information updated<br>• Tools: use tools to record meetings, use asynchronous communication technology<br>• Agile practices: use *Scrum* practices, use local *Scrum Master* instead of *Product Owner* to answer questions, encourage offshore representative participation in onshore daily *Scrum* meetings, synchronize the sprint length |

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

beginning of each project [S14], promoting periodic meetings between different teams, and using synchronous communication (telephone, video, etc.). Moreover, other authors recommended using a simple design and promoting small releases, creating local teams and using clear roles and responsibilities for each site, avoiding distributed pair programming, and systematically checking for refactoring [S13,S18]. In addition, it was recommended that more attention be paid to the importance of the role of the *Product Owner* among GDAD teams and inside a co-located team [S14]. Others suggested increasing effective formal communication (e.g., inception meetings at the beginning of the project, weekly or daily meetings with other GDAD members, and meetings with customers) and encouraging mutual trust among team members [S4].

**Customer communication:** This study identified a number of techniques for addressing issues related to customer communication. These techniques include using a customer representative to play the role of the customer [S8,S11], enhancing rapid communication, promoting regular agile meetings and active customer engagement [S10], and promoting the role of *Product Owner* (*Scrum* roles) [S14].

**Project characteristics:** This study identified some techniques for addressing project characteristics issues. These techniques include using an overall architecture vision among GDAD teams, coordinating the overall strategy with the local team's process [S3], and encouraging offshore representatives to participate in onshore daily meetings to help resolve any misunderstandings [S1]. Other suggestions include using a reference architecture and ensuring agreement on the project's requirements by all teams and relevant members at the beginning of the project [S3].

**Organizational factors:** This study also identified a number of techniques for addressing organizational factors issues: offering different communication tools, using available project management tools, using different communication models [S4,S11,S21], employing *Scrum Master* and *Product Owner* for each site, and enabling frequent visits of *Scrum* roles to other sites [S19]. In addition, use of the communication technology assessment tool (CTAT) to assess and select an appropriate tool for supporting GDAD communication was suggested [S5]. Moreover, a corporate organizational culture that supports rapid communication and trust between agile stakeholders was considered a success factor for GDAD [S10,S11].

**Human factors:** Previous authors have recommended some techniques to address human factors problem: synchronizing the sprint length, encouraging offshore representatives to participate in onshore daily *Scrum* meetings [S1], using *Scrum of Scrum* practices [S14], speaking slowly and clearly, informing team members about a meeting's topics before the meeting, addressing the language issue as early as possible in the project, using less detailed communication [S4], keeping shared information up-to-date, using a local *Scrum Master* instead of *Product Owner* to answer questions in the case of large teams [S12], enhancing trust in relationships and shared understanding through exchanging visits [S7,S11,S15], and using multiple communication modes and tools to record meetings [S7,S12].

### 4.7. GDAD communication patterns

Our meta-analysis revealed that some GDAD communication patterns have been identified in the selected studies. These patterns include strategies, tools, and agile practices (e.g., *Scrum* and *XP*).

In the strategy patterns, frequent deliveries and systematic coordination to the small release were recommended as successful GDAD asynchronous communication loops that can serve as a sufficient surrogate for synchronous communication [S11,S13]. Moreover, synchronizing work hours, reducing the number of GDAD team meetings, hosting team gatherings at the beginning of each project, exchanging visits between GDAD teams, and maintaining key documentation were found to support GDAD communication [S1,S3,S7,S12,S16]. In addition, corporate culture is considered as a success factor for GDAD communication [S10].

In the tool patterns, access to appropriate means or tools of communication that match the situational needs of the user (e.g., developer) was found to be a key factor for enhancing GDAD communication [S2,S5,S6,S18]. Other successful GDAD communication tool patterns included using a central digital repository for the customer requirements and the assigned tasks for each agile developer [S5,S13,S18]. This repository is accessible by all team members and allows all team members to view and manipulate the progress made in satisfying the requirements in real-time [S13].

In the agile practice patterns, team members play important roles in GDAD communication. For example, component leads and project managers have a significantly higher communication overhead than development team members [S17]. The coordination responsibilities of component leads and project managers make them the focal point of many communication links in GDAD. Thus, together with technical acumen, component leads and project managers need to possess strong social skills to enhance communication and collaboration across sites [S17]. They need to be able to travel when needed [S18]. Moreover, pair programming in GDAD was found to enhance communication among distributed team members [S16]. Pair programming in GDAD can be achieved using screen sharing and audio calls [S16]. GDAD *Scrum* practices offer a distinctive advantage in mitigating GDAD communication challenges [S1,S18]. The following practices were found very effective for addressing poor communication in GDAD [S18]: the *Product Owner* should represent the customer and interact with the development team, local *Product Owners* and *Scrum Masters* are needed for each site to facilitate communication, clear *Product/Sprint Backlog* items should be specified in detail, and specifications should be able to contain rich information to facilitate understanding (e.g., using figures, diagrams) among team members.

### 4.8. Impact of communication on GDAD success

Communication efficiency and effectiveness have been reported to play a critical role in the success of GDAD projects or releases [13,S2,S4,S10,S11]. While communication is crucial, communication technologies should also be an integral part of the GDAD environment to facilitate communication and thereby successful GDAD projects [S4,S5,S9,S20]. There are a number of communication tools to choose from (e.g., *Lync*, *Skype*). However, using a combination of communication tools that fits the GDAD context is considered more appropriate than using one single communication tool because tools differ in scope [S5,S6]. Effective customer communication should be considered a key factor for successful GDAD projects [S8,S9]. It is important for a successful GDAD project to have efficient and effective knowledge sharing and transfer, which allow team members to understand the customer requirements and help team members plan and perform development activities [S4,S10]. Poor communication can cause severe problems for GDAD projects, including leading to inferior-quality products and delayed software delivery [S8,S11]. Moreover, agile practices in GDAD were found to be useful for enhancing communication and product quality [S7,S15].

## 5. Discussion

It is still arguable whether agile practices can be effectively scaled up and used in GDAD environments due to communication challenges. Despite its acknowledged importance, we found that

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

our knowledge about GDAD communication in practice is limited. This is the result of the study results being scattered, inconclusive, and ambiguous and scarcely any studies opening up the communication process or focusing on the social interaction and behaviour of teams as part of the research. It is not well-established how high communication efficiency and effectiveness can be achieved in this context. However, this SLR study confirms that there is a growing interest in GDAD communication, and most of the selected studies have reported the possibility of a successful implementation of GDAD and overcoming communication challenges (e.g., [S4,S20]).

Our findings reveal that *Distance Differences* were the most reported challenges for GDAD communication. Additionally, *Team Configuration*, *Organizational Factors* and *Human Factors* were reported to have significant negative impacts on GDAD communication. Other challenges were rarely reported, but they may have a strong effect on GDAD communication. For example, *Project Characteristics* were reported in only two studies despite software architecture being considered a substantial challenge in the traditional distributed software development environment.

Our findings also indicate that certain techniques should be used to support GDAD communication. GDAD communication requires some temporal overlap between GDAD teams or stakeholders to allow meetings to be held. Communication time must also be reduced to avoid temporal overlap, and supported distributed strategies should be used. These strategies can be implemented by synchronizing work hours, creating local teams, increasing the use of local meetings, using strict communication policies, reducing the number of GDAD team meetings, hosting team gatherings at the beginning of each project, exchanging visits between GDAD teams, requiring presentations by all team members, maintaining key documentation, and so on. Moreover, the current available technologies (e.g., *Skype*, *IM*) can be used to support and enhance GDAD communication.

Many studies have reported that using *Scrum* practices in GDAD may enhance communication and collaboration between GDAD teams (e.g., [S15,S18,S21]). This is consistent with the fact that the *Scrum* method is the most preferred agile method among GDAD teams. These studies have argued that *Scrum* practices enhance the GDAD communication through: (1) *Sprint* planning meetings, which can be arranged by teleconference or using exchange visits between teams, and dividing meetings into several parts (i.e., onsite and offsite), (2) daily *Scrum* meetings between distributed team members, (3) retrospective meetings, (4) a *Scrum Master*, who shapes the information flow between onsite and offsite teams, (5) a *Product Owner*, who can represent the customer, and (6) the use of a *Backlog* that all distributed members can access. However, Valone et al. [S19] reported that the members of one *Scrum* team should not be distributed over several sites and that every site should have at least one *Scrum Master* and one *Product Owner*.

According to the quality criteria reported in this study, some of the selected papers did not clearly report the recruitment strategy used for data collection, bias issues or the control group technique. Thus, there is a need for more quality empirical studies employing high-quality methodologies in the context of GDAD communication. In turn, such studies may enhance the usefulness of the research for agile practitioners and the academic community.

Similar to any other SLR study, this study has some implications and limitations, as discussed in the following sections.

### 5.1. Implications for research and practice

This systematic review has a number of implications for research and practice. For research, the findings highlight a vital research gap that needs more attention, namely, how efficient and effective GDAD communication is achieved in practice and what techniques are used to enhance GDAD communication. This is expected to increase interest among researchers as to what the effects are of the communication challenges that emerge in response to GDAD, what enables efficient and effective communication, and how this communication contributes to the GDAD project success. Hence, there is a need for more empirical studies in the field of GDAD communication.

Moreover, it has been found that the reviewed articles extensively focus on the challenges of GDAD and techniques to overcome those challenges. The results of this review paper show that empirical evidence on most GDAD communication categories (i.e., Distance Differences, Team Configuration, Customer Communication, Organizational Factors and Human Factors) is available. However, empirical evidence regarding the Project Characteristics challenges to GDAD communication is lacking. Future research may also investigate whether there are other critical challenges/factors that affect the GDAD communication efficiency and effectiveness.

In addition, we found that there is an abundance of qualitative and exploratory studies but not confirmatory and explanatory studies. This is consistent with the findings of [31]. We call for more confirmatory and explanatory research because many GDAD communication challenges and techniques have already been identified by practitioners and exploratory case studies. Similarly, design-oriented studies and action-oriented research using the available communication tools could help clarify the effects of the use of different GDAD communication practices. Furthermore, to obtain a deeper understanding of GDAD communication, previous authors have recommended that the research design needs to fit the current state of theory and research [18,19]. Therefore, the role of communication in GDAD could be investigated from the perspectives of related theories, such as *Activity Theory* [20] and *Coordination Theory* [38]. All such studies would increase the knowledge about GDAD communication and its impact on project success.

Finally, it has been noted that although knowledge about how to manage traditional distributed software teams' communication has accumulated over the past few decades, the governance of GDAD teams' communication (e.g., using software architecture guidelines) has not received much attention. Indeed, the incorporation of agile methods into traditional software distributed development environments poses many new challenges, highlighting the need for a better understanding of human and communication issues in the GDAD context [12]. For example, it is not clear whether GDAD communication is characterized by a higher or lower intensity, frequency, or quality relative to traditional distributed software development. None of the selected studies have attempted to compare or contrast communication in distributed traditional development to GDAD communication. Evidence suggests that instead of abandoning traditional distributed project communication principles, one should take advantage of these principles and combine them with agile communication-focused project management practices [11,19].

For practitioners, this review shows that many promising studies of the use of agile methods in GDAD have been reported. Despite the many GDAD communication challenges, some studies have reported that using *Scrum* practices (e.g., 2-week review meetings and monthly retrospective meetings) enhances communication between distributed sites (e.g., [S1,S14]). Other studies have reported that *XP* pair programming could be enabled between distributed teams using screen sharing and audio calls (e.g., [S7,S16]). Other studies have reported that the role of an on-site customer is unsustainable in GDAD; however, a customer representative may play the role of the customer in GDAD (e.g., [S10,S11]). In summary, the results of the review suggest that it is possible to achieve efficient and effective communication in GDAD, which facilitates project success and increases customer satisfaction. Moreover, this research provides evidence for practice



Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

suggesting that it is necessary to focus on human and organizational factors to increase GDAD communication efficiency and effectiveness. Specifically, GDAD communication requires a strong emphasis on personal communication and trust as well as a high level of tacit knowledge embedded in the team [18,19,40].

### 5.2. Limitations

Given the scope and time constraints of this research project, this study is limited to the number of selected search databases and the finite number of search strings. However, as discussed earlier, we are fully confident that the selected databases and search strings provided sufficient recent literature for identifying the current GDAD communication challenges and relevant recommendations. Moreover, other than *XP* and *Scrum*, we may have missed findings on other agile methods because they were not included in the search strings.

It is important to mention that there was no relationship bias between the researchers and the authors of the selected empirical studies used in this review. To help ensure that the selection process was unbiased, we followed the research questions, the identified keywords and the search terms that we carefully developed and reviewed to rectify the weaknesses and refine the selection process. As in any other SLR study, this approach does not guarantee that the keywords and search strings used have not caused the omission of other relevant studies. To further ensure the unbiased selection of articles, we utilized a multistage process to document and discuss the reasons for inclusion/exclusion at every stage.

The analysis, coding and labelling of concepts and categories are subject to human error and mistakes, which may lead to inconsistencies. The concepts and categories and their interconnections were identified by the first two researchers independently, and the third researcher was continuously involved in reviewing the findings. Checking was performed iteratively to minimize any possible omissions, errors or coding bias. The extracted data were then reviewed, and disagreements were resolved by consensus in review meetings by involving the third researcher, who has more experience in SLR.

## 6. Conclusions

Communication has been widely recognized as one of the key challenges of GDAD. The previous literature has reported on the concepts of GDAD communication challenges as well as the techniques for mitigating these challenges. However, no study has systematically reviewed and synthesized the empirical studies on GDAD communication. This paper presented a SLR of the communication challenges identified in empirical studies on GDAD. This study identified a number of challenges that need to be addressed for establishing efficient and effective GDAD communication. The data were analyzed and interpreted from the selected studies to answer the research questions at hand. The analysis and interpretation enabled us to construct the current state of the art regarding GDAD communication challenges.

The identified challenges of GDAD communication were categorized into six key categories: Distance Differences, Team Configuration, Customer Communication, Project Characteristics, Organizational Factors, and Human Factors. Each category includes some communication challenges (Table 8). The Distance Differences category was the most frequently mentioned category in the literature. It includes "time-zone" and "geographic" differences between GDAD teams, which were the most commonly mentioned communication challenges in the literature. Surprisingly, the most rarely mentioned category was Project Characteristics, which refers to the architecture used in GDAD and the domain of the project under development. The negative impacts of GDAD communication challenges were identified in this study (Table 10). The techniques for mitigating those impacts were also identified (Table 11). In addition, the impact of communication on GDAD success was identified, with a positive effect of communication efficiency and effectiveness on GDAD success being supported by the literature.

The findings of this review provide a knowledge base that can be helpful to agile practitioners who use or intend to use agile approaches in a GDAD environment. Practitioners may use these findings in the beginning of the GDAD project and make informed decisions about adopting context-specific policies, strategies, practices, roles and tools to support GDAD communication.

## Acknowledgment

We would like to thank the associate editor and the reviewers for their constructive comments and suggestions regarding the earlier drafts of this paper.

## Appendix A. Selected studies (studies selected for this SLR)

[S1] P.L. Bannerman, E. Hossain, R. Jeffery, Scrum practice mitigation of global software development coordination challenges: a distinctive advantage?, 45th Hawaii International Conference on System Science (HICSS), IEEE, 2012, pp. 5309–5318.
[S2] A. Boden, B. Nett, V. Wulf, Coordination practices in distributed software development of small enterprises, Second IEEE International Conference on Global Software Engineering (ICGSE 2007), IEEE, 2007, pp. 235–246.
[S3] J. Bosch, P. Bosch-Sijtsema, Coordination between global agile teams: from process to architecture, in: D. Šmite et al. (Ed.), Agility Across Time and Space, Springer, Berlin Heidelberg, 2010, pp. 217–233.
[S4] S. Dorairaj, J. Noble, P. Malik, Effective communication in distributed Agile software development teams, in: A. Sillitti et al. (Ed.), Agile Processes in Software Engineering and Extreme Programming, Springer, Berlin Heidelberg, 2011, pp. 102–116.
[S5] A.Q. Gill, D. Bunker, Towards the development of a cloud-based communication technologies assessment tool: an analysis of practitioners' perspectives, VINE 43 (2013) 57–77.
[S6] R. Green, T. Mazzuchi, S. Sarkani, Communication and quality in distributed agile development: an empirical case study, in: Proceedings of the World Academy of Science, Engineering and Technology (WASET), 2010, pp. 322–328.
[S7] H. Holmström, B. Fitzgerald, P.J. Ågerfalk, E.Ó. Conchúir, Agile practices reduce distance in global software development, Inf. Syst. Manag. 23 (2006) 7–18.
[S8] M. Korkala, P. Abrahamsson, Communication in distributed agile development: a case study, 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE, 2007, pp. 203–210.
[S9] M. Korkala, F. Maurer, Waste identification as the means for improving communication in globally distributed agile software development, J. Syst. Softw. 95 (2014) 122–140.

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

[S10] M. Korkala, M. Pikkarainen, K. Conboy, A case study of customer communication in globally distributed software product development, in: Proceedings of the 11th International Conference on Product Focused Software, ACM, 2010, pp. 43–46.
[S11] L. Layman, L. Williams, D. Damian, H. Bures, Essential communication practices for Extreme Programming in a global software development team, Inf. Softw. Technol. 48 (2006) 781–794.
[S12] S. Lee, H.-S. Yong, Distributed agile: project management in a global environment, Empir. Softw. Eng. 15 (2010) 204–217.
[S13] L.M. Maruping, Implementing extreme programming in distributed software project teams: strategies and challenges, in: D. Šmite et al. (Ed.), Agility Across Time and Space, Springer, Berlin Heidelberg, 2010, pp. 11–30.
[S14] M. Paasivaara, C. Lassenius, D. Damian, P. Raty, A. Schroter, Teaching students global software engineering skills using distributed scrum, 35th International Conference on Software Engineering (ICSE), IEEE, 2013, pp. 1128–1137.
[S15] 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), IEEE, 2009, pp. 195–204.
[S16] H. Sharp, R. Giuffrida, G. Melnik, Information flow within a dispersed agile team: a distributed cognition perspective, in: C. Wohlin (Ed.), Agile Processes in Software Engineering and Extreme Programming, Springer, Berlin Heidelberg, 2012, pp. 62–76.
[S17] R. Sindhgatta, B. Sengupta, S. Datta, Coping with distance: an empirical study of communication on the jazz platform, in: Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, ACM, 2011, pp. 155–162.
[S18] A. Välimäki, J. Kääriäinen, Patterns for distributed Scrum – a case study, in: Enterprise interoperability III, Springer, London, 2008, pp. 85–97.
[S19] R. Vallon, S. Strobl, M. Bernhart, T. Grechenig, Inter-organizational co-development with Scrum: experiences and lessons learned from a distributed corporate development environment, in: H. Baumeister and B. Weber (Eds.), Agile Processes in Software Engineering and Extreme Programming, Springer, Berlin Heidelberg, 2013, pp. 150–164.
[S20] V. Yadav, M. Adya, D. Nath, V. Sridhar, Investigating an 'agile-rigid' approach in globally distributed requirements analysis, in: Proceedings of the 11th Pacific-Asia Conference on Information Systems, 2007, pp. 151–165.
[S21] F. Zieris, S. Salinger, Doing Scrum rather than being agile: a case study on actual nearshoring practices, 8th International Conference on Global Software Engineering (ICGSE), IEEE, 2013, pp. 144–153.

## Appendix B. Data extraction form (adopted from [19])

| Study description | | |
|---|---|---|
| 1 | Study identifier | Unique ID for the study |
| 2 | Date of data extraction | |
| 3 | Bibliographic reference | Author, year, title, source |
| 4 | Type of article | Journal article, conference paper, workshop paper, book section |
| 5 | Study aims | What were the aims of the study? |
| 6 | Objectives | What were the objectives? |
| 7 | Design of study | Qualitative, quantitative (experiment, survey, case study, action research) |
| 8 | Research hypothesis | Statement of hypotheses, if any |
| 9 | Sample description | Size, students, professionals (age, education, experience) |
| 10 | Setting of study | Industry, in-house/supplier, products and processes used |
| 11 | Control group | Yes, no (number of groups, sample size) |
| 12 | Data collection | How were the data obtained? (questionnaires, interviews, forms) |
| 13 | Data analysis | How were the data analyzed? (qualitative, quantitative) |
| Study findings | | |
| 1 | Findings and conclusions | What were the findings and conclusions? |
| 2 | Validity | Limitations, threats to validity |
| 3 | Relevance | Research, practice |

## Appendix C. Aims of the selected studies

| Study | Study aim |
|---|---|
| S1 | Report the impact of temporal, geographical and sociocultural distance and seven *Scrum* practices on the use of *Scrum* practices in global software development (GSD) |
| S2 | Show to what extent small and medium enterprises rely upon situated coordination practices to warrant their agility |
| S3 | Study the relation between large-scale and agile approaches to software development as well as current problems |
| S4 | Explore distributed agile software development communication challenges from the perspective of agile practitioners |
| S5 | Investigate and present the developers' needs for communication technologies in the context of distributed adaptive development environments (DADE) |
| S6 | Explore the relevancy of communication richness in various agile in distributed agile development phases and its impact on quality |
| S7 | Explore how agile practices can reduce three types of "distance", namely, temporal, geographical, and sociocultural, in global software development (GSD) |
| S8 | Compare case study findings against existing recommendations about communication in distributed agile development |
| S9 | Identify non-value-producing communication elements and solutions to mitigate them |
| S10 | Describe results from a case study of customer communication in a globally distributed product development programme applying both traditional and agile approaches |

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

**Appendix C** (*Continued*)

| Study | Study aim |
|---|---|
| S11 | Understand how globally distributed team created a successful project in a new problem domain using a methodology that is dependent on informal, face-to-face communication |
| S12 | Highlight successful practices and challenges that have been overcome by the globalization project and suggest a framework for software globalization project management using a distributed agile approach |
| S13 | Outline some of the strategies and challenges associated with implementing agile methods in distributed software project teams |
| S14 | Describe distributed *Scrum* augmented with best practices in global software engineering (GSE) |
| S15 | Present findings of successfully adopted project in distributed *Scrum* |
| S16 | Report on a field study of one successful partially dispersed agile team |
| S17 | Study the communication characteristics of team members in a large, globally distributed software development project that uses the IBM Jazz platform (*Scrum*) |
| S18 | Find the best practices for distributed *Scrum* |
| S19 | Investigate an agile approach from a real world project in a distributed development environment |
| S20 | Report an exploratory quasi-experimental study, which investigates the performance of requirements analysis projects in an 'agile-rigid' distributed environment |
| S21 | Present the actual *Scrum* process and practices of the external teams and contrast them with the intended way of proceeding |



## References

[1] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods: review and analysis, VTT Technical Report, Finland, 2002.
[2] P. Abrahamsson, K. Conboy, X. Wang, Lots done, more to do: the current state of agile systems development research, Eur. J. Inf. Syst. 18, 2009, pp. 281–284.
[3] P.J. Agerfalk, B. Fitzgerald, H. Holmstrom, B. Lings, B. Lundell, E.O. Conchúir, A framework for considering opportunities and threats in distributed software development, International Workshop on Distributed Software Development, Citeseer, 2005, pp. 47–61.
[4] P. Agerfalk, B. Fitzgerald, S. Slaughter, Flexible and distributed information systems development: state of the art and research challenges, Inf. Syst. Res. 20, 2009, pp. 317–328.
[5] Agile Manifesto, Manifesto for Agile Software Development, 2001 http://www.agilemanifesto.org/ , [20.9.13].
[6] M. Agrawal, K. Chari, Software effort, quality, and cycle time: a study of CMM level 5 projects, IEEE Trans. Softw. Eng. 33, 2007, pp. 145–156.
[7] A.M. Aladwani, An integrated performance model information systems projects, J. Manag. Inf. Syst. 19, 2002, pp. 185–210.
[8] M. Ali Babar, T. Ihme, M. Pikkarainen, An industrial case of exploiting product line architectures in agile software development, in: Proceedings of the 13th International Software Product Line Conference, Carnegie Mellon University, 2009, pp. 171–179.
[9] V. Balijepally, R. Mahapatra, S. Nerur, K.H. Price, Are two heads better than one for software development? The productivity paradox of pair programming MIS Quart. 2009, pp. 91–118.
[10] J.B. Barlow, J. Giboney, M.J. Keith, D. Wilson, R. Schuetzler, P.B. Lowry, A. Vance, Overview and guidance on agile development in large organizations, Commun. Assoc. Inf. Syst. 29, 2011, pp. 25–44.
[11] D. Batra, W. Xia, D. VanderMeer, K. Dutta, Balancing agile and structured development approaches to successfully manage large distributed software projects: a case study from the cruise line industry, Commun. Assoc. Inf. Syst. 27, 2010, pp. 379–394.
[12] B. Boehm, R. Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley Professional, 2003.
[13] R.P. Bostrom, Successful application of communication techniques to improve the systems development process, Inf. Manag. 16, 1989, pp. 279–295.
[14] H.H. Clark, S.E. Brennan, Grounding in communication, Perspect Soc. Shar. Cognit. 13, 1991, pp. 127–149.
[15] K. Conboy, Agility from first principles: reconstructing the concept of agility in information systems development, Inf. Syst. Res. 20, 2009, pp. 329–354.
[16] A.R. Dennis, R.M. Fuller, J.S. Valacich, Media, tasks, and communication processes: a theory of media synchronicity, MIS Quart. 32, 2008, pp. 575–600.
[17] T. Dingsøyr, S. Nerur, V. Balijepally, N.B. Moe, A decade of agile methodologies: towards explaining agile software development, J. Syst. Softw. 85, 2012, pp. 1213–1221.
[18] T. Dingsøyr, T.E. Fægri, J. Itkonen, What is large in large-scale? A taxonomy of scale for agile software development in: A. Jedlitschka (Ed.), et al., Product-Focused Software Process Improvement, Springer, Switzerland, 2014, pp. 273–276.
[19] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Inf. Softw. Technol. 50, 2008, pp. 833–859.
[20] Y. Engeström, Activity theory and individual and social transformation, in: I.Y. Engeström, R. Miettinen, R.-L. Punamäki (Eds.), Perspectives on Activity Theory, Cambridge University Press, 1999, pp. 19–38.
[21] B. Fitzgerald, G. Hartnett, K. Conboy, Customising agile methods to software practices at Intel Shannon, Eur. J. Inf. Syst. 15, 2006, pp. 200–213.
[22] A. Fruhling, G.-J.D. Vreede, Field experiences with eXtreme programming: developing an emergency response system, J. Manag. Inf. Syst. 22, 2006, pp. 39–68.
[23] A.Q. Gill, D. Bunker, Conceptualization of a context aware cloud adaptation (CACA) framework, 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), IEEE, Sydney, Australia, 2011, pp. 760–767.
[24] A.Q. Gill, D. Bunker, P. Seltsikas, Evaluating a communication technology assessment tool (Ctat): a case of a cloud based communication tool, in: Proceedings of the PACIS 2012, paper 88, 2012.
[25] J.D. Herbsleb, Global software engineering: the future of socio-technical coordination, 2007 Future of Software Engineering, IEEE Computer Society, 2007, pp. 188–198.
[26] J.D. Herbsleb, A. Mockus, An empirical study of speed and communication in globally distributed software development, IEEE Trans. Softw. Eng. 29, 2003, pp. 481–494.
[27] J.D. Herbsleb, D. Moitra, Global software development, IEEE Softw. 18, 2001, pp. 16–20.
[28] G. Hofstede, G. Hofstede, M. Minkov, Cultures and Organizations: Software of the Mind, McGraw-Hill, New York, 2010.
[29] H. Holmstrom, E.Ó. Conchúir, 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), IEEE, 2006, pp. 3–11.
[30] E. Hossain, P.L. Bannerman, R. Jeffery, Towards an understanding of tailoring scrum in global software development: a multi-case study, in: Proceedings of the 2011 International Conference on Software and Systems Process, ACM, 2011, pp. 110–119.
[31] M. Hummel, C. Rosenkranz, R. Holten, The role of communication in agile systems development, Bus. Inf. Syst. Eng. 5, 2013, pp. 343–355.
[32] H. Keskin, Antecedents and consequences of team memory in software development projects, Inf. Manag. 46, 2009, pp. 388–396.
[33] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Ver. 2.3 EBSE Technical Report, EBSE, 2007.
[34] K. Kuusinen, T. Mikkonen, S. Pakarinen, Agile user experience development in a large software organization: good expertise but limited impact, in: M. Winckler, P. Forbrig, R. Bernhaupt (Eds.), Human-Centered Software Engineering, Springer, Berlin Heidelberg, 2012, pp. 94–111.
[35] F. Lanubile, Collaboration in distributed software development, in: A. De Lucia, F. Ferrucci (Eds.), Software Engineering, Springer, Berlin Heidelberg, 2009 , pp. 174–193.
[36] G. Lee, W. Xia, Toward agile: an integrated analysis of quantitative and qualitative field data, MIS Quart. 34, 2010, pp. 87–114.
[37] K. Lyytinen, G.M. Rose, Information system development agility as organizational learning, Eur. J. Inf. Syst. 15, 2006, pp. 183–199.
[38] T.W. Malone, K. Crowston, What is coordination theory and how can it help design cooperative work systems? in: Proceedings of the 1990 ACM conference on Computer-supported cooperative work, ACM, 1990, pp. 357–370.
[39] A. Martini, L. Pareto, J. Bosch, Communication factors for speed and reuse in large-scale agile software development, in: Proceedings of the 17th International Software Product Line Conference, ACM, 2013, pp. 42–51.
[40] L.M. Maruping, V. Venkatesh, R. Agarwal, A control theory perspective on agile methodology use and changing user requirements, Inf. Syst. Res. 20 (2009), 2009, pp. 377–399.
[41] D. McQuail, Mass Communication Theory: An Introduction, Sage Publications, Inc., 1987.
[42] D. Mishra, A. Mishra, S. Ostrovska, Impact of physical ambiance on communication, collaboration and coordination in agile software development: an empirical evaluation, Inf. Softw. Technol. 54, 2012, pp. 1067–1078.
[43] S. Modi, P. Abbott, S. Counsell, Negotiating common ground in distributed agile development: a case study perspective, 8th International Conference on Global Software Engineering (ICGSE), IEEE, 2013, pp. 80–89.
[44] S. Nerur, R.K. Mahapatra, G. Mangalaraj, Challenges of migrating to agile methodologies, CACM 48, 2005, pp. 72–78.
[45] M. Niazi, S. Mahmood, M. Alshayeb, B. Baqais, A. Ahmed, A.Q. Gill, Motivators of adopting social computing in global software development: initial results, in: Proceedings of the World Congress on Engineering and Computer Science, London, UK, 2013, pp. 1–5.
[46] M. Paasivaara, C. Lassenius, Using Scrum practices in GSD projects, in: D. Šmite (Ed.), et al., Agility Across Time and Space, Springer, Berlin Heidelberg, 2010, pp. 259–278.

Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003

[47] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, J. Still, The impact of agile practices on communication in software development, Empir. Softw. Eng. 13, 2008, pp. 203–337.
[48] A. Qumer, B. Henderson-Sellers, Crystallization of agility back to basics, in: Proceedings of the First International Conference on Software and Data Technologies (ICSOFT), 2006, pp. 121–126.
[49] A. Qumer, B. Henderson-Sellers, An evaluation of the degree of agility in six agile methods and its applicability for method engineering, Inf. Softw. Technol. 50, 2008, pp. 280–295.
[50] B. Ramesh, L. Cao, K. Mohan, P. Xu, Can distributed software development be agile? CACM 49, 2006, pp. 41–46.
[51] F. Rowe, What literature review is not: diversity, boundaries and recommendations, Eur. J. Inf. Syst. 23, 2014, pp. 241–255.
[52] S. Sarker, S. Sarker, Exploring agility in distributed information systems development teams: an interpretive study in an offshoring context, Inf. Syst. Res. 20, 2009, pp. 440–461.
[53] K. Siau, Y. Long, M. Ling, Toward a unified model of information systems development success, J. Database Manag. 21, 2010, pp. 80–101.
[54] M. Staron, J. Hansson, J. Bosch, Performance in software development, Inf. Softw. Technol. 56, 2014, pp. 463–464, (Special issue editorial).
[55] J. Sutherland, A. Viktorov, J. Blount, N. Puntikov, Distributed scrum: agile project management with outsourced development teams, 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), IEEE, Hawaii, United States, 2007, pp. 274a–284a.
[56] J. Sutherland, G. Schoonheim, M. Rijk, Fully distributed scrum: replicating local productivity and quality with offshore teams, 42nd Hawaii International Conference on System Sciences (HICSS'09), IEEE, Hawaii, United States, 2009, pp. 1–8.
[57] R. Vidgen, X. Wang, Coevolving systems and the organization of agile software development, Inf. Syst. Res. 20, 2009, pp. 355–376.
[58] L. Wallace, M. Keil, A. Rai, Understanding software project risk: a cluster analysis, Inf. Manag. 42, 2004, pp. 115–125.



Please cite this article in press as: Y.I. Alzoubi, et al., Empirical studies of geographically distributed agile development communication challenges: A systematic review, Inf. Manage. (2015), http://dx.doi.org/10.1016/j.im.2015.08.003