

Scrum Practices in Global Software Development: A Research Framework

Emam Hossain, Paul L. Bannerman, and D. Ross Jeffery

NICTA, Australian Technology Park, Sydney, Australia
UNSW, The University of New South Wales, Sydney, Australia
{Emam.Hossain, Paul.Bannerman, Ross.Jeffery}@nicta.com.au

Abstract. Project stakeholder distribution in Global Software Development (GSD) is characterized by temporal, geographical and socio-cultural distance, which creates challenges for communication, coordination and control. Practitioners constantly seek strategies, practices and tools to counter the challenges of GSD. There is increasing interest in using Scrum in GSD even though it originally assumed collocation. However, empirically, little is known about how Scrum practices respond to the challenges of GSD. This paper develops a research framework from the literature as a basis for future research and practice. The framework maps current knowledge and views on how Scrum practices can be used to mitigate commonly recognized challenges in GSD. This research is useful as a reference guide for practitioners who are seeking to understand how Scrum practices can be used effectively in GSD, and for researchers as a research framework to validate and extend current knowledge.

1 Introduction

Global Software Development (GSD) is a major trend in software engineering. Rapid advances in computer networks, telecommunications, internet technologies and collaboration tools have provided enough infrastructures to enable firms to take advantage of high-skilled, low-cost offshore resources in developing and maintaining new software. It is claimed that GSD can reduce time to market, increase productivity, improve quality, and provide cost efficiencies for software developing organizations [28]. However, as well as expected benefits, GSD challenges have also been identified [29]. In particular, GSD is usually characterized by engagements with different national and organizational cultures in different geographic locations and time zones, using various traditional and IT-enabled means to collaborate. Such arrangements may encounter major difficulties in team communication, coordination, control, infrastructure compatibility, conflicting expectations, achieving shared understanding and building trust [8]. Therefore, GSD practitioners need to employ suitable context-specific mechanisms to mitigate these problems. Various solutions have been proposed, such as dividing work into separate well-specified independent modules to minimize communications between sites [30]. However, in practice, practitioners still experience significant issues and continue to look for more effective mechanisms to mitigate inherent GSD challenges and risks [31].

Agile methods are gaining popularity in software development contexts characterized by high uncertainty. They promise handling requirements changes throughout the development lifecycle; promote extensive collaboration between customers and developers; and support early and frequent delivery of products [32]. However, an issue for GSD is that the design of Agile methods assumes collocated development teams whereas GSD assumes distributed teams. Notwithstanding this, practitioners have begun to adapt Agile practices to GSD to leverage the advantages of both approaches [33]. Among the current Agile methods, Scrum practices are increasingly being considered and trialled in GSD projects. Initial empirical findings suggest that using Scrum practices in GSD can improve communication, trust, motivation and product quality [1]. Also, industry experience suggests that Scrum practices may promote communication, collaboration, frequent product delivery, and reduce some GSD challenges and risks [8, 24].

We acknowledge that GSD challenges apply to any software development method used in GSD projects, including non-Agile methods, and that Scrum may share some mitigation mechanisms, including complementary enabling tools, with other methods. However, our focus in this study is solely on Scrum practices, not other methods or the relative capabilities of Scrum versus other software development methods.

Little empirical research has been done on how Scrum practices can mitigate the challenges presented by the distributed nature of GSD projects [34]. A small number of case studies discuss the benefits and difficulties of using Scrum practices in GSD [1, 2], but most reports in the literature record only industrial experiences. Therefore, a knowledge gap exists. There is a need for more empirical studies on how Scrum is being used in GSD and how Scrum practices might be applied and with what tool support to overcome the challenges encountered in GSD.

To begin to address this gap, we propose a research framework that maps current literature-based prescriptions on how Scrum practices might mitigate commonly recognised GSD challenges. The framework draws on views from 27 primary papers that were identified in a Systematic Literature Reviews (SLR), following the guidelines prescribed in [35].

Since most current software engineering literature on the topic comprises industry experience reports, the proposed framework offers two main contributions: first, it provides a consolidated representation of current views on how Scrum practices might be applied to overcome the recognized challenges of GSD, and; second, it provides an integrated set of initial propositions about the potential role of Scrum practices in GSD that can be validated in subsequent empirical studies and used to guide practice.

The remainder of this paper is organized as follows. Section 2 overviews the literature and describes the framework development. Section 3 describes the research framework. Section 4 then discusses the contribution and future research before concluding in the last section.

2 Research Background

This section overviews the literature on GSD challenges and Scrum practices in GSD as background to describing the development of the proposed research framework.

2.1 GSD Challenges

There is a growing body of literature that focuses on challenges in GSD. Communication, coordination and control challenges are recognised to arise due to geographic, temporal and socio-cultural distances encountered in GSD [33]. A recent literature review also identified group awareness, knowledge management, coordination, collaboration, project and process management, process support, and risk management as likely issues in distributed software development [28].

Temporal distance may create communication issues such as reduced hours of collaboration, difficulties in holding synchronous meetings, and response delays [40]. As a result, GSD coordination processes may also be significantly affected [33]. *Geographic distance* makes communication difficult because of the reduced ability to hold face-to-face meetings [32]. Lack of face-to-face meetings reduces informal contact and this can lead to a lack of critical task awareness, lack of “teamness” and reduced trust [32, 37]. Therefore, a fundamental GSD problem is that many of the mechanisms that function to coordinate work in a collocated setting are absent or disrupted [12]. *Socio-cultural distance* may create issues relating to inconsistent work practices, different perceptions of authority, and lack of mechanisms for creating shared understanding and avoiding misunderstandings and reduced cooperation [1, 32, 33, 38, 44].

Several GSD issues frameworks and models already exist in the literature [24, 33, 34, 39, 40]. For example, one maps communication, coordination and control process issues against temporal, geographical and socio-cultural distance dimensions [33]; others present risk frameworks for the use of Agile practices in GSD [24, 31]; and another presents a model of how remote communication and knowledge management, cultural diversity and time differences negatively impact requirements engineering in GSD [39]. Our proposed framework integrates the GSD issues classification scheme of [33] as one dimension and uses Scrum practices [1] as the other dimension.

2.2 Scrum Practices in GSD

Scrum is an iterative, time-boxed, incremental project management method based on a simple “inspect and adapt” framework [29]. A major reason for the success of Scrum is the physical collocation of development team members [32]. However, the literature also reports instances of success in using Scrum practices in GSD [34]. For example, a recent study found that using Scrum practices in GSD improves communication, trust, motivation and quality [1]. Some reports also claim that some Scrum practices can mitigate some of the recognized GSD challenges [8, 24]. For example, based on experience, one report claims that Scrum practices such as daily scrum, scrum of scrums, sprint planning and retrospective meetings engage distributed team members in collaboration, help visualization of hidden problems, develop trust and increase team spirit [23]. Similarly, [8] claims that sprint planning provides shared visualization of project activities and increases “teamness”. Furthermore, [21] suggests that daily scrum meetings bring transparency and encourage informal communication among distributed stakeholders; sprints provide frequent offsite work monitoring opportunities; sprint planning meetings provide shared understanding of common goals and improve task awareness, and; sprint ‘demos’ bring transparency to stakeholders and prevent problems early.

Our proposed research framework draws on the Scrum practice set identified in one of the few empirical studies of the use of Scrum in GSD, namely [1].

2.3 Research Framework Development

In sum, a review of the developing literature on the use of Scrum practices in GSD indicates encouraging results [34]. However, there is a paucity of empirical studies in the literature. Most current studies are industry experience reports so there is a need for further empirical research.

An initial way forward is to construct a research framework from the current reports as a basis on which to empirically validate and build on current knowledge and experience. The framework is intended to integrate current literature-based results and views about how Scrum practices have been found to be effective (or are expected to contribute) in mitigating the challenges that arise in GSD. We propose such a framework and approach in the remainder of this paper.

The framework, in Table 4, shows GSD challenges as rows and Scrum practices as columns. The rows list challenges facing software development in global contexts identified in [33] and [40]. Twelve GSD challenges were identified from [33] and [40] in three broad categories (communication, coordination, and control), each relating to a particular source characteristic (temporal, geographical or socio-cultural distance). These challenges are summarized in Table 1, adapted from [33] (and each challenge is further described in [40]). For example, *reduced opportunities for synchronous communication* are attributed to communication and feedback delays due to offset time zones between locations (refer to [40] for the remainder).

Table 1. Summary of GSD challenges

Process Category	Challenge	Source Category
Communication	Reduced opportunities for synchronous communication	Temporal distance
	Face to face meetings difficult	Geographical distance
	Cultural misunderstandings	Socio-cultural distance
Coordination	Increased coordination costs	Temporal distance
	Reduced informal contact can lead to lack of critical task awareness	Geographical distance
	Inconsistent work practices can impinge on effective coordination	Socio-cultural distance
	Reduced cooperation arising from misunderstanding	Socio-cultural distance
Control	Management of project artefacts may be subject to delays	Temporal distance
	Difficult to convey vision and strategy	Geographical distance
	Perceived threat from training low-cost 'rivals'	Geographical distance
	Different perceptions of authority can undermine morale	Socio-cultural distance
	Managers must adapt to local regulations	Socio-cultural distance

The columns in the reference framework in Table 4 list seven Scrum practices identified in an empirical study of the use of Scrum in GSD [1]. These are: sprint planning meeting, sprints, daily scrum, scrum of scrums, sprint demo (or review), retrospective meeting, and backlog (refer to [1] for a description of these practices).

The table cells are populated from our literature review, with the literature-based challenge mitigation mechanisms summarized in categories. The categories, described in Tables 2 and 3, were derived as follows. The initial SLR [34] and a subsequent

search using the same protocol found 27 primary papers [1-27]. These papers were examined for empirical evidence, unsupported theory, and/or observations from experience indicating how Scrum practices might mitigate GSD challenges. The search resulted in 202 scrum-related mechanisms being identified. Due to the large number, the mechanisms were then examined to find common themes, which enabled them to be clustered into eight categories. The categories are described in Table 2. The primary papers that identified one or more mechanisms within a category are marked with an X in Table 3 (a spreadsheet of the full detailed list of practices, reference sources and categorizations is available from the first author upon request).

The search protocol, paper selection and practice extraction and categorisation were enacted, validated and cross-checked by two researchers.

Table 2. GSD Challenge Mitigation Mechanism Categories

Category	ID	Description
Synchronized work hours	GSD_P1	Increase overlapping working hours between sites to enable synchronous communication for meetings; for example, adjust working hours at sites to create some overlap or participate in meetings from home
ICT-mediated synchronous communication	GSD_P2	Practices that enable synchronous formal or informal communication between teams; for example, use individual or conference phone calls, teleconference, video conference, web conference, or application
ICT-mediated asynchronous communication	GSD_P3	Practices that enable asynchronous communication between team members; for example, email, Instant Messaging, or Wiki
Visit	GSD_P4	Face-to-face meeting made possible by travelling between sites. Two main kinds: seeding visits to build relationships, and; maintaining visits to sustain relationships
Frequent (or Improved) communication	GSD_P5	Enable frequent formal and informal communication among team members through tools and/or face-to-face meetings
Iteration	GSD_P6	Activities that involve cyclical repetition enable multiple incremental opportunities to monitor progress and resolve issues
Review	GSD_P7	Formal or informal activities that enable reflection on prior activities, assessment of completed work, and the opportunity for stakeholders to provide feedback to the teams
Planning	GSD_P8	Activities that establish the scope of work, resourcing, scheduling, and the processes to be employed

The categories described in Table 2 reflect a range of approaches found in the literature to overcome challenges arising from the temporal, geographical, and socio-cultural distances encountered in GSD. For example, mechanisms to address *temporal distance* include adjusting working hours [37], increased use of asynchronous collaboration tools [39] and use of bridging across sites [40]. To address *geographical distance*, mechanisms include using synchronous and asynchronous communication tools [39], visits (travel) [40] and modularization of work [38]. To address *socio-cultural distance*, mechanisms include liaisons [40], increased use of asynchronous tools [39], and planned rotations (visits) [39]. While these mechanisms appear to be generic (that is, they are equally applicable to any development method), the underlying principles of Agile methods imply that Scrum practice may leverage additional benefits from their use. For example, the last four categories (GSD_P5 to GSD_P8), represent mechanisms that reflect inherent characteristics of many Scrum practices themselves (that is, the Scrum practices themselves may mitigate the GSD challenge).

Table 3. GSD challenge mitigation mechanism categories referenced in the literature

Mechanism Categories Paper Reference	GSD_P1	GSD_P2	GSD_P3	GSD_P4	GSD_P5	GSD_P6	GSD_P7	GSD_P8
[1]	X	X	X		X	X	X	X
[2]	X		X		X		X	
[3]	X	X	X	X	X		X	X
[4]	X	X				X	X	X
[5]	X	X				X	X	X
[6]	X	X	X	X	X	X	X	X
[7]					X			
[8]		X	X		X			X
[9]	X		X			X	X	
[10]			X				X	X
[11]		X					X	X
[12]	X	X			X	X		
[13]	X	X	X					X
[14]	X	X	X				X	X
[15]		X		X	X	X	X	X
[16]	X	X	X		X		X	X
[17]	X	X			X	X	X	
[18]					X		X	X
[19]	X		X		X	X	X	
[20]		X		X	X	X	X	
[21]	X	X	X	X	X	X	X	X
[22]					X		X	
[23]			X		X		X	X
[24]		X			X		X	
[25]	X	X	X	X	X		X	X
[26]	X	X			X			
[27]	X	X						

The findings on how Scrum practices might contribute to overcoming each GSD challenge are described in the next section. Empty cells in Table 4 indicate that no mitigation mechanism was found relating to that combination of variables in the literature. The framework contributes an integrated summary of the literature-based findings and expectations of how common Scrum practices might mitigate (minimize or reduce to zero) the distance-based challenges that can arise in developing software in dispersed global locations.

The research framework provides value as an independent research outcome, as a reference guide to inform current practice, and as a basis upon which to conduct further empirical research (starting with validation of the literature-based views contained within it).

3 Research Framework

This section describes the proposed research framework.

3.1 Communication Challenges

According to the literature, temporal, geographical and socio-cultural distance may impact GSD communication processes by creating three main challenges: reduced

opportunities for synchronous communication; face-to-face meeting [are] difficult, and; cultural misunderstandings (Table 1). The mechanisms that may mitigate each challenge, summarised by category in Table 4, are discussed following.

Reduced opportunities for synchronous communication (due to temporal distance):

The effect of time zone offsets can be so great that there is little or no opportunity for direct contact between dispersed team members. The literature suggests that Scrum practices can mitigate this challenge by *synchronizing work hours* (GSD_P1) and/or *ICT-mediated asynchronous communication* (GSD_P3). For example, Scrum team members may participate in daily scrums in common working hours or outside normal hours by adjusting working hours, or by emailing answers to the three questions before the meeting (What did I do yesterday? What will I do today? What impediments are in my way?), and reviewing minutes emailed back after the meeting. Similarly, for most (if not all) of the other meeting practices, team members from each site may synchronize work hours to achieve a suitable meeting time and/or use various asynchronous communication tools (such as wiki or email).

Face-to-face meetings difficult (due to geographical distance): Due to developers being located in different countries, it can be difficult to hold face-to-face meetings. The literature suggests that this challenge can be met by using *ICT-mediated synchronous communication* (GSD_P2) to hold meetings online. It also suggests, in the case of sprint planning, that *ICT-mediated asynchronous communication* (GSD_P3), and/or *visits* (GSD_P4) can be used, and; in the case of sprints, that the challenge can be mitigated by *visits* (GSD_P4) to another site. Distributed teams may use a wide range of ICT-mediated synchronous communication tools to support their Scrum practices, such as: teleconference; videoconference; web-conferencing; audio/video Skype; Microsoft Office Communicator with Live Meeting; and/or desktop sharing and Microsoft Net Meeting for application sharing. Similarly, ICT-mediated asynchronous communication tools such as email, Internet Relay Chat (IRC), and Wiki may be used to support meetings. Distributed team members may also physically visit a counterpart site to plan and/or conduct a sprint as a collocated team; or, in some cases, when new knowledge needs to be transferred, some offshore team members may visit the onshore site to work for a few sprints.

Cultural misunderstandings (due to socio-cultural distance): With developers located in different countries, misunderstandings can occur due to language and cultural differences. The literature suggests that Scrum practices may reduce cultural misunderstandings by using: *ICT-mediated asynchronous communication* (GSD_P3); *visits* (GSD_P4), and/or; *frequent (or improved) communication* (GSD_P5). For example, distributed Scrum team members could post answers to the three Scrum questions into a wiki (an ICT-mediated asynchronous communication tool) which can help to reduce misunderstandings. Similarly, visits by distributed team members to other sites for sprints can also increase familiarization and understanding between teams, and; frequent participation by distributed teams in daily scrums and sprint planning meetings can provide many communication opportunities to break down cultural barriers and increase cultural awareness.

Table 4. Research Framework: GSD Challenge Mitigation Mechanisms by Categories

	Scrum practice GSD challenges	Sprint Planning	Sprint	Daily Scrum	Scrum of Scrums	Sprint Review	Retro- spective	Backlog
Communication	<i>Temporal:</i> Reduced opportunities for synchronous communication	GSD_P1, GSD_P3		GSD_P1, GSD_P3	GSD_P1	GSD_P1, GSD_P3	GSD_P1, GSD_P3	
	<i>Geographical:</i> Face to face meetings difficult	GSD_P2, GSD_P3, GSD_P4	GSD_P4	GSD_P2	GSD_P2	GSD_P2	GSD_P2	
	<i>Cultural:</i> Cultural misunderstandings	GSD_P5	GSD_P4	GSD_P3, GSD_P5				
Coordination	<i>Temporal:</i> Increased coordination costs	GSD_P1, GSD_P3		GSD_P1, GSD_P3	GSD_P1	GSD_P1, GSD_P3	GSD_P1, GSD_P3	
	<i>Geographical:</i> Reduced informal contact can lead to lack of critical task awareness	GSD_P2, GSD_P8	GSD_P4, GSD_P6	GSD_P3, GSD_P5, GSD_P7, GSD_P8	GSD_P5, GSD_P7, GSD_P8	GSD_P2, GSD_P7	GSD_P3, GSD_P7	GSD_P3, GSD_P7, GSD_P8
	<i>Cultural:</i> Inconsistent work practices can impinge on effective coordination		GSD_P6	GSD_P5				
	<i>Cultural:</i> Reduced cooperation arising from misunderstandings	GSD_P3, GSD_P5, GSD_P8	GSD_P4, GSD_P6	GSD_P5	GSD_P5	GSD_P5, GSD_P7	GSD_P5, GSD_P7	
Control	<i>Temporal:</i> Management of artifacts may be subject to delays							
	<i>Geographical:</i> Difficult to convey vision and strategy		GSD_P4, GSD_P8	GSD_P5, GSD_P8		GSD_P7		
	<i>Geographical:</i> Perceived threat from training low cost 'rivals'							
	<i>Cultural:</i> Different perceptions of authority can undermine morale	GSD_P5		GSD_P5		GSD_P5		
	<i>Cultural:</i> Managers must adapt to local regulations							

3.2 Coordination Challenges

According to the literature, temporal, geographical and socio-cultural distance may impact GSD coordination by creating four main challenges: increased coordination costs; reduced informal contact, leading to a lack of critical task awareness; inconsistent work practices that can impinge on effective coordination, and; reduced cooperation arising from misunderstandings (Table 1). The mechanisms that may mitigate each challenge, summarised by category in Table 4, are discussed following.

Increased coordination costs (due to temporal distance): The effect of time zone differences can be so great that project coordination complexity and costs increase. As for the communication challenge, the literature suggests that Scrum practices can be used to mitigate coordination costs by: *synchronizing work hours* (GSD_P1), and/or; using *ICT-mediated asynchronous communication* (GSD_P3). For example, distributed team members may adjust work hours to participate in Scrum meetings, thereby

improving project coordination by enabling progress to be reviewed, work to be planned and open issues resolved between sites in a cost effective manner. Costs may also be reduced by distributing relevant material before and/or after meetings via asynchronous means (such as email, audio and video recordings, and wikis).

Reduced informal contact can lead to lack of critical task awareness (due to geographical distance): Due to geographical dispersion, lack of close interaction between developers may reduce team awareness. The literature suggests that Scrum practices can mitigate this challenge by: *ICT-mediated synchronous* (GSD_P2) and/or *asynchronous communication* (GSD_P3); *frequent (or improved) communication* (GSD_P5); *iteration* (GSD_P6); *review* (GSD_P7), and/or; *planning* (GSD_P8). For example, distributed team members may use synchronous communication tools such as video-conferencing to increase their sense of presence in meetings with other team members. They may also use ICT-mediated asynchronous communication in support of meetings to increase task awareness. For example, for retrospectives, one approach is to implement a transition backlog into which distributed team members can record and save ad-hoc improvement ideas. A significant benefit of the Scrum model is that all practices offer the opportunity to review what is required and/or what is being done, which increases task awareness. Similarly, the iterative nature of sprints and work planning also provide opportunities for, and encourage, informal interactions as well as formal contacts that can improve task awareness within the distributed teams.

Inconsistent work practices can impinge on effective coordination (due to socio-cultural distance): Due to developers being located in different countries, there may be differences in national culture, language, motivation and work ethics that can impede effective project coordination. The literature suggests that daily scrum and sprint practices can mitigate this challenge through *frequent (or improved) communication* (GSD_P5) and/or *iteration* (GSD_P6), respectively. For example, regular participation of distributed team members in daily scrum meetings can help to maintain consistent work practices between sites. Similarly, iterations of sprints reinforce practice consistency and improve coordination between distributed sites.

Reduced cooperation arising from misunderstandings (due to socio-cultural distance): Similarly, team member cooperation might be reduced due to cultural and language differences creating misunderstandings. The literature suggests that Scrum practices can mitigate this challenge by: *ICT-mediated asynchronous communication* (GSD_P3); *visits* (GSD_P4); *frequent (or improved) communication* (GSD_P5); *iteration* (GSD_P6); *review* (GSD_P7), and/or; *planning* (GSD_P8). For example, sprint planning meetings may be recorded by a web-conferencing tool and played back by the offshore team to reinforce its understanding. Distributed team members may also travel to other sites to participate in a few sprints as a collocated team to develop their understanding and increase cooperation and team coordination. Also, participation of distributed team members in Scrum meetings and reviews can provide frequent opportunities for communication to reduce misunderstandings between team members and improve cooperation and coordination. Similarly, the iteration of sprints and related meetings enables misunderstandings to be identified and resolved early, thereby reinforcing ongoing cooperation and improving project coordination.

3.3 Control Challenges

Finally, according to the literature, temporal, geographical and socio-cultural distance may impact GSD control processes by creating five main challenges: management of project artefacts may be subject to delays; difficulty in conveying vision and strategy; perceived threat from training low cost rivals; different perceptions of authority that undermine morale, and; managers must adapt to local regulations (from Table 1). The mechanisms that may mitigate each challenge, summarised by category in Table 4, are discussed following.

Management of project artefacts may be subject to delays (due to temporal distance): When a project involves members from different sites, enforcing process and artefact standards can be particularly important in maintaining consistency and interoperability. There was no evidence in the literature, however, of the use of Scrum practices to mitigate this specific GSD challenge.

Difficult to convey vision and strategy (due to geographical distance): Due to stakeholders being located in different countries, it can be difficult for onshore-based managers to convey the project vision and strategy to offshore sites. The literature suggests that Scrum practices can variously mitigate this challenge by: *visit* (GSD_P4); *frequent (or improved) communication* (GSD_P5); *review* (GSD_P7), and/or; *planning* (GSD_P8). For example, daily scrums enable team members to communicate frequently, enabling the project's vision and strategy to be reinforced within the project. In sprint reviews, offshore team members are able to ask product- and project-related questions of the onshore team and, in some cases, the product owner or customer, thereby increasing their understanding of the project's mission. Also, distributed team members may visit the onshore site and/or participate in a Sprint "zero" (project kickoff Sprint), which can develop an understanding of the product vision and strategy and project goals.

Perceived threat from training low cost 'rivals' (due to geographical distance): Employees in higher cost economies can feel that their jobs are under threat from resources sourced from lower cost economies. However, there was no evidence in the literature of the use of Scrum practices to mitigate this specific GSD challenge.

Different perceptions of authority can undermine morale (due to socio-cultural distance): Perception of authority in a team environment can vary between cultures. For example, in some cultures (e.g. Irish), developers may require their superiors to earn their respect and in others (e.g. US culture) give more unquestionable respect to figures of authority [40]. The literature suggests that Scrum practices can mitigate this challenge by *frequent (or improved) communication* (GSD_P5). For example, participation of onshore and offshore teams in daily scrums, sprint planning and review meetings can help members develop a sense of individual worth and value as a team member. These meetings can also clearly establish the boundaries of responsibility for the sprint within and outside the Scrum team.

Managers must adapt to local regulations (due to socio-cultural distance): When working in a global setting, onshore-based managers must be aware of the limitations that local regulations can bring to the project. There was no evidence in the literature, however, of the use of Scrum practices to mitigate this specific GSD challenge.

4 Discussion

Industry experience suggests that use of Scrum practices can provide communication, coordination and control benefits in GSD. As a result, there is increasing interest in using Scrum practices for GSD. However, empirical research on Scrum practices in GSD is scarce. Therefore, a knowledge gap exists on the use of Scrum in GSD in research and practice. Based on findings and experiences reported in the literature, we have proposed a research framework as a basis for developing further understanding. The framework integrates current results and views on how Scrum practices mitigate the challenges of distributed and global software development. The mechanisms were found to fit into eight categories. The framework contributes a research outcome in itself, a reference guide for current practice, and a basis for future empirical validation and investigation for research.

Examining the table overall, the literature appears to suggest that Scrum practices have no distinctive advantage over other development methods in mitigating temporal distance-based challenges (since adjusting working hours for meetings and using ICT-mediated asynchronous communications tools are available to any method). However, while communication tools (which are common mechanisms) are also prescribed to mitigate geographical challenges, these other challenges appear to also be mitigated by mechanisms such as frequent communication, iteration, planning and review that are distinctively inherent in Scrum practices.

4.1 Limitations

The proposal has some limitations. First, the framework is a theoretical contribution that remains to be empirically validated. Since few of the source articles from the literature are empirical studies (indeed, most are industry experience reports), the framework represents mostly theoretical propositions that need to be empirically validated. This work is proceeding in related research.

Second, it may be possible, in some instances, that the researchers misinterpreted an author's intent. For example, the researchers may have incorrectly interpreted an effect of a Scrum practice on a GSD challenge that is inferred in a paper rather than explicitly stated. The use of two researchers reviewing the papers reduced the likelihood of this occurring. Furthermore, subsequent use of the framework in empirical testing will validate its legitimacy from the perspectives of both the researchers who developed it and the authors of the papers on which it was based.

Third, the challenges and mechanisms contained in the framework are not exhaustive so may not be complete. For example, relevant studies may also exist in the systems engineering and/or information systems literatures, which were not included in this study. Also, this is an evolving field in software engineering; other studies may have emerged since the literature review was completed. The framework, however, is easily extendible to take on new challenges and mitigation mechanisms as new research emerges.

Finally, the framework implicitly assumes a generic GSD context so it may obscure project-specific variations in mechanisms. GSD takes many forms, depending on contextual factors, which may heavily influence how and which Scrum practices are used [42]. For example, contextual variables such as team size, collaboration

mode, number of distributed sites, degree of overlap in time zones, and degree of socio-cultural-economic compatibility will influence the configuration of a particular GSD project. Our expectation is that the framework will evolve to encompass a base set of challenges and mitigation mechanisms that will be relevant in most GSD projects using Scrum practices.

4.2 Implications

This review and the framework it has produced raise implications for research and practice. First, for research, questions arise from some initial observations of the framework:

1. Three challenges (in the Control section of Table 4) are not mitigated by any mechanisms in the literature. Is this because of gaps in the literature? Are these challenges invalid? or Is Scrum unable to respond to these challenges at all?
2. Two other control challenges are mitigated by only a few mechanisms. Does this mean that these challenges are significant threats to the use of Scrum in GSD?
3. Six of the nine rows that contain mitigating mechanism categories, include one or more mitigation mechanism from categories GSD_P5 to GSD_P8. This implies that for these GSD challenges, the nature of the Scrum practice itself is sufficient to mitigate the challenge. For example, the literature suggests that the challenge *“Inconsistent work practices can impinge on effective coordination”* can be mitigated by the fact of holding daily (frequent) scrums and/or iterative sprints. Does this mean that Scrum practices are particularly effective in mitigating these challenges?
4. Maintaining backlogs appears to have a limited role in reducing GSD challenges. Alternatively, does this represent a gap in the practice and research literature?

These observations point to areas for future research.

Second, at a higher level of thinking, the framework reinforces the contingency theory view of development methodologies that there is no one “ideal” method for every context [43]. Every development context is different, requiring an adapted method and, to be practically useful, every method needs to be adaptable to different development contexts. Scrum was originally conceived for Agile development in collocated teams. However, with some support and enablement of various tools and mechanisms, Scrum may also be directly or indirectly effective in developing software in globally distributed team environments.

Third (and more concretely), the framework raises other questions for ongoing software engineering process research. For example, given that it may be possible to use Scrum in GSD, is this approach more effective than other development methods? What comparative set of GSD challenges does Scrum mitigate in contrast to other methods? What (if any) differences exist in the toolsets used (or needed)? What can each approach learn from the other? Does it matter which approach is used?

Finally, for practice, the framework provides practitioners with an initial understanding of how Scrum practices may be used effectively in mitigating some of the challenges in developing software collaboratively across borders. It can serve as a reference framework of current knowledge and experience to guide practices in current and future projects.

4.3 Future Work

Related and future research will conduct multiple case studies in real life industry settings to validate and extend the framework. Due to the scale of the framework, these studies may focus separately on challenge-based sections of the framework. Based on these studies and other emergent findings in the literature, we will modify and extend the framework as a reference map for research and practice.

For example, work is underway on qualitative analysis of four case studies of GSD projects using Scrum practices from three internationally known corporations involving industrial, telecommunications and software engineering applications. This study is focusing on validating the prescriptions for the four *coordination* challenges in Table 4. Data was analysed and coded using NVivo. First the data was examined for evidence of the four coordination challenges (which was found in each case) and then for evidence of how Scrum practices were used to mitigate each challenge (mitigation mechanisms were found for each practice/challenge combination; that is, for each cell in this section of the table, thus potentially extending the framework). Currently, the case-based findings are being compared to the literature-based prescriptions.

5 Conclusion

Software engineering cannot escape the trend towards globalisation that faces many business endeavours today. To effectively and efficiently develop and maintain high quality products collaboratively across borders, traditional software development and project management processes need to be re-examined and re-thought. Agile methods such as Scrum need to be sufficiently ‘agile’ to adapt to new usage domains for them to continue to be relevant and to survive. GSD needs adaptable processes to overcome the significant challenges that can arise in this environment. Practice is likely to continue to inform research as software developers try out different ways of succeeding in GSD projects. As presented in this paper, the literature indicates that there is a substantial need for research to “catch up” and support the needs of practice. In addition to offering an integrated view of the current literature to support practitioners, the literature-based analysis and resulting framework presented in this paper offers some forward directions for this research.

References

1. Paasivaara, M., Durasiewicz, S., Lassenius, C.: Distributed agile development: Using Scrum in a large project. *Software Process Improvement and Practice* 13(6), 527–544 (2008)
2. Sutherland, J., Viktorov, A., Blount, J., Puntikov, N.: Distributed Scrum: Agile project management with outsourced development teams. In: *Proceedings of HICSS-40*, p. 274 (2007)
3. Sutherland, J., Schoonheim, G., Rijk, M.: Fully distributed Scrum: Replacing local productivity and quality with offshore teams. In: *Proceedings of HICSS-42*, pp. 1–8 (2009)
4. Cho, J.: Distributed Scrum for large-scale and mission-critical projects. In: *Proceedings of AMCIS 2007* (2007)

5. Williams, W., Stout, M.: Colossal, scattered, and chaotic (planning with a large distributed team). In: *Proceedings of Agile 2008*, pp. 356–361 (2008)
6. Drummond, B., Unson, J.F.: Yahoo! Distributed Agile: Notes from the world over. In: *Proceedings of Agile 2008*, pp. 315–321 (2008)
7. Cristal, M., Wildt, D., Prikładnicki, R.: Usage of Scrum practices within a global company. In: *Proceedings of ICGSE 2008*, pp. 22–226 (2008)
8. Holmstrom, H., Fitzgerald, B., Agerfalk, P.J., Conchuir, E.O.: Agile practices reduce distance in global software development. *Information Systems Management*, 7–26 (Summer 2006)
9. Vax, M., Michaud, S.: Distributed Agile: Growing a practice together. In: *Proceedings of Agile 2008*, pp. 310–314 (2008)
10. Smits, H.: Implementing Scrum in a distributed software development organization. In: *Proceedings of the Conference on Agile 2007*, pp. 371–375 (2007)
11. Jensen, B., Zilmer, A.: Cross-continent development using Scrum and XP. In: *Proceedings XP 2003*, pp. 146–153 (2003)
12. Kussmaul, C., Jack, R., Sponsler, B.: Outsourcing and offshoring with agility: A case study. In: *Proceedings of XP/Agile Universe*, pp. 147–154 (2004)
13. Sureshchandra, K., Shrinivasavadhani, J.: Adopting Agile in distributed development. In: *Proceedings of ICGSE 2008*, pp. 217–221 (2008)
14. Danait, A.: Agile offshore techniques: A case study. In: *Proceedings of Agile Development*, pp. 214–217 (2005)
15. Summers, M.: Insights into an Agile adventure with offshore partners. In: *Proceedings of Agile 2008*, pp. 333–338 (2008)
16. Therrien, E.: Overcoming the challenges of building a distributed agile organization. In: *Proceedings of Agile 2008*, pp. 368–372 (2008)
17. Berczuk, S.: Back to basics: The role of Agile principles in success with a distributed Scrum team. In: *Proceedings of Agile 2007*, pp. 382–388 (2007)
18. Karsten, P., Cannizzo, F.: The creation of a distributed Agile team. In: *Proceedings of XP 2007*, pp. 235–239 (2007)
19. Cottmeyer, M.: The good and bad of Agile offshore development. In: *Proceedings Agile 2008*, pp. 362–367 (2008)
20. Paasivaara, M., Lassenius, C.: Could global software development benefit from Agile method? In: *Proceedings of ICGSE 2008*, pp. 109–113 (2006)
21. Paasivaara, M., Durasiewicz, S., Lassenius, C.: Distributed Agile development: Using Scrum in a large project. *Proceedings of ICGSE 2009*, 195–204 (2009)
22. Bondi, A.B., Ros, J.P.: Experience with training a remotely located performance test team in a quasi-Agile global environment. In: *Proceedings of ICGSE 2009*, pp. 254–261 (2009)
23. Hansen, M.T., Baggesen, H.: From CMMI and isolation to Scrum, Agile, Lean and collaboration. In: *Proceedings of Agile 2009*, pp. 283–288 (2009)
24. Hossain, E., Babar, M.A., Verner, J.: How can agile practices minimize global software development co-ordination risks? In: O'Connor, R.V., Baddoo, N., Cuadrado Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) *EuroSPI 2009. Communications in Computer and Information Science*, vol. 42, pp. 81–92. Springer, Heidelberg (2009)
25. Lee, S., Yong, H.: Distributed agile: project management in a global environment. *Empirical Software Engineering* 15(2), 204–217 (2010)
26. Sutherland, J., Schoonheim, G., Kumar, N., Pandey, V., Vishal, S.: Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India. In: *Proceedings of the Agile Conference 2009*, pp. 277–282 (2009)

27. Therrien, I., Lebel, E.: From Anarchy to Sustainable Development: Scrum in Less Than Ideal Conditions. In: Proceedings of the Agile Conference 2009, pp. 289–294 (2009)
28. Jimenez, M., Piattini, M., Vizcaino, A.: Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, Article ID 710971, 1–14 (2009)
29. Herbsleb, J., Moitra, D.: Global software development. *IEEE Software* 18(2), 16–20 (2001)
30. Herbsleb, J., Grinter, R.: Architectures, coordination, and distance: Conway’s law and beyond. *IEEE Software* 16(5), 63–70 (1999)
31. Hossain, E., Babar, A.M., Paik, H., Verner, J.: Risk identification and mitigation processes for using Scrum in global software development: A conceptual framework. In: Proceeding of the Asia Pacific Software Engineering Conference, APSEC 2009, pp. 457–464 (2009)
32. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods: review and analysis. Technical Report # 408, VTT Publications, Espoo (2002)
33. Ågerfalk, P.J., Fitzgerald, B.: Flexible and distributed software processes: old petunias in new bowls? *Communication of the ACM* 49(10), 27–34 (2006)
34. Hossain, E., Babar, A.M., Paik, H.: Using Scrum in global software development: A systematic literature review. In: Proceedings of ICGSE 2009, pp. 175–184 (2009)
35. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report, EBSE-2007-01 (2007)
36. Herbsleb, J.D., Mockus, A., Finholt, T.A., Grinter, R.E.: Distance, dependencies, and delay in a global collaboration. In: Proceeding of CSCW 2000, pp. 319–327 (2000)
37. Moe, N.B., Šmite, D.: Understanding a lack of trust in global software teams: A multiple-case study. *Software Process Improvement and Practice* 13(3), 217–231 (2008)
38. Carmel, E.: *Global software teams: Collaborating across borders and time zones*. Prentice-Hall, NJ (2009)
39. Damian, D., Zowghi, D.: Requirements engineering challenges in multi-site software development organizations. *Requirements Engineering Journal* 8(1), 149–160 (2003)
40. Ågerfalk, P.J., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B., O’Conchuir, E.: A framework for considering opportunities and threats in distributed software development. In: International Workshop on Distributed Software Development 2005, pp. 47–61 (2005)
41. Schwaber, K., Beedle, M.: *Agile software development with Scrum*. Prentice Hall, Upper Saddle River (2001)
42. Hossain, E., Ali Babar, M., Verner, J.: Towards a Framework for Using Agile Approaches in Global Software Development. In: Bomarius, F., Oivo, M., Jaring, P., Abrahamsson, P. (eds.) PROFES 2009. Lecture Notes in Business Information Processing, vol. 32, pp. 126–140. Springer, Heidelberg (2009)
43. Avison, D., Fitzgerald, G.: *Information Systems Development: Methodologies, Techniques and Tools*, 4th edn. McGraw-Hill, Maidenhead (2006)
44. Krishna, S., Sahay, S., Walsham, G.: Managing cross-cultural issues in global software outsourcing. *Communication of the ACM* 47(4), 44–47 (2004)