2012 45th Hawaii International Conference on System Sciences

Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?

Paul L. Bannerman
NICTA and UNSW, Australia
paul.bannerman@nicta.com.au

Emam Hossain
UNSW and NICTA, Australia
emam.hossain@nicta.com.au

Ross Jeffery
NICTA and UNSW, Australia
ross.jeffery@nicta.com.au

Abstract

Global software development is a major trend in software engineering. Practitioners are increasingly trying Agile methods in distributed projects to tap into the benefits experienced by co-located teams. This paper considers the issue by examining whether Scrum practices, used in four global software development projects to leverage the benefits of Agile methods over traditional software engineering methods, provided any distinctive advantage in mitigating coordination challenges. Four temporal, geographical and socio-cultural distance-based coordination challenges and seven scrum practices are identified from the literature. The cases are analyzed for evidence of use of the Scrum practices to mitigate each challenge and whether the mitigation mechanisms employed relate to any distinctive characteristics of the Scrum method. While some mechanisms used were common to other/ traditional methods, it was found that Scrum offers a distinctive advantage in mitigating geographical and socio-cultural but not temporal distance-based GSD coordination challenges. Implications are discussed.

1. Introduction

Software development methods continue to evolve. The most recent major trend, in Agile methods, appears to be having a positive impact on software project outcomes. For example, the latest Standish Group industry survey attributes an improvement in project performance to an increase in 'agile projects' (growing at 22% per annum) and a decrease in 'Waterfall' development projects (growing at just 1% per annum) [18].

However, the context of software engineering is also changing, taking on increased complexity in pursuit of cost efficiencies, improved product quality and speed to market. Distributed and global software development (GSD) is a notable example of this. GSD is software development that is dispersed across two or more locations that are separated by national or continental borders. GSD can offer benefits to the onshore organization of access to qualified and skilled resources at a lower cost, proximity to markets, access to local knowledge, and flexibility in responding to diverse local opportunities [8]. However, GSD also brings particular challenges that are not present (or not as significant) in traditional co-located software development projects. In particular, GSD typically involves stakeholders located in different time zones and geographic locations, from different national and organizational cultures, using different and, at times, unreliable technologies to collaborate. Such temporal, geographical and socio-cultural distances can result in significant communication, coordination and control challenges that need to be overcome for the benefits of GSD to be realized [9].

Agile methods such as Scrum are increasingly being applied in GSD to leverage the benefits of both methods. Agile methods promise benefits of handling requirements changes throughout the development lifecycle; extensive collaboration between customers and developers; and early and frequent delivery of products [1]. However, Agile methods typically assume a project context that allows close interaction between team members. Indeed, one of the principles behind the *Agile Manifesto* is that face-to-face conversation is the most efficient and effective method of sharing information in a development team [4]. This presents a significant problem in GSD. Development teams and/or team members can be spread around the globe. Overcoming temporal, geographical and socio-cultural distance-based communication, coordination and control challenges in such context sets significant hurdles to overcome to realise the benefits of Agile methods in GSD.

To investigate this issue, and examine the ability of Agile methods to be effectively deployed in GSD projects (in comparison to traditional software engineering methods), this paper empirically examines four case studies in which Scrum practices were adopted and used. In each case, the project manager sought to leverage the benefits of the Scrum method over the traditional methods previously used to gain improved visibility of project activities and improved

978-0-7695-4525-7/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2012.512

stakeholder collaboration, trust, team awareness and shared understanding. Of the GSD communication, coordination and control challenges commonly referred to in the literature, the study focused on examining the use of Scrum practices to mitigate GSD *coordination* challenges. This approach contains the scope of the study and focuses on a significant issue (coordination) in managing dispersed developers that appears to receive more attention in the literature than the other two categories. Four temporal, geographical and socio-cultural distance-based coordination challenges were identified from the literature along with seven scrum practices typically used in GSD. These two dimensions (GSD coordination challenges and Scrum practices) formed the axes of a tabular research framework which was then populated, via qualitative case study analysis of the four cases, with mitigation mechanisms used in the cases. These mechanisms were then examined for their relative specificity to the Scrum method to determine whether the Scrum practices made a distinctive contribution to mitigating the GSD coordination challenges studied. The testimony of key case study informants on the perceived benefits of using Scrum practices over the traditional methods previously used is also considered.

Empirical investigation of the use of Scrum practices in GSD is under-represented in the literature [10]. Most current published studies are industrial experience reports. Consequently, this paper makes two main contributions to knowledge. First, an empirical study of the use of Scrum practices in GSD, and; second, a critical investigation of the distinctive contribution (or otherwise) of Scrum practices in managing important GSD challenges.

The paper is structured as follows. The next section briefly reviews prior related research, before the research method is described. Section 4 then describes the case studies and key findings on the use of Scrum practices; Section 5 analyzes these findings, and; Section 6 highlights the results. The findings are then discussed and implications and conclusion drawn from the study.

2. Related research

This section reviews the literature on GSD challenges and the use of Scrum practices in GSD.

2.1. GSD challenges

There is a growing body of literature that focuses on challenges in GSD. Communication, coordination and control challenges are recognized to arise due to temporal, geographic, and socio-cultural distances encountered in GSD [3].

Temporal distance, a measure of the dislocation in time between two people wishing to interact [2], may create communication issues such as reduced hours of collaboration, difficulties in holding synchronous meetings, and response delays [2]. As a result, GSD coordination processes may be significantly affected [3]. *Geographic distance*, a measure of the effort required to visit another person's home site [2], makes communication difficult because of the reduced ability to hold face-to-face meetings [1]. Lack of face-to-face meetings reduces informal contact which can lead to a lack of critical task awareness, "teamness" and reduced trust [1] [7] [15]. A fundamental problem in GSD is that many of the coordination mechanisms that work in a collocated setting are absent or disrupted [14]. *Socio-cultural distance*, a measure of a person's understanding of another's values and normative practices [2], 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] [3] [5] [13] [16].

Other recognized issues that may arise in GSD relate to collaboration, group awareness, project and process management and support, risk management, and knowledge management [12].

Several GSD issues frameworks and models exist in the literature (for example, [2] [3] [6]). A recent systematic literature review identified twelve communication, coordination and control challenges resulting from temporal, geographical and socio-cultural distances [11]. The four coordination challenges, which are the focus of this paper, are:

Increased coordination costs (due to temporal distance): The effect of time zone differences can be so great that project coordination complexity and costs increase.

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.

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.

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.

2.2. Scrum practices in GSD

Scrum is an iterative and incremental Agile project management approach comprising a range of common practices. Seven Scrum practices are identified by [16] that might typically be used in GSD, namely: Backlog, Sprint Planning, Sprint, Daily Scrum, Scrum of Scrums, Sprint Review (or Demo), and Retrospective. These practices largely depend on close interactions between developers, business stakeholders and customers. Indeed, it is claimed that a major reason for the success of Scrum is the collocation of development team members [1]. This interaction is difficult to achieve in GSD. However, regardless of apparent differences between the underlying principles of Scrum and distributed development approaches, there is a growing interest in investigating the use of Scrum practices for GSD to realize the benefits of this Agile method [10]. For example, an empirical study found that using Scrum practices in GSD improved communication, trust, motivation and product quality [16]. Furthermore, industrial experience suggests that using Scrum practices promotes communication and collaboration, ensures frequent delivery of product, and provides an opportunity to reduce some GSD challenges [20].

However, empirical studies on the use of Scrum practices in GSD are under-represented in the research literature; most are industrial experience reports [10]. So far, empirical studies have focused on establishing the feasibility of using Scrum practices in GSD (e.g., [16] [17] [19] [20]). The present study aims to extend this knowledge base by examining the distinctiveness of the contribution of the seven Scrum practices (identified by [16], above) in mitigating the four GSD coordination challenges (identified by [11], also above) in multiple case studies in which the project manager sought to overcome limitations previously encountered in using traditional development methods in GSD projects. In particular, these related to poor visibility of the dispersed project activities and problems in establishing and maintaining stakeholder collaboration, trust, team awareness and shared understanding.

3. Research method

Qualitative case study research was chosen as the method, following Yin's guidelines [21], to provide access to a rich and deep dataset on how Scrum is used in practice to mitigate GSD coordination challenges. A case study protocol was prepared to guide the study process, which was trialed in a pilot case study.

For the main study, companies were selected opportunistically (via industry contacts), that were engaged in GSD and had chosen to use Scrum practices to overcome limitations experienced previously in using traditional methods. Four projects were chosen involving industrial, software engineering and telecommunications applications from three internationally known corporations (Table 1). At the request of the companies, pseudonyms are used to protect the privacy of the study participants.

Table 1: Case Studies

Project (Company No.)	Product	Onshore Country
PaperSys (1)	Paper mill quality control	Finland
EnergySys (1)	Energy production control	
ToolSoft (2)	Collaboration software	Australia
TestSoft (3)	Telecom test platform	Finland

The primary data collection method was interviews, supplemented and triangulated by qualitative data from documentation, direct observation, informal discussion and artifacts (such as screenshots of tools used). Fifteen semi-structured interviews were conducted, each lasting 1-2 hours. Eleven interviews were face-to-face and four were conducted over Skype. Informants spanned multiple roles (development manager, product owner, project manager, Scrum master, architect, developer). Interviews were recorded, transcribed and verified by key informants. All data was loaded into and analyzed using NVivo (a qualitative data management and analysis software tool that enables related data from multiple sources to be interlinked).

Within-case analysis examined each case for evidence of how each Scrum practice was used and whether its use mitigated any of the four GSD coordination challenges. Cross-case analysis compared practice usage and mitigation mechanisms across the four cases to identify similarities and/or differences. Summary descriptions of mitigation mechanisms were prepared for each practice, within each case, for each GSD coordination challenge. These were then analyzed for evidence of distinctiveness to the Scrum method.

4. Case studies

The contextual characteristics of the cases are summarized in Table 2. Each case was found to experience the temporal, geographical and socio-cultural distance-based coordination challenges identified in the literature. Each case was also found to adapt its project structure and processes, including its use of Scrum practices, to suit the circumstances of the project. These adaptations served to mitigate the coordination (and other) GSD challenges faced by the projects. The use of Scrum practices is summarized, following, for each project case.

Table 2: Summary of case contextual characteristics

Contextual Factors	PaperSys	EnergySys	ToolSoft	TestSoft
Project-specific:				
Contract nature	Offshore contractor	Offshore contractor	Intra-organizational	Intra-organizational
Product domain	Automation industry	Automation industry	Collaboration software	Telecommunications
Requirements changes	Moderate	Moderate	Moderate	Low
Team-Specific:				
Team size	10 (4 onshore; 6 offshore)	11 (4 onshore; 7 offshore)	15 (11 onshore; 4 offshore)	15 (1 onshore; 14 offshore)
Distributed sites	4 (3 offshore)	2 (1 offshore)	2 (1 offshore)	4 (3 offshore)
Experience	Onshore: high; Offshore: mostly high	Onshore: high; Offshore: mostly low	Onshore: high; Offshore: low	Onshore: high; Offshore: Varied
Distance:				
Temporal	Moderate (up to 5 hrs)	Low (1 hour)	High (up to 19 hours)	High (up to 9.5 hours)
Geographical	Moderate (no direct flights)	Moderate (no direct flights)	High (no direct flights)	High (no direct flights)
Socio-cultural	High	High	Low	High

4.1. Case 1: PaperSys

The project operated as a single Scrum, distributed across two onshore- and offshore-based organizations. At the start of the project, the offshore team visited the onshore site for training and familiarization.

Since the domain and product knowledge was based onshore, the product *Backlog* was developed and updated onshore, but the sprint backlog was developed and maintained by the offshore team. Backlogs were stored centrally and accessed via globally accessible software tools.

Sprint Planning comprised three meetings: a joint 'goal introduction' meeting held online; an offshore team sprint planning meeting, and; a joint 'plan introduction' online meeting to validate the offshore team's sprint plan. *Sprints* were four weeks in duration.

The offshore team held traditional *Daily Scrum* meetings, online (the onshore team did not participate because its main focus was on project specifications). Since the teams operated as a single Scrum, a *Scrum of Scrums* meeting was not necessary. However, onshore and offshore 'Scrum masters' held a weekly status meeting as a proxy Scrum of Scrums.

Since the project was producing code for a product line that had to be centrally tested and integrated, the *Sprint Review* was adapted, with offshore code being released to an onshore tester for acceptance testing. *Retrospective* review meetings were held during early Sprints but discontinued because the Scrum model was working effectively and any issues or changes could be handled through the other meetings.

Unless otherwise stated, online meetings were held via Live Meeting for joint onshore/offshore meetings and Skype for offshore team meetings. Meeting times were synchronized between the four sites. Dispersed team members could communicate with each other whenever needed via Office Communications Server (OCS), telephone or email.

Overall, stakeholders were satisfied with the Scrum method compared to the previous plan-driven method. Because they had met their counterparts in the other country, team members found the communication tools to be sufficient means for interaction and collaboration, and less expensive and time-consuming than travel. The adapted Scrum model worked well for the parties, enabling each to focus on its own areas of expertise, with quick access for issue discussion and resolution if needed but, otherwise, with minimal disruption in achieving delivery goals. The Scrum meeting regime was found to improve coordination, team awareness and socio-cultural understanding between individuals in each country. Also, the short release cycle and frequent interaction helped to overcome the offshore team's lack of domain knowledge. Reviewing activities and achievements on weekly (Scrum master status meetings) and monthly (end of Sprint code acceptance testing) cycles provided high project visibility and the opportunity to resynchronize if necessary. It also engendered confidence and trust in the partnership.

4.2. Case 2: EnergySys

This project was from a different business division of the same company as PaperSys. EnergySys operated as a single project and team but held separate onshore and offshore Scrums.

The onshore-based product owner and architects maintained the product *Backlog* while the offshore team developed and maintained their sprint backlog

(the onshore team did not maintain a sprint backlog). Initially Team Foundation Server (TFS) was used for both but, mid-project, they had to change to a new company standard, Lotus Notes.

Similarly to PaperSys, an online Sprint pre-planning meeting was held with the project manager/ product owner, followed by a local offshore *Sprint Planning* meeting, but no post-planning confirmation meeting was held. However, the pre-planning meeting was recorded for later replay by offshore team members, if needed, and onshore stakeholders could be contacted at any time via OCS or other tools, as necessary. *Sprints* were 1.5 weeks in duration.

Only the offshore team held a *Daily Scrum*, which was co-located, but a formal joint *Sprint Review* was held, online via OCS and Live Meeting, at the end of each Sprint. The *Scrum of Scrums* and *Retrospective* practices were not used. However, ideas flowed freely within the team, exchanged via email and other communication mechanisms. Also, with the short sprint cycle, the other meetings reinforced coordination and resolution of issues and misunderstandings.

Compared to PaperSys, EnergySys struggled with low domain knowledge and experience in the offshore team, communication issues and problems transferring to Lotus Notes and its project repository. However, the short Sprint cycle adopted helped to identify issues so that they could be addressed early before significant downstream damage was done to the project.

4.3. Case 3: ToolSoft

The project operated as a single development team and worked on a single commercial product marketed by the company. However, to minimize dependencies and maximize delivery frequency, it was divided into four onshore sub-teams in Sydney and one offshore sub-team in USA, segmented on a product architecture or feature basis. Each sub-team had its own sub-goals, backlog and nominated team coordinator. Product and sub-team *Backlogs* were maintained in an issue tracking tool (another of the company's products).

Sprints were two-weeks long, to increase delivery, but *Sprint Planning* meetings were held weekly, one at the start of the Sprint and one in the middle, to review and revise the product road map, prioritize the product Backlog, select and confirm what would be done in the coming week in each sub-team and estimate the work effort. Only the sub-team coordinators participated in these meetings, with the project manager. Video conferencing was used so that onshore and offshore participants could see each other's face. These weekly meetings also served as *Scrum of Scrums* meetings, in which any cross-team issues and dependencies were discussed and resolved.

Due to time zone differences, separate onshore and offshore *Daily Scrum* meetings were held, and two onshore meetings were held to enable each member of the four sub-teams to speak. An offshore representative participated in the onshore Daily Scrums via Skype. Furthermore, two onshore and offshore team members were selected each week to talk via telephone or online chat about what each other was doing.

Sprint Reviews were merged with the weekly meetings as both planning and review meetings. Code produced by each team was reviewed by another sub-team before being sent to a separate QA team for testing and integration into the next product release. QA team members could participate in daily or weekly meetings to provide feedback, as necessary.

Onshore and offshore *Retrospectives* were held separately every four weeks because two weeks was found to be too frequent for the number of problems encountered. Meeting results were posted on the project wiki for information and discussion.

Scrum was considered to be highly effective for the purposes of the project and given the distribution of resources across two countries. As the project manager explained, "with Waterfall, you don't know if anything has been done for months; with Scrum, you know in a matter of weeks. The best thing about Scrum is project visibility – you have to deliver every two to four weeks for inclusion in a release so it is very visible if you don't. Also, you don't have to start all processes at once; you can progressively add them at each cycle, as a piloting/learning exercise". Furthermore, "having a distributed development team, especially on the other side of the world, makes the project more complicated, but our 'divide and conquer' approach of sub-teams, and rationalizing participation in meetings, makes it possible to still have good visibility over the project at the same time as maintaining a focus on delivery."

4.4. Case 4: TestSoft

This project involved two offshore teams: the main (but less-experienced) development team located at one site in Brazil, and a smaller European team located in Germany and India. These teams ran separate Scrums but used the same Sprint cycle. The 'onshore' domain knowledge-based site was in Finland.

The product *Backlog* was maintained centrally by the product owner/project manager, in conjunction with the Scrum masters of the two development teams, and the sprint backlogs and burndown charts were maintained separately by each Scrum team. In each case, the tool used was ScrumWorks.

Sprints were two weeks long. *Sprint Planning* involved a pre-planning meeting between the product owner/project manager and the two Scrum masters.

This was followed by separate local Sprint planning meetings (the Brazilian team meeting was co-located and the European team meeting was distributed, held via teleconference).

Separate *Daily Scrum* meetings were held every second day, via teleconference, by the European team (because there were only three people in that team), and daily in Brazil. Onshore personnel were not involved in either meeting.

At the end of each Sprint, each team held its own separate *Sprint Review*/*Retrospective* meeting, without onshore involvement. The results of these meetings were posted on the project wiki for the benefit of other project stakeholders.

Two additional teleconference meetings were held: a twice weekly de facto *Scrum of Scrums* management status meeting involving the product owner/project manager, Scrum masters and, sometimes, customers, and; a biweekly architectural/technical meeting involving both Scrum teams (this was the only regular joint meeting of both development teams).

Team members occasionally travelled to another site for joint workshops or to work with another team on a Sprint to support knowledge dissemination and familiarization across the teams.

Informants believed that Scrum was effective in the project. It enabled them to overcome many problems encountered with the previous plan-driven method. Scrum enabled frequent delivery of working software and increased project visibility. Furthermore, the project manager was able to incrementally adjust the plan according to Sprint outcomes. Scrum also helped reduce overwork, improved team collaboration and created trust between team members. However, issues remained to be resolved, particularly relating to limited Scrum and product domain knowledge in Brazil and unreliable communications with Brazil and India.

5. Case study analysis

Analysis proceeded as follows. For each project case (within-case analysis), a statement was prepared, for each practice used, on how the practice mitigated each of the four GSD coordination challenges. This resulted in a table of 28 (7 x 4) statements per case.

The statements were then categorized. Each statement was examined and found to match one or more of eight categories of mitigation mechanism, described in Table 3. Note that categories M1 to M4 involve various mechanisms which, combined with a Scrum practice, have the effect of mitigating various distance-based GSD coordination challenges. For example, M2 and M3 are communication mechanisms that are enabled by information and communication technology (ICT) tools such as videoconferencing or a wiki. Similarly, M1 adjusts working hours to enable a meeting to be held and M4 reduces distance barriers through visits to another site. By contrast, M5 to M8 represent characteristics of various Scrum practices themselves (frequent communication, iteration, review, planning) that mitigate GSD challenges, particularly around the Sprint cycle.

In categorizing the statements, it was found that primary and secondary mechanisms applied to most practices. For example, to mitigate the temporal distance challenge, Scrum meeting practices typically synchronized work hours (M1) as the primary mechanism, to enable an online meeting to be held (M2), as the secondary mechanism. The statements were coded with the secondary mechanism(s) in parentheses (i.e., as M1 (M2)).

The mitigation mechanisms for each practice were then examined across the four cases (cross-case analysis) and a summary statement was prepared and categorized representing the intersection of the four cases. This yielded a summary table of mechanisms used by the four cases for each practice to mitigate each of the four distance-based GSD challenges. Table 4 represents the results summarized by mitigation mechanism category (the individual statements are withheld due to space limitations)[1]

Working down each row of Table 4, the GSD coordination challenges were mitigated as follows:

Temporal Challenge: Increased coordination costs. Consistently, the Scrum meeting practices mitigated this temporal distance-based challenge by arranging a suitable common time between team members (and any other relevant project stakeholders) (M1) to hold an online meeting via an ICT-mediated synchronous communication tool (M2). In some cases for Retrospectives, individual Scrum teams met in isolation of the whole project stakeholder group (but still online if the team was distributed) and posted their results on the project wiki for the advice of other stakeholders (M3). The Backlog practice (not a meeting practice) similarly employed globally accessible ICT-mediated asynchronous communication tools (M3) to maintain and update task details on product and sprint backlogs and burndown charts (this was consistent for all four challenges so will not be repeated for the other challenges). These mechanisms enabled the projects to mitigate the effects of time zone differences and thereby contain coordination costs (ignoring communications costs that are common to GSD, such as channel and tool costs). As a practice, Sprints did not feature in mitigating this challenge.

[1] The detailed intermediate and final analysis tables are available from the first author by request.

Table 3: GSD coordination challenge mitigation mechanism categories

Category	ID	Description
Synchronized work hours	M1	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	M2	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	M3	Practices that enable asynchronous communication between team members; for example, email, Instant Messaging, or Wiki
Visit	M4	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	M5	Enable frequent formal and informal communication among team members through tools and/or face-to-face meetings
Iteration	M6	Activities that involve cyclical repetition enable multiple incremental opportunities to monitor progress and resolve issues
Review	M7	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	M8	Activities that establish the scope of work, resourcing, scheduling, and the processes to be employed

Table 4: Summary of case mitigation mechanisms by category

Scrum Practices / Coordination Challenges	Backlog	Sprint Planning	Sprint	Daily Scrum	Scrum of Scrums	Sprint Review	Retro-spective
Increased coordination cost (Temporal)	M3	M1 (M2)		M1 (M2)	M1 (M2)	M1 (M2)	M1, M3 (M2)
Reduced informal contact can lead to lack of critical task awareness (Geographical)	M3	M2 (M1)	M5, M6, M7, M8 (M5, M7)	M2 (M1)	M2 (M1)	M3, M7 (M1, M2, M3)	M2, M3 (M1)
Inconsistent work practices can impinge on effective coordination (Socio-cultural)	M3	M5 (M1, M2)	M6	M5 (M1, M2)	M5 (M1, M2)	M5, M7, (M1, M2, M3)	M5 (M1, M2, M3)
Reduced cooperation arising from misunderstandings (Socio-cultural)	M3	M3, M4, M5 (M1, M2, M7)	M4, M5, M6, M7, M8 (M5, M6, M7)	M5 (M1, M2)	M5 (M1, M2)	M5, M7 (M1, M2, M3)	M3

Geographical Challenge: Reduced informal contact can lead to lack of critical task awareness. Scrum meeting practices were mostly consistent in mitigating this geographical distance-based challenge by holding online meetings, enabled mostly by ICT-mediated synchronous communication tools (M2). Some Retrospectives also used ICT-mediated asynchronous communication tools (M3) to post results on the project wiki. In the case of Sprint Reviews, these mechanisms were also variously used but it was more the review function of this practice that mitigated the challenge than the meeting *per se* (M7). Finally, the fundamental characteristics of Sprints in fostering frequent communication (M5), iteration (M6), review (M7) and planning (M8) significantly contributed to mitigating this challenge. Overall, these mechanisms mitigated the reduced informal contact imposed by the geographical dispersion of the project teams by ensuring project-wide understanding and awareness of critical tasks.

Socio-cultural Challenge: Inconsistent work practices can impinge on effective coordination. Socio-cultural differences can bring entrenched divergent behavior patterns to projects that can be difficult to harmonize. As evidenced in the cases studied, one of the great benefits of the Scrum method is that it provides a consistent set of practices, mostly around meetings, that can avoid or counter negative

impacts on project coordination arising from socio-cultural differences. The Scrum meeting practices were mostly consistent in mitigating this socio-cultural distance-based challenge via the frequency of communication enabled by these meetings (M5) and repetitive iteration of Sprint-related activities (M6), variously enabled by online meetings and offline communication (M1, M2 and M3). These mechanisms provided a consistency of practice that substantially avoided coordination issues resulting from divergent socio-cultural behaviors.

Socio-cultural Challenge: Reduced cooperation arising from misunderstandings. Finally, culture and language differences, in particular, can easily lead to misunderstandings and/or offense, alienating people and resulting in issues of cooperation in projects. Again, the Scrum meeting practices were mostly consistent in mitigating this socio-cultural distance-based challenge by the frequent communication associated with these meetings (M5), variously enabled by online meetings and offline communication (M1, M2 and M3) and, on limited occasions, inter-site visits (M4). In the case of Sprint Reviews, these mediating mechanisms were also variously used but it was more the review function itself (M7) and the frequency of its practice (M5) that mitigated the challenge, rather than the meeting *per se*. In Retrospectives, posting the results on the project wiki and eliciting a response by other team members if misunderstandings were detected, both via ICT-mediated asynchronous communication tools (M3), mitigated the challenge. These mechanisms enabled misunderstandings to be identified and resolved early, avoiding flow-on impacts on project cooperation and coordination.

6. Results

The key findings from the preceding analysis are as follows.

First, the *temporal distance* challenge, whereby developers operated in different time zones, was typically mitigated by synchronizing work hours to arrange a time when all parties could meet online. However, where the time zone difference was too great to make meeting time synchronization practical, the Scrum model was adapted to minimize the need for online meetings involving all team members.

Second, the *geographical distance* challenge, which prevented interaction through physical presence, was typically mitigated by ICT-mediated synchronous communication tool-based meetings and, in some cases, with support from ICT-mediated asynchronous communication tools such as wikis. However, the fundamental characteristics of sprints (M5 to M8) also significantly contributed to mitigating this challenge by maintaining task awareness, despite reduced informal contact, through the constant cycle of interactions.

Third, the *socio-cultural distance* challenges, reflecting different norms, values and national perspectives, were typically mitigated by the frequency of communication enabled by Scrum-related meeting practices – sometimes supported by recording meetings for later replay, posting meeting results on a project wiki and/or inter-site visits – and the inherent characteristics of those practices of ongoing iterative review and planning (that is, M5 to M8). These meetings enforced common work practices on the multi-cultural participants and provided frequent checkpoints to ensure shared understanding and ongoing collaborative cooperation throughout the project.

In sum, in the case studies examined, temporal distance was mitigated by synchronizing work hours to hold meetings or by structuring the project to limit the need for synchronized meetings; geographical distance was mitigated by communicating online via ICT-mediated tools and iterative sprint-based interactions, and; socio-cultural distance was similarly mitigated by the characteristics inherent in Scrum practices of frequent tight iterations and interactions, enabled by ICT-mediated communications.

Considering the distinctiveness of these mitigation mechanisms, the first (temporal distance) mitigation mechanism is clearly not specific to the Scrum method. Any management method used for GSD could synchronize working hours to arrange suitable meeting times. Consequently, based on the cases studied, *Scrum offers no distinctive advantage in mitigating the effects of temporal distance on coordination in GSD projects*.

Second, for geographical distance, two types of mitigation mechanism were found. First, ICT-mediated synchronous communication (online meetings), is also not specific to Scrum. It could be used with any GSD management method. However, the value of this mechanism in facilitating coordination was enhanced when used in conjunction with the Scrum method because of the high proportion of meeting practices inherent to, and characterizing, Scrum. The cases suggest that the frequency of communication, iterative nature, and inherent focus on planning and review of Scrum meeting practices significantly contributed to mitigating the geographical distance-based GSD challenge (by providing many rich opportunities for interaction, thereby increasing task awareness). On this basis, *the Scrum model does make a distinctive contribution to mitigating the effects of geographical distance on coordination in GSD projects.*

Finally, the socio-cultural distance mitigation mechanisms stemmed directly and distinctively from the inherent iterative, incremental, time-boxed

characteristics of Scrum. The frequently iterated Sprint cycle of plan, build and review provides an ideal mechanism for maintaining cohesion between socio-culturally diverse teams and identifying and resolving misunderstandings early, thereby minimizing downstream impacts. As an integrated package, these features are typically not characteristic of other development management methods. Therefore, based on the case studies, *the Scrum model has a distinctive advantage in mitigating the effects of socio-cultural distance on coordination in GSD projects.*

7. Discussion

This paper has aimed to make two main contributions to knowledge. First, an empirical study of the use of Scrum practices in GSD (to help overcome a gap in empirical research), and; second, a critical study of the distinctive contribution of Scrum practices in overcoming important GSD coordination challenges. While some mechanisms used are commonly available to other/traditional methods, it was found that Scrum offers a distinctive advantage in mitigating geographical and socio-cultural but not temporal distance-based GSD coordination challenges.

The study has validity threats and limitations. Considering validity threats, reliability was enhanced by multiple design techniques, including using a case study protocol; conducting a pilot case study; using at least two researchers to conduct interviews, code and analyze the data, and; using NVivo as a database for management and analysis of case data. Internal validity was enhanced by triangulating data from multiple types of data and multiple sources within type, and supported by a high degree for consistency in findings across the four case projects. With respect to external validity, four cases are indicative but not necessarily fully representative. Therefore, no claims are made that the findings generalise, statistically, to other settings, only that they generalize to emergent theory on how Scrum practices mitigate GSD coordination challenges.

With respect to limitations, first, only four of the twelve GSD challenges identified from the literature by [11] were examined. Others remain to be investigated. Also, other GSD challenges may exist in practice that have not yet been identified in the literature. Second, the study focused on the Scrum practices, rather than the tools and mechanisms that mediate the challenge mitigation. Third, the study did not directly compare Scrum practices with traditional development methods in mitigating these coordination challenges. Rather, it chose projects in which project managers were seeking to gain benefits from the use of Agile/Scrum over the previous traditional methods that they had used. Fourth, project contextual factors were found to influence how Scrum was tailored to fit the project context. The influence of these factors was not directly controlled or systematically examined in this study. Finally, as implied above, further validation is needed, from other case studies and other research methods, to further substantiate the conclusions.

The study has implications for research and practice. First, for both research and practice, the study strongly suggests that viewing Agile methods such as Scrum as only a method for managing co-located projects imposes an unnecessary and artificial constraint on the method. Through the enablement of ICT-mediated communication tools and other mechanisms, Scrum brings many of its inherent benefits to GSD *and* overcomes many of the inherent challenges of GSD. Second, for research, the findings summarized in Table 4 contribute to an emergent body of knowledge on the use of Scrum practices in GSD. Third, the findings provide focus for future research by highlighting the distinctive advantages (and otherwise) of the Scrum method in GSD.

For practice, the findings summarized in Table 4 have great utility in informing business, IT and project managers considering using (or already trialing) Scrum in GSD on how Scrum practices can be effectively deployed and common problems that might be encountered.

For future research, each of the above limitations represents areas for future investigation. In particular, the study points to an opportunity to further research the contingency drivers that shape the adaptation of the Scrum method in GSD contexts and the decisions available in practice for managers to make in tailoring Scrum-based GSD projects.

8. Conclusion

As software development becomes more dispersed globally, to take advantage of lower-cost resources and skills wherever they are located, new opportunities and challenges arise for applying new and traditional software engineering and development management methods. This study has focused on four projects in which managers sought to leverage the benefits of Agile/Scrum methods in the GSD context and found distinctive benefits over the traditional plan-driven methods they had previously used. This is a welcomed though unexpected finding for a software development method based on assumptions of close interactions between developers and other project stakeholders. However, advances in information and communication technologies, and other mechanisms, have enabled the temporal and geographical distances challenging GSD to be substantially mitigated, largely independent of the method used. The real contribution of this study,

however, is to show that the distinctive characteristics of the Scrum method can benefit distributed as well as co-located software development projects by mitigating geographical and socio-cultural distance-based coordination GSD challenges. For the projects managers in the cases studied, this was something the previous traditional methods struggled to do.

9. Acknowledgments

NICTA is funded by the Australian Government via the Department of Broadband, Communications & Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

10. References

[1] P. Abrahamsson O., Salo, J. Ronkainen & J. Warsta, Agile Software Development Methods: Review and Analysis, Technical Report # 408, VTT Publications, Espoo, 2002.

[2] P. Ågerfalk, B. Fitzgerald, H. Holmström, B. Lings, B. Lundell, & E. O'Conchuir, "A Framework for Considering Opportunities and Threats in Distributed Software Development", in Proceedings of the International Workshop on Distributed Software Development, 2005, pp. 47-61.

[3] P. Ågerfalk, & B. Fitzgerald, "Flexible and Distributed Software Processes: Old Petunias in New Bowls?" Communication of the ACM, 49(10), 2006, pp. 27-34.

[4] Agile Manifesto. "Principles behind the Agile Manifesto", http://agilemanifesto.org/principles.html, and "Manifesto for Agile Software Development", http://www.agilemanifesto.org), accessed June, 2011.

[5] E. Carmel, Global Software Teams: Collaborating Across Borders and Time Zones, Prentice-Hall, NJ, 2009.

[6] D. Damian & D. Zowghi, "Requirements Engineering Challenges in Multi-Site Software Development Organizations", Requirements Engineering Journal, 8(1), 2003, pp. 149-160

[7] M. Hansen & H. Baggesen, "From CMMI and Isolation to Scrum, Agile, Lean and Collaboration", in Proceedings of the Agile Development Conference, 2009, pp. 283-288.

[8] J. Herbsleb & D. Moitra, "Global Software Development", IEEE Software, 18(2), 2001 pp.16-20.

[9] H. Holmström, B. Fitzgerald, P. Ågerfalk, & E. Ó Conchúir, "Agile Practices Reduce Distance in Global Software Development", Information Systems Management, 23(3), 2006, pp. 7-18.

[10] E. Hossain, A. Babar & H. Paik, "Using Scrum in Global Software Development: A Systematic Literature Review", in Proceedings of the International Conference on Global Software Engineering, 2009, pp. 175-184.

[11] E. Hossain, P.L. Bannerman & R Jeffery, "Scrum Practices in Global Software Development: A Research Framework", in Proceedings of the International Conference on Product Focused Development and Process Improvement (PROFES 2011), Springer-Verlag, LNCS 6759, 2011, pp. 88-102.

[12] M. Jimenez, M. Piattini & A. Vizcaino, "Challenges and Improvements in Distributed Software development: A Systematic Review", *Advances in Software Engineering*, Article ID 710971, 2009, pp. 1-14, accessed 16 May 2011http://www.hindawi.com/journals/ase/2009/710971.htm.

[13] S. Krishna, S. Sahay, & G. Walsham, "Managing Cross-Cultural Issues in Global Software Outsourcing", Communication of the ACM, 47(4), 2004, pp. 44-47.

[14] C. Kussmaul, R. Jack & B. Sponsler, "Outsourcing and Offshoring with Agility: A Case Study", in Proceedings of XP/Agile Universe, 2004, pp. 147-154.

[15] N. Moe & D. Šmite, "Understanding a Lack of Trust in Global Software Teams: A Multiple-Case Study", Software Process Improvement and Practice 13(3), 2008, pp. 217-231.

[16] M. Paasivaara, S. Durasiewicz & C. Lassenius, "Using Scrum in a Globally Distributed Project: A Case Study", Software Process Improvement and Practice, 13(6), 2008, pp. 527-544.

[17] M. Paasivaara, S. Durasiewicz & C. Lassenius, "Using Scrum in Distributed Agile Development: A Multiple Case Study", in Proceedings of the International Conference on Global Software Engineering, 2009, pp. 195-204.

[18] Standish. CHAOS Manifesto: The Laws of CHAOS and the CHAOS 100 Best PM Practices, The Standish Group International Inc., Boston, 2011.

[19] J. Sutherland, A. Viktorov, J. Blount & N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams", in Proceedings of the Hawaii International Conference on System Sciences (HICSS'40), 2007, pp. 1-10.

[20] J. Sutherland, G. Schoonheim & M. Rijk, "Fully Distributed Scrum: Replacing Local Productivity and Quality with Offshore Teams", in Proceedings of the Hawaii International Conference on Systems Sciences (HICSS '42), 2009, pp. 1-8.

[21] R. Yin, Case Study Research: Design and Methods; Fourth Edition, Sage, Thousand Oaks, 2009.