Towards an Understanding of Tailoring Scrum in Global Software Development: A Multi-case Study

Emam Hossain  
The University of New South Wales  
Kensington, NSW  
Australia  
+61 423 918 804

Paul L Bannerman  
The University of New South Wales  
Kensington, NSW, Australia  
+61 2 9376 2169

Ross Jeffery  
The University of NSW,  
Kensington, NSW, Australia  
+61 2 9376 2188

Emam.Hossain@nicta.com.au  
Paul.Bannerman@nicta.com.au  
Ross.Jeffery@nicta.com.au

ABSTRACT

There is growing interest in applying Scrum practices in Global Software Development to leverage the advantages of both. However, the effective use of Scrum practices largely depends on close interactions between project stakeholders. The distribution of project stakeholders in GSD provides significant challenges related to project collaboration processes that may limit the use of Scrum. However, project managers increasingly seek to use the Scrum model in their distributed projects. While there is an emerging body of industrial experience, there are limited empirical studies that discuss Scrum tailoring in GSD. The paper reports a multi-case study that investigates the impact of key project contextual factors on the use of Scrum practices in GSD. This study is relevant to researchers and practitioners who are seeking ways to use Scrum in GSD and improve project effectiveness.

Categories and Subject Descriptors

D.2.9 [Management]: Software Process models, programming teams.

General Terms

Management, Economics, Theory.

Keywords

Global Software Development, Scrum, Case study.

1. INTRODUCTION

There is growing interest in applying Agile practices in Global Software Development (GSD). Among the Agile methods available, an increasing number of project managers are considering using Scrum as an Agile project management method in their distributed projects [1]. However, the effective use of Scrum practices significantly depends on close interactions between project stakeholders. In GSD, geographical, temporal and socio-cultural distances create challenges that may restrict team collaboration [2]. Moreover, other GSD project contextual factors such as project personnel, number of distributed sites, and product domain may further exacerbate team collaboration [1]. As a result, some researchers note that Scrum is difficult to scale up to support distributed arrangements [3] and the fundamental question of whether Scrum practices can be used in a distributed setting has been a subject of debate [4].

However, despite apparent significant differences between the fundamental principles of Scrum and distributed development, a Systematic Literature Review (SLR) found success cases in using Scrum practices in GSD [1]. However, the SLR findings indicated that in most GSD projects, Scrum use was adapted to fit the circumstances of the project [5, 6]. The study also found that empirical research on the use of Scrum in GSD is scarce and that description of how project contextual factors impact the tailoring of Scrum in distributed projects in the literature is limited [1]. Nevertheless, the Scrum model offers the potential of improved visibility and control over distributed developments through its practices of frequent development and review iterations and collaborative communication between developers. Therefore, given the relative novelty of the phenomenon and the paucity of the empirical evidence, there is a need for more empirical studies to better understand how Scrum might be tailored to operate effectively in GSD projects. Specifically, this paper considers the impact of key project contextual factors in tailoring the Scrum model for GSD.

A multi-case study design is used for the study. Four GSD projects in which Scrum practices were used are examined for the influence of project-, team- and distance-specific contextual factors on the adaptation of the Scrum model deployed. These factors are based on an analytical framework developed in [19]. The research is important and relevant to both researchers and practitioners in understanding how an emergent software development management method might be extended beyond the bounds of its original operating setting to bring similar benefits to GSD.

The paper is structured as follows. Section 2 provides background to the research and Section 3 describes the research methodology used. Section 4 describes the four cases and provides initial within-case analysis and Section 5 details the cross-case analysis. Finally, Section 6 discusses the limitations and implications of the study and reaches conclusions about its contributions.
2. RESEARCH BACKGROUND
This section briefly reviews the literature on the Scrum method in GSD and the research context of the study.

2.1 Scrum Method in GSD
GSD (including outsourcing and partnerships across national borders) is a major trend in software development. It has gained significant adoption as a means of reducing time to market, increasing productivity, improving quality and gaining cost effectiveness and efficiency [7]. In addition to these expected benefits, however, several challenges of GSD have also been identified [8]. In particular, GSD is normally characterized by stakeholders from different national and organizational cultures, located in separate geographic locations and time zones, using different information and communication technologies to collaborate. Such conditions usually result in major problems in relation to team communication, coordination and collaboration [9]. Furthermore, key project-specific, team-specific and distance-specific contextual factors may also impact on team effectiveness [1]. These include, for example, the nature of the contract; application domain; requirements volatility; project personnel; distribution of sites; team experience; and temporal, geographical and socio-cultural distance between partners and sites [1].

Scrum is an iterative and incremental project management approach that has a number of practices that largely depend on close interactions between developers, business stakeholders and customers. Indeed, [10] claims that a major reason for the success of Scrum is the collocation of development team members. This interaction is difficult to achieve in GSD. However, Scrum is a flexible Agile method that can be tailored according to the project context [11]. Despite apparent and significant differences between the fundamental principles of Scrum and distributed development approaches, there is a growing interest in assessing the viability of using Scrum practices for GSD [1]. One recent empirical study found that using Scrum practices in GSD improved communication, trust, motivation and product quality [5]. In addition, industry experience suggests that using Scrum practices promotes communication and collaboration, ensures frequent delivery of product and provides an opportunity to reduce some GSD challenges [6]. To identify Scrum practices that can be used in GSD, we examined a number of studies (e.g. [5], [12], [13]) and chose the seven Scrum practices identified in [5], as indicative of Scrum use in GSD for examination in our study. These practices are: sprint, sprint planning, daily Scrum, Scrum of Scrums, sprint review, retrospective, and backlog.

2.2 Research Context
Communication and collaboration processes are at the heart of using Scrum practices. However, GSD project contextual factors may create additional challenges that restrict the use of Scrum practices [1]. Some solutions have already been proposed in the literature for using Scrum in GSD. Sutherland et al. [6] propose three Scrum models that can be used in GSD: isolated Scrums; distributed Scrum of Scrums and; totally integrated Scrums. For example, in their proposed distributed Scrum of Scrums model, the Scrum team is geographically separated but integrated by a “Scrum of Scrums” practice. Similarly, in a totally integrated Scrum model, Scrum teams are cross-functional with team members distributed across geographical locations. In addition, a related stream of research has focused on the tailoring of Agile methods to the actual needs of the GSD context. For example, one approach suggests developing hybrid Scrum and plan-based development methods, with criteria for deciding when (under what conditions) each should be used [14]. Moreover, a number of practitioners more generally suggest that the effective use of Agile largely depends on the tailoring process in GSD [2, 4, 15].

Despite some discussion of the tailoring of Scrum methods in GSD, there is little focused empirical research on tailoring in GSD in the literature [1]. To contribute to this research gap, the objective of our research is to explore and understand how Scrum practices are used (based on the seven practices identified in [5]) and how project contextual factors impact the tailoring of Scrum practices in GSD projects (based on the key project contextual factors in [19]). We also investigate how collaboration tools and supporting mechanisms are used in tailoring the Scrum practices.

3. RESEARCH METHODOLOGY
Investigating the research problem in this study requires rich and deep description of GSD projects using Scrum practices in actual industry settings. Case studies are the preferred strategy for this type of study; that is, for “when (a) ‘how’ or ‘why’ questions are being posed, (b) the investigator has little control over events, and (c) the focus is on a contemporary phenomenon within some real-life context” [16, p3]. To carry out the study we followed Yin’s guidelines [16]. This started with development of a research protocol as a record of the procedures to be followed. This helped maintain consistency and reliability in the research. The unit of analysis is a Scrum practice used in GSD. We used purposeful sampling, selecting projects that involve software development using Scrum in a global setting. To preserve privacy, identities are withheld and the four selected projects are given pseudonyms: PaperInfo, EnergyInfo, CollaborationSoft, and TestSoft.

The primary source of data was interviews, supplemented by informal conversations, observations, tool demonstrations, documentation and photographs. Fifteen semi-structured interviews were conducted; each lasting 1-2 hours. The interviews were carried out by two researchers, one interviewing and the other taking notes. Most interviews were conducted face-to-face in industry settings. For PaperInfo, the onshore-based project manager and architect were interviewed. The offshore-based team lead and architect were also interviewed via Skype. For EnergyInfo, the onshore-based project manager, team lead, process manager and offshore Scrum master (on an occasion when he was visiting the onshore site) were interviewed. For CollaborationSoft, four project team members were interviewed, including the onshore-based project manager. For TestSoft, the project manager was interviewed and two offshore team members via Skype. Documents made available to the research team included system specifications, project plans, testing scripts and the completed software. Documentary information was also used to corroborate (triangulate) and augment evidence found from the interviews and discussions that focused on the use of Scrum practices in distributed projects. The raw data was stored in and analyzed using NVivo™ (a qualitative data analysis tool), as the case study database.

Data analysis comprised qualitative content analysis [17] complemented by framework analysis [18] aimed at identifying, describing and making sense of how Scrum practices were tailored by considering the influence of key project contextual factors. NVivo™ was used for coding, grouping and analyzing the textual data. Evidence was categorized according to relevant
themes and apparent relationships. The coded data (with cross references to sources) was then charted to summarize the findings. Charting helped to understand the whole dataset. To improve the quality of the analysis, initial findings were reported to the key informant on the project (the Project Manager/Scrum Master), who provided feedback identifying omissions and rectifying misunderstandings.

Case study design was based upon the four criteria recommended by Yin [16] for ensuring high integrity case study research. First, construct validity, which involves establishing valid operational measures for the concepts being studied, was facilitated by using multiple sources of evidence, establishing a chain of evidence, and having key informants review draft case study reports. Second, internal validity was supported by use of three forms of triangulation: multiple data types (interviews, documents, etc.); multiple sources within type (multiple team members, multiple documents, etc.); and; observer triangulation. For the third, external validity, the multi-case design enhances the possibility of generalizing findings but the study aimed only to generalize to theory, not to other settings. Finally, reliability was supported by use of a case study protocol and a case study database: a detailed case study protocol was developed, tested and applied consistently across all cases, and; NVivo was used to ensure consistency in handling, coding and analyzing data from each case and enable chains of evidence to be established.

4. CASE DESCRIPTIONS

Using the framework developed in [19], the key project contextual factors characterizing the four projects are summarized in Table 1 and the Scrum practice utilization is summarized in Table 2. Each project is described, following.

4.1 PaperInfo

We start by describing the PaperInfo development project.

4.1.1 Project Description

PaperInfo is a project of a global supplier of process industry machinery and systems. The project was part of a large product development comprising different projects that produced both hardware and software products. The project contributed to the development of an information system product to maintain quality control in paper mill industries. Many large customers all over the world use the product. Requirements changed moderately during the project. The project was distributed in two countries: Finland and an offshore country (see the project structure in Figure 1). The four-person domain knowledge-based Finnish team included the project manager/scrum master, project architect, developer and test engineer at one onshore location. The main roles of the offshore team were to maintain the product backlog, generate and maintain specifications for the offshore development team’s sprints (the sprint backlog), and verify the quality of completed software before delivery to customers. Due to its domain expertise, the offshore team also did some development that required specific domain knowledge. The offshore team was highly experienced. For example, the project manager had over thirty years experience in software development. The six-person offshore team was distributed across three sites in the offshore country. Two people were located at each site. All six worked together as a single Scrum team. The offshore development team members’ experience varied from six to ten years.

The project involved temporal, geographical and socio-cultural distances. The sites spanned multiple time zones. The main and another offshore site were one hour ahead of the Finnish site and the third offshore site was four hours ahead of the other two offshore sites (i.e., five hours ahead of Finland). Hence, the project featured a low temporal distance (less than four hours) except for one offshore site (less than eight hours). There was no direct flight between the onshore and offshore main site, resulting in a flight time of five hours. Travel was further exacerbated as one of the offshore sites needed an additional four hours flight time. Therefore, based on the main offshore site, it can be concluded that the project experienced moderate geographical distance (less than eight hours). Based on Hofstede’s indices of socio-cultural distance [20] and participant opinions, the project experienced significant socio-cultural distance.

Figure 1. PaperInfo Project Structure

4.1.2 Use of Scrum Practices

Scrum was adapted to suit the project. Some practices were used in a ‘pure’ distributed form (sprint, daily Scrum and backlog), some were tailored (sprint planning, Scrum of Scrums and sprint review) and others were rarely used (retrospectives). Sprints were mostly standardized. Initially, sprint length varied from 2-4 weeks but after a few cycles they were fixed at 4 weeks. However, the length of some sprints had to be modified to accommodate different holiday practices between the two countries. Sprint planning was also adapted. At the beginning of the sprint, a sprint pre-planning meeting was held (called a “goal introduction meeting”), in which the onshore team presented the goals and prioritized backlog for the upcoming offshore sprint through Office Communications Server (OCS) and Live Meeting tools. This meeting lasted for 1-2 hours and was recorded. In the meeting, offshore team members were able to ask relevant questions to minimize misunderstanding. The offshore team then conducted its internal sprint planning meeting using Skype, which lasted for up to 4 hours. During this meeting, the offshore team could replay the recorded “goal introduction meeting” if needed or communicate with onshore team members through OCS to clarify any issues. The next day, the offshore team presented their detailed sprint plan to the onshore team in a “plan introduction meeting”, lasting thirty minutes, through OCS and Live Meeting. In this meeting, the onshore team verified the offshore team’s plan and provided feedback as necessary. Based on the offshore feedback, the offshore team finalized its sprint plan and developed the sprint backlog in Team Foundation Server (TFS) tool.
Table 1. Key project contextual factors

<table>
<thead>
<tr>
<th>Contextual Factors</th>
<th>PaperInfo</th>
<th>EnergyInfo</th>
<th>CollaborationSoft</th>
<th>TestSoft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Specific</td>
<td>Offshore Subcontractor</td>
<td>Offshore Subcontractor</td>
<td>Intra-organizational</td>
<td>Intra-organizational</td>
</tr>
<tr>
<td>Contract nature</td>
<td>Automation industry</td>
<td>Automation industry</td>
<td>Enterprise software solutions</td>
<td>Telecommunication</td>
</tr>
<tr>
<td>Product Domain</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Low</td>
</tr>
<tr>
<td>Requirements changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Specific</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project personnel (team size)</td>
<td>10 (onshore management team 4; offshore development team 6)</td>
<td>11 (onshore management team 4; offshore development team 7)</td>
<td>15 (onshore management and development team 11; offshore development team 4)</td>
<td>15 (management team 1; offshore development team 14)</td>
</tr>
<tr>
<td>Distributed sites</td>
<td>4 (onshore 1; offshore 3)</td>
<td>2 (onshore 1; offshore 1)</td>
<td>2 (onshore 1, offshore 1)</td>
<td>4 (onshore 1, offshore 3)</td>
</tr>
<tr>
<td>Experience</td>
<td>Highly experienced management team; development team experience varied from 5 to 15 years</td>
<td>Highly experienced management team, offshore development team mostly inexperienced</td>
<td>Highly experienced management team; low experienced development team (varied from 1 to 5 years)</td>
<td>Highly experienced management team, mostly experienced development team (varied from 1 to 15 years)</td>
</tr>
<tr>
<td>Distance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>Low (1 hour); except 1 offshore site (5 hours)</td>
<td>Low (1 hour)</td>
<td>High (19 hours in winter and 17 hours in summer)</td>
<td>High (no overlapping working hours between two of the sites)</td>
</tr>
<tr>
<td>Geographical</td>
<td>Moderate distance; Finland and a nearshore country, 5 hours flight time, one of the offshore site needs 9 hours</td>
<td>Moderate distance; Finland and a nearshore country, Moderate distance, 5 hours flight time</td>
<td>High geographical distance; Australia and US more than 15 hours flight time</td>
<td>High geographical distance; Finland, Germany, India and Brazil. Flight time from India to Brazil 30 hours</td>
</tr>
<tr>
<td>Socio-cultural</td>
<td>Significant, differences in language, culture and norms</td>
<td>Significant, differences in language, culture and norms</td>
<td>Low; similar language, culture and norms</td>
<td>Significant; differences in language, culture and norms</td>
</tr>
</tbody>
</table>

Daily Scrum was used in a standard manner by the distributed offshore team via Skype. The meeting focused on answering the three standard Scrum questions: What has been accomplished since the last meeting? What is going to be done before the next meeting? and What obstacles are in the way? Questions and discussion between team members then followed. The meeting lasted 5-15 minutes depending on the extent of discussion. The onshore team did not participate in the daily Scrum as their focus was on generating and maintaining project specifications. The offshore teams operated as a single Scrum so the Scrum of Scrum practice was not used. However, to stay on track, update the offshore team with any changes and resolve any cross-site issues and dependencies, the onshore and offshore Scrum masters held a weekly status meeting via Skype. This meeting served as a proxy for a Scrum of Scrum meeting.

The sprint review practice was also tailored. At the end of the sprint, rather than hold a traditional review meeting, the completed code from the sprint was released to the onshore test engineer for acceptance testing. Retrospectives were held in the initial 5-6 sprints using OCS. However, the practice was discontinued because the Scrum model was working effectively and any issues or changes could be adequately handled through the other meetings. The backlog practice was used in the project. The onshore site developed and updated the product backlog, maintained in TFS using ScrumWorks. At the beginning of the sprint, the backlog was prioritized in the sprint pre-planning meeting. Based on this meeting, the offshore team then developed the sprint backlog, also in TFS. During the sprint, the developers maintained the status and updated the remaining time of the allocated tasks on a daily basis. TFS was also integrated with the burn down chart, indicating project progress.

4.2 EnergyInfo

Next, we describe the EnergyInfo development project.

4.2.1 Project Description

The EnergyInfo project was part of a large product development at the same company as PaperInfo but involved a different product line. The project developed an information system to control a power, energy and oil refinery system. It was a new development that had moderate change requirements. The project manager hired a subcontractor company from a nearshore country as the main development team. As in PaperInfo, the project was distributed in two countries: Finland and the same offshore country (see the project structure in Figure 2). The Finland-based four-person onshore team held the project domain knowledge. It comprised a product owner/project manager, architect, technical lead and test engineer. The onshore management team’s main task was to generate and maintain specifications provided to the offshore development team. However, due to the offshore team’s lack of domain knowledge, the onshore team also did some development that required domain knowledge. The seven-person offshore team comprised the main project developers. Experience varied. The onshore team had over ten years of experience in software development. However, the offshore team was less experienced, although the development lead also had more than ten years of experience in software development.

As the project was distributed (similarly to the PaperInfo’s main onshore and offshore sites), it involved temporal, geographical and socio-cultural distances similar to those of PaperInfo.

4.2.2 Use of Scrum Practices

As in PaperInfo, Scrum was adapted to suit the circumstances of the project. Some practices were used in a distributed form (sprint, sprint review, backlog), some were used in a collocated form (daily Scrum), others were tailored (sprint planning), and some
Table 2. Summary of Scrum practice usage in case projects

<table>
<thead>
<tr>
<th>Practice Case</th>
<th>Sprint planning</th>
<th>Daily Scrum</th>
<th>Scrum of Scrums</th>
<th>Sprint review</th>
<th>Retrospective</th>
<th>Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>PaperInfo</td>
<td>Used (length initially varied; then fixed at 4 weeks)</td>
<td>Use modified (added pre- and post-planning meetings with onshore team)</td>
<td>Used (by offshore distributed teams via Skype; onshore team not directly involved)</td>
<td>Use modified (weekly status meeting between offshore project manager and offshore team lead)</td>
<td>Ultimately not used (used in initial sprints but later discontinued)</td>
<td>Used (maintained in ScrumWorks); product backlog maintained onshore; sprint backlog developed and burndown chart maintained by offshore Scrum team</td>
</tr>
<tr>
<td>EnergyInfo</td>
<td>Used (length initially varied; then fixed at 1.5 weeks)</td>
<td>Use modified (held an additional sprint pre-planning meeting)</td>
<td>Used in collocated form (used by offshore team without onshore team’s involvement)</td>
<td>Not used (other meetings and communication opportunities were considered to be sufficient)</td>
<td>Not used (tries but discontinued due to lack of feedback from both teams)</td>
<td>Used (maintained in Lotus Notes); product backlog maintained onshore; sprint backlog developed and burndown chart maintained by offshore Scrum team</td>
</tr>
<tr>
<td>Collabora-</td>
<td>Used (initially 4 weeks then fixed at 2 weeks)</td>
<td>Use modified (held weekly with project manager and sub-team coordinators)</td>
<td>Use modified (onshore and offshore separate; two daily onshore meetings; offshore representation via Skype)</td>
<td>Use modified (combined with weekly sprint planning and review meetings)</td>
<td>Use modified (sprint output reviewed by onshore-based QA team)</td>
<td>Used (maintained in Jira); product backlog maintained centrally; sprint backlog developed and burndown chart maintained by Scrum sub-teams</td>
</tr>
<tr>
<td>tionSoft</td>
<td>Used (fixed at 2 weeks)</td>
<td>Use modified (held an additional sprint pre-planning meeting)</td>
<td>Use modified (Europe- and Brazil-based teams held separate Scrums; Europe-based Scrum team held meeting every second day)</td>
<td>Use modified (weekly status meeting was held involving project manager and the two offshore Scrum masters)</td>
<td>Use modified (only Scrum masters, project manager and customer involved)</td>
<td>Used (Europe-based team’s results published in Wiki)</td>
</tr>
</tbody>
</table>

were not used at all (Scrum of Scrums, retrospective). After some variation, sprints were set at 1.5 weeks, although some sprint lengths needed to be varied to synchronize with different onshore holiday patterns. The shorter period enabled tight scrutiny and feedback on work completed by the offshore team.

Sprint planning was adapted. Due to the offshore team’s lack of domain knowledge, at the beginning of every sprint, the offshore team participated in a sprint pre-planning meeting with the product owner/project manager through OCS and Live Meeting. In this meeting (which took up to two hours and was recorded), the product backlog was prioritized for the offshore teams’ upcoming sprint and nominated item specifications explained. During the meeting, offshore team members were able to ask questions about the proposed tasks. Based on the prioritizations, the offshore team members then held their own detailed sprint planning meeting, which lasted up to four or more hours. To clarify issues and reduce misunderstandings, offshore team members could replay the recorded sprint pre-planning meeting. During and after the planning meeting, offshore team members could also communicate one-on-one with onshore stakeholders to clarify any issues, mostly via OCS. The sprint backlog was developed in this local planning meeting and maintained in the company standard collaboration tool, Lotus Notes (which replaced TFS).

The daily Scrum was used in a collocated rather than distributed form at the offshore site. The onshore team had no daily Scrum as their development team was small and the site’s focus was mainly on generating and maintaining project specifications. The daily Scrum process was similar to that of PaperInfo. The sprint review meeting was a key practice in this project in distributed form. At the end of a sprint, the offshore team presented what they had accomplished during the sprint to the onshore team through OCS and Live Meeting tools. As proxy customer, the product owner / project manager checked the completed sprint and provided feedback as necessary. The Scrum of Scrums practice was not used because, even though the two sites operated as separate ScrumS, most development was centered at the offshore site. Moreover, the short sprint length provided frequent opportunity to discuss and resolve any cross-site issues and dependencies. Similarly, while retrospective had been attempted, the practice was discontinued due to lack of feedback from both sites. The backlog practice was used in this project. The product backlog was created and maintained by the onshore team, in Lotus Notes, and was prioritized for the offshore team’s sprints by the onshore team in the sprint pre-planning meeting. In turn, the offshore team developed the sprint backlog in their local sprint planning meeting, which was also maintained in Lotus Notes. To support the burndown chart, a separate application on top of Lotus Notes was used to indicate detailed project progress.

4.3 CollaborationSoft

Here, we describe the CollaborationSoft project.

4.3.1 Project Description

CollaborationSoft is a project of an enterprise collaboration software vendor that is well-known for serving Agile software development. The CollaborationSoft project was developing an enterprise wiki for intranets and a knowledge management application. The project was distributed in two countries: Australia and the US. The eleven-person Sydney-based onshore site had the domain knowledge-based management team and main developers, including the project customer, QA team and other relevant stakeholders. The core product was developed onshore. The onshore team was divided in four sub-teams, each of which were allocated a separate feature-based part of the product. Experienced team members were selected as coordinators of each sub-team. A four-person offshore-based sub-team, in San Francisco, worked independently on various application plug-ins to the main product. However, the offshore team’s deliveries were
integrated with the main product regularly. Team members had a variety of skills and experience. For example, the project manager had over ten years experience in software development.

The project’s distribution in Australia and the US created geographical, temporal and socio-cultural distances. There is a seventeen or nineteen hour time difference (depending on the time of the year) between Sydney and San Francisco. At best, this resulted in half a day’s overlap in business hours between the two sites, four days a week. Also, Australia and the US are on opposite sides of the Pacific Ocean. There was no direct flight between the two sites and it took almost a day to travel from the onshore to offshore site. However, based on Hofstede’s [20] socio-cultural indices for Australia and US and informant opinions, the project involved low socio-cultural distance.

The sprint review meeting was also tailored due to the way work was allocated to development teams and the nature of the product produced by the project. Code from each sub-team was reviewed by another sub-team. Also, at the end of each sprint, completed software was passed to the onshore QA team for review. Feedback was provided to the Scrum teams as necessary.

**Retrospectives** were also tailored due to the temporal distance between sites and the segmentation of work. Onshore and offshore teams conducted retrospectives separately. In the retrospective, sub-teams analyzed team performance by focusing discussion on what the team had achieved, what impediments were encountered, what tasks could not be finished or started at all, and any other relevant topic (such as, what did not go well and what needed to be improved). The obstacles were noted. Each site’s retrospective results were posted in the project wiki, which was accessible by all stakeholders. Initially, retrospectives were held at the end of every sprint. Later, however, as operations began to run smoothly, the time interval was increased to the end of every second sprint.

The Backlog practice was integrated with the weekly planning cycle (as described above). Product and sub-team backlogs were maintained in Jira. The burndown chart was maintained in a related tool, GreenHopper, to indicate project progress in detail.

### 4.4 TestSoft

Finally, we describe the TestSoft GSD project.

#### 4.4.1 Project Description

TestSoft was a project in a large market leader in the manufacture of telecommunications products for a global market. The project was developing software for a product testing platform for the company’s own use and the use of other large customers around the world. The project was a new development and had moderate levels of requirements changes in its initial release cycles. However, the project manager was expecting more requirements changes in coming releases as the needs of other customers were considered. The project involved 15 people distributed in four countries. The Finland-based onshore product owner had more than ten years’ project management experience. The project involved two “offshore” development teams which they referred to as the Brazil-based and Europe-based teams. The Brazil-based eleven-person subcontractor team was located in a single site and was considered to be the project’s main development team. In this team, Scrum experience varied from highly experienced to inexperience but domain knowledge was generally low. The Europe-based team had three people, two of whom were located in Germany and the other in India. Finland had the main domain knowledge (vested in the product owner/project manager), although the Europe-based team was also strong in domain understanding. In this team, experience in software development varied from 7 to 20 years. In general, each offshore team was allocated independent architectural subsystems (although modules were cycled around between teams, in different sprints, to spread and share knowledge within the project).

The project distribution in four countries created temporal, geographical and socio-cultural distances. The project involved high temporal distance. For example, there was no overlap in business hours between sites in India and Brazil. The project also involved high geographical distance as there were no direct flights between distributed sites and travel could take up to 30 hours.

---

**Figure 3. CollaborationSoft Project Structure**

#### 4.3.2 Use of Scrum Practices

In this project, Scrum was also adapted. Some practices were used in a traditional form (sprint and backlog) while the rest were tailored in some manner (sprint planning, daily Scrum, Scrum of Scrum, sprint review and retrospective). Sprints were mostly standardized. Initially, the sprint length was 4 weeks but was then fixed at 2 weeks to increase project visibility and delivery frequency. Sprint planning was adapted. At the beginning of the sprint, representatives from the five sub-teams participated in a sprint planning meeting, via video conferencing, in which the backlog was reviewed and prioritized for the upcoming sprint. Scrum team representatives also negotiated what backlog items would be assigned to their Scrum teams. Based on the prioritized backlog, Scrum team members then planned their own sprint backlog for the upcoming sprint, which was maintained in the Jira tool. Mid-sprint, another meeting reviewed issues and progress, and discussed new ideas and proposals. This meeting also served as a Scrum of Scrum meeting, to resolve any cross-team issues and dependencies. Also, two people from each site participated in additional chat- or telephone-based meetings, twice a week, to discuss sprint progress and any other matters of relevance.

The daily Scrum was modified to suit the project design (into sub-teams) and the time zone distance between sites. Separate daily Scrums (stand-ups) were held in each site but representative offshore team members could participate in the onshore daily Scrums via Skype with video (and increasingly did so towards the end of release cycles). Two daily Scrum meetings were held onshore due to the large project personnel and to limit the meeting duration to ten minutes.
Visa requirements further increased geographical distance. Based on Hofstede’s [20] definitions of cultural dimensions for Brazil, India, Finland and Germany, and informant opinions, the project also involved significant socio-cultural distance.

5. CASE ANALYSIS

This section presents and discusses the results from the cross-case analysis of the data gathered from the four cases.

5.1 Impact of Project Contextual Factors on Using Scrum Practices

This sub-section discusses the impact of key project-, team- and distance-specific contextual factors on how the seven Scrum practices were used.

5.1.1 Sprint

The multi-case data indicates that the project-specific factors “requirement changes” and “product domain” impacted use of the sprint practice. For example, in CollaborationSoft, due to frequent requirements changes, the Project Manager changed the 4-week sprint length to 2 weeks. This shorter duration enabled the project to take on and deliver requirements changes in more frequent iterations throughout the project. Case data also indicates that the offshore teams’ lack of domain knowledge influenced tailoring sprint practices. For example, in EnergyInfo, the offshore team’s lack of domain knowledge created some misunderstandings. As a result, the sprint length was shortened to 1.5 weeks to enable the project manager to verify the offshore sprint more frequently, increasing project visibility. The data also shows that the distance-specific factor “socio-cultural distance” impacted use of the sprint practice. For example, in PaperInfo, sprint length had to be synchronized at the offshore development site due to different onshore holiday patterns.

5.1.2 Sprint Planning Meeting

The case studies show that the project-specific factor “product domain” had an impact on using sprint planning meeting. In all four cases, due to the offshore team’s lack of domain knowledge, the sprint planning meeting became a sprint planning process that involved additional meetings. For example, in PaperInfo, to complete the offshore team’s sprint planning meeting, onshore and offshore team members participated in two joint meetings. First, in a sprint pre-planning meeting (called a “goal introduction meeting”, which was recorded), the onshore team presented the goals and prioritized backlog for the upcoming offshore sprint. In this meeting, offshore team members were able to ask relevant questions to the onshore domain knowledge-based team members which minimized misunderstandings. Based on the onshore teams’ prioritization of the backlog, offshore team members then held their sprint planning meeting. During this meeting, the
offshore team could replay the recorded “goal introduction meeting” if needed. The next day, the offshore team presented its detailed sprint plan to the onshore team in a sprint post-planning meeting (called a “plan introduction meeting”). In this meeting, the onshore team verified the offshore team’s plan and provided feedback as necessary.

The data also indicates that the team-specific factors “project personnel”, and “distributed sites” impacted tailoring of the sprint planning practice. For example, in CollaborationSoft, due to the large project personnel, key representatives from each sub-team participated in a sprint pre-planning meeting. In this meeting, onshore management team members, together with the Scrum team representatives, prioritized the product backlog for the upcoming sprint. Based on the product backlog prioritization, each sub-team then held their own detailed sprint planning meetings. Similarly, due to the distribution of sites, the project manager divided work into separate modules and allocated these modules to the site-based sub-teams. Thus each sub-team conducted its own site-based sprint planning meeting.

The data also indicates that the distance-specific factors “temporal”, “geographical” and “socio-cultural” distance had an impact on use of sprint planning meetings. It shows that project managers used a wide range of collaboration tools and supporting mechanisms to conduct sprint planning meetings due to these distances. For example, due to high temporal distance in CollaborationSoft, only one or two US-based team members used the mechanism “adjust working hours” to join in the weekly sprint planning meeting via video conferencing. Similarly, in PaperInfo, due to geographical distance, offshore distributed team members participated in their sprint planning meeting through Skype. Moreover, in the case projects, team members used the mechanism “visit” so that they could participate in sprint planning face-to-face. For example, in TestSoft, offshore team members sometimes visited the onshore site to participate in sprint pre-planning meetings. Also, in PaperInfo, to avoid challenges due to socio-cultural distance, the sprint pre-planning meeting was recorded using the OCS tool. In that case, to clarify misunderstandings or confusion, the offshore team was able to replay the recorded presentation.

5.1.3 Daily Scrum

The case data indicates that the project-specific factor “product domain” may have impacted use of the daily Scrum practice. For example, in CollaborationSoft, towards the end of the project, an offshore team representative participated in onshore daily Scrum meetings to help resolve any misunderstandings due to lack of domain knowledge. The data also suggests that the team-specific factor “team size” had an impact on the daily Scrum practice. For example, in CollaborationSoft, the onshore team members initially participated in a single daily Scrum. However, due to the large team size, the project manager organized two separate daily Scrum meetings. Similarly, in TestSoft, due to the small size and spread of one Scrum team, daily Scrums were held every second day. Case findings also suggest that the team-specific factor “experience” impacted the daily Scrum. For example, in TestInfo, the Brazilian team members were less experienced in both Scrum and domain knowledge. As a result, their daily Scrum meetings were much longer than the usual 10-15 minutes because they went into too much detail, as a group, in the meeting (rather than to resolve issues via collaborative discussions after the meeting).

The study also found that project managers used collaboration tools and mechanisms to support the use of daily Scrum and mitigate challenges due to the distance-specific factors “temporal”, “geographical” and “socio-cultural” distance. For example, in PaperInfo, due to temporal distance, offshore team members used the mechanism “adjust working hours” to participate in the daily Scrum. Similarly, in TestSoft, due to the geographical distance, Europe-based Scrum team members used the collaboration tool “teleconference” to participate in daily Scrums. Case data also suggests that socio-cultural distance had an impact on the daily Scrum. For example, in CollaborationSoft, a socio-cultural issue, “doubtful of others capabilities”, arose between the onshore and offshore sites. As a result, an offshore team representative participated in onshore daily Scrums, particularly at the end of release cycles, to help reduce mistrust and misunderstandings.

5.1.4 Scrum of Scrums

The study indicates that the team-specific factor “team size” had an impact on the Scrum of Scrums practice. For example, in PaperInfo, the offshore teams operated as a single Scrum so, strictly speaking, the Scrum of Scrum practice was not used in that project. However, to stay on track, update the offshore team with any changes and resolve any cross-site issues and dependencies, the onshore project manager and offshore project lead held a weekly status meeting. Evidence suggests that the distance-specific factor “temporal” and “geographical” distance also impacted the Scrum of Scrums practice. In CollaborationSoft, for example, due to the high temporal distance, representatives from each sub-team used the mechanism “adjust working hours” to participate in a weekly status meeting. Similarly, in TestSoft, due to the geographical distance, Scrum masters and the project manager participated twice weekly in a status meeting via a “teleconferencing”. In addition, due to geographical distance, team representatives used the mechanism “visit” to travel between sites to participate in status meetings face-to-face. For example, in PaperInfo, the offshore project lead frequently visited the onshore site to participate in the project’s weekly status meetings.

5.1.5 Sprint Review

The case evidence also indicates that the team-specific factor “team size” had an impact on use of the sprint review practice. For example, in TestSoft, rather than the entire Scrum team (due to one team being too small and the other too large), only the Scrum master of each team participated with the project manager and customers in the sprint demo. The sprint review meeting was also tailored due to geographical and temporal distance. However, Project Managers employed a range of collaboration tools and mechanisms to support sprint review meetings in their distributed project environments. For example, in TestSoft, due to the temporal and geographical distances involved, each team held separate reviews locally and posted the results on the project wiki. By contrast, EnergyInfo used the mechanism “adjust working hours” to participate in sprint review meetings via OCS and Live Meeting tools (the only common meeting of all team members). In some projects, distributed team members also had an opportunity to participate in some sprint review meetings face-to-face, through the “visit” mechanism. For example, in EnergyInfo, sometimes offshore team members participated in review meetings face-to-face, while visiting the onshore location.
5.1.6 Sprint Retrospective

The case data indicates that the team-specific factor “team size” and “distributed sites” impacted use of the retrospective practice. For example, in CollaborationSoft, due to the large team size, retrospectives were held after every second sprint, locally at each site, with the results posted on the project wiki. Similarly, due to the distribution of teams in TestSoft, each team similarly held a separate retrospective and posted the results in the project wiki. The retrospective practice was also affected by the distance-specific factors “temporal” and “geographical” distance. For example, in TestSoft, due to the temporal and geographical distances involved, the Europe-based Scrum team members used the mechanisms “adjust working hours” and “teleconference” to hold their retrospective meetings.

5.1.7 Backlog

One case also indicates that the project-specific factor “product domain” had an impact on use of the backlog practice. Due to the specific interests of diverse customers, in TestInfo, the Scrum management team met biweekly with customers to identify new backlog items, eliminate noise, and clean and refine the items list for future sprints. The study also indicates that use of the backlog practice was supported by collaborative tools due to the distance-specific factor “geographical distance”. For example, in PaperInfo and TestInfo, product and sprint backlogs were maintained in a globally accessible tool ScrumWorks. Similarly, the burndown chart was also maintained in globally accessible tools. For example, in CollaborationSoft, the burndown chart was maintained in GreenHopper, which was integrated with the backlog tool Jira.

5.2 Benefits of Tailored Scrum Model

In this multi-case study, although the projects faced various challenges resulting from the distribution of project stakeholders, informants indicated that they were satisfied with their tailored Scrum model. Informants from PaperInfo, EnergyIInfo and TestSoft noted that the existing tailored Scrum enabled them to minimize some of the project challenges that they could not, or better than they could, under the previous plan-driven model. For example, lack of offshore work visibility was identified as one of the key challenges. However, the sprint model provided iterative cycles and many formal and informal contact opportunities that ensured the frequent delivery of working software, increasing project visibility. Moreover, all of the project managers noted that in their sprint model, they were able to incrementally adjust the plan according to sprint outcomes. Informants also stated that tasks were well-organized under Scrum, minimizing overwork, and that project teams were required to be largely self-organized, increasing motivation and a sense of teamness. As a result, the use of Scrum improved coordination in the distributed project and gave the project manager a greater sense of control and improved transparency of progress towards goals. The Scrum meeting practices also enabled team members to participate regularly, on a predictable cycle, improving team collaboration and creating an environment of high trust. However, informants also noted that the effective use of Scrum in GSD critically depends on the use of good collaboration tools (e.g. wiki, chat, conferencing) and other supporting mechanisms (e.g. “adjust working hours” and “visit”).

6. DISCUSSION

In this paper, we presented a multi-case study on tailoring Scrum practices in four globally distributed projects. We also considered the use of collaboration tools and supporting mechanisms in tailoring Scrum in GSD projects. The contributions of the paper to knowledge are threefold. First, supported by collaboration tools and enabling mechanisms, it illustrates how key project contextual factors can influence the adaptation of the Scrum model in a manner that can mitigate significant GSD challenges. Second, it contributes to a paucity of empirical literature on the use of Scrum in distributed environments. Third, it presents experiences and views of project managers who have used Scrum in GSD projects that may inform practitioners seeking to improve their own project performance.

6.1 Limitations and Further Research

The paper has some limitations. As a qualitative study, causality is implied by the relationships between constructs as established from analysis of the accounts of informants (some of the analyses are only partially explicated due to the limitations of space, but are coded and charted in our dataset). Consequently, the authors make no claims that the findings are generalizable to other GSD projects. Indeed, an implicit assumption of the paper is that there is no one way to adapt the Scrum model in a GSD project. Rather, project managers will respond to the challenges of GSD by crafting a Scrum model to fit the circumstance of the project (as represented here by certain key projects contextual factors). Instead, the paper contributes to a body of knowledge from which general principles may emerge on how Scrum can be used in non-collocated environments.

Furthermore, GSD projects take many forms. Four case studies are informative but not necessarily indicative or representative. More case studies are needed to provide cumulative insight into tailoring the Scrum method in GSD by considering a range of project settings. Also, other research methods such as surveys could be employed as the volume of industry experience grows.

Another limitation of the paper is that it focuses only on tailoring the Scrum method in GSD. Effective management of GSD projects may be achievable by adapting other project management and development methods, with the support of enabling tools and mechanisms. The paper makes no claim that Scrum is the ‘only’ or ‘best’ method for GSD. Rather, it is a response to growing interest by practitioners and researchers in deploying Scrum, as an Agile method, outside of its original domain of development by a collocated team.

6.2 Implications

The main implication of the study, for both theory and practice, is that viewing Scrum as a process for managing collocated software project teams may be imposing an artificial and unnecessary constraint on the method. Today, electronic conferencing and computer-mediated work tools, and mechanisms such as “adjust working hours” and “visit” are increasingly becoming standard operational support capabilities as trends towards globalization and inter-organisational collaboration become part of the normal landscape of business. Electronic conferencing may provide a different experience to face-to-face contact and meetings in terms of quality of knowledge exchange. However, the cases examined in this study suggest that the Scrum method, which presupposes collocation, may be tailored in dispersed environments, and with the help of collaboration tools and supporting mechanisms, be used to effectively manage software development in challenging global landscapes. If this is so in general, it challenges the traditional proximity assumptions of Agile methods and opens
opportunities to interpret and develop them more broadly in research and practice.

6.3 Conclusion
This paper reports an empirically-based qualitative research study of four projects that has aimed to contribute knowledge and understanding on the use of Scrum practices in GSD and on how Scrum can be tailored in GSD by considering key project contextual factors, including collaboration challenges resulting from temporal, geographical and socio-cultural distances. The findings support emergent views that the Scrum model offers the potential of improved visibility and control over distributed developments through its practices of frequent development and review iterations and collaborative communication between developers. Adapting the model to the characteristics of the project and finding suitable collaboration tools and enabling mechanisms would appear to be critical to realizing these benefits.

7. ACKNOWLEDGMENTS
NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

8. REFERENCES