Towards a Framework for Using Agile Approaches in Global Software Development

Emam Hossain[1,3], Muhammad Ali Babar[2], and June Verner[3]

[1] UNSW-NICTA, Australian Technology Park
Sydney, Australia
Emam.Hossain@nicta.com.au
[2] Lero,University of Limerick
Castletroy, Limerick, Ireland
malibaba@lero.ie
[3] School of Computer Science and Engineering,UNSW
Sydney, Australia
jverner@cse.unsw.edu.au

Abstract. As agile methods and Global Software Development (GSD) are become increasingly popular, GSD project managers have been exploring the viability of using agile approaches in their development environments. Despite the expected benefits of using an agile approach with a GSD project, the overall combining mechanisms of the two approaches are not clearly understood. To address this challenge, we propose a conceptual framework, based on the research literature. This framework is expected to aid a project manager in deciding what agile strategies are effective for a particular GSD project, taking into account project context. We use an industry-based case study to explore the components of our conceptual framework. Our case study is planned and conducted according to specific published case study guidelines. We identify the agile practices and agile supporting practices used by a GSD project manager in our case study and conclude with future research directions.

Keywords: Agile approaches, Global Software Development, Case study.

1 Introduction

Agile Software Development (ASD) and Global Software Development (GSD) are promoted as a means of reducing time to market, increasing productivity, improving quality and gaining cost effectiveness and efficiency [1]. ASD has gained significant popularity because of a promise to handle requirements volatility throughout the development life cycle, promotion of extensive collaboration between customers and developers, and support for early and frequent delivery of a product [2]. GSD is also considered to be a cost effective software development paradigm driven by a number of factors, such as time to market pressures, taking advantage of using distributed resource pools, use of multiple time zones, shared best practices, and closer proximity to customer [3].

F. Bomarius et al. (Eds.): PROFES 2009, LNBIP 32, pp. 126–140, 2009.

GSD project managers have recently begun using agile practices in their development environments [4, 5]. However, despite the expected benefits of using agile approaches with GSD, the overall combining mechanisms of the two approaches are yet to be fully understood [6]. To address this problem, we propose a conceptual framework that describes the use of various strategies to use agile approaches in GSD projects, based on the research literature. We also conducted an industry-based case study to test our framework and to help us better understand the use of agile strategies in a GSD project in a real life setting. We used a framework-based data collection approach, adopted from [7], to record project context factors as we believe that GSD project agility depends substantially on project context. In our case study, we identified the agile strategies used by the project manager to aid in a discussion of our framework components.

In the next section we discuss the background to our research. Section 3 describes our research problem in detail, while section 4 presents the proposed framework. Section 5 discusses our research methodology and case study, and we conclude with section 6, which discusses our future research directions.

2 Research Background

The fundamental concepts of an agile approach in software development are described in the agile manifesto. This manifesto states that the agile community values individuals and interactions over process and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [8]. The heart of an agile approach is in using agile practices that emanate from agile methods, such as XP and Scrum. On the other hand, GSD is a contemporary form of software development where project stakeholders are dispersed in distributed locations where socio-cultural distances may be involved. Because of geographical, temporal and in some cases, socio-cultural differences, GSD may suffer from a number of difficulties related to communication, coordination, control [12]. Thus, it is apparently difficult to apply many of the key agile concepts in distributed developments as agile development promotes a close collaboration and communication environment. However, despite the apparent difficulties in applying agile approaches within distributed settings, a number of GSD projects are currently using agile practices in their development environments [5, 9].

3 Our Research

As noted earlier, the combination of agile approaches with GSD is not fully understood although such a combination is expected to be beneficial [6]. Because of geographically dispersed teams, it can be very difficult to apply some agile practices, for example pair programming, daily stand up meetings, the planning game, and onsite customer participation in distributed projects. Also project contextual factors (for example, project size, collaboration modes, number of distributed sites, and team size) may limit the use of agile practices. Hence using agile approaches in distributed

settings is not straightforward; rather practitioners need to fully understand what agile strategies will be effective for their specific development environment.

To address this challenge, the broad objective of our research is to explore and understand effective strategies to help GSD project managers with agility. To this end we develop and describe the components of our framework. As noted earlier, to understand, explain and explore agile strategies, we conduct an industry-based case study in a real life setting. We use the case study findings to further understand the components of our framework. We also investigate if there are any elements not present in the framework that should be considered for inclusion in a revised framework. Finally our research concludes with future research directions.

4 Conceptual Framework

This section defines, explains and describes the various elements of our proposed framework.

4.1 Development Process

In order to support agile approaches in GSD, we have identified a number of components from a literature survey that are included in our framework. The framework includes components and factors that need to be taken into account when a project manager considers using agile approaches for a GSD project. Our framework includes an agility assessment process that covers both project contextual factors and an organization's previous agile experience. The framework also includes agility supporting strategies that are categorized into five groups: plan, policy, people, process and infrastructure. To develop our conceptual framework, we taken several carefully planned steps, as described below:

- To identify framework components we studied existing frameworks and methodologies that discuss agility issues in a number of development scenarios (e.g. [10-11, 33]); and conducted an extensive survey of the GSD literature where agile approaches were used, as well as analyzing the heuristics of experienced GSD researchers and practitioners (e.g. [14]).
- We identified and categorized factors that drive project managers towards choosing agile strategies for their GSD projects (e.g. [14-27]).
- We investigated research papers (e.g. [7, 29]) that describe how project contextual information is recorded. After reviewing these studies, we identified a number of key project contextual factors useful when assessing a possible degree of project agility. Agility assessment is important; it is usually difficult for a GSD project to be agile and a project manger needs to carefully balance agility with a defined development process [6]. An organization's past agile experience is also considered to be part of the agility assessment.
- We reviewed reports describing agile approaches in GSD projects (e.g. [9, 12, 15-28]) and identified a number of agility supporting strategies used by GSD project managers. We categorized these strategies as plan, policy, infrastructure, process and people elements.

- Finally, we consolidated the components into a framework to aid a GSD project manager in deciding what agile strategies could be used within a project.

Figure 1 presents GSD project managers with a framework to assist them in assessing the degree of agility possible for a project. This framework also provides a basis for the consideration of suitable agility supporting strategies in GSD projects.

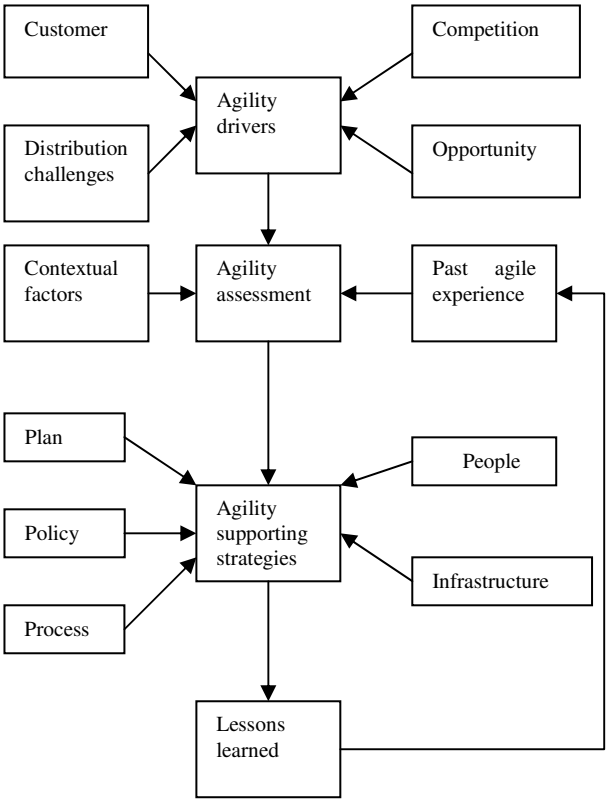

Fig. 1. Proposed conceptual framework for using agile approaches in GSD projects

4.2 Framework Usage

We do not claim that we have developed an exhaustive list of components that influence the agility of all GSD projects. Rather the proposed framework will help guide the identification of effective agile strategies for such projects. The framework is easily modifiable and extensible, as is necessary in research that is still in its inception stage. Our framework will also help to identify missing agile strategies in a GSD project when an agile approach is being considered. To provide an initial validation for our research, we conducted an industry-based case study of a project that uses some agile practices and some GSD supporting practices. From the case study, we identified effective, poorly executed and missing agile strategies in a GSD project that

claimed agility. In future, a series of case studies will help to build up a body of knowledge that will guide GSD project managers in choosing effective agile strategies suitable for their particular development environment.

4.3 Framework Components

Framework components are classified as 1) agility drivers, 2) contextual factors, 3) past agile experience, 4) agility assessment, 5) agility supporting strategies and 6) lessons learned. We discuss each of these below.

Agility Drivers. Agility drivers are factors that influence GSD project managers to use agile approaches. We found that agility drivers can be broadly categorized into the following:

Competition. This category includes factors such as: increasing demand for project quality, a rapidly changing market, cost savings, maximizing project productivity, time, competitors, and the enhancement of technical capabilities, mergers and acquisitions, and scalability [4, 14-16]. Other factors such as, leveraging the distributed team's expertise, facilitating knowledge transfer, and supporting the international nature of a company, can also drive a GSD project manager toward using an agile approach [18].

Customer. The customer is one of the most important drivers in choosing an agile approach for a GSD project [15, 20]. For example, a customer may want to develop his product using XP [22]. Requirements volatility and rapid changes also drive GSD customers toward choosing an agile approach [17, 21].

Distribution challenges. GSD project managers also choose agile approaches to minimize various challenges related to communication, coordination and control [8, 12]. For example, to minimize delays in communication and to increase communication quality, a GSD project manager may use some agile practices (e.g. pair programming) [23].

Opportunity. Because an organization may wish to change its existing GSD development processes, it may perceive an opportunity to choose an agile approach as the most optimal method [15, 17, 21]. Previous project experience success also drives GSD project managers to use agile approaches [15, 16, 24]. A GSD project manager may also use some agile practices in a pilot project or in an experimental study in order to investigate the risks and benefits of agile methodologies [18, 25, 28]. Earlier development method failure (e.g. with the waterfall model), may drive GSD project managers towards agile approaches [27, 26]. A GSD project manager may use an agile approach to: increase a project's visibility, allow for early project estimation, or to help provide client business security [4, 19, 27]. Agile approaches are also used as an opportunity for distributed teams to standardize their processes and tools [19]. A shortage of onshore expertise and a match with an outsourced partner's development methodology, as well as a desire to capture domain knowledge and expertise, may also encourage a GSD project manager toward an agile approach [16].

Contextual Factors. Software projects can be influenced by as many as 250 different contextual factors; although most projects are affected by 10-20 major factors (for example project complexity, size, uncertainty, staff experience, contract nature etc.) [29]. Project stakeholder distribution provides an additional contextual factor in GSD projects and is related to geographical, temporal and socio-cultural distances [12]. Project contextual factors heavily influence the use of agile strategies and we consider the following project contextual factors should be considered as a minimum [7, 30].

Software classification. The software to be developed can be classified as [12, 30]. 1. System, 2. Commercial, 3. Information system, 4. Military, 5. End user/Private, 6. Other.

Project specific. Factors that include development quality, schedule, and risk etc should be considered. But initially we suggest: 1. Contract nature, 2. Project domain, 3. Requirements changes, 4. Staff months, 5. Budget, 6. Complexity, 7. Criticality.

Team characteristics. Specific distributed team information needs to be considered. 1. Team size, 2. Team experience, 3. Project manager's experience, 4. Team work culture.

Distance. The geographical, temporal and socio-cultural distances caused by project stakeholder distribution in a particular GSD project will also be recorded.

1. Geographical distance: This distance is considered as a directional measure of the effort required for one actor to visit another, at the latter's home site [6].
2. Temporal distance: This is a directional measure of the dislocation in time experienced by two actors wishing to interact [12].
3. Socio-cultural distance: Any differences in organizational culture, national culture and language, politics, individual motivation, and work ethics, etc [3].

Technology. Technology required for developing the project also needs to be considered, e.g. graphical packages, specific programming languages etc.

Past Agile Experience. A project experience repository can provide lessons learned and effectiveness ratings for agile approaches used in the past, as well as appropriate tools and agility supporting distributed practices. At the start of a project, a GSD project team with extensive prior project experience will have effective agility coping strategies [13]. Previous agile experience can also help to decide on the extent of agility, and need for 1) formal communication, 2) training, or 3) extensive documentation [20].

Agility Assessment. Agility assessment is an evaluation of a particular project's degree of agility based on several parameters. The project manager should make an assessment of a project's need for agility and the organization's capabilities before deciding to use an agile approach for the project [10]. Conboy and Fitzgerald [10] note that an appropriate agile approach should be selected based on the project's contextual factors. Boehm and Turner's [30] risk based model provides a good example of the assessment of a project's degree of agility; it determines required agility by analyzing a project's environmental risks, agile risk and plan-driven risk. We believe that a GSD project manager can estimate the extent of a project's possible agility by analyzing both project contextual factors and past agile project experiences.

Agility Supporting Strategies. Based on an initial assessment of the extent of agility, a GSD project manager needs to apply agility supporting strategies in an effective way. In any development environment, the core of an agile approach is based on a gradual evolution of effective processes, infrastructure, teams, plans, and policies [11]. For the effective use of agile strategies a GSD project manager should ensure appropriate plans are developed, effective teams formed, and provide the necessary infrastructure, ensure appropriate policies and follow defined processes.

Plan. A GSD project manager should develop plans that will initially describe how the project will be carried out. For example: a project manager can plan to distribute the project work based on the nature of the work (for example, highly volatile work should be co-located), form a team close to the business, provide agile training etc. [20].

People. The success of agility in a GSD project is very dependent on its people [16]. A number of research papers mention that in considering the use of agile approaches, a GSD project manager should decide if he can build distributed teams that include experienced agile developers [16-18].

Infrastructure. A GSD project manager should also ensure the necessary project infrastructure (hardware, software, licenses, tools etc.) to support the agile practices used in the globally distributed project is available [16, 20]. A GSD project manager should also carefully chose appropriate software tools relevant to communication, collaboration, project management, testing, and metrics/measurement etc.

Policy. A project manager should maintain policies to tackle any GSD project challenges that emerge. For example: a project team member training policy (technical, domain, process etc), a documentation maintenance policy or policies for using various agility supporting distributed practices (e.g. cultural liaisons) [15-19].

Process. Balancing agility within defined processes is one of the major challenges for GSD project managers [6]. A GSD project manager should ensure sufficient processes for the effective use of the agile approaches, and should also choose suitable agile practices for the development environment. Like many other researchers, we believe that not all agile practices are suitable for use in globally distributed projects. Thus, a GSD project manager should carefully choose appropriate agile practices suitable for their development environment. Although the agile manifesto suggests less emphasis on process and tools, we found that a GSD project manager must define processes for the use of agile practices in a GSD environment [6].

Lessons Learned. Both the degree of success achieved, and the challenges faced in a project, while using specific agile approaches, should be monitored and stored in a project experience repository [20]. Practices and tools which are proven to be effective for a particular project should be recorded and managed in the process repository.

5 Research Methodology and Case Study

In this section, we report on our industry-based exploratory case study. The case study is considered as a robust research method with a range of data collection approaches

when a holistic in-depth investigation of a social phenomenon in its real life context is required [31]. The research question we investigate is: how can GSD project managers decide what agile strategies will be effective in a particular project environment? To make such a decision, a project manager needs an understanding of the everyday mechanics of team processes, project characteristics, distance and technology involved in existing GSD projects and the agile practices that are used effectively in such projects. In this research, we do not provide formal hypothesis testing or draw any general conclusions as GSD has many forms based on project contextual factors. We use a case study to gain greater understanding of how specific agile strategies were used within a particular GSD project context.

Our primary data collection method was the interview, supplemented by a review of relevant project documentation, onsite demonstrations of the software and informal conversations. We performed two face-to-face interviews (one with a project manager and one with an actively involved customer); each interview lasted about two hours. We provided a brief research plan to both participants before beginning the interview sessions. We asked our respondents about the facts of the matter as well as gaining their opinions about events that occurred. Project artifacts, such as documentation, were also used as an important data source. System specifications, the project plans, testing scripts and the completed software were made available to the researchers. Documentary information was also used to corroborate and augment evidence found from the interviews and discussion. A qualitative content analysis technique was used to extract the framework component data from the interviews. Qualitative data analysis was done by the lead author who coded both interviews, and developed separate codes for addressing each of the framework components. In doing the data analysis, our aim was to identify, describe and make sense of how the chosen strategies were used to make a GSD project agile. To improve the quality of our interpretation, we reported our initial findings back to both interviewees who provided us with valuable feedback that rectified omissions and misunderstandings.

5.1 Case Description

AusBest is an Australian based software development company that develops a range of software products. For some time the company has had developers in Australia and Malaysia. We call the project we investigated the "AGI project"; it is a service-based commercial software product developed by a team distributed in Australia and Malaysia. In a later section, we discuss our case in detail while describing elements of our framework. We discuss our various case study findings, in particular how our case GSD project's agile strategies differ from our proposed conceptual framework. This helps us to understand how a real life GSD project manager did or did not use agile strategies effectively.

Agility Drivers. AusBest's senior manager decided to go with an agile approach in order to minimize project cost. He felt that this would enable his company to release the product within the set time limits, to use their distributed resource pool effectively and help with cost reduction. Thus we can argue that as cost is a component of competition and that *"Competition"* was the key driver for the AGI project in its use of agile strategies.

Contextual Factors. Project contextual factors were as follows.

Software classification. The "AGI project" software is a graphical software engineering tool developed for *commercial* use for external customers.

Project specific factors. Key project specific factors are as follows.

1. Contract nature: The developer's contract specified that the project would be developed within a fixed price and schedule.
2. Project Domain: The project is a web based graphical service application.
3. Requirements change: There were a number of initial requirements changes but after that the requirements were stable and clear to the development team.
4. Staff months: In terms of effort the AGI project required approximately 20 person months and was developed over six months.
5. Budget: The project budget was slightly more than one hundred thousand Australian dollars.
6. Complexity: Although the project was small in size it was a complex graphical application; this increased the development complexity.
7. Criticality: The software was critical to setting up a new business venture.

Team characteristics. The characteristics of the project teams included:

1. Number of distributed teams: There were two development teams, distributed in Australia and Malaysia. The customer team was based in Australia.
2. Team size: The project manager was based in Sydney; the Sydney part of the team consisted of the project manager, two full time developers and one part time test engineer. The time involvement of Malaysian developers varied. They were mostly involved in back end development work, and had around 25 developers with one local development lead. The project work was assigned based on skills and availability as the Malaysian developers were involved in several other projects at the same time.
3. Overall team experience: Both Australian and Malaysian team members had previously worked on several distributed projects and had also participated in agile projects in the past. However, no team members had any prior experience of using agile practices in distributed projects.
4. Project manager experience: Although the project manager was experienced in project management he had no prior experience using agile approaches in a distributed project.
5. Team culture: The working culture of both sites tended toward informal. Later on the project manager tried to impose more explicit processes to provide better project visibility to the teams.

Distance. The project involved low geographical and temporal distance but had significant socio-cultural distances.

1. Geographical distance: The development team was distributed between Malaysia and Australia. Malaysia and Australia are relatively closely located and there are convenient air links and regular flights between the two countries.

2. Temporal distance: There was a two hours time difference (three hours in summer) between Sydney and Malaysia; hence we can argue that the project had a low temporal distance.
3. Socio-cultural: we used Hofstede's study [32] to measure the cultural differences between the Malaysian and Australian team. Hofstede's study use five cultural dimension (Power distance, Individualism, Masculinity, Uncertainty Avoidance, Long term orientation) to describe national cultural differences between two countries. From that study's indexes, we found there were significant differences in the cultural dimensions, power distance, individualism and moderate differences in masculinity, and uncertainty avoidance. Thus based on Hofstede's study [32], we can claim that there were significant socio-cultural distances between the Malaysian and Australian team members.

Technological. A variety of programming languages, methods and tools were used to develop the complex graphical software engineering tool. Initially, the development team members had a low level of expertise with the technologies used for the project. The Sydney team members received training in the use of the new programming languages and tools.

Past Agile Experience. The project manager claimed that AusBest had considerable previous agile project experience although applying agile practices in distributed projects was new for this company. The company's previous agile project success encouraged them to use some agile practices in this distributed project.

Agility Assessment. The project manager did not follow any particular model or method when assessing the possible degree of project agility during the initial stage of the development; this was due to the lack of agile experience of the project manager in distributed projects.

Agility Supporting Strategies. Although the project manager used a number of agile supporting strategies, we believe that the lack of defined project processes, plans and policies made project success difficult. The project manager had a number of difficulties because some processes were ignored.

Plan: Poor project planning was identified as a major challenge and this caused a number of difficulties. The project manager eventually developed a concrete plan for the project which was agreed with the customers. For example: minimizing task interdependence was an important focus in that plan.

People. The people were an important driver in this project's success. Although the project suffered from staff turnover the project manager utilized the company's large offshore resource pool effectively (a large part of the team was in off shore). The project manager moved as much work as he could to Malaysia which had a number of talented programmers and a team lead who was very committed to the project; the leader of the Malaysian team played a key role in making the project a success.

Infrastructure. The project manager ensured sufficient infrastructure was available to support the agile approaches used in the project. He also ensured that a number of tools were available to the teams including communication tools such as**:** telephone,

VOIP (Skype), email, teleconferencing, video conferencing, IM. The project manager used a wiki as a collaborative tool, and a tool called "Jira" as an issue tracker, bug tracker and project management tool. For testing purposes, automated testing tools (e.g. continuous integration tools) were also used.

Policies. The project manager maintained different policies for different agile strategies. For example, a training policy for the Sydney based front-end developers. There were also a number of policies for using agility supporting distributed practices, for example, policies for maintaining "just enough documentation" in order to minimize project misunderstanding and miscommunication. He frequently used the practice "visits" for both development teams, and spent every second month in Malaysia. The project manager also used other supporting practices such as multiple communication modes (e.g., sufficient communication and collaborative tools), and synchronized work hours to increase communication and reduce misunderstandings among distributed team members.

Process. Project management suffered due to a lack of defined processes; the project manager appeared to have no control over the project at some stages. Later, to take control, and to make project management visible to project stakeholders, the project manager set up some standard work processes and took greater ownership with more direct supervision. The project manager claimed that he used a number of agile practices including daily Scrum meetings, Scrum planning meetings, Scrum review meetings, Retrospectives, continuous code integration, and test driven development within their normal defined development process.

Lessons learned. Although the project manager did not maintain a formal project experience repository, he documented what practices were effective and any problems incurred while using the agile practices. He also documented effective agility supporting distributed practices and tools used in the project.

5.2 Discussion

The project was completed within time and budget and was considered a success by the customers and vendors. Both the customer and project manager were happy with the product developed. However, our framework-based analysis aids us in exploring a number of issues in relation to the use of agile strategies in this project. We found that this agile project contract was fixed price although this type of contract has not previously proven effective in an outsourcing environment [34]. The project manager also did not use some agile practices effectively and did not properly consider the possible extent of agility for the project during the initial stages of development. We believe that this was due to the project manager's inexperience in using agile methods in a distributed environment. Plans and policies were also lacking for the effective use of the agile strategies chosen. The customer maintained continuous pressure on the project manager to properly define project processes and to use the agile practices in a planned way. The customer actively participated in the project and had a very good relationship with both the project manager and AusBest's senior management. The project contract specified that payment was to be made incrementally, and when the customer was unsatisfied with the release, payments were not made. This meant that

senior management investigated what was going on, and this was one reason for improvement in the processes and for a successful project outcome.

Although the agile manifesto opposes a focus on processes and tools, these are very important when using agile practices in distributed projects. Although poor project management and staff turnover are common risks for any software development project, our study reveals they are more challenging when using agile approaches in a distributed project. Experienced agile developers are also a key requirement for the successful use of agile approaches in distributed projects; our GSD project suffered substantially when experienced agile developers left the project.

Improper use of agile practices creates problems. The project manager did not use some agile practices effectively. The customer (who had extensive software development experience and actively participated in the project development) commented that some of the practices, including daily standup meetings, Scrum planning, and Test Driven Development (TDD) were not used properly. For example, one of the customers commented that the development team released code (working software) too frequently (almost every day!) with a large number of errors. This made it very difficult for the customer to perform acceptance testing. The customer complained that the testing took an unreasonable amount of time, and that the releases should be fewer, of higher quality, and with previously identified problems and bugs fixed before the next release. This led to a number of confrontational phone calls and visits to the vendor by the customer although later the project manager was more careful with the releases. This case study reinforces the importance of using teams of skilled developers for the effective use of agile approaches. Lack of developer language expertise can also impact on the effective use agile practices.

Thus, in our case study project the GSD project manager struggled with the framework components, "People", "Plan" and "Process". The project manager did not find any major difficulties due to socio-cultural and time zone differences. He successfully minimized project distribution challenges by ensuring a sufficient communication environment by providing suitable communication tools and supporting practices, for example, synchronized work hours, and frequent visits. The work culture of the Malaysian site also made it relatively easy for the project manager, as the Malaysians frequently started their work early to increase the number of overlapping hours with Sydney team. The GSD project manager helped to build trust by establishing a feeling of "teamness", and increased project visibility, and reduced misunderstandings and misinterpretations by using agile practices and appropriate supporting practices.

The project manager was also happy with the tools used in the project. The communication tools: telephone, VOIP (Skype), email, teleconferencing, video conferencing, and Instant Messaging (IM) were found effective. The wiki used by the project was also found effective for collaboration. The project manager effectively used an integrated testing tool and was happy with the tool "Jira" which was used as an issue tracker, bug tracker and also to support project management.

5.3 Case Study Limitations

The design of this case study is based upon the four criteria for judging the quality of research design recommended by Yin [31]. Construct validity, which is involved with

establishing correct operational measures for the concepts being studied was not a limitation in our study. We developed a sufficient operational set of measures for collecting data. As our case study is exploratory in nature, not explanatory or causal, we need not consider internal validity. Our study is also not concerned with external validity as our study findings are not generalized to other GSD projects. Our single case study initiated an exploration of the use of various agile strategies in a GSD project. In this case study, we must consider reliability; data was collected based on a framework developed from the literature. However we cannot exclude bias on the part of our interviewees who reported what they thought happened. However, we did use multiple sources of evidence (documentation, discussion, interaction etc) to help us ensure sufficient reliability.

6 Conclusions and Future Research

There is an increasing interest in using agile approaches for GSD projects. GSD takes many forms, based on project contextual factors but process issues around using agile approaches in GSD are not clearly understood. GSD practitioners need further research to help them understand what agile strategies are likely to be effective for a particular type of project. We propose a conceptual framework based on existing GSD literature that describes the effective use of various agile strategies. Our goal is to introduce the different elements of our framework and to provide an initial test of its effectiveness with an industry-based case study. Various issues were identified, including the project manager's lack of experience, and failure to provide suitable processes, plans and policies. Such processes, plans and policies would help a GSD project manager to understand how to achieve effective agility in a development environment. Our case study reveals that the project manager did not assess project contextual factors for an appropriate degree of agility prior to starting this project.

A GSD project manager can reduce some project stakeholder distribution challenges by using appropriate agile practices; for example: we agree with the project manager's claim that distributed Scrum meetings, using video conferencing, reduces GSD project communication and coordination overhead. On the other hand, our case study has identified elements missing from our proposed conceptual framework as we did not initially include some important project contextual factors, such as staff turnover, contract type, customer-project management/management relationship, technology expertise, nature of the day to day work, and documentation practices.

In future, we will conduct a series of case studies to aid us in better understanding the use of agile strategies in GSD projects. This will include appropriate agile practices, agility supporting distributed practices and tools within a defined GSD project context. Thus, our research will contribute to answering a current GSD research question: What agile strategies are effective and when?

Acknowledgments. M. Ali Babar's research is partially supported by Science foundation Ireland under grant number 03/CE2/I303-1.

References

1. Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be agile? Communications of the ACM 49, 41–46 (2006)
2. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods - Review and analysis. VTT Electronics (ed.). VTT Publications (2002)
3. Conchuir, E.O., Holmstrom, H., Agerfalk, P.J., Fitzgerald, B.: Exploring the Assumed Benefits of Global Software Development. In: Proceedings of the Conference on ICGSE 2006, pp. 159–168. IEEE Press, New York (2006)
4. Nisar, M.F., Hameed, T.: Agile methods handling offshore software development issues. In: Proceedings of the conference on INMIC 2004, pp. 417–422 (2004)
5. Taylor, P.S., Greer, D., Sage, P., Coleman, G., McDaid, K., Keenan, F.: Do agile GSD experience reports help the practitioner? In: Proceedings of the Conference on GSD 2006, pp. 87–93 (2006)
6. Agerfalk, P., Fitzgerald, B.: Flexible and Distributed software processes: Old Petunias in new bowls? Communications of the ACM 49, 41–46 (2006)
7. Williams, L., Kerbs, W., Layman, L., Anton, A.I., Abrahamsson, P.: Toward a Framework for Evaluating Extreme Programming. In: Proceeding of the Conference on EASE 2004, pp. 11–20 (2004)
8. Manifesto for Agile Software Development, `http://www.agilemanifesto.org`
9. Paasivaara, M., Durasiewicz, S., Lassenius, C.: Distributed Agile Development: Using Scrum in a Large Project. In: Proceedings of the Conference on ICGSE 2008, pp. 87–95 (2008)
10. Conboy, K., Fitzgerald, B.: Toward a Conceptual Framework for Agile Methods. In: Zannier, C., Erdogmus, H., Lindstrom, L. (eds.) XP/Agile Universe 2004. LNCS, vol. 3134, pp. 105–116. Springer, Heidelberg (2004)
11. Qumer, A., Hendersom-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Information and Software Technology 50, 280–295 (2008)
12. Holmstrom, H., Fitzgerald, B., Agerfalk, P.J., Conchuir, E.O.: Agile Practices Reduce Distance in Global Software Development. In: Information Systems Management, pp. 7–26 (summer 2006)
13. Lee, G., Delone, W., Espinosa, J.A.: Ambidextrous coping strategies in globally distributed software development projects. Communications of the ACM 49, 41–46 (2006)
14. Herbsleb, J.D.: Global Software Engineering: The Future of Socio- technical Coordination. In: Proceeding of the Conference on Future of Software Engineering, FOSE, pp. 188–298 (2007)
15. Sutherland, J., Viktorov, A., Blount, J., Puntikov, N.: Distributed Scrum: Agile Project management with Outsourced Development Teams. In: Proceedings of the Conference on HICSS'40 (2007)
16. Sutherland, J., Schoonheim, G., Rustenburg, E., Rijk, M.: Fully distributed Scrum: The secret sauce for Hyperproductive Outsourced Development Teams. In: Proceedings of the Conference on Agile 2008, pp. 339–343 (2008)
17. Berczuk, S.: Back to Basics: The Role of agile Principles in Success with a Distributed Scrum Team. In: Proceedings of the Conference on Agile 2007, pp. 382–388 (2007)
18. Hogan, B.: Lessons Learned from an eXtremely Distributed project. In: Proceedings of the conference on Agile 2006, pp. 321–326 (2006)
19. Yap, M.: Follow the Sun: Distributed Extreme Programming Development. In: Proceedings of the Conference on ADC 2005, pp. 218–224 (2005)

20. Sureshchandra, K., Shrinivasavadhani, J.: Adopting Agile in Distributed Development. In: Proceedings of the Conference on ICGSE 2008, pp. 217–221 (2008)
21. Farmer, M.: DecisionSpace Infrastructure: Agile Development in a Large, distributed team. In: Proceedings of the Conference on Agile Development Conference (ADC 2004), pp. 95–99 (2004)
22. Sison, R., Yang, T.: Use of Agile Methods and Practices in the Philippines. In: Proceedings of the Conference on 14th APSEC 2007, pp. 462–469 (2007)
23. Xiaohu, Y., Bin, X., Zhijun, H., Maddineni, S.R.: Extreme Programming in Global Software Development. In: Proceedings of the Conference on CCECE - CCGEI 2004, pp. 1845–1848 (2004)
24. Smits, H.: Implementing Scrum in a Distributed Software Development Organization. In: Proceedings of the Conference on Agile 2007, pp. 371–375 (2007)
25. Urdangarin, R., Fernades, P., Avirtzer, A., Paulish, D.: Experiences with Agile Practices in the global studio project. In: Proceedings of the Conference on ICGSE 2008, pp. 77–86 (2008)
26. Paasivaara, M., Lassenius, C.: Could Global Software Development Benefit from Agile Method? In: Proceedings of the Conference on ICGSE 2006, pp. 109–113 (2006)
27. Cottmeyer, M.: The Good and Bad of Agile Offshore Development. In: Proceedings of the Conference on Agile 2008, pp. 362–367 (2008)
28. Simon, M.: Internationally Agile. In: Inform IT 2002 (2002), http://www.informit.com/articles/article.aspx?p=25929
29. Jones, C.: Software Assessments Benchmarks and Best Practices. Addison Wesley, Boston (2000)
30. Boehm, B., Turner, R.: Balancing Agility and Discipline: A Guide for the Perplexed. Addison Wesley, Boston (2003)
31. Yin, R.K.: Case Study Research. Sage publications, Thousand Oaks (1994)
32. Geert Hofstede™ Cultural Dimensions, http://www.geert-hofstede.com/
33. Sharifi, H., Zhang, Z.: A methodology for achieving agility in manufacturing organizations: An introduction. Production Economics 62, 7–22 (1999)
34. Martin, A., Biddle, R., Noble, J.: When XP met outsourcing. In: Eckstein, J., Baumeister, H. (eds.) XP 2004. LNCS, vol. 3092, pp. 51–59. Springer, Heidelberg (2004)