2009 16th Asia-Pacific Software Engineering Conference

Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework

Emam Hossain
CSE, The University of New South Wales and National ICT Australia
Sydney, Australia
Emam.Hossain@nicta.com.au

Muhammad Ali Babar
Lero, University of Limerick
Limerick, Ireland
Muhammad.AliBabar@lero.ie

Hye-young Paik, June Verner
CSE,The University of New South Wales, UNSW
Sydney, Australia
{hpaik, jverner}@cse.unsw.edu.au

Abstract— ***There is growing interest in applying agile practices in Global Software Development (GSD) projects. But project stakeholder distribution in GSD creates a number of challenges that make it difficult to use some agile practices. Moreover, little is known about what the key challenges or risks are, and how GSD project mangers deal with these risks while using agile practices. We conduct a Systematic Literature Review (SLR) following existing guidelines to identify primary papers that discuss the use of Scrum practices in GSD projects. We identify key challenges, due to global project distribution, that restrict the use of Scrum and explore the strategies used by project managers to deal with these challenges. Our findings are consolidated into a conceptual framework and we discuss various elements of this framework. This research is relevant to project managers who are seeking ways to use Scrum in their globally distributed projects.***

Keywords- Global software development, Risk, Scrum

I. INTRODUCTION

Global Software Development (GSD) is a recent trend in the software development industry. GSD practitioners claim that it is possible for organizations to gain time-zone effectiveness, leverage a large skill pool, develop software closer to the customer, and exploit low labor costs in certain parts of the World [1]. The Agile Software Development (ASD) paradigm has also gained significant attention due to its flexible approach to managing requirements volatility, fostering of close collaboration between customers and developers, and early and frequent delivery of products [2]. Among the various agile methods, eXtreme Programming (XP) primarily focuses on development practices, and Scrum focuses on project management, are the most well known [2]. There is growing interest in applying agile practices in GSD projects to leverage the combined advantages of both approaches [3]. But project stockholder distribution in GSD projects is often characterized by temporal, geographical and socio-cultural distance and creates a number of challenges that may impact on project communication, coordination and collaboration processes [4]. Using agile practices in GSD projects may also exacerbate risks as agile practices are based on the philosophy of close, frequent and collocated collaboration [2]. In addition, some GSD project contextual factors, for example, collaboration modes, increased number of sites, a large number of project personnel, lack of tool support etc., may also impact on project communication and collaboration processes and restricts the use of agile practices. Thus, some researchers [4] argue that the question of using agile practices in a distributed setting is still open to debate. Despite a number of risks when using agile practices in GSD, we found a few instances of success in the literature when some agile practices were used by global teams [5]. But the current literature does not clearly address what the key risks are due to project distribution, and how GSD project managers can deal with these risks when using agile practices in their development environment.

To address this research challenge, we propose a conceptual framework based on an extensive review and systematic analysis of the GSD research literature. The framework graphically presents the key risks due to GSD project contextual factors, and current strategies to deal with these risks while using Scrum practices. For identifying the risks and related strategies, we identified primary papers that discuss the use of Scrum in GSD based on a previous Systematic Literature Review (SLR) [6-7]. Our SLR, which followed the guidelines reported in [8], identified 20 primary papers (from 366 papers) that discuss the use of Scrum in GSD projects. We chose "Scrum", as GSD project managers have shown interest in using Scrum practices in their day-to-day project management [7].

II. BACKGROUND AND MOTIVATION

We discuss the agile method "Scrum", the risks of using Scrum in GSD, and the context of our research.

A. Scrum

Scrum is an iterative and incremental project management approach that provides a simple "inspect and adapt" framework [5]. When using Scrum, software is delivered in increments called "Sprints" (usually 2-4 week iterations). A sprint starts with planning and ends with a

1530-1362/09 $26.00 © 2009 IEEE
DOI 10.1109/APSEC.2009.56

review. A sprint planning meeting is a time-boxed meeting dedicated to developing a detailed plan for a sprint. Project stakeholders attend sprint review meetings to review the state of the business, the market and technology. A retrospective meeting may be scheduled to assess the completed sprints. A daily Scrum meeting is a short daily meeting (usually up to 15 minutes long) in which each team member is expected to address three questions: what did I do yesterday, what will I do today and what impediments are in my way? Three artefacts, namely: product backlogs, sprint backlogs and burn-down charts are produced. Backlogs consist of customer requirements while daily burn down charts show what cumulative work remains.

B. Risk of Using Scrum in Global Software Development

Scrum is usually considered to be effective for co-located projects with a small team size as Scrum teams are self organized and focused on rich team communication and collaboration [2]. But project stockholder distribution in GSD projects is often characterized by temporal, geographical and socio-cultural distances and creates a number of challenges or risks that may impact on communication and collaboration processes [7]. In addition, some GSD project contextual factors, for example, collaboration modes (e.g., sub contractor involvement), increased number of sites, large number of project personnel involvement, lack of tool support may also significantly impact on a project's communication processes. Thus, we can argue that GSD project contextual factors may create a number of challenges or risks and may restrict the use of Scrum.

But it is not clear from the current GSD literature what the key risks are, and how these risks are minimized while using Scrum. For example, it is not clear from the literature what the risks of using Scrum are in a GSD project that involves sub-contractors. Similarly, we do not know the risks of using Scrum in a GSD project with a large number of project personnel, or an increased number of distributed sites. Thus, we can argue that there are a number of risks when using Scrum in GSD and these risks may vary according to project contextual factors.

Despite the apparent risk of using Scrum in GSD, we have found a few successful instances of using Scrum. We have found, in the GSD literature, that due to project contextual factors, project managers are using a number of risk minimization strategies or practices to support the use of Scrum. But the effectiveness of those strategies is not clearly described or understood. Additionally, we do not know how the risk behavior varies from one GSD project to another because of different project contextual factors. For example, it is still not clear if Scrum is suitable for a GSD project with a large number of sites. Moreover, it is also not clear how effective the practice of using "Scrum of Scrums" is if a GSD project involves a large number of sites. Hence, there is a vital need of understanding the risks involved and their mitigation processes to support the use of Scrum practices by considering GSD project contextual factors.

C. Research Context

The objective of the reported research was to identify and understand, from the research literature, the challenges or risks that need to be considered when using Scrum in GSD projects and corresponding strategies to minimize these risks. In order to identify the primary GSD papers that discuss the use of Scrum, we conducted a Systematic Literature Review (SLR), which involved several activities, such as the development of a review protocol, the identification and selection of primary papers, data extraction and synthesis, and reporting the results. Our SLR identified twenty primary papers [9-28] that discuss the use of Scrum practices in GSD. A comprehensive description of our SLR and findings can be found in Hossain et al [6]. We identify the key challenges or risks and also corresponding strategies to reduce these risks. Our findings are consolidated into a conceptual framework that is expected to provide some useful insights into the mechanics of using Scrum for GSD projects.

III. CONCEPTUAL FRAMEWORK

We discuss the elements of conceptual framework with its possible usage and limitations in this section.

A. Framework Development Process

To develop our conceptual framework, we took several carefully planned steps, as described below:

- In order to identify the framework components, we studied a number of existing frameworks and models that discuss various aspects of GSD (e.g. [1, 17, 29-32]. We 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. [5, 32-34]) to identify the key components.
- We identified and categorized key challenges or risks from the twenty primary papers [9-28] that discuss Scrum processes in GSD projects.
- From these papers, we also identified GSD project manager strategies or practices to reduce these challenges to support the use of Scrum. We categorize these practices by the risks identified.
- Finally, we consolidated the components into a framework in order to help a GSD project manager to understand the key risks that may have negative impact when using Scrum practices. We also present the strategies used by GSD project managers to minimize the identified risks.

B. Framework Components

Our framework's components are broadly classified as 1) Major risks, 2) Current strategies to reduce these risks (as

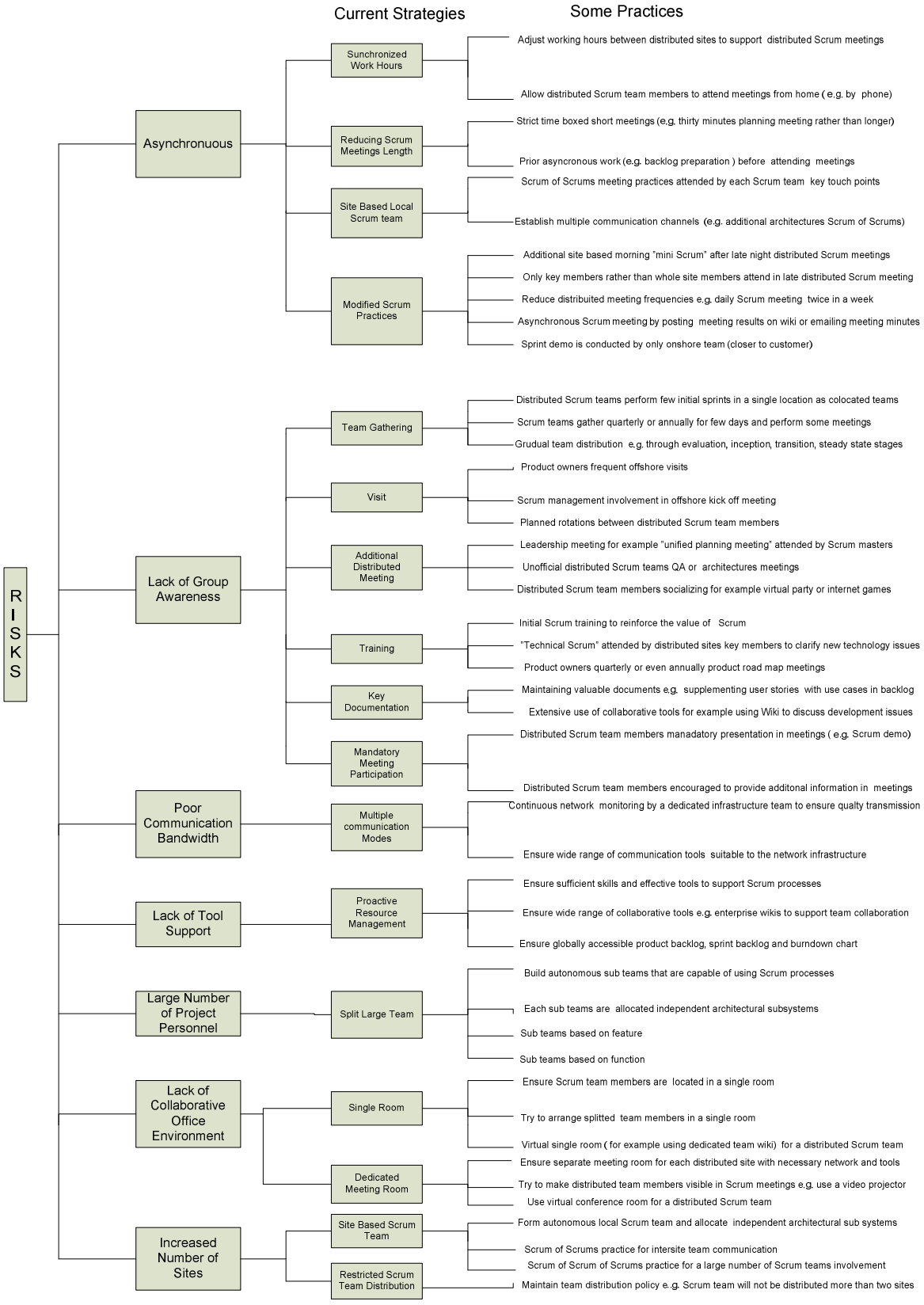

Figure1. Risks identification and mitigation processes for globally distributed projects using Scrum

shown in Figure 1. We discuss each of the identified risks and the strategies to reduce these risks.

1) Asynchronuous: Distributed Scrum meeting practices are found difficult when a GSD project involves a lack of overlapping working hours [9-10, 15-16, 18-19, 25-26, 28]. GSD project managers use a number of strategies to ensure synchronous communication while using various meeting practices. We have identified some existing strategies and categorize them as follows.

a) Synchronized work hours: This strategy is widely used to increase overlapping hours between distributed sites. Some practices, for example, adjusting working hours between distributed sites, allowing distributed Scrum team members to attend meetings from home via phone may ensure the required level of synchronous communication to support Scrum meeting practices [9-10, 15, 18, 22-23, 25-26,27-28].

b) Reduce Scrum meeting length: Scrum teams can also address the asynchronous challenge by using strategies that can help to reduce meeting length. For example, a Scrum team can perform strict time boxed short meetings (e.g. thirty minute Scrum planning meetings). It is also possible to make a meeting short by some prior asynchronous work, for example, through posting three daily Scrum questions, or preparing a backlog, before attending a distributed meeting. These length reducing processes can be used to avoid late night or early morning distributed meetings [15, 17, 19, 21, 24].

c) Site based local Scrum team: Due to a lack of overlapping hours between distributed sites, Scrum teams are formed locally. In this case, each Scrum team has site based team members and who perform their own local Scrum [15-18,19-20, 27]. The practice Scrum of Scrums is attended by a key touch point member of each Scrum team to ensure inter-team communication. Thus to conduct a local Scrum, the PM needs to build autonomous local teams and allocate independent architectural subsystems with well defined interfaces to each team in order to reduce inter site communication [15-18, 22]. To establish multiple communication lines, along with the Scrum of Scrums meetings, additional meetings, for example, for the technical lead or design architect of each local team are also allowed [18].

d) Modified or extended Scrum meeting practices: It has been reported that Scrum is a flexible method that can be modified according to a distributed project's requirements. To reduce asynchronous challenges, Scrum team also modify or extend existing Scrum practices to reinforce the value of Scrum such as a local "mini-scrum" in the morning, after a late distributed scrum meeting can be very effective in reinforcing the value of a Scrum within the local team [16]. Other practices, e.g., instead of the whole team's presence at a late night (or early morning) Scrum meeting, only key members attend the distributed meeting [14, 16, 22]. The distributed daily Scrum meetings are cut down to twice-a-week meetings [25]. Modified Scrum practices, e.g., asynchronous retrospective meetings (e.g., posting comments and results on Wikis, emailing the minutes of a local Scrum meeting to other teams), and conducting a sprint demo by an onshore team only (later the onshore team briefs the offshore team) are also used to address asynchronous challenges [9-11, 18, 22, 25].

2) Lack of Group Awareness: GSD project managers use a number of practices to increase teamness to facilitate better team collaboration in various Scrum meeting sessions. We categorize those practices are as follows.

a) Team Gathering: Project stakeholders of Scrum teams are gathered in a single location and perform initial sprints as a collocated team before the teams are distributed [22, 24-28]. Distributed project stakeholders of Scrum teams are also gathered quarterly or annually for few days [9, 15, 19, 27]; here Scrum teams have meeting sessions including scrum planning, review meetings, retrospectives, and even various social activities [27].

b) Visits: Project stakeholders' visits between distributed sites are a common practice to increase teamness. Product owners regularly visit offshore sites to help increase project domain knowledge [24-25, 28]. Planned rotations among distributed team members also facilitates cultural exchange, improves shared understanding, reduces miscommunication and improves distributed meeting sessions [23-24]. Practices like product owners organizing quarterly product roadmap meetings are also effective for teams to fully understand the project vision and reinforce the value of Scrum [25].

c) Additional distributed meetings: For increased team collaboration, along with formal meetings, distributed Scrum team members may also use informal meetings for clarifying issues. The unofficial meetings may involve leadership meetings, testing, architectural meetings, distributed team lead meetings, and "unified planning meetings" attended by Scrum masters of each sub-team, peer meetings, as well as social meetings (for example, virtual party or games) or even "coffee talks" for collocated team members[9, 23].

d) Training: The practice, training, including "initial Scrum training" or even a "technical Scrum" to clarify new technology issues also reinforces the value of Scrum and improves GSD team collaboration [18, 25].

e) Key documentation: Maintaining valuable documentation may also improve GSD team collaboration processes while using Scrum practices [16, 18, 25, 28]. For example, supplementing user stories with Use Case diagrams in globally accessible backlogs helps to reduce misunderstandings and improves team collaboration [25]. Scrum teams use a number of tools, for example, issue

tracker (e.g. Jira), enterprise wikis (e.g. Confluence), and a project management tool (e.g. Scrum works) to maintain good documentation and project transparency.

f) Mandatory participation: A practice like "mandatory participation" may reduce risks such as "offshore silence" due to socio-cultural distance. For example, each site provides a thirty-minute mandatory demo presentation during retrospective sessions; this ensures better team collaboration and helps build empowered distributed Scrum teams [25, 27]. To increase team collaboration, offshore teams are also encouraged to provide additional information during the Scrum meeting sessions [9].

3) Poor Communication Bandwidth: To support the rich communication and collaboration environment requirement required for Scrum, GSD project managers need to provide high communication bandwidth and reliable network support throughout the development life cycle [14-16, 24-25, 27-29]. Some GSD projects in the literature found that communication networks are often slow and unreliable, with poor transmission quality. These cause problems for Scrum meetings by hampering communication processes. This is especially true when some specific types of communication tools are used (e.g. video conferencing) [24-25, 28].

To provide a rich communication environment and also to avoid slow, unreliable, and poor transmission, we found Scrum teams use practices like "multiple communication modes". The practice ensures that a distributed Scrum team is supported with various communication tool options, such as phone, web camera, teleconference, video conference, web conference, net meeting, email, shared mailing list, IM, Short Message Service (SMS), and Internet Relay chat (IRC) [9]. Thus, a Scrum team can choose the appropriate tool from a wide range of communication tools suitable to the communication bandwidth during their meetings. For example, if a Scrum team found videoconferencing is not facilitated by the existing communication bandwidth, they may choose a teleconference for their distributed meeting sessions.

4) Lack of Tool Support: Our literature review reveals that GSD project managers must ensure sufficient communication, collaboration and project management tools that include, for example, a suitable bug tracker, issue trackers, a globally accessible backlog tool, and burn down chart while using Scrum [13, 19-20, 24-27]. We found that GSD project managers use practice like "proactive resource management". With this practice, Scrum teams are ensured available tools and skills are suitable to their development environment as a support for their Scrum processes. We also found that, along with communication tools, GSD project managers use a number of collaborative tools, including Wikis, Blogs, social book marking, expertise finders, whiteboards, electronic work space, desktop and application sharing, photo charts, knowledge bases, experience databases, and lessons learned repositories, while using Scrum[9-28]. An enterprise wiki (e.g. Confluence) has been found to be very effective while using Scrum practices as distributed team members can post their comments there and publish the results of various Scrum meeting minutes [28]. To increase project transparency and visibility and to support the Scrum practice "Backlog", Our findings also reveal that distributed Scrum teams use a number of tools including globally accessible project management tools (e.g. "Rally"), issue tracker, bug tracker (e.g. "Jira"), backlog management tools (e.g. "Scrum works"), and various tools for supporting the Scrum artifacts such as "Burn down charts" [9-13, 16, 19, 26, 27-28].

5) Large Number of Project Personnel: A Scrum team is typically five to ten people, although teams as large as fifteen and as small as three have also reported benefits [5]. For this reason, using Scrum for a team of a large number of project personnel is considered to be a risk. It is even more risky to use Scrum in a large team distributed over multiple sites [10,14, 16, 19, 25].

However, based on our literature review, we note that Scrum can be used in GSD projects in different forms. For example, in a GSD project only a particular distributed site rather than all distributed sites may use Scrum. This type of Scrum team is defined as an Isolated Scrum team [10]. It is not clear from the literature how these teams are integrated with other plan driven (or other agile) teams and how effective these Scrum teams are in real life settings. We believe that this type of isolated Scrum team will face a number of challenges due to the different methods involve in the project. We also found that a large GSD project can consist of a number of site based local Scrum teams. In this type of GSD project, each Scrum team is assigned independent architectural subsystems and performs their own Scrum. The practices Scrum of Scrums, attended by the key touch points (e.g. Scrum master) from each sub-team or other informal distributed meetings ensure effective project coordination [9]. If the number of sub-teams increases, in some cases a nested Scrum of Scrums practice (e.g. Scrum of Scrum of Scrums) provides effective coordination for a large number of Scrum teams [16]. This form of Scrum, in GSD, is sometimes described as a distributed Scrum of Scrums team model [10]. We found that this type of Scrum team is very common in GSD projects. It is widely used because it is effective in reducing GSD communication and collaboration overhead. But some GSD projects found there was a risk in using the Scrum of Scrum practice because it may create miscommunication, misunderstanding or misinterpretation and may also involve information loss [5]. The challenges arise due to an over reliance on one person per team for communication. To reduce these problems, in addition to the Scrum of Scrum practice, some Scrum teams also allow additional Scrum of

Scrums communication. For example, in addition to a Scrum master, Scrum of Scrums, the architecture group arranges their own regular Scrum of Scrums. This type of practice ensures multiple communication channels and increases project communication. In some GSD projects, we found cross-functional Scrum teams with team members distributed across geographical locations. This type of Scrum team organization is defined as a fully integrated Scrum team model [10]. Geographical, temporal and socio-cultural distance involved in a GSD project significantly impacts on this type of Scrum team model as all of the distributed Scrum team members need to attend and participate in every Scrum meeting practice. We found in some cases that a project with several fully integrated Scrum teams follows the practice "centrally located management team" in which the management person for each team is centrally located [9-10]. In this case, frequent meetings among a centrally located product owner team, a team of Scrum masters, and architects from the sub-teams, ensures effective multiple sub-team communication and collaboration [10].

Whilst we have identified that different Scrum team models are used in different GSD projects, we have found that a commonly used strategy for managing a large distributed Scrum team is to split it into small manageable sub-teams [9-10, 14]. Thus, a large Scrum GSD project may consist of a number of teams (or sub-teams). The sub-teams may use any of the previously identified Scrum team models, including the isolated Scrum team model, a distributed Scrum of Scrums team model, a fully integrated Scrum model or even a combination of these models suitable to their development environment [9]. However, GSD project managers usually have some strategies they use to build sub-teams. Project managers build autonomous sub-teams and allocate each sub-team independent architectural subsystems with well defined interfaces [15-18, 22]. We found that GSD project manager builds feature or function based sub-teams, which may be collocated or distributed. For example, highly volatile features need frequent interaction with business users and such features can be developed with a sub-team close to customers [9, 11, 14, 22].

6) Lack of Collaborative Office Environment: To support the practice of distributed Scrum meetings, each site needs good collaborative meeting facilities. Our literature review revealed that Scrum teams had difficulties in their distributed meetings due to the lack of a collaborative office environment [5]. In a number of studies, we note that for distributed Scrum team communication and collaboration processes, a dedicated meeting room with appropriate infrastructure and tool support for each site is considered necessary [24-26]. We found some strategies to support better team collaboration.

a) Single room: This practice ensures each Scrum team is allocated to a single room so that they can communicate with each other [9, 18, 20]. In this case, if a person switches teams, he or she is also relocated to the new team's room [9]. If the Scrum team is divided into multiple sub-teams, then the project manager should try to ensure that all co-located sub-teams are able to work in a single room [9].

b) Dedicated meeting room: This practice ensures that each site has a separate meeting room with all necessary network connectivity and tools for a distributed meeting [9, 11]. To make Scrum meetings visible to everyone, each site may also use a video projector [23]. In some cases, a virtual conference room can be used as a dedicated meeting room for Scrum meeting sessions [13].

7) Increased Number of Sites: Usually in any distributed software development, the greater the number of sites, the more difficult it is to manage a project. Our literature review revealed that using Scrum with a team distributed over a number of sites is more difficult because project stakeholders' distribution over multiple sites, with different time zones may restrict team communication and collaborative processes [5]. Using Scrum practices with a team distributed over more than two sites with different time zones is reported as risky [17]. Following are some of the strategies that can be used to tackle the challenges of GSD projects involving multiple sites.

a) Local Scrum team: Autonomous site-based local Scrum teams are formed and are allocated tasks with independent architectural subsystems, and well-defined interfaces [14-16, 17, 21]. The practice Scrum of Scrums is attended by a key site touch point (e.g. Scrum master) and is used to provide inter-team coordination [22].

b) Restricted team distribution: With this practice, a fully integrated Scrum team is restricted to a limited number of sites. For example, one of the studies reported a project that was distributed over multiple sites but each Scrum team was distributed between only two sites [17].

C. Framework Limitations

We do not claim that we have developed an exhaustive list of components that enable us to identify all the risks and corresponding strategies to reduce these risks, when using Scrum in GSD projects. Our conceptual framework is based on twenty primary papers identified through a SLR that addressed the use of Scrum in GSD projects. One of the main limitations of this framework is that it has a small and narrow project specific focus. However, real life GSD projects have some more important issues than particular project focus issues. In many cases, a GSD consists of a number of projects (sometimes ten or even twenty) and all projects are part of a product integration effort. Hence, a product portfolio view may be more important than a single project point of view for some GSD projects. However, practically it is very difficult and complex to identify all the

risks and the corresponding strategies to reduce these risks when using Scrum in a GSD product consisting of several projects. Because of this complexity, our research focuses on a single GSD project which can provide us with a deeper understanding of the risks and their corresponding risk reduction strategies. A project specific investigation will help us to better understand possible risks from a GSD product point of view.

IV. Summary and Future Work

There is an increasing interest in using Scrum practices for GSD. However, GSD project contextual factors can make it difficult to use Scrum in a particular project environment. We observe a gap in the literature when we investigate issues around the actual Scrum processes used in GSD projects as project distribution creates a number of significant challenges. Our objective is to identify key risks due to project contextual factors while using Scrum in GSD projects. Commonly used strategies to reduce these risks are also explored.

We have identified, through a SLR, a number of risks that may restrict the use of Scrum in GSD projects. Corresponding strategies to reduce these risks and that support the use of Scrum have also been explored. We have identified a number of risks or challenges of using Scrum in GSD projects and categorized them in different GSD issues. Based on the literature findings, we categorized seven broad classifications of identified risks which are not complete yet. We have consolidated these findings into a conceptual framework and discussed various elements of that framework. A discussion of the framework components is expected to help GSD practitioners to understand the risk factors they may need to consider while using Scrum. In addition, GSD project managers can benefit from synthesized information incorporating the strategies used to deal with these risks. However, the framework reported here cannot be described as complete in its current form, as the strength of evidence from the literature is low and the framework is constructed from only twenty research papers. In addition, our framework does not address risks from a GSD product point of view, as a product may comprise a number of GSD projects. However, we have identified a number of research issues related to using Scrum in GSD through our literature review and framework construction. Our future research will continuously modify the proposed framework based on the literature findings.

To enhance the findings of our preliminary work, we intend to conduct multiple in-depth industry-based case studies in real life settings in order to understand project based risks through a consideration of project contextual factors. Moreover, our future research will also explore effective strategies to reduce the risks identified and support the use of Scrum in GSD projects. We believe the presented framework and our future research will help GSD practitioners to understand the effective use of Scrum processes appropriate to their project development environment.

Acknowledgment

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

References

[1] E. Carmel, *Global software teams: collaborating across borders and time zones*: Prentice-Hall, 1999.

[2] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, *Agile software development methods - Review and analysis*, VTT Electronics ed.: VTT Publications, 2002.

[3] B. Ramesh, L. Cao, K. Mohan, P. Xu, "Can distributed software development be agile?" *Communications of the ACM,* vol. 49, October 2006, pp. 41-46.

[4] F. Abbattista, F. Calefato, D. Gendarmi, F. Lanubile, "Incorporating Social Software into Distributed Agile Development Environments" in proceedings of the 23rd ASE workshop, pp.46-51,2008.

[5] J. Sutherland, K. Schwaber, "The Scrum Papers: Nuts, Bolts, and Origin of an Agile Process" *http://scrumtraininginstitute.com/home/stream_download/scrumpapers, Last accessed 25 March 2009.*

[6] E., Hossain, M., A., Babar, H., Paik, "Using Agile Practices in Global Software Development: A Systematic Review," UNSW CSE Technical Report, TR 904, (2009).

[7] E., Hossain, M., A., Babar, H., Paik, "Using Scrum in Global Software Development: A Systematic Literature Review," accepted for the conference ICGSE July (2009).

[8] B. Kitchenham, S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering" EBSE Technical Report, EBSE-2007-01, 2007.

[9] M. Paasivaara, S. Durasiewicz, C. Lassenius, "Distributed Agile Development: Using Scrum in a Large Project," *Software Process Improvement and Practice*, Vol. 13, Issue 6, pp. 527-544, 2008.

[10] J. Sutherland, A. Viktorov, J. Blount, N. Puntikov, "Distributed Scrum: Agile Project management with Outsourced Development Teams" in Proceedings of the Conference on HICSS'40, pp. 274, 2007.

[11] J. Sutherland, G. Schoonheim, M. Rijk, "Fully distributed Scrum: Replacing Local Productivity and Quality with Offshore Teams," in proceedings of the Conference on HICSS'42, pp. 1-8, 2009.

[12] J. Cho, "Distributed Scrum for Large-Scale and Mission-Critical Projects," in Proceedings of the Conference on AMCIS 2007, paper 235, 2007.

[13] W. Williams, M. Stout, "Colossal, Scattered, and Chaotic (Planning with a Large Distributed Team)," in Proceedings of the Conference on Agile 2008, pp. 356-361, 2008.

[14] B. Drummond, J. F. Unson, "Yahoo! Distributed Agile: Notes from the World Over," in Proceedings of the Conference on Agile 2008, pp. 315-321, 2008.

[15] M. Cristal, D. Wildt, R. Prikladnicki, "Usage of SCRUM Practices within a Global Company," in Proceedings of the conference on ICGSE 2008, pp. 22-226, 2008.
[16] H. Holmstrom, B. Fitzgerald, P. J. Agerfalk, E. O. Conchuir, "Agile Practices Reduce Distance in Global Software Development" *Information Systems Management*, Summer, pp. 7-26, 2006.
[17] M. Vax, S. Michaud, "Distributed Agile: Growing a Practice Together" in Proceedings of the Conference on Agile 2008.pp.310, 2008.
[18] H. Smits, "Implementing Scrum in a Distributed Software Development Organization," in proceedings of the Conference on AGILE 2007, pp.371-375, 2007.
[19] B. Jensen, A. Zilmer, "Cross- continent Development using Scrum and XP" in Proceedings of the Conference on XP 2003, pp.146-153, 2003.
[20] C. Kussmaul, R. Jack, B. Sponsler, "Outsourcing and Off shoring with Agility: A Case Study" in Proceedings of the Conference on XP/Agile Universe, pp. 147-154, 2004.
[21] K. Sureshchandra, J. Shrinivasavadhani, "Adopting Agile in Distributed Development" in Proceedings of the conference on ICGSE'08,pp. 217-221, 2008.
[22] A. Danait, "Agile offshore techniques- A case Study" in proceedings of the Conference on Agile Development Conference, pp.214-217, 2005.
[23] M. Summers, "Insights into an Agile Adventure with Offshore Partners," in Proceedings of the Conference on Agile 2008, pp. 333-338, 2008.
[24] E. Therrien, "Overcoming the Challenges of Building a Distributed Agile Organization," in Proceedings of the Conference on Agile 2008, pp. 368-372, 2008.
[25] S. Berczuk, "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.
[26] P. Karsten, F. Cannizzo, "The Creation of a distributed Agile Team," in proceedings of the Conference on XP 2007, pp.235-239, 2007.
[27] M. Cottmeyer, "The Good and Bad of Agile Offshore Development," in proceedings of the Conference on AGILE 2008, pp.362-367, 2008.
[28] M. Paasivaara, C. Lassenius, "Could Global Software Development Benefit from Agile Method?" in proceedings of the Conference on ICGSE 2006, pp. 109-113, 2006.
[29] P., J., Ågerfalk, B., Fitzgerald, H., Holmström, B., Lings, B., Lundell, E. O'Conchuir, "A Framework for Considering Opportunities and Threats in Distributed Software Development," In Proceedings of the Conference on *The International Workshop on Distributed Software Development* Paris, France 29th Aug 2005 (pg 47 - 61).
[30] D., Damian, D., Zowghi, " An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations", in proceedings of the 36th Hawaii Conference on System Sciences (HICSS'36), 2003.
[31] D., Smite, C., Wohlin, R., Feldt, T., Gorscheck, "Reporting Empirical Research in Global Software Engineering: a Classification Scheme," in Proceedings of the Conference on ICGSE 2008, pp. 173-181, 2008.
[32] B., Sengupta, S., Chandra, V., Sinha, " A Research Agenda for Distributed Software Development," in proceedings of the Conference on ICSE, pp. 731-739, 2008.
[33] J. D. Herbsleb "Global Software Engineering: The Future of Socio- technical Coordination" in proceeding of the Conference on Future of Software Engineering, FOSE, pp.188-298, 2007.
[34] P. J. Ågerfalk, B. Fitzgerald, "Flexible and distributed software processes: Old petunias in new bowls?" *Communications of the ACM,* vol. 49, Oct 2006, pp. 26-34.
[35] J., Ralyte, X., Lamielle, N., Arni-Bloch, M., Leonard, "Distributed Information Systems Development: A Framework for Understanding and Managing," *International Journal of Computer Science and Applications*, Vol. 5, No. 3b, pp. 1-24, 2008.