Global requirements engineering on the cloud: PhD research proposal

Global Requirements Engineering on the Cloud

PhD Research Proposal

Sajid Ibrahim Hashmi
Lero-the Irish Software Engineering Research Centre
University of Limerick, Ireland
Sajid.Hashmi@lero.ie
Advisor: Dr. Ita Richardson; Lero, University of Limerick {Ita.Richardson@lero.ie}

***Abstract*— Global software development is a business model that involves software development distributed beyond national boundaries. However, distributed nature of the processes makes it very challenging to communicate and collaborate. Requirements engineering is an intensive software development life cycle activity and involves frequent communication among the stakeholders. In global software development, tight project schedules and global distance give rise to incomplete requirements handovers from one site to another. Therefore, the need for an efficient mechanism becomes inevitable as information available to one project team can often contradict what is available to the other.**

On the other hand, Software as a Service (SaaS) is one of the deployment models of the cloud that can provide multiple users with a web space to collaborate on things of mutual interest. In this research, we propose a SaaS based mechanism to facilitate globally distributed software development teams working on the requirements engineering process. Our emphasis is on the situation that occurs after requirements are handed to another software development site.

My Research Areas, Keywords — global software development (GSD); global requirements engineering (GRE); software as a service (SaaS) clouds

I. INTRODUCTION

GSD deals with teams dispersed across multiple geographical locations to carry out and accomplish software development tasks [2]. Organizations expect to benefit from enhanced business value through advantages such as round the clock software development, availability of skills and labour, and a reduction in overall project costs. This kind of development model emphasizes the need for communication among the development teams, which is good for software development but at the same time highlights challenges, as communication among GSD teams remains an issue [1].

A typical software project involves technical as well as non-technical development processes. Technical processes are the engineering processes within software project development phases such as requirements analysis, design, coding, and testing and their sub activities. Non-technical processes are those processes which are non-engineering, but are vital to the support of technical processes. Examples of non-technical process are *communication* and *coordination*. Due to distance and time zone issues, both types of processes would be equally important for GSD.

II. PROBLEM STATEMENT

In GSD, the situation gets worse when teams are located in different time zones. Moreover, distributed requirements engineering has been a problem area [4][8][9], and a key challenge when carrying out GSD projects is how to effectively communicate and collaborate when requirements are handed over from one software development team to the other one [3]. This requires an effective requirements engineering process especially when the teams are in different time zones. Our review of research literature a reveal that the teams involved in GSD projects often appoint one of their colleagues to work beyond the normal working hours to answer queries from the other team. The strategy may work well for certain situations but the use of natural language to discuss clarity on requirements is likely to be ambiguous [6].

III. RESEARCH METHODOLOGY AND OBJECTIVE

Our research question is how we can facilitate globally distributed requirements engineering process. In order to perform this research, we carried out a literature review and investigated the existing communication mechanisms and their limitations of coping with requirements engineering phase of GSD. Within this, we first examined alleviation of oral communication so that the teams in different time zones can be facilitated. Second, the proposed mechanism should facilitate a requirements validation and verification mechanism which accounts for missing and caters for conflicting situations.

In addition, industrial interviews were conducted to collect data on experiences and problems specific to the topic under discussion. The emphasis was on investigation of GRE process related barriers. Based on the deficiencies with the existing mechanisms in GRE, we propose a methodology that we argue can deal with the issue under discussion. The objective of this research is threefold: first, it focuses on the need for communication during requirements handover while addressing deficiencies with the existing communication mechanisms being used for GSD. Second, it highlights situations where incomplete requirements could be handed over. Third, it proposes a SaaS cloud based mechanism to facilitate GRE process that aims to incorporate requirements validation and verification as well.

TABLE I. CASE STUDY FINDINGS

Interviewee Role(s)	Questions Asked	Details	Recommendations
Business consultant, onsite with client – involved with requirements gathering and to lead up the configuration	What was your project specific role? What were the issues you faced while working with distance team? What practices went well and what would you recommend?	It could be the time zone that is a big issue; we had to work out of the hours to facilitate others. Human related issues were involved in communication.	Central development group offsite did not have a development process, but we worked out a specific process would work for us the best. It is really important to have that. Moreover, we instigated communication with the offsite team and configured processes within the application.
Software practitioners – offsite (in a different time zone)	What was your specific role? Did you face any challenges with the time zone issue? What practices went well and what would you recommend? How you would have liked to receive requirements?	Lack of documentation caused ambiguities. Due to the time zone issue, things took time before get corrected by the onsite team, software development was delayed as a result.	Do proper documentation and analysis and determine what needs to be developed. Requirements related artifacts should be developed upfront. We would want requirements to be well thought out, easy to understand, so that with a quick handover, you would be good to go. Requirements in a structured format would have saved the time.
Technical Consultant, onsite with client - Moreover, was involved in other capacities as well, like requirements and business consultancy	What was your specific role in the project? What type of the development process you were following? What practices went well and what would you recommend? Advantages and disadvantages of working with people from different cultures?	In addition to the time zones, there was other communication overhead as well. As the communication between the client and the practitioners (offsite) was going on through us	Requirements document should depict the technical details as well, like how a specific requirement could be implemented. Initially, developers had problems understanding our documents because those outlined in a high level. You have to make sure that your requirements documents contain sufficient level of details.
Business consultant – onsite with client	-Could you give an overview of the project and your specific role? -What type of the development processes you were following? -What practices went well and what would you recommend? What issues did you face?	The Process was not well documented; Communication between the two sites was a problem due to different time zones.	In addition to the requirements artifact documents, what the offsite practitioners really wanted was a high level solution. We generally understood that GRE process is a challenging thing. The issues would be with the documents and we had to solve those verbally.

IV. RELATED WORK

Lack of research studies and their limitations motivated us to perform this research work. Research on requirements engineering of GSD projects have mainly been through empirical studies justifying the significance of the domain [8][9]. The results of those studies confirm the GRE problems but do not provide solutions to the issues that cover technical as well as non-technical facets. Moreover, the existing work on the domain mainly focuses on early phase activities such as requirements planning and elicitation. [10] Have listed some measures which enhance collaboration through emails, instant messaging, and screen sharing but its usefulness cannot be established during requirements handovers. Some industry based tools [11] have also been proposed to facilitate collaboration among software development teams. However, those facilitate task level collaboration only and are mainly focused on the software development phase of the life cycle, i.e. keeping track of the task level audit trail in addition to a basic collaboration environment. The use of wikis [12] has also been instrumental in software engineering but wikis have their own challenges and limitations as they have mainly been used to store and retrieve documents or to allocate tasks. In short, we have not found any proven methodology or automated technique that could facilitate GRE process.

V. INITIAL RESULTS

The proposed methodology is based on the findings of a case study of FiSCo we under took in order to find real industrial problems in terms of GRE and the factors it could be influenced by. The company had a very good track record of successfully delivering business value to their clients. Employees from two different GSD projects were interviewed. The Interviews were of different duration and lasted around 1 and 2 hours. We asked several questions from the participants (some of the related ones are listed in Table I). The sample included project managers, technical development leads, solution architects, business consultants, and software practitioners. The objective of this exercise was to augment the existing literature findings on the domain with input from the practitioners. An analysis of those interviews revealed several processes, people, and business related issues. In total we interviewed 24 professionals, Table I list down a few with

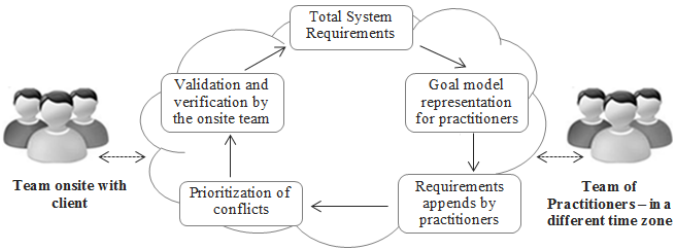

Figure 1. An Overview of the SaaS based Proposed Process

multiple roles in order to support our claim for GRE being a problem area. As part of the business model, the organization would appoint a team onsite with client to elicit project requirements but as mentioned above, the teams faced problems in order to cope with the situation (Table I).

Figure 1 shows our proposed methodology. In order to make representation of requirements simple to comprehend, we opt for a goal based methodology [7]. This entails representing requirements in the form of goals that are easy to manage. The idea is to goal model the requirements handed over to the team of software practitioners. Since requirements are transferred from one development site to another, the conflicts are likely to be established once the software practitioners input to the system requirements. Goal model representation would allow the teams to identify the missing as well as the conflicting requirements. Moreover, it will allow them more visibility into the system and they can append the goal models with the requirements which went missing earlier. The team with the client has to verify and validate the additions before the practitioners' team could proceed with the implementation.

VI. Expected Results

Previously [5], we developed the idea that GSD processes could be facilitated through the Cloud paradigm. Moreover, in [3] we figured out GRE as a very challenging process phase within GSD. The reasoning we presented in [5] made us come up with the SaaS clouds having potential to address the requirements engineering related issues of GSD projects.

As part of the future work, this research proposal will be built as workflow on top of the cloud. We aim to use SaaS rather than a simple web based architecture as its use will ensure that a collaborative *as you go* space is provided to the software development team members, the scalability feature ensures that it can handle variable number of communication work flows and it can also provide *as you go* storage of requirements artifacts.

VII. Evaluation

To demonstrate the applicability and usefulness of the work, concrete validation of our project should be a two-step procedure. First, we will extend the case study and incorporate some additional GRE related issues and the causal factors through an in-depth study. Second, look into the possibilities of incomplete requirements handovers using a real time scenario. Third, SaaS-based formalization of the proposed methodology for requirements completeness and validation so that requirements trade-offs can be made on the basis of a well-defined criterion. That will assist team members with their decision making on the *to be implemented requirements* for an optimal solution.

VIII. Acknowledgement

I want to thank Dr. Sarah Beecham for her support of this project. This research has received funding by Science Foundation Ireland grant 10/CE/I1855 to Lero (www.lero.ie), a Higher Education Authority grant PRTLI 4 to the Lero Graduate School in Software Engineering, and from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

IX. References

[1] I. Richardson, V. Casey, F., Mc-Caffery, J. Burton, S. Beecham, "A process framework for global software engineering teams," *Information and Software Technology (IST)*, Elsivere, Vol .54, Issue 11, 2012, pp. 1175 – 1191.

[2] J. D. Herbsleb, "Global software engineering: the future of sociotechnical coordination," In *proceedings of the Future of Software Engineering* (FOSE'07), 2007, pp. 188-198.

[3] S. I. Hashmi, F. Ishikawa, and I. Richardson, "A communication process for global requirements engineering," International Conference on Software and System Process (ICSSP), 2013, pp. 136-140.

[4] D. E. Damian, D. Zowghi, "RE challenges in multi-site software development organisations," *Req. Eng. J.* 8(3): 149-160 (2003).

[5] S.I. Hashmi et al., "Using the Cloud to facilitate global software development challenge," ICGSE Workshop on *Tool Support Development and Management in Distributed Software Projects*, 2011, pp. 70-77.

[6] F. Chantree, B. Nuseibeh, A.D. Roeck, A. Willis, "Identifying nocuous ambiguities in natural language requirements," In *Proc. of the IEEE Int. Req. Eng. Conf.* (RE), 2006, pp. 59-68.

[7] A.V. Lamsweerde. Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons, Chichester (2009).

[8] F. Calefato, D. Damian, and F. Lanubile, "Computer-mediated communication to support distributed requirements elicitations and negotiations tasks," *Empirical Software Engineering,* 17(6): 640-674 (2012).

[9] D. Damian, D. Zowghi , L. Vaidyanathasamy, Y. Pal, "An industrial case study of immediate benefits of requirements engineering process improvement at the Australian center for Unisys software," *Empirical Software Engineering*, Vol. 9, No. 1-2, March 2004, pp. 45-75.

[10] Li-Te Cheng, C. R. B. de Souza, S. Hupfer, J. Patterson, S. Ross. –T, "Building collaboration into IDEs," *ACM Queue*, 1(9):40–50, 2004.

[11] www.ibm.com/developerworks

[12] P. Louridas, "Using Wikis in Software Development,". IEEE Software, Volume 23, Issue 2, 2006, pp. 88-91.

[13] A.V. Lamsweerde. Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons, Chichester (2009).