DISTRIBUTED REQUIREMENTS SPECIFICATION: MINIMIZING THE EFFECT OF GEOGRAPHIC DISPERSION[i]

Leandro Lopes, Rafael Prikladnicki, Jorge L. N. Audy
School of Computer Science, PUCRS, 6681 Avenida Ipiranga, Porto Alegre, Rio Grande do Sul, Brazil

Azriel Majdenbaum

Keywords: requirements engineering, requirements specification, distributed software development, global software development

Abstract: Requirements specification is an important phase of the requirements engineering area in the software development process. In geographically distributed environments, this phase becomes critical due to the characteristics of the distributed development (physical and temporal distance, cultural differences, trust, communication, etc). The objective of this paper is to analyze the requirements specification in geographically distributed environments, identifying the main challenges and proposing a process to minimize the impacts of this scenario. The results are based on a case study carried on a multinational organization that has software development units in multiple countries, and was recognized as a SW-CMM level 2 organization in 2 of them. The results suggest the necessity to adapt the requirements specification phase to the distributed software development environment, addressing the main existing challenges. The problems and the solutions adopted are presented, aiming to relate these solutions to the organization distribution level, considering where the project team, users and customers are located.

1 INTRODUCTION

Software development has become part of business globalization. This is mainly due the need for cost reduction, increased competitiveness and the possibility to share resources on a global scale (Herbsleb, 2001). As a consequence, communication between the project team, users and customers occurs in a geographically distributed way.

Requirements engineering, recognized as a critical phase in software engineering, presents several new difficulties, as well as the rise of fundamental difficulties when in distributed environments (Damian, 2002; Zowghi, 2002).

This study has as objective to understand what kind of problems the project teams face during requirements engineering and how these problems have been addressed. With this objective, a case study was conducted in a multinational organization with software development units distributed globally, identifying the difficulties and proposing a solution to address the problems identified. The results are analyzed and the existing challenges are identified. Some of the solutions that are being implemented with the objective of minimizing the problems found are presented. This paper has the following structure: section 2 presents the theoretical base; section 3 describes the research method; section 4 describes and presents results of the case study; section 5 presents the process proposed to minimize the problems identified; section 6 present the conclusions, future studies and the research limitations.

2 THEORECTICAL BASE

2.1 Distributed Software Development

As part of the globalization efforts currently pervading society, software project teams have also become geographically distributed on a worldwide scale. Many companies are distributing their

Lopes L., Prikladnicki R., L. N. Audy J. and Majdenbaum A. (2004).
DISTRIBUTED REQUIREMENTS SPECIFICATION: MINIMIZING THE EFFECT OF GEOGRAPHIC DISPERSION.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 531-534
DOI: 10.5220/0002640205310534

software development process in countries such as India, China, Singapore, Russia and Brazil. Frequently this process also occurs inside a country, particularly in regions with tax incentives or critical mass in some skill or resource areas. This characterizes the Distributed Software Development (DSD) or Global Software Development (GSD) when the distribution becomes global.

Despite the benefits, GSD is one of the biggest business-oriented challenges that the current environment presents under the software development process point of view (Herbsleb, 2001; Carmel, 1999, Prikladnicki, 2002).

2.2 Requirements Engineering

Requirements engineering (RE) plays an important role in the software development. A requirement is the condition or capacity that a system that is being developed must satisfy (Oberg, 2000). Therefore, the compliance with requirements determines the success or the failure of a software project.

The steps of requirements engineering (Pressman, 2001; Sommerville, 1997) are: requirements elicitation, requirements analysis and negotiation, requirements specification, system modelling, requirements validation and requirements management.

IEEE recommends the use of a Software Requirements Specification (SRS) as the basis for agreement between customers and suppliers on what the software product do (IEEE, 1998). The SRS is also fundamental to cost and schedule estimation. Some other artefacts can be linked to requirements engineering process, like Vision/Scope in RUP (Rational Unified Process) (Kruchten, 2001).

3 RESEARCH METHOD

This research is characterized as a study mostly exploratory, since the main research method was the case study (Yin, 1994). The research has two main phases (Figure 1).

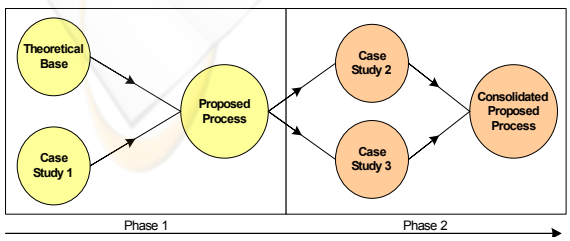

Figure 1. Research Method

Phase one is already completed and the results are presented in this paper. Besides the theoretical base, a case study was developed aiming to identify the main challenges of requirements specification in DSD and propose a process to address those challenges. In phase two, we intend to apply the proposed process in 2 case studies and, based on the results, formalize a consolidated proposed process.

4 CASE STUDY

The study was developed in a global software development unit located in Brazil and owned by a multinational organization. The purpose of the case study was to analyse two projects developed in the organization, aiming at the identification of problems, advantages and disadvantages considering the requirements specification in both projects in a geographically distributed context.

During the project development, considering the requirements engineering, many observations were done and interviews with the Program Managers were performed. After the analysis of these two projects, it can be concluded that the requirements specification in global software development can become an arduous task if the process are not well defined and if the teams are not previously prepared to work in this scenario.

The categories identified in this study are culture, communication, knowledge management and technical aspects, as shown in Figure 2.

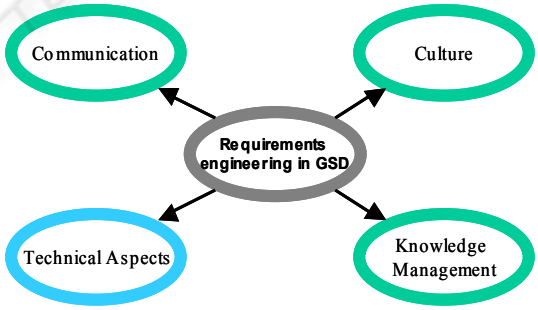

Figure 2. Categories related to requirements engineering in GSD

Each of these categories has several factors involved and the relationship between them is close, what makes it difficult to define the limits of each one. The main factors related to each category were also identified, and are presented in sequence.

The requirements engineering process is highly dependent on communication. Clear communication is critical to avoid misunderstandings and conflicts. The core factors related to communication found in this study are language, time zone and communication medium.

Requirements engineering in distributed environments is directly influenced by cultural differences. Both, organizational and national culture can affect requirements engineering. The main factors related to culture are context, attitude and values.

The requirements engineering process deals with large volume of information. Collect, process, store and make available the knowledge related to the requirements process, as well as unify the organizational vision are needs that should be addressed with knowledge management. The main factors identified related to knowledge management are expectations, awareness and management of cultural information.

Several technical aspects affect requirements engineering in distributed environments. The requirements process depends on coordination and control mechanisms, for example, that can help to reduce the impact caused by team distribution. The main factors found are patterns, process and configuration management.

The need for a requirements process that address distributed development issues was already identified (Zowghi, 2002). This study conducted in the same direction. It was found several factors that impact requirements engineering in distributed environments. It is clearly necessary to create new or adapt current process to address these factors. In next section it is presented a proposal of requirements process based on the challenges found.

5 PROCESS PROPOSAL

This process proposal is based on one common distribution and division of roles of global software development with specification and development teams. The specification team is responsible for conduction the elicitation, analysis, negotiation, specification and validation of requirements with stakeholders. The development team is responsible for modeling software based on the specification and maintaining requirements traceability with code and test.

It is important to recognize that beyond distribution between specification and development teams, it is possible to have distribution between the specification team and users or clients, what affects the requirements engineering phases related to those groups (Damian, 2003; Lloyd, 2002).

Another important problem commonly faced by organizations that develop software for more than one geographically distributed (internal or external) client is the diversity of structures, patterns and level of detail used in requirements documentation. Once there are several different clients, the diversity leads to difficulties in obtaining metrics, estimative and organizational standards.

5.1 Process

The proposed process intends to reduce distribution difficulties in requirements process by introducing tasks of adaptation and understanding the SRS in development team. It is composed of five steps, as presented in Figure 3 and described below.

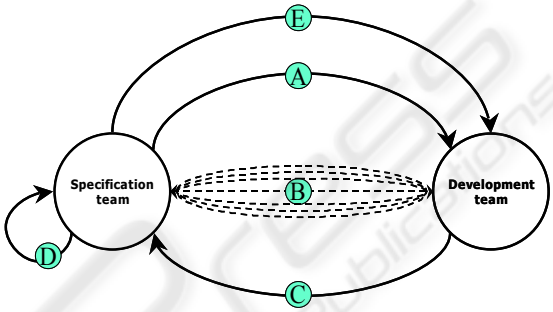

Figure 3. Proposed process

A. SRS first version is concluded and sent to development team.

While executing tasks of elicitation, negotiation and specification, the specification team writes the first versions of SRS. After concluding these tasks the document is sent to development team for adaptations.

B. SRS analysis and adaptation by development team.

After receiving SRS from specification team, the first action of development team is try to deep understand the requirements and its context. During this phase the SRS is adapted to reduce potential sources of problems. Ambiguity and lack of clearness are likely between cultures and languages. In cases of great difficulties, SRS can be completely rewritten.

The process of adapt or rewrite the SRS permits the development team to get a larger amount of information than the explicitly written in the document. Several questions arise and shall be cleared with specification team. In this step the SRS can be also be adapted for standardization.

Communication between team is heavy during this step. Beyond clearing information, communication occurs to build agreements, simplifying step D.

C. SRS adaptation is concluded. SRS is sent to specification team approval.

Once the SRS is completely adapted, the new document must be approved. In this step the

document is sent back to be verified by specification team.

D. SRS validation and approval by specification team.

Specification team shall verify SRS to assure that after adaptation it still reflects the needs and objectives of stakeholders. Communication occurred in step B maintains specification team aware of the adaptation process, reducing the effort need to validate SRS.

E. SRS Final version is defined.

After the approval by specification team, the final version of SRS is defined as approved SRS. Then, development team uses this version as basis for modeling, coding and testing software.

6 CONCLUSION

Requirements engineering has been considered a critical area in software development. Its study in distributed software development environments offers excellent research opportunities once it is a new area, growing fast. New techniques and processes are clearly needed.

This research presents evolutions towards a RE process to distributed software development, once it identifies challenges of requirements engineering in distributed software development and proposes a process to address part of those challenges. The process model proposed address the challenges identified in section 4.

Communication issues due to language are addressed during SRS adaptation (step B). Problems with time zone and communication media tends to arise during the process, but as development team increases its comprehension of requirements, problems tend to reduce considerably.

Cultural issues in requirements like context and values tend to be reduced when SRS is adapted. However, teams must be aware of cultural differences while communicating to avoid conflicts.

Knowledge management is partially addressed through the use of common documents and process, once teams have common expectations and are aware of each other roles.

Technical aspects are addressed through the process definition. Patterns like phrase and document structure can be applied during adaptation. The negotiation and definition of a common final SRS enhances software configuration management.

Moreover, another important contribution of the process is helping to achieve SRS standardization in the beginning of development process, concerning to phrase structure, patterns, and level of detail, for instance.

The next research step involves the validation of the proposed process. We intend to apply the process in three global projects to investigate its effectiveness in distributed requirements engineering. To do this, we are beginning empirical studies in two multinationals of information technology.

REFERENCES

Herbsleb, J., Moitra, D., 2001. *Global Software Development*, IEEE Software.

Damian, D., 2002. The study of requirements engineering in global software development: as challenging as important, In *ICSE 2002 - International Workshop on Global Software Development.* Florida, USA.

Zowghi, D., 2002. Does Global Software Development Need a Different Requirements Engineering Process?, In *ICSE'2002 - International Workshop on Global Software Development*, Orlando, Florida, USA.

Carmel, E., 1999. Global Software Teams – Collaborating Across Borders and Time Zones. *Prentice Hall*.

Prikladnicki, R., Peres, F., Audy, J., Móra, M., and Perdigoto, A., 2002. Requirements specification model in a software development process inside a physically distributed environment, In *ICEIS'2002,* Ciudad Real, Spain.

Oberg, R., Probasco, L., and Ericsson, M., 2000. Applying Requirements Management with Use Cases, *Rational Software White Paper*, Cupertino, CA.

Pressman, R. S., 2001. *Software Engineering: A Practitioner's Approach.* 5th Edition.

Sommerville, I., Sawyer, P., 1997. Requirements Engineering – a good practice guide. *Wiley*.

IEEE, 1998. IEEE Recommended Practice for Software Requirements Specification, Std 830-1998. *IEEE Computer Society.* New York, USA

Kruchten, P., 2001 The Rational Unified Process: An Introdution. 2nd Edition. *Addison Weasley*.

Yin, R., 1994. Case study research: design and methods. *Sage*.

Damian, D., Zowghi, D., 2003. An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations. In *HICSS'03 - 36th Hawaii International Conference on Systems Science.* Hawaii, USA.

Lloyd, W., Rosson, M., Arthur, J., 2002. Effectiveness of elicitation techniques in distributed requirements engineering. In *RE'02 - IEEE Joint International Conference on Requirements Engineering*.