

# Does Global Software Development Need a Different Requirements Engineering Process?

Didar Zowghi  
Faculty of Information Technology,  
University of Technology, Sydney,  
P O Box 123 Broadway  
NSW 2007 Australia  
+61 2 9514 1860  
[didar@it.uts.edu.au](mailto:didar@it.uts.edu.au)

## ABSTRACT

Global software development exhibits certain features that make it fundamentally different from traditional co-located software development. These characteristics have an impact on the requirements engineering process. The traditional requirements engineering processes do not cater for the peculiarities of developing software over distance and across different time zones with different cultural boundaries. This paper advocates the development of a different RE process for global software development outlining some preliminary suggestions on what such process model would include.

## 1. INTRODUCTION

Requirements Engineering (RE) plays a very important role in software development. For some years now, it has been recognized that problems associated with requirements engineering are among the major reasons for software project failures where the end product does not meet the real needs of the problem owners [1][2][3]. Errors in requirements specifications can have a major impact on software costs. One estimate given states that 40% of requirements need rework during the course of the software development project [4]. It is evident that early detection and correction of potential problems during requirement stage may alleviate many much larger problems later on during testing and maintenance. Despite this, in comparison with the rest of software development activities, relatively little time and effort is expended during the requirements phase. Furthermore, while much research has been conducted for supporting, improving and automating many of the different activities making up the software development process, elicitation, analysis, specification, validation and management of requirements remain one of the least explored and have the least satisfactory scientific foundations.

A process is an organized set of activities that transforms inputs to outputs. It is difficult to write down a sequential plan of activities that adequately describes the RE process as suggested by activity-oriented approaches to process modelling such as the Waterfall model. It is generally believed that RE has its own life cycle. In the RE literature many different definitions have been given for the RE process as well as for the activities that take place during

that process. In some cases, RE process is defined at a very fine level of details and the steps in the process must be carried out exactly as described [5]. However, this form of process description usually applies to very simple processes, for more complex processes, the description is usually less detailed and it is up to the person or team who are executing the process to carry it out (enact the process) in their own environment.

We have recently investigated RE practices in global software development through conducting a field study in a geographically distributed organisation [6]. We have found that some of the fundamental problems associated with various activities making up the requirements engineering process are exacerbated when the software development teams are geographically distributed [6]. This is due to the fact that global software development (GSD) possesses certain distinguishing characteristics that challenge and directly impact the RE process activities. According to Carmel [7] the distinguishing features of global software development teams are *distance*, *time-zone differences*, and *cultural differences*. In this paper, I briefly discuss the impact of these characteristics on the RE process and argue that because of their impact on RE and hence on software development there is a need to fully investigate and develop RE processes that specifically support and are tailored to global software development.

## 2. Impact of GSD on RE Process

RE is one of the most important phases because it is at this stage where purpose, functionality and boundaries of the software are meant to be fully identified, analysed and defined. RE is also the most *communication* rich activity of software development since it is at this stage of development that most of the interactions between the problem owners and problem solvers take place. Geographical *distance* between development sites has a direct impact on all forms of communication. That is, communication is usually less frequent and more constrained [7]. If the customers and developers are not co-located, clearly distance impacts the communication between the problem owning and problem solving communities who are the two essential groups of stakeholders in RE. Global software teams often deal with distance through using different media such as video conferencing, phone, email and groupware. It is important to identify the differences between communication during requirements engineering and other stages

of software development. Why is using email alone (for example) not effective enough to facilitate the necessary communication? The issue here is not just facilitating asynchronous communication among stakeholders but ongoing management of the discussions and ultimate requirements decisions made by email (i.e. the management of the RE process as well as RE product). Given the communication intensive nature of RE process, it is surprising that none of the existing RE process models (e.g. [8][9][10]) make systematic or rigorous mention of synchronous or asynchronous tools in their process descriptions. Clearly, there is a need to include tool support in RE process to global software teams when developing requirements. Such tools not only provide the usual traceability and record keeping functionality that the commercially available requirements management tools<sup>1</sup> offer but are specifically suited to integrate well with the type of tools that are commonly used by global software teams. The expectation from such an integrated toolset would be a seamless flow of requirements information among stakeholders that are geographically distributed.

Distance also affects all levels of team *coordination* and *control*, which are the main ingredients for effective team management. Usually, managing software teams depends little on controls and commands, relying instead on informal means of coordination (such as face to face contact), which is very communication intensive [7]. Furthermore, managing the RE process is hard enough when development teams are co-located let alone when they are geographically dispersed. During RE process the participants have to work towards coming up with a common vision of the requirements for the system to be built. Distance not only slows down this process but also prevents congruence to be achieved by consensus and consultation.

Another challenge for RE that is impacted by distance is knowledge management. The sheer volume of requirements information from multiple sources needs to be shared with all the stakeholders. Much of this form of knowledge remains tacit and undocumented particularly at earlier phases of requirements engineering process such as elicitation, negotiation and prioritisation. This problem is magnified further when stakeholders are geographically distributed and are not able to communicate informally with one another in order to pass on the knowledge. Once again, the existing RE processes do not take this factor into account in order to develop RE techniques and methods that would address the issue of knowledge management over distance.

*Time differences* are known to exacerbate the communication related issues especially when there is no natural overlap of working hours for voice or video conversations [7]. In this way, one side almost always has to compromise. Time difference affects elicitation, negotiation and prioritisation activities of RE process. It hinders appropriate participation from stakeholders such as system users and field personnel. Speed is regarded as one of the critical success factors in information technology and is of major concern in global software development [11]. RE process is often considered to be time consuming because of its total reliance on communicating the requirements from one group of people to the other. Time differences thus introduce a further

delay in requirements negotiation and prioritisation. Once more, traditional RE processes (for nonglobal software development) obviously do not pay particular attention to this issue. In many organizations project planning and management is often linked with the RE process since all software development projects commence with RE activities. Hence a new RE process for global software development must accommodate different time zones in such a way that managers can then capitalise on the information provided in the RE process for their project planning.

Finally, issues related to *cultural difference* have been partially addressed in the RE literature (see for example [12][13][14]). Macaulay [12] argues that any RE process model depends to a large extent on the customer-supplier relationship. This relationship becomes even more important when we deal with differences in national and ethical traditions, customs, norms of behaviour, as well as language. It has been observed [7] that cross-cultural teams (as opposed to a more uniform cultural groups) have more potential for productivity as well as more potential for problems. Moreover, differences between functional cultures in the same organization (i.e. marketing versus developers versus engineers) often cause major problems in RE process. This is mainly due to lack of effective communication and lack of trust between diverse groups (with different backgrounds) in organisations. These are the so-called “soft” issues of requirements management that mostly necessitate changes in organisational and project culture. These issues cannot be adequately addressed by computer science or software engineering research alone and should more appropriately be studied using research methodologies and theories originating from management and social sciences discipline. There is, however, an opportunity to investigate these issues with respect to global software development and propose a set of guidelines to alleviate these problems in practice. In contrast there is much more scope to do something about providing tools, methods and techniques to assist practitioners in more effective management of requirements artifacts and the RE process over distance.

### 3. Conclusion

In this paper I have briefly discussed the impact of global software development over RE process and identified some of the issues that can be addressed by investigating and devising a new RE process model for geographically distributed requirements engineering. Because social issues are at the heart of many of the problems in RE and because they cannot solely be addressed by the currently available technical methods, novel approaches and paradigms need to be sought. Furthermore, the relative immaturity of the RE field suggests an eclectic approach to RE process issues especially those related to distance, communication and coordination problems. The answer to the question: “Does Global Software Development Need a Different Requirements Engineering Process?” is clearly “yes”. To this end, we are commencing a research project to develop an integrated RE tool environment that addresses the challenges of communication and coordination identified in [6] with an existing industry partner who is engaged in global software development. Another two related important question that needs to be investigated are: “Should we really be doing distributed RE?” and “Does the cost of doing RE in distributed settings is justifiable?” Many geographically distributed systems development organizations try

---

<sup>1</sup> Tools such as DOORS or Rational’s RequisitePro etc.

to elicit and specify requirements in one location where the customers physically are using the traditional face-to-face meetings. They then pass the specifications to the rest of the development team to be designed and implemented in a different location. What is not clear, however, is that how does the organization manage the requirements throughout the software development lifecycle. To be able to answer these questions and address any of the related issues that is influenced by GSD further research is needed.

## REFERENCES

- [1] Jackson, M., The meaning of requirements, the annals of software engineering, special issue on requirements engineering, 3:5-21, 1997.
- [2] Sawyer, P., Sommerville, I., and Viller, S., Improving the requirements process, Proceedings of the 4<sup>th</sup> International workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'98), Pisa, Italy, July 1998.
- [3] Thayer, R.H., and Dorfman, M., Software requirements engineering, IEEE Computer Society Press, Second Edition 1997.
- [4] Hutchings, A., and Knox, S., Creating products customers demand. Communications of the ACM, 38(5), 1995.
- [5] Zowghi, D. A logic-based framework for the management of changing software requirements, PhD thesis, Macquarie University, Sydney, Australia 1999
- [6] Damian, D. E., Zowghi, D. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, Department of Software Engineering Technical Report, 2002.1. University of Technology, Sydney, Australia.
- [7] Carmel, E. Global software teams, Prentice Hall, 1999
- [8] Kotonya, G., and Sommerville, I., Requirements Engineering, processes and techniques, Wiley, 1997
- [9] Houdek, F., and Pohl, K., Analysing requirements engineering processes, a case study, Proceedings of the 2nd International Workshop on Requirements Engineering Process (part of DEXA 2000), England, September 2000.
- [10] Zowghi, D., A requirements engineering process model based on *Defaults* and *revisions*, Proceedings of the 2nd International Workshop on Requirements Engineering Process (part of DEXA 2000), England, September 2000.
- [11] Herbsleb, J., Mockus, A., Finholt, T.A. and Grinter, R.E. An empirical study of global software development: distance and speed, *International Conference on Software Engineering*, Toronto, 2001
- [12] Macaulay, L. *Requirements Engineering*, Springer, 1996
- [13] Al-Rawas, A. and Easterbrook, S. Communication problems in requirements engineering: a field study, *Proc. of Conf. on Prof. Awareness in Software Engineering*, London, 47-60, 1996
- [14] Jirotko, M., and Goguen, J., Requirements Engineering social and technical issues, Academic Press, 1994.