An empirical study on Global Software Development: Offshore Insourcing of IT Projects

Rafael Prikladnicki, Jorge L. N. Audy, Roberto Evaristo
School of Computer Science, PUCRS, Porto Alegre, Brazil; University of Illinois, Chicago, USA
rafael@inf.pucrs.br, audy@pucrs.br, evaristo@uic.edu

Abstract

The objective of this paper is to present lessons learned from a case study conducted in a Brazilian software development unit owned by a multinational organization. The focus of this study is to understand the factors that enable multinationals and virtual corporations to operate successfully across geographic and cultural boundaries. Since the number of organizations distributing their software development processes worldwide keeps increasing, this change is having a profound impact not only on marketing and distribution but also on the way products are conceived, designed, constructed, tested, and delivered to customers. Our results show empirical results towards the identification of problems the organizations involved in offshore insourcing of IT projects have faced when going global.

1. Introduction

Software has become a vital component of almost every business. Success increasingly depends on using software as a competitive advantage [1]. More than a decade ago, many organizations began to experiment with remotely located software development facilities (also called Distributed Software Development - DSD) seeking lower costs and access to skilled resources. Economic forces are relentlessly turning national markets into global markets and spawning new forms of competition and cooperation that reach across national boundaries [2].

This change is having a profound impact not only on marketing and distribution but also on the way products are conceived, designed, constructed, tested, and delivered to customers. For these reasons, DSD has attracted a large research effort in software engineering (i.e., [1]; [2]; [3]; [4]; [5]; [6]). The search for such competitive advantage forces organizations to search for external solutions in other countries, and foster the Global Software Development (offshore sourcing). This epitomizes the traditional problems and the existing challenges.

The two main options currently under use include offshore outsourcing (contracting services with an external organization located in another country) as well as offshore insourcing (contracting with a wholly owned subsidiary also located in another country). The first has become fairly common, but difficulties abound in trying to develop a relationship with an unknown foreign partner that is time and geographically distant. Such issues have led select organizations to create their own software development centers in countries like India, Russia, Brazil, Ireland, etc. Although offshore insourcing bypasses some of the tough contracting difficulties found in organizations that are involved with traditional offshore outsourcing, a whole different set of issues is created. And this is what we are trying to address in this paper.

This paper focus on problems that organizations (specifically those involved in offshore insourcing) have faced when going global in software development and how these problems have been addressed. The research question can be defined as: What are the main issues related to the performance of IT projects when developed in an offshore insourcing environment, and how each issue can be addressed?

In order to answer the research question, a case study was conducted identifying some of the difficulties, and solutions involved. This research has as purpose to explore the main issues found in the case study, looking for improvements in projects being developed in this environment. The results are analyzed and the existing challenges identified. Our contributions are the lessons learned from the case study.

This paper has the following structure: section 2 presents the theoretical base; section 3 describes the research method; section 4 describes the case study; section 5 discuss the results found in the case study and presents the lessons learned; section 6 presents the conclusions, future studies and the research limitations.

2. Theoretical Base

2.1. Global Software Development (GSD)

As said by Pressman [7], software process is defined by a set of activities, methods, practices and technologies that people and companies use to develop and to keep related software and products. The interest in the software process is based on the following premises:

- The software quality is strongly dependent on the quality of the process used in its preparation;
- The software process can be defined, managed, measured and improved.

However, even using a well-defined development process it is not a simple task to develop software. As part of the globalization efforts currently pervading society, software project teams have also become geographically distributed on a worldwide scale. This characterizes Global Software Development (GSD).

Organizations search for competitive advantages in terms of cost, quality and flexibility in the area of software development [8], looking for productivity increases as well as risk dilution [9]. Many times the search for these competitive advantages forces organizations to search for external solutions in other countries (offshore sourcing). This epitomizes the traditional problems and the existing challenges in GSD.

GSD causes a profound impact on the way the products are conceived, designed, constructed, tested, and delivered to customers [1]. Thus, the structure needed to support this kind of development is different from the one used in collocated environments. Different characteristics and technologies are needed, increasing the necessity of considering some details not perceived before. GSD has diverse effects on many levels, including strategic issues, cultural issues, knowledge management and technical issues.

Tools and technological environments have been developed over the last few years to help in the control and coordination of the development teams working in distributed environments. Many of these tools are focused in supporting procedures of formal communication such as automated document elaboration, processes and other non-interactive communication channels.

Nowadays, some studies can be found in the literature, proposing some models for global software development. These studies consider both technical and non-technical factors.

2.2. Offshore Sourcing

Offshore sourcing of IT work is increasingly occupying the attention of IT managers in U.S.-based firms. The term “offshore sourcing” includes both offshore outsourcing to a third-party provider as well as offshore insourcing to an internal group within a global corporation [10]. Organizations that avail themselves of outsourcing services can concentrate in its core businesses, potentially reducing the software development team. The combination of these factors results in a significant reduction in time and cost of software development. Insourcing organizations have as advantages the domestic accountability, since they utilize their own resources from the organization software development centers.

IT managers are being pressured to contain costs in addition to ramping up projects quickly, finding experienced staff in fast-moving technologies, and innovating constantly with IT. To acquire the IT competencies that address these challenges, IT managers can choose one of two strategies: either outsource to a domestic supplier or go offshore. The foreign sourcing of IT work is growing based on some reasons:

- Technologies for managing and coordinating work across geographic distances have matured considerably;
- Offshore organizations (both internal and third-party) have improved their software development and project management capabilities.

In the study conducted by Carmel and Agarwal [10], it was identified that the offshore IT sourcing is maturation process and have some stages. The authors proposed four stages in a model called SITO (Sourcing of IT Work Offshore). Each stage in this model is characterized by a set of strategic imperatives and internal firm dynamics.

In the study, the authors understand that technology companies that are in the stage four have different organizational structures and mechanisms than the other companies. The first idea was that these firms have accumulated considerably more experience in offshore IT sourcing, but they usually preferred to own their IT units, and this can lead to many difficulties in terms of software development.

3. Research Method

This research is exploratory in nature based on case study [11]. The case study was developed in a software development unit owned by a multinational organization with worldwide units. The organization works with computer manufacturing and support and is recognized as SW-CMM[1] level 2. It has software development units responsible for **internal client demand** worldwide. Its headquarters are located in the U.S.

The data collection was constituted of primary sources (interviews) and secondary sources (document reviews

[1] SW-CMM is one of the CMM models used for software engineering organizations (http://www.sei.cmu.edu).

and software development process). Considering the respondents, we interviewed 11 people – located in Brazil – from two projects. They represented project team members, development managers, quality assurance team members, software process improvement responsible and individuals representing the organization strategic level. We developed two questionnaires, each considering a specific dimension to be explored: organizational dimension, containing information about the organization as a whole and the strategies involving GSD, and the project dimension, containing information related to the projects selected to be part of this study.

4. Case Study

The case study was developed in the software development unit located in Porto Alegre, south of Brazil. This center aims to perform worldwide technological development for the organization. Almost all projects are distributed, mainly global, since customers and users are located in offices around the world. It has 120 collaborators working in software development and all clients are internal to the organization. The software development process is based on the MSF (Microsoft Solutions Framework), and also on known methodologies, like RUP (Rational Unified Process), PMI (Project Management Institute). The unit studied is recognized as SW-CMM level 2 since January of 2003.

Considering the reasons to invest in global software development, the individual representing the strategic level of the organization pointed out the following items:

- Cost reduction;
- Expanding strategy to global markets;
- Consolidate the organization trademark outside the U.S.;
- Global standard of software development.

The interviews were conducted considering two global projects each one from one department in the organization. For both projects we considered the interaction among (inter-group) project team, users and customers and the interaction inside (intra-group) each group.

5. Case Study Results

5.1. Difficulties found

According to the interviews conducted in the organization, the GSD difficulties are related to the requirements engineering, lack of standards of the activities between distributed teams, the difficulty of share information and the lack of a well-defined software development process. Besides that, corroborated by Carmel [2], and Evaristo [6], there were difficulties concerning language barriers and communication, cultural differences, context sharing and trust acquisition between teams (Table 1).

Table 1. GSD difficulties found

GSD difficulties
Requirements engineering
Software development process
Standards
Communication and language
Culture and context sharing
Trust

Requirements engineering was considered as a continuous difficulty, involving requirements elicitation, analysis, specification, validation, and management. Some individuals interviewed mentioned that since the project is distributed in multiple sites, there is a necessity of having as detailed a set of requirements as possible.

The software development process itself was considered a large difficulty since sometime distributed teams are not using the same process. In addition, software configuration is a critical issue, being the source of many problems related to the development (artefacts with different versions and content in each site).

Communication and language problems were motivated by the cultural differences between both the dispersed individuals and the sites. Finally, trust was also a problem, mainly the necessity of a distributed trust acquisition.

5.2. Solutions

Although there are many possible solutions for each difficulty identified, the organizations focused their solutions mainly on the need for work standardization, investment in planning, and process engagement. It was also mentioned the integration and ways to increase trust between global teams, and continuous training, also mentioned by [6]. (Table 2):

Table 2. Solutions implemented

Solutions
Planning
Training
Standardization
Requirements Engineering
Trust and integration

The initial planning was a necessity identified to select the projects to be distributed, evaluating its characteristics and the unit availability to receive it. Moreover, it was

perceived that the process engagement plays an important role to start the interaction between distributed teams.

Another solution implemented was training in soft skills (non-technical factors). Topics explored included leadership, communication, culture, context sharing, project management, and technical training. Standardization was adopted when the distributed teams were not using the same process. Three strategies were considered: forcing standardization; blending methodological components from the various sites into one "new" methodology; and imposing high-level guidelines.

The organization is investing in face-to-face requirements elicitation. But this depends on the project characteristics and travel limitations. There was a big effort in having formal approvals for artefacts in every project. Finally, integration activities are being conducted, aiming at trust acquisition. Some of these activities are developed virtually, but most of them occur when teams (or part of it) meet each other face-to-face.

5.3. Critical Success Factors

The critical success factors identified are directly related to the organizational "modus operandi". For the same activity we can have different factors, each one related to the strategy adopted by each organization. Consolidating the results of this study, we identified the following critical success factors (Table 3):

Table 3. Critical Success Factors

Critical Success Factors
Software Development Process
Training
Planning and Engagement
Infra-Structure
Team integration
Communication and Feedback

The software development process was considered one of the most important success factors for distributed projects. A large investment in training resulted in an improving relationship. The initial planning was important to evaluate distributed projects correctly and to select the proper unit to receive each project. The process engagement was considered a success factor because it was the first contact between the teams in some projects. Likewise, integration activities were also a success factor because it improved individuals' soft skills, increasing trust and minimizing cultural differences. Finally, integration improved the communication and feedback.

6. Lessons Learned

The study conducted in the organization shows many characteristics of GSD (section 4). These characteristics were identified based on the difficulties, solutions and critical success factor found and listed previously.

In spite of being an offshore insourcing software development center that has been recently set up (two years), when we examine how the head office and its Brazilian branch conceived and conducted the offshore project development, we can infer a series of lessons.

In this section we will present some of the lessons learned based on the empirical results found.

Table 4. Lessons Learned

No.	Lesson
#1	The existence of a global and well-defined software development process is very important in distributed projects
#2	Requirements engineering is the main challenge for the software development process point of view
#3	The planning phase is important to organize and manage the distributed projects properly
#4	The project management, and in particular risk management need additional effort and steps
#5	The investment in recruiting and training global teams can minimize the difficulties related to the non-technical dimension

Lesson 1: The existence of a global and well-defined software development process is very important in distributed projects

According to Pressman [7], a well-defined process is a process that has a good documentation, detailing what is being done (product), when (steps), for whom (actors), the artefacts used (input) and the developed artefacts (output/results). Moreover, a life cycle must be selected as the starting point for any project.

The study showed that all projects without a well-defined process had many difficulties, some of them related to the process (requirements, configuration management, testing, etc.), and others inherited, as communication, synchronization and trust. Thus, a single and well-defined process in accordance with the project environment can be the solution for many difficulties in global development.

Lesson 2: Requirements engineering is the main challenge for the software development process point of view

Requirements engineering plays an important role in the software development. A requirement is the condition or capacity that a system that is being developed must

satisfy [12]. Therefore, the compliance with requirements determines the project success or failure. Requirements are identified, registered, organized and verified during the project development, and are essential to keep the agreements among project team, users and customers.

The problems related with requirements engineering are one of the main reasons for software projects failures [12]. Research has identified [12] that 70% of the requirements were difficult to identify and 54% were not clear and well organized. Therefore, it is not difficult to find errors in requirement specifications with a resulting large impact in the project costs. It is clear that the earlier a problem is detected and solved (especially during the requirements phase) the earlier other problems are minimized in the following phases [12].

Almost all project managers and technical leaders interviewed pointed out difficulties related to requirements engineering activities. One project had the requirements instability as the main problem, mainly because the distance between teams, compromising understanding and agreement between parties. In all projects the requirements were identified as a challenge, involving activities like meetings, requirements documentation as soon as defined, traceability, requirements control and management.

Lesson 3: The planning phase is important to organize and manage the distributed projects properly

To define strategies of an organization in the information systems area based on a formal planning process is a challenge [13]. The lack of a formal planning phase can be one of the main problems before the software development process. According to [14], the lack of a formal planning phase causes a great number of problems in the next phases.

In the study, it was identified the initial planning as a formal and basic phase to decide if a project has characteristics to be distributed and to plan its development. Thus, the planning basically involves the definition of the strategies, which will lead the development of the whole process. Based on the case study, it is possible to consider the planning phase as a former cycle of many projects cycles derived from the planning process.

Lesson 4: The project management, and in particular risk management need additional effort and steps

According to the Project Management Body of Knowledge [15], project management is the application of knowledge, abilities and techniques to plan activities that can reach the needs and expectations of all stakeholders involved in a project. Bad project management can mean the loss of the project and the resources involved. Therefore, risk management is one of the most important activities in a project, involving the identification; treatment and elimination of risk sources before it become a concrete threat for the project. Risks can also be treated in different levels.

In the study, all activities involving project management and risk management have a huge importance for distributed projects and the managers interviewed said that in distributed projects these activities take longer than in traditional projects (collocated), requiring a larger effort and some additional steps in the traditional models. Additionally, all risks concerning the decision of sending a project to be developed offshore were considered in the project risk management.

Lesson 5: The investment in recruiting and training global teams can minimize the difficulties related to the non-technical dimension

In global development, project managers have to organize and manage projects with a team composed by individuals from different cultures, with different customs. According to Kiel [16], the technical barriers are diminishing rapidly. On the other hand, the human factors are less studied. Therefore, when distribution ultimately fails, it can be a web of social, cultural, linguistic and political factors, rather than use or misuse of specific tools or techniques [16]. There are other factors that can be added to this list (communication, context, interpersonal relationship), but this study brought a very important conclusion. The lack of investment in the recruiting and training of project teams to be global teams can lead to unexpected problems in the project development.

Organization's policy included investing in team training, focusing communication, cultural differences, trust, and context sharing. As a result of this initiative, the interactions between distributed teams (including customers, users and project team), were easier. Problems identified before the training started to occur less frequently, showing that the management of distributed teams is a key to the project success.

7. Conclusions

Any software professional knows that even collocated software development is fraught with difficulties. The entire field of software development, or software engineering, is still maturing. It is becoming harder to justify completing a software development project inside company walls.

As the software community appreciates the economy of merging diverse development skills and domain expertise, and as communication media become more sophisticated, the cost and technology pushes more companies toward global software development. It is becoming less and less

cost-effective or competitive to develop a software product in the same building, company, or even country.

Improvements in tools and methods over the last several decades are allowing groups from different locations and backgrounds to come together as a global software development team. Moreover, GSD is leading the researchers to acquire new knowledge and to be more interdisciplinary.

This paper advances the knowledge in the GSD area by identifying important characteristics of this recent and growing field, focusing on the offshore insourcing of IT projects. We discussed lessons learned based on a case study in a software development unit from a multinational organization. These sets of results give us indication that the search for greater formalism and the selective utilization of international patterns will provide full conditions to overcome the problems originating from the dispersion specifically in the case of wholly-owned subsidiaries. Planned follow up studies in this topic will continue to analyze the organizations difficulties and solutions and will going deep in the study of specific factors found in this work, like requirements engineering, risk management, and project allocation, despite of analyzing how other organizations in similar situations are dealing with all these problems.

Finally, this project is not only a landmark study in the area of offshore insourcing, something until recently not been researched, but also has strong implications to the more traditional offshore outsourcing. The key reason is that most of the work currently being done in offshore outsourcing is seen under the perspective of contracting; although obviously very relevant, eventually such studies will need to go further past that issue – which is exactly what we are proposing to do in the near future.

8. References

[1] Herbsleb, J. D., and Moitra, D. "Global Software Development", IEEE Software, March/April, USA, 2001, p. 16-20.

[2] Carmel, E. "Global Software Teams – Collaborating Across Borders and Time-Zones". Prentice Hall, USA, 1999, 269p.

[3] Karolak, D. W. "Global Software Development – Managing Virtual Teams and Environments". Los Alamitos, IEEE Computer Society, USA, 1998, 159p.

[4] Damian, D. "The study of requirements engineering in global software development: as challenging as important", Proceedings of International Workshop on Global Software Development at ICSE, Florida, USA, 2002.

[5] Prikladnicki, R.; Audy, J. L. N.; Evaristo, R. "Distributed Software Development: Toward an understanding of the relationship between project team, users and customers". *Proceedings of ICEIS*, Angers, France, 2003.

[6] Evaristo, J. R., Scudder, R., Desouza, K. and Sato, O. "A Dimensional Analysis of Geographically Distributed Project Teams: A Case Study," forthcoming in the Journal of Engineering Technology and Management, 2003.

[7] Pressman, R. S. "Software Engineering: A Practitioner's Approach". Fifth Edit, USA, 2001.

[8] Prikladnicki, R., Peres, F., Audy, J., Móra, M. C., and Perdigoto, A. "Requirements specification model in a software development process inside a physically distributed environment", Proceedings of ICEIS, Ciudad Real, Spain, 2002.

[9] McConnel, S. "Rapid Development". Microsoft Press, Canada, 1996.

[10] Carmel, E.; Agarwal, R. "The Maturation of Offshore Sourcing of Information Technology Work", *MIS Quarterly Executive*, Vol. 1, No. 2, June 2002, 65-77.

[11] Yin, R. K "Case study research: design and methods", Sage, USA, 1994.

[12] Oberg, R., Probasco, L., and Ericsson, M. "Applying Requirements Management with Use Cases", Rational Software White Paper, Cupertino, CA, USA, 2000.

[13] Audy, J. L. N. "Strategic Planning Model of Information Systems: contributions of decision process and organizational learning (in Portuguese)". Ph. D. Tesis, PPGA – UFRGS, Brazil, 2001.

[14] Martin, J. "Information Engineering (in Portuguese)". Rio de Janeiro, Campus, 1991.

[15] A guide to the project management body of knowledge (PMBOK guide). Project Management Institute, USA, 2000. 216p.

[16] Kiel, L. "Experiences in Distributed Development: A Case Study", Proceedings of International Workshop on Global Software Development at ICSE, Oregon, USA, 2003, 4p.