

RISK MANAGEMENT IN DISTRIBUTED SOFTWARE DEVELOPMENT: A PROCESS INTEGRATION PROPOSAL¹

Rafael Prikladnicki

School of Computer Science, PUCRS, rafael@inf.pucrs.br
BRAZIL

Marcelo Hideki Yamaguti

School of Computer Science, PUCRS, yamaguti@inf.pucrs.br
BRAZIL

Dante Carlos Antunes

antunesdancec@aol.com

The goal of this paper is to propose a process to manage risks in globally distributed software development environments. We propose a process integration, aiming to reflect in the operational level all information and decisions taken in the strategic and tactical levels. The operational level is defined as the project risk management process, and the strategic and tactical levels are defined as all work concerning the decision to develop a project using geographically distributed resources. The integration was proposed based on the results found in a case study conducted in a software development center from a multinational organization located in Brazil. Thus, the focus of this study is to help in the understanding of what it is involved when managing risks in distributed software development environments.

1. INTRODUCTION

In the last decade, a great investment has been made to convert national markets to global ones. This reality creates new opportunities for competition and collaboration (Herbsleb, 201). However, it also creates some problems such as a great number of project failures, and the scarcity of good resources. In this environment, software development organizations found in Distributed Software Development (DSD) an alternative for these problems. DSD is causing a great impact not only in the market, but also in the way software products are conceived, designed, constructed, tested, and delivered to customers (Prikladnicki, 2003). In this context, risk management becomes a more sensible activity with a great importance.

Risk management in DSD should happen at the operational level, as well as at the strategic and tactical levels. In the strategic and tactical levels, risk management identifies the risks involved in projects, which will be developed in a distributed way. The proposition in this paper is to integrate the risk management from strategic

and tactical levels to the operational level. In this way, a project manager, at the operational level should be aware of identified risks at the higher levels and he/she could plan response actions for these risks. The results are based on a case study conducted in a software development center (wholly-owned subsidiary) from a multinational organization located in Brazil. This paper has the following structure: section 2 presents the theoretical base; section 3 presents the research method, and the case study; section 4 presents the process integration proposal; section 5 presents the final remarks, future studies and the research limitations.

2. THEORETICAL BASE

2.1 Risk Management

The word “risk” comes from old Italian word “risicare”, which means, “to dare” (Bernstein, 1997). In the software area, Boehm represented risk in a systematic way in the 1980’s through the spiral model. This model has as principle to be iterative and risk analysis driven in each iteration (Bohem, 1991). But successfully managing risk requires more than good processes and the ability to think intuitively, it also requires discipline. This discipline is called risk management.

Nowadays, risk management in software engineering is an evolution of the risk concept, and should pervade all the processes in the software lifecycle. The risks cannot be just simple details in the project, but they should be the core of the business (Kerzner, 2000). Also, risk management has a proactive focus on preventing problems, is continuous (in the whole software lifecycle), and concurrent (many risks are managed at the same time).

2.2 Distributed Software Development (DSD)

More than a decade ago, seeking lower costs and access to skilled resources, many organizations began to experiment with remotely located software development facilities (DSD). Several factors have contributed to build this scenario:

- The business market proximity advantages, including knowledge of customers and local conditions;
- Pressure to improve time-to-market by using time zone differences in “round-the-clock” development;
- The need to have a global resource pool to successfully and cost-competitively have resources, wherever located.

Organizations search for competitive advantages in terms of cost, quality and flexibility in the area of software development (Carmel, 1999), looking for productivity increases as well as risk dilution (Karolak, 1998). Sometimes, organizations view DSD as an increasing risk, since involves a management overhead, global standards, etc. Many times the search for these competitive advantages forces organizations to search for global solutions (offshore software development). This epitomizes the traditional problems and the existing challenges.

Software is being developed in a multisite, multicultural and globally distributed scenario. Engineers, managers, and executives are facing many challenges on many levels, from the technical to the social, political and cultural. And this change is

having a profound impact on the way products are conceived, designed, tested, and delivered to customers (Herbsleb, 2001).

2.3 Risk Management in Distributed Software Development

Risk management in distributed software development is an important and necessary activity. In research by Prikładnicki (Prikładnicki, 2003), it was detected that effective risk management was an alternative to solve existent problems in distributed projects. This result from the fact that it is hard to deploy, execute and control projects in DSD environments because non-technical factors such as social, cultural, behavioral, and political (Kiel, 2003). Other studies (Herbsleb, 1999) present the same difficulties but also suggest technical factors such as the software development process, project management, project size and complexity.

Therefore, the risk management becomes important in projects that are developed with distributed teams (from the same or different organizations). Whether a project is developed globally or in the same city, having geographically dispersed teams using collaboration technologies and developing specific solutions to distributed projects also adds more risk factors to the projects.

According to Karolak (Karolak, 1998), risk management is part of any project and risks in DSD projects tend to be less visible. Also, according Karolak, there are three categories of risks in DSD projects: organizational, technical and of communication. Additionally, some risks belong to more than one category, and these should be in the top of the priorities list.

In Prikładnicki (Prikładnicki, 2003), the risk management in DSD projects should be done not only in the project level, but also in the organizational level. First of all, to decide if a particular project can be developed by globally dispersed teams is difficult (strategic level). Moreover, the decision of where the project will be better developed can also be a problem (tactical level). Some analysis considering the risk and benefit of project dispersion can be necessary. A number of models are possible and appropriate under different circumstances. As an example, there is a study conducted by Morstead (Morstead, 2003), where he developed a roadmap and a risk management framework for reinventing the IT function into a low-cost, high-service organization leveraging remote resources.

Additionally, it is suggested that all the identified risks in the strategic and tactical levels should be reflected at the project development, or operational, level. It means, since the risk analysis was made and the decision of distributing a project was taken, the identified risks must be passed to the project manager. In this way, the project manager can plan response actions to these risks and add the risks to those of the whole project, following the risk management process defined for software development.

In short, there are a set of inherent problems and challenges to software development. The DSD, by adding factors like geographic dispersion, temporal dispersion and cultural differences, has accentuated some challenges and added new ones to the development process. Among these challenges we can cite as important ones: strategic issues, cultural issues, knowledge management and risks management. As a result, the work in DSD environments is more problematic than in centralized ones, and the effective risk management can never be depreciated. The

importance of risk management must be emphasized and its use is critical in projects in DSD environments.

4. CASE STUDY: TOWARDS THE PROCESS INTEGRATION PROPOSAL

This study was developed to evaluate risk management in an offshore software development organization, trying to understand how the decision to develop a project using globally dispersed teams was taken and how it was reflected in the project development itself. It analyzed specifically the risk management process, since decisions at the strategic and tactical levels impact project development at the operational level.

4.1 Research Method

This research is exploratory in nature based on case studies. Qualitative methods are appropriate when studying state of the art situations where practice precedes theory (Yin, 1994). The case study was developed in a software development center owned by a multinational organization.

The data collected was constituted of primary and secondary sources. As primary sources, we conducted individual interviews. As complement, secondary sources were also used, such as document reviews, strategic mission analysis, business processes, meeting minutes, and software development process description.

4.2 The Organization

The case study was developed in the organization software development center located in *Porto Alegre*, Brazil. This center performs worldwide software development for a multinational organization. Almost all projects are geographically distributed globally since customers and users are located in offices around the world. All customers are internal to the organization. The software development process is based on the MSF (Microsoft Solutions Framework), and on known methodologies such as RUP (Rational Unified Process), and PMI (Project Management Institute). The center studied is recognized as a level 2 organization in the SW-CMMⁱⁱ model and is working to achieve level 3 in the CMMIⁱⁱⁱ model.

4.3 Project allocation

In order to have better control of the distributed project allocation and planning, the organization created a set of activities to be implemented in all projects being developed in the organization's development centers. These activities include offshore demand definition through resource allocation. Once the project is planned and is able to be sent to the development centers, the project execution is started, following the organization's defined software development process. These activities generate a process called the Software Development Distribution Model (Figure 1), where all strategic and tactical decisions are made following a systematic process

with well-defined steps (generally performed by senior managers and development center directors).

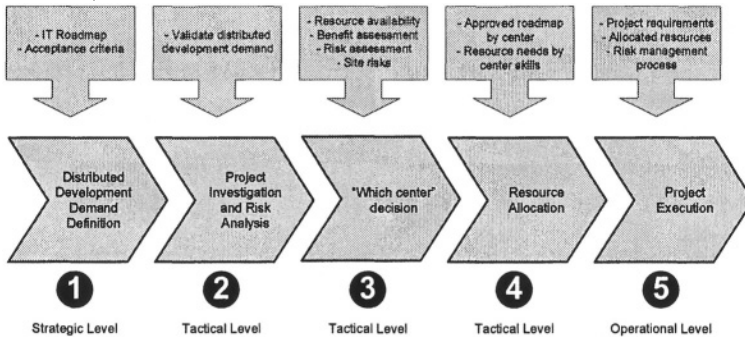


Figure 1 - Software Development Distribution Model

The risk analysis is represented in step two where, after the distributed development demand definition (step one), a risk and benefit analysis must take place in order to decide if the project can be allocated to a remote development center. Once the decision is made, a risk assessment is conducted in step three to verify which center among all the organization’s development centers, can better develop each project. Once the center is selected, resource allocation activity occurs in step four and finally, in step five, project execution occurs following the organization’s defined software development process.

4.4 Risk Management in the software development process

Once the project is evaluated and it is in the roadmap to be developed by the organization’s distributed software development centers, its execution is started. Project execution involves all work concerning project development by the project team following the organization’s software development process. This process includes risk management activities (performed by project managers). The risk management process in the operational level (project execution) was defined based on the CMMI model and has eight activities (Figure 2). The organization uses the PMBOK (PMBOK, 2000), with inputs from the risk management defined in the MSF and the organization’s defined software development process. The risk management process was defined in order to be simple to implement, and to improve the communication with all stakeholders involved.

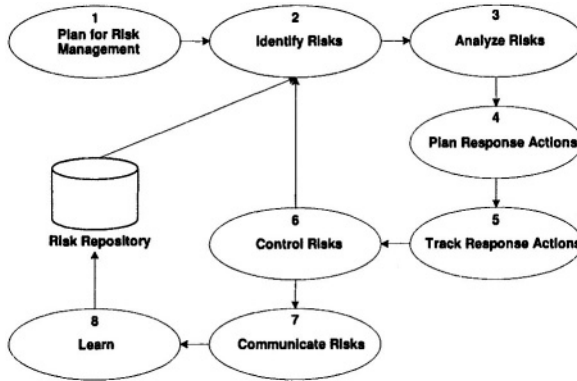


Figure 2 – Risk Management Process, based on the CMMI model

The first activity involves the planning and definition of steps that will be used in the project risk management and identifying stakeholders. The project manager must define strategies to be used based on the strategies available in the organization software development process, which risks will be managed (top risks), who the risk status will be communicated to, and so on.

After, the second activity must take place, where all risks will be identified. Risk identification includes searching for common risks and past risks in the risk repository, using the techniques planned in the first activity, and may involve all of the project team, including the ones globally dispersed, customers and/or users. Risk identification must consider as input project requirements, assumptions and constraints. Once the risks are identified, they have to be analyzed (third activity). Risk analysis involves determining the probability and impact of each risk, and its exposure.

The fourth activity is one of the most important in the risk management process, since it involves the planning of response actions for each risk. All actions must define the type of response (mitigation, avoid, transfer, etc.), who is responsible for the action and the timeline. All planned response actions must be tracked, as defined in the fifth activity. To track all response actions means to monitor if each response action is being implemented on time, and to track periodically the probability and impact of each risk. If a risk occurs, a contingency action (planned in the fourth activity) must take place, and the risk must be controlled. The risk control is defined in the sixth activity, and also involves identifying new risks. All risks being managed in a project will have a status that must be communicated (seventh activity) to specific stakeholders as defined in the first activity. Finally, when a project is finished, lessons learned should be registered in the risk repository for use in the future projects (eighth activity).

4.5 Critical Analysis

Both the software development distribution model (subsection 4.3) and the software development process (subsection 4.4) have activities related to risk management. The software development distribution model implements a risk analysis and a risk assessment at the strategic and tactical levels in order to help in the decision to distribute development and the selection of the best center for a specific project.

On the other hand, risk management in the software development process involves risk management concerning the project itself, or at the operational level. But a key point in the whole process is the integration of the risk analysis and assessment done in the strategic and tactical levels (generally performed by senior managers and development center directors) with the risk management process done in the operational level (performed by project managers). Despite the process to select appropriate centers to develop each project, and a process to manage risks when a project being executed, the processes are not typically integrated, according to the interviews conducted in the organization studied.

Almost all interviews indicated that when the organization works with globally dispersed teams and distributed projects, the risk management process must be improved. For this reason, we propose a process for integration, where all risks analyzed in the strategic and tactical levels must be reflected in the operational level, through a systematic and documented procedure as detailed in the next section.

5. PROCESS INTEGRATION PROPOSAL

Based on the case study developed (Section 4), we have developed a process integration proposal. The integration considers the creation of models and activities to help the organization have a better and more effective risk management system. The main idea of this proposal is to document all decisions made in steps two and three of the Software Development Distribution Model (Figure 1), and to reflect in step five, where the project will be executed and the risk management process based on the software development process will begin (Figure 3).

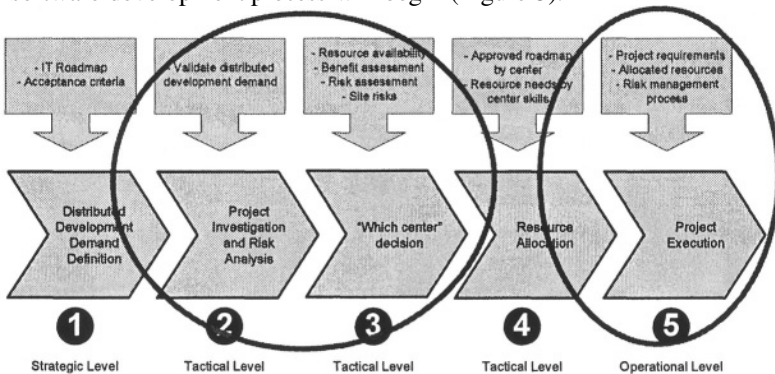


Figure 3 – Process Integration

The integration was proposed using UML^{iv}. Use cases diagrams were used to represent the main activities in each process. Also, activity diagrams were used to perform in-depth analysis of a specific activity. This step was important in order to create a standard representation between the processes. After the standardization through the UML representation, we created another use case diagram showing integration possibilities, and another activity diagram used to identify some activities to integrate the processes. All diagrams were developed based on the information collected from the case study. We present here part of the use case diagram (Figure 4), and the activity diagram (Figure 5) with some additional

activities proposed based on the use case and activity diagrams from the original processes.

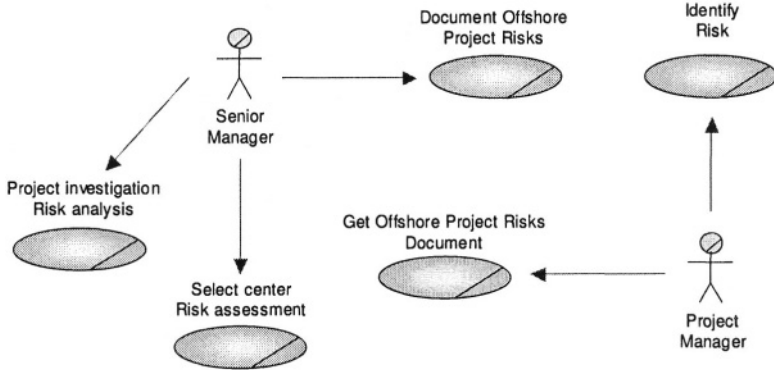


Figure 4 – Use case diagram integrating both processes

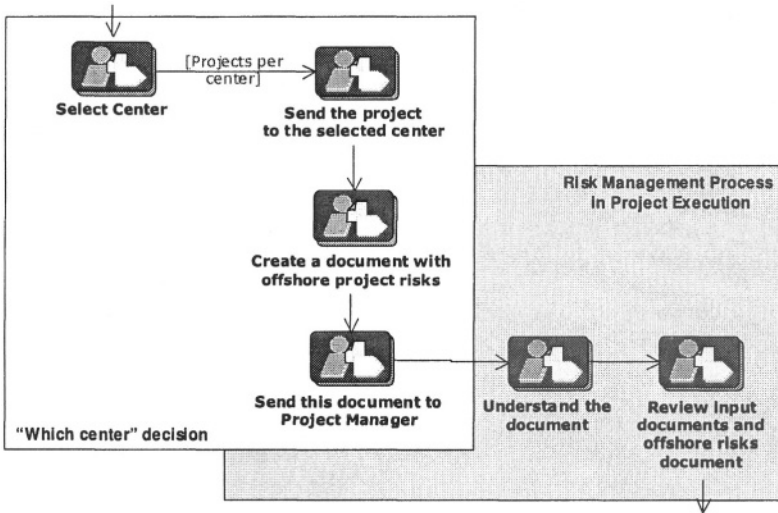


Figure 5 – Activity diagram integrating both processes

The last integration effort was the creation of an artifact to document all risk analysis and risk assessment conducted at the strategic and tactical level. The purpose of this artifact was to give to the project manager all information necessary from discussions by senior managers. The artifact is composed of two main sections: the first section describes details of the project investigation and decision of which development center to use; the second session is related to the risk description, including strategies to reduce the risk, the exposure from the organizational point of view and general comments. This document should be completed after step three in the Software Development Distribution Model, and is sent to the project manager to be used as another input for the risk management process. Finally, when the project is finished, all lessons learned should be stored in the risk repository and used when beginning the Software Development Distribution Model for new projects to be developed by globally dispersed teams.

6. FINAL CONSIDERATIONS

Software projects are dynamic and unique, which lead to the existence of many risks that must be managed. In order to have success in all projects, organizations need to manage risks effectively. But one of the main reasons that risk management is inefficient or is not implemented in many organizations is the lack of documentation of both success and failures in projects. Knowledge about risk management alone is not sufficient.

The practice of learning from past experiences is mandatory to help senior managers and project managers to plan for and control risks (Kwak, 2003). DSD provides a good opportunity to exploit the benefits of knowledge management since the DSD model involves additional steps not found in traditional risk management models. And sometimes, risk management in this kind of project can take longer than in traditional projects, because of the geographic dispersion and time zone differences.

The process integration proposed in this paper helps the organization improve risk management in all distributed software development projects since it includes standard procedures to document all decisions made in the software development distribution model and is directly reflected in project execution. Planned follow-up studies on this topic will analyze other organizations in the same situation (geographically distributed software development within wholly owned subsidiaries), in order to improve and validate this process integration.

7. REFERENCES

1. Herbsleb, J. D., Moitra, D. "Global Software Development", IEEE Software, March/April, USA, 2001, p. 16-20.
2. Prikladnicki, R. "MuNDDoS: A Reference Model for Distributed Software Development (in Portuguese)". 145 f. 2003. Master Thesis, PPGCC – PUCRS, Porto Alegre, Brazil, 2003.
3. Boehm, B. "Software risk management: principles and practices", Piscataway: IEEE Software, v. 8, p. 32-41, jan. 1991..
4. Bernstein, P. "Challenge to God: the risk history (in Portuguese)". Rio de Janeiro: Campus, 1997.
5. Kerzner, H. "Project Management: a systems approach to Planning, Scheduling, and Controlling". John Wiley & Sons Inc., USA, 2000.
6. Carmel, E. "Global Software Teams – Collaborating Across Borders and Time-Zones". Prentice Hall, USA, 1999, 269p.
7. Karolak, D. W. "Global Software Development – Managing Virtual Teams and Environments". Los Alamitos, IEEE Computer Society, USA, 1998, 159p.
8. Kiel, L. "Experiences in Distributed Development: A Case Study", Proceedings of International Workshop on Global Software Development at ICSE, Oregon, USA, 2003, 4p.
9. Herbsleb, J. D, Grinter, R. "Splitting the organization and integrating the code: Conway's Law revisited". In: ICSE, 1999, Carolina do Norte. Proceedings... EUA, 1999. 11 p.
10. Kwak, Y. H., Stoddard, J. "Project Risk Management: lessons learned from software development", Technovation, In Press, Corrected, 2003
11. PMBOK Guide "A Guide to the Project Management Body of Knowledge". Project Management Institute, Pennsylvania, US, 2000.
12. Yin, R. K "Case study research: design and methods", Sage, USA, 1994.
13. Morstead, S., Blount, G. "Offshore Ready: strategies to plan & profit from offshore IT enabled services", ISANI Press, US, 2003.

ⁱ Research developed at CDPe research center at PUCRS University, and funded in terms of the Brazilian Federal Law for Information Technology (Law No. 8.248/91).

ⁱⁱ The SW-CMM describes the principles and practices underlying software process maturity and is intended to help software organizations improve the maturity of their software processes (<http://www.sei.cmu.edu/cmm/cmms/cmms.html>).

ⁱⁱⁱ The CMMI models improve upon the best practices of previous models, like SW-CMM (<http://www.sei.cmu.edu/cmmi/cmmi.html>)

^{iv} UML is a standard and graphical language of the Object Management Group for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system (<http://www.omg.org/technology/documents/formal/uml.htm>).