



IDENTIFICATION AND MITIGATION OF RISKS IN DISTRIBUTED DEVELOPMENT OF SOFTWARE PROJECTS

Pedro Henrique Veras Coelho^a, Germano Fenner, Alberto Sampaio Lima^a

^aFederal University of Ceará (UFC)

Abstract

By every minute markets become more globalized, impacting in different areas and making way for new means of competition and cooperation that transcend borderlines. As a result, the software development field was also influenced, which lead many organizations to try the development of software products on establishments located in other countries. Distributed Development of Software (DDS) then emerged; a field that presents many risk factors, such as geographic spread and communication issues. Although organizations where DDS is used tend to face a great deal of difficulties, it was found that risk mitigation measures make benefits overcome difficulties. Moreover, embracing it not only reduces costs but also increases competition between organizations. A case study and bibliographic research were the chosen methods to identify DDS most current risks and the how to handle them.

Keywords: Distributed Development of Software, Risk Management, Project Management

1. INTRODUCTION

Due to the current economic outlook, reducing projects cost and time demanded to place new products in the market is now of great need. Defined by the distribution of software development among cities, districts and even countries, Distributed Development of Software (DDS) emerged to meet organizations competition needs.

As infrastructure and communication tools became accessible, DDS management turned viable (Persson *et al.*, 2009). Nonetheless, due to its complexity, DDS still presents numerous risks in the communication, culture, technology, contribution and requirements engineering fields.

According to Kwak and Stoddard (2004), risk management is the least practiced method among project management areas. Because of multiple factors - such as limited social interaction between team

members - risk management cannot be neglected when talking about DDS, since its inexistence can lead the project to failure.

This study researched DDS related risks associated to five fields (communication, culture, technology, contribution and requirements engineering), also suggesting measures to mitigate them. Promising results were obtained in a real case study, propitiating better comprehension of the identified risks and of the strategies used to overcome them.

2. RELATED WORK

Over the last years, a great progress could be noticed in the globalization of business. Organizations, seeking costs reduction and contact with qualified workers, started experimenting outsourcing. This led to the construction of software development facilities located in remote sites.



Steinmacher *et al.* (2013) says that DDS operates based on geographically spread teams that work collaboratively on a software project. In the DDS system, developers interact in an allocated manner during the software life cycle, creating a network of dispersed teams. In some cases, these teams already include members of the same organization, although others can combine workers of different companies. The authors also stated that DDS brings new challenges to the managers, such as contextual, cultural, organizational, geographical and chronological ones, not to mention language and political discrepancies between the software developers. It was also mentioned that the growing adoption of DDS and its features represents not only challenges and new opportunities to explore, but also the chance to investigate the applicability of existing scientific researches. The authors' work consisted of a systematic review of the DDS system. It could be identified breaches between teams and opportunities for research in areas like communication, administration and cooperation. It could also be found a lack of connection between appraised scientific researches and other areas of study, such as ubiquitous computing. Project management was presented as a potential study field, therefore being the focus of the present work.

Jimenez *et al.* (2009) mentioned that face-to-face meetings have become less frequent in DDS projects; also, interaction between members required technology support to help teams communicate, manage and cooperate.

Besides costs reduction, DDS has many advantages, such as a better application of the labor law, proximity to local markets, government tax incentives, time zone overlaps (becoming possible to work 24 hours a day) which allow the implementation of round-the-clock working system. Round-the-clock working, on its turn, reduces the amount of time needed to place a product in the market (Hersleb *et Moitra*, 2001). According to Sakthivel (2007), many American companies ranked in the Fortune 500 list have been benefiting by developing their information systems in countries with cheap labor costs such as India and China.

Many investments have been drawn towards DDS due to economics advance, refinement of technology and pressure for lowering costs. The improvement of tools and methods used on software engineering will create a constant pressure to embrace globalized

approaches on software creation. These may be presented as outsourcing formal contracts; begin through a partnership between partitions of different international organizations, or even be expressed by a small group of developers who work on a same project but reside in different cities (Kiel, 2003).

Without further questioning, both traditional software development and DDS present complications. However, distributed development adds some challenges to project managing, such as how to manage people who are geographically dispersed and in different time zones, not to mention language barriers and sociocultural differences. Herbsleb *et Moitra* (2001) state that the physical segregation of project members can reflect on several aspects of the project, such as deciding to do it in the distributed manner or not; the way tasks will be assigned based on available resources, infrastructure and technology expertise; issues regarding cultural values and principles between team members; issues related to process management, such as having a hard time synchronizing and the need of reaching the same milestones during the project; and finally, issues related to knowledge management; for instance, information sharing and storage.

Bubeck *et al.* (2012) proposed using DDS for the distributed development of robot platforms, helping in the process of integrating the system. Projects for the development of ten robots located in seven different places were successfully managed. The aspects related to code development were the project outlook.

Estler *et al.* (2012) presented a case study regarding the impact structured and agile developing processes have on globally distributed software. Success, the extent of money saved, the amount of importance shown by clients, team motivation, amount of time spent real-time communicating and asynchronous necessary during projects execution were evaluated in the appraised researches. The authors debated aspects such as offshore versus nearshore, communication patterns and other critical subjects. Outcomes suggested that opting for agile or structured methods on DDS projects is not essential.

The study of Farias Junior *et al.* (2012) used an empirical approach to identify and present communication risks found on DDS projects. By interviewing project leaders and managers of numerous companies, it could be obtained ways to mitigate the



risks. The authors declared companies lack efficiency and many don't have a formal method of work defined, even though its managers are aware of the importance of risk management on DDS projects. It was also stated that when working with distributed workforce, if the software process is not well defined and the teams are not prepared for working under these conditions, the risk increases substantially.

3. RISK MANAGEMENT IN DISTRIBUTED DEVELOPMENT OF SOFTWARE

Software development is creating something that has never been done before, even though the process may resemble some used on other projects (Kwak *et Stoddard*, 2004). As a result, a scenario of uncertainty is likely to emerge, possibly combined to risks.

Boehm (1991) investigated 600 companies, finding out that 35% of them have had at least one challenging project. Many of these challenges could have been avoided or abbreviated if identifying and solving high risks factors were a concern at that time. According to PMI (2008), proactive approaching the project uncertainties may prevent them to turn into threats.

One of the major factors that lead to deadline and costs overrun on software developing projects is the incorrect management of risks. The danger is even greater when these systems are developed by team members who speak different languages, are of different nationalities and are geographically and chronologically dispersed (different time zones) (Salthivel, 2007).

Managing risks when working on DDS projects becomes even more essential as the distributed projects are of difficult implementation, execution and control due to non-technical reasons, such as cultural, social and political differences (Priladnick *et Yamaguti*, 2004). However, if the risks are effectively managed – opting for actions, tools and measures that help in the communication process – DDS can be extremely advantageous.

3.1 Risks identification in Distributed Development of Software projects

Risks typical to distributed projects were divided into five areas and listed in Board 1. Some of them can

also appear on co located projects; however, here the analysis will focus on distributed development only.

3.2 Communication

Communication is crucial when working in software development, making it possible to solve problems, to understand requirements and to comprehend tasks that need be Distributed Development of Software projects. To Herbsleb *et Mockus* (2003), when distance between team members is over 30 meters or more, participants begin communicating as if they were kilometers apart. A case study conducted by Perry *et al.* (1994) determined that developers spend at least 75 minutes a day communicating informally.

Communication between team members can be a great challenge when working in great challenge when working in Distributed Development of Software projects. To Herbsleb *et Mockus* (2003), when distance between team members is over 30 meters or more, participants begin communicating as if they were kilometers apart.

Board 1. Identified Risks on DDS Projects

Field of Risk	Risk
Communication	Number of places involved
	Geographic dispersion
	Chronological Dispersion
Culture	Sociocultural differences
	Language differences
Technology	Technological expertise
	Configuration management
	Infrastructure
	Tools compatibility
Collaboration	Process harmonization
	Data sharing
	Project management
Requirements Engineering	Task knowledge
	Requirements change
	Poor quality

Source: Compiled by authors

Depending on the numbers of places involved and on chronological dispersion, synchronous communication can become a challenge, making it necessary to use asynchronous communication methods.

Among aspects that interfere on DDS communication, the most prominent are:



- **Number of places involved:** in a distributed project, the higher the number of places involved is, more complicated task management and distribution tend to be. For example, there could be a national holiday in a certain country causing the local team to interrupt work while the other teams keep on going. Besides that, certain legislations can block hardware and software importation.
- **Geographical dispersion:** Distributed projects may present different levels of geographical dispersion, preventing face-to-face interaction. Audy et Prikladnick (2008) present these levels as:
 - *national dispersion* – defined by members located in the same country. In some countries, there are time zone differences; however, teams still manage to gather in a short amount of time.
 - *continental gap* – defined by teams located in different countries but located in the same continent. Cultural, language and time zone changes are more accentuated, which might make interacting harder.
 - *global gab* - defined by teams located in different countries and continents. The different time zone can prevent synchronous interactions, and cultural disparities can make work complicated.
- **Chronological Dispersion:** the project can engage teams located in different time zones, turning managing and synchronous exchange of data into hard tasks; this increases dependency of asynchronous forms of communication, such as email exchange. Given that, it may be unavoidable for a team to wait more than a day to hear from another team, located in a different time zone, and finally obtain the solution for their problems or doubts. Scheduling meetings can also be a risk when working with chronological dispersed teams, since there might be misunderstandings regarding the scheduled time due to the time zone difference.
- Sociocultural differences – gathering people with different values, religious beliefs and traditions in the same project environment requires considerable cooperation; if not, conflicts and disagreements may happen. Different ways to work, intonation when speaking and even the language used when writing emails can lead to conflicts, damaging the sense of teamwork and negatively affecting the project (Kiel, 2003). Besides, discrepancies between religious beliefs can also exist: if a certain area has the tradition of not working through a particular time period, the project will lack synchronism.
- Language differences – language is an important barrier of communication for geographically spread teams, especially if verbal communication is involved. Accent and intonation differences can lead to misunderstandings and to communication delays even on teams that speak the same language (Kiel, 2003).

3.4 Technology

An adequate technological infrastructure is a fundamental piece of the DDS system; without it, communication and interaction between those involved in the development of the project would be impossible. Besides having a network and server configuration related physical infrastructure, one must also know how to manage the tools used on the project to assure it is standardized. The most prominent technology related risks are:

- **Technical expertise:** according to Sakthivel (2007), some countries have little interest in investing in scientific researches and progress; this ends up producing information technology professionals without any technical or process development expertise and little productivity. Besides, it is impossible for teams to have the same technical knowledge; so inevitably, one of them will know a programming language that the others teams don't.
- **Configuration Management** – a configuration management software system consists in a very powerful tool that reduces communication issues, as it makes one look at the project and at the work process as unique (Carmel et Argwal, 2001). This technology, however, can present as a great challenge to the dispersed teams due to the use of different tools and unawareness of changes in the product or existing malfunctions (Persson et al., 2009).

3.3 Culture

Working with teams spread in numerous countries leads to culture related challenges. Each country has its own way to communicate, meaning conflicts between teams of different countries might happen due to misinterpretations. Between risks related to culture, the most prominent are:



- **Infrastructure** - some countries may have inadequate electric power and Internet infrastructure, making distributed projects a hard job to do due to slow Internet connection and electrical fluctuations.
- **Tools compatibility** – Persson *et al.* (2009) says that when multiple developers from around the world are working together, tools compatibility can be a great problem, since not everyone is likely to opt for the same programming language, operating system or software development tools.

3.5 Collaboration

Collaboration demonstrates how well the members involved in the project can share information. According to Persson *et al.* (2009), the highest risk happens when the collaboration structure doesn't match the distributed development one. Among the risks are:

- **Process alignment** – nescience of organizational culture, differences of business processes, developing methods and of project management can result in conflicts, disagreements and in having difficulties creating a sense of teamwork between project members (Persson *et al.* 2009).
- **Data sharing** – failing in sharing data can happen due to distributed development different levels of dispersion and unawareness of who is responsible for a determined task or even of who has the technical expertise needed in a certain area. Also, if there is lack of confidence between the teams, a fear of losing possession of the solutions found by one of them may happen; this, then, makes teams unwilling to share results with others. Hence, the time and money that would be saved reusing codes and solutions found is jeopardized.
- **Project Control** – unawareness of the project status and of which tasks are following the critical path may happen as a result of the communication challenges imposed by distributed development. Furthermore, distributed development also demands different ways to estimate effort, as those used on co allocated projects may give an inaccurate evaluation due to distributed development many levels of communication and managing.

3.6 Requirements engineering

Communication between requirements engineering specialists and stakeholders is less frequent because

of the distance between them. Facing that, there is a great risk of misinterpreting requirements, which may lead to modifications, raising costs and overrunning the project deadline. Some of the risks related to requirements engineering are:

- **Task knowledge** – as asynchronous communication methods are frequently used and requirements stipulations tend to be quiet inconsistent, improper task designations are bound to happen, bringing along incomprehension about what must be developed.
- **Requirement changes** – requirement changes are a great danger to the software development process. According to Herbsleb *et al.* (2003), these changes are 2.5 times faster to apply in co allocated projects when compared to distributed projects.
- **Poor quality** – a project that involves people located in different countries and lacks proper methods, tools or development processes can lead to poor quality products. This, in turn, generates rework and increases mistrust amongst project members (Ebert *et al.*, 2008).

4. CASE STUDY

A case study consists in a methodological approach of an investigation over a sustained period of time. This type of study seeks comprehending, exploring or describing events or scenarios of complex nature where multiple factors are simultaneously involved. Yin (2005) said this approach better fits researches in which the investigator faces complex situations, making it hard to determine the relevant variables.

To better identify the risks involved, a real DDS project scenario was used. Both risks and the measures taken to mitigate them were presented at the time.

The scenario consisted in a project financed using tax exemption resources guaranteed by *Lei da Informática* (roughly translated to Informatics Law), a Brazilian law that grants tax incentives to technology companies who want to invest in research and progress. The project goal was to develop a social networking website for scientists employed by an American company with offices located in Brazil and Israel. To respect the business confidentiality, the company name will not be disclosed.



4.1 Methods

The improvement, elucidation and assessment of concepts and ideas through existing theories about risk management on DDS projects were the purpose of this project.

In order to achieve that, a qualitative and exploratory field research was done (Gil, 2002). During the course of the DDS project evaluated, relevant material was retrieved using literature review, observation of everyday activities and information obtained through semi-structured interviews made with project managers, team members and company leaders.

During the research design and interview planning stages, recommendations of Runerson *et Host* (2009) and Wohlin *et al.* (2000) were followed.

4.2 Procedure and Data Analysis

The project analyzed had one global dispersion level: a team was located in Fortaleza, Brazil; one in Porto Alegre, also in Brazil; while the third one was located in Haifa, Israel. An overall perception of the physical distance that separates those cities is represented on Figure 1.

The team located in Fortaleza was responsible for developing the system presentation tier; this tier connected to the business logic one, which included algorithms developed by the Israel team.

The team based in Porto Alegre mediated the process and took care of the project contract details.

A 10 days' life cycle incremental approach of the software development methodology was chosen. At the end of each life cycle phase, a teleconference meeting was held between all teams to discuss what had been developed at that phase, lessons that were learned and what could be improved in the next phases.

One of the issues identified during this research was the chronological dispersion. Israel and Brazil are separated by a 5-hour time zone difference (as represented on Figure 2), reducing communication window and time for solving doubts.



Figure 1. Teams setting
Source: Compiled by authors

Sociocultural differences were another determined issue. The team based in Israel followed the *Shabbat* Judaic tradition, which starts on Friday evenings, going until Saturday nights. During this period, considered as the Judaism's day of rest and seventh day of the week, Jews refrain from work activities. Moreover, Israel's workweeks start on Sundays, unlike Brazil's. Thus, there were only four workdays the teams shared during the project.

Combined to the risks related to geographic and chronological dispersion, it could also be identified in the project a technical expertise related risk, discussed in section 3.4. The teams located in Porto Alegre and Haifa did not know how to work with the programming language used by the Fortaleza team, while the last one could not work with the algorithms developed by the team located in Israel.

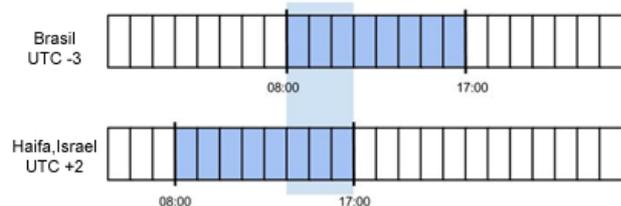


Figure 2. Time Zones
Source: Compiled by authors

Initially, the Haifa team had concerns over providing the algorithm they had developed to the other teams. As presented in section 3.5, refusing to share data is a common risk on distributed projects.

The team situated in Fortaleza requested some changes in order to improve the algorithm developed



by the Haifa team, which may imply there was a lack of knowledge of the task, a risk mentioned on section 3.6 of this research.

Difficulties related to configuration management and to the poor quality of the software developed were two pertinent risks to the project. Because of them, policies for trials and for configuration management were used as mitigation measures – which will be discussed further below.

As soon as the project was launched, the technical leader of the Fortaleza team had to go to Israel. Thanks to that, it became possible to reduce risks related to the lack of process alignment and to accomplish a better detailing of process requirements – both would hardly happen without face-to-face interaction.

Language differences were a major risk to the project. English was chosen as the standard language of the project; however, some members of the Fortaleza team presented a very basic level of the language. The project members were asked to attend English classes, even during work hours, as a way to help improving their communication skills.

Due to the time zone difference and to the Jewish tradition of *Shabbat*, meetings could only be held from Monday to Thursday until 9 A.M (Brasília time). This risk had to be incorporated to the project, as mitigation measures could not be established. It became necessary to the Fortaleza team to begin their workdays earlier on days meetings were scheduled due to the time zone difference. Because of the *Shabbat* tradition, teams only worked 4 days together, affecting their interaction.

All the project meetings were held via teleconference, using a screen sharing software as well. These were organized to evaluate the progress of the project and to add new requirements and features to it. That not only created risks such as misunderstanding the tasks that had to be developed, but also risks related to the lack of knowledge of the project current status. As a way to reduce them, at the end of each meeting Minutes were produced; these were delivered to team conferences, and then, finally, the assignments were registered on the task management software. The information inserted on the software was made available for all the participants of the project.

Making the Fortaleza team responsible for developing the programming language, which was their dominant field of knowledge, reduced technical expertise related risks. Meanwhile, the Israel team promoted an algorithm study. Also, the team based in Haifa worked on development processes that didn't demand the programming language used by the other team.

Along with trials policies, a configuration management system was embraced; it consisted on using a local data repository in Fortaleza. At the end of each development life cycle, the local data repository was synchronized with the central one, located in Porto Alegre. If the software developer was starting a new activity, he created a branch. After the testing process and at the end of the activity, the developed product was sent to the central data repository. This method avoided the introduction of error codes in the central data repository, which would compromise the software quality. Through the configuration management policy, it was possible to eradicate the risk of lacking source control of files and codes.

The risks identified in the project and how they were handled are listed in Board 2:

5. CONCLUSIONS

It is undeniable DDS brings benefits such as reducing labor cost, a better use of labor laws and round-the-clock development. However, DDS also comes with a number of risks related to communication issues, language barriers and, finally, time zone and sociocultural differences.

The risks associated with DDS go beyond technological matters, involving social and humane factors. Among the identified risks, communication is the most mentioned in the scientific literature, since as dispersion levels increase more difficult it becomes to manage it.

Based on the experience obtained from a real case study scenario combined with literature review, this work provided a broad perspective of the risks involved when dealing with DDS projects.

The mitigation strategies presented can be used to reduce arising complications when developing



Board 2. Scenario risks and mitigations

Field of Risk	Risk	Risk Management
Communication	Language Differences	The team was encouraged to take classes and to study English
	Time zone differences	Mitigation measures could not be taken; the risk was, then, incorporated to the project
Culture	Shabbat Judaic Tradition	Maximizing shared work days (monday to thursday)
Technology	The Haifa team lacked technical knowledge of the programming Language used by the Fortaleza team	The team based in Fortaleza was responsible for developing the programming language they were familiar with, while the Israel team worked on a task that did not demand any knowledge of it
	Fortaleza team's nescience of the algorithms developed by the Israel team	In order to guide the Fortaleza team, an in-depht study and clarification of doubts were held with the help of the team based in Israel.
	Source controlling files and developed software	Using a local data repository in Fortaleza, which was synchronized with the central data repository at the end of each development life cycle. Developers created branches everytime they initiated a new activity
Cooperation	Israel team's refusal to provide the algorithm developed by them	A deal was made between the teams of Fortaleza and Israel; the first promised to keep total confidentiality of the algorithm developed by Israel and agreed to use passwords and criptology to protect the computers of their developers.
	Project management difficulties due to geographically spread teams	The teams based in Fortaleza, Porto Alegre and Israel could observe the progress of the activity through the Atlassian Jira software and the web interface
Requirements Engineering	Misundertanding tasks	A task management software was used, allowing not only the attachment of images and files but also indexing a thoroughly description of the task.
	Adding and changing features	Minutes were produced at the end of meetings; they were then verified and aproved by all teams.
	Poor quality of the software developed	After doing unit testings and automated integration testings, the software was manually tested.

Source: Compiled by authors

Distributed Development Software; this allows companies to have successful outcomes when managing this kind of projects.

The risks identified in this study are recurrent in distributed projects. However, as each project has its own peculiarity, teams must learn how to determine related threats and keep working constantly to spot them.

Despite the promising results obtained in this research, which indicated the need of an effective

risk management in DDS projects, it is important to point out that the qualitative paradigm is submitted to human limitations as it uses information based on people's experiences. It should also be mentioned that this research included only one company in its analysis, which may make generalizing results harder.

Reproducing this study in other companies, technically evaluating risk management solutions and inspecting the quality of the product developed could be some of many possibilities presented as prospective works.



REFERENCES

- Audy, J. et Prikladnicki, R. (2008), *Desenvolvimento distribuído de software: desenvolvimento de software com equipes distribuídas*, Elsevier, Rio de Janeiro, RJ.
- Boehm, B. W. (1991), "Software risk management: principles and practices", *Software, IEEE*, Vol.8, No.1, pp.32-41.
- Bubeck A., Weisshardt F., Sing T., Reiser U., Hagele M., Verl A. (2012), "Implementing best practices for systems integration and distributed software development in service robotics - The Care-O-bot robot family proceedings of International Symposium on System Integration, Fukuoka, Japão, Dec 16-18, 2012.
- Carmel, E. et Agarwal, R. (2001), "Tactical approaches for alleviating distance in global software development", *Software, IEEE*, Vol.18, No.2, pp.22-29.
- Ebert, C., Murthy, B. K., Jha, N.N. (2008), "Managing risks in global software engineering: principles and practices", proceedings of International Conference on Global Software Engineering, Bangalore, India, Ago 17-20, 2008.
- Estler H.C, Nordio M., Furia C. A., Meyer B., Schneider J. (2012), "Agile vs. structured distributed software development: a case study", proceedings of 7th International Conference on Global Software Engineering, Porto Alegre, RS, Ago 27-30, 2012.
- Farias Junior I H., Azevedo R. R., Moura H. P., Silva D. S. M. (2012), "Elicitation of communication inherent risks in distributed software development", proceedings of 7th International Conference on Global Software Engineering, Porto Alegre, RS, Ago 27-30, 2012.
- Gil, A. C. (2002), *Como elaborar projetos de pesquisa*, Atlas, São Paulo, SP.
- Herbsleb, J. et Moitra, D. (2001), "Global Software Development", *Software, IEEE*, Vol.18, No.2, pp.16-20.
- Herbsleb, J. et Mockus, A. (2003), "An empirical study of speed and communication in globally distributed software development", *IEEE Transactions on Software Engineering*, Vol.29, No.6, pp.481-494.
- Jimenez M., Piattini M., Vizcaíno A. (2009), "Challenges and improvements in distributed software development: a systematic review", *Advances in Software Engineering*, Vol. 2009, pp. 1-14.
- Kiel, L. (2003), "Experiences in distributed development: a case study", proceedings of the International Workshop on Global Software Development, 2003, Portland.
- Kwak, Y. H. et Stoddard, J. (2004), "Project risk management: Lessons learned from software development environment", *Technovation*, Vol.24, No.11, pp. 915-920.
- Perry, D. E., Staudenmayer, N. A., Votta, L. G. (1994), "People, organizations, and process improvement", *Software, IEEE*, Vol.11, No.4, pp.36-45.
- Persson, J. S., Mathiassen, L., Boeg, J., Madsen, T.S., Steinson, F. (2009), "Managing risks in distributed software projects: an integrative framework", *Engineering Management, IEEE Transactions on*, Vol.56, No.3, pp.508-532.
- Prikladnicki, R et Yamaguti, M. H. (2004), *Risk Management in global software development: a position paper. proceedings of 3th International Workshop on Global Software Development*, Edinburgh, Scotland, 24th May, 2004.
- Project Management Institute – PMI (2008), *Um guia do conjunto de conhecimento sem gerenciamentos de projetos: Guia PMBOK. 4 ed. Four Campus Boulevard, Pennsylvania*.
- Runerson, P. et Host, M. (2009), "Guidelines for conducting and reporting case study research in software engineering", *Empirical Software Engineering*, Vol.14, pp. 131-164.
- Sakhthivel, S. (2007), "Managing risk in offshore systems development", *Communications of the ACM*, Vol.50, No.4, pp. 69-75.
- Steinmacher I., Chaves A. P., Gerosa M. A. (2013), "Awareness support in distributed software development: a systematic review and mapping of the literature", *Computer Supported Cooperative Work*, Vol.22, No.2, pp. 113-158.
- Wohlin C., Runerson P., Host M., Ohlsson B. R., Wesslen A. (2000), *Experimentation in software engineering - an introduction*, Kluwer Academic Publishers, Norwell, MA.
- Yin, R. K. (2005), *Estudo de caso: planejamento e métodos*, Bookman, São Paulo, SP.