Applicability of Risk Process in Software Projects in Accordance with ISO 31.000:2009

Marcelo Nogueira, Ricardo Machado

Applicability of Risk Process in Software Projects in Accordance with ISO 31.000:2009

Marcelo Nogueira[1], Ricardo J. Machado[2]

[1] Software Engineering Research Group, University Paulista, UNIP, Campus of Tatuapé, São Paulo, Brasil
marcelo@noginfo.com.br
[2] ALGORITMI Center, School of Engineering, University of Minho, Campus of Azurém, Guimarães, Portugal
rmac@dsi.uminho.pt

Abstract. In a progressively competitive global market, software development companies, under the pressure for conquering new market shares, subject themselves to business demands where the inherent risks to these operations are diversified and of exposure not always calculated. Given that a minority of such companies adopt risk management into their business processes, such exposure may affect the participation and success of these projects. To assure the quality of the software risk analysis and risk assessments are required. Among the uncertainties of software design, some risk factors should be treated: timeline, estimated costs and compliance to business requirements, among others, can be mentioned. Through a bibliographical review it was possible to produce a risk roadmap to provide to the professional in the field the understanding of risks process in a friendly way. To contribute to these software projects, this work presents the activities of a risk management process, in order to insert the culture and capacity of professionals who work in such projects, can objectively target to the mitigation of risks into which such projects are exposed. In addition, the adopted approach is in accordance to ISO 31000 standard.

Keywords: software engineering; risk management; software crisis; quality software, information systems.

1 Introduction

In a competitive environment of increasingly complex change, the appropriate management of information is crucial in the process of decision making in organizations (Nogueira, 2009).

Being this subject both comprehensive and specialized, the adoption of the practices of software engineering as a baseline of information management enables the development and consolidation of knowledge in the production of software.

These practices also prepare professionals to confidently face new challenges in the business world, strengthening their skills and abilities and keeping them up to date

on the potential of information systems and new technologies in a globally competitive business perspective.

The objective this paper is present applicability of risk management through the roadmap with critical points the process of software production identified in the literary review.

This literary review consists of a merger between the classical scientific references in the area production software and the recent consolidation the ISO 31000.

2 Software Crisis versus Software Quality

Software engineering can be defined as a set of methods, procedures and tools aimed at the production of software with quality, in other words, in accordance with customer requirements (Nogueira, 2009).

Software engineering has as primary objective the quality improvement of software products and the increase of the productivity of software engineers, in addition to meeting the requirements of efficiency and effectiveness (Maffeo, 1992).

In the study of software engineering, author Roger S. Pressman (2006) mentions the "Software Crisis", where numbers are given that express the problem with non-completion of software projects. The same author points out that one of the main factors that cause such "Software Crisis" is the lack of adoption of methods, procedures and tools in building software.

The term "Software Crisis", which began to be used in the 60s, historically alludes to a set of problems recurrently faced in the process of software development (construction, deployment and maintenance) (Maffeo, 1992).

In general terms, the "Software Crisis" occurs when the software does not meet the customers, users, developers or enterprise needs and exceeds cost and time estimates (Nogueira, 2009).

Despite the enormous variety of problems that characterize the software crisis, in computer systems development field, engineers and project managers tend to focus their concerns on the following point: "There is huge uncertainty of estimates of timelines and development costs" (Nogueira, 2009).

Many of these errors could be avoided if organizations could have a software engineering process defined, controlled, measured and improved. However, it is clear that for many IT professionals these concepts are not very clear, which certainly hampers the action of managers in the improvement of their production processes (Blaschek, 2003).

There are several techniques, methodologies and quality standards to contribute to the development of software, including risk management. Professionals who do not embrace them find difficulties in performing software projects which are free of maintenance and re-work, so directly condemning the product quality.

Adoption of software engineering leads the individual to perform the activities related to their professional role through systematic methods throughout the software life cycle, allowing the developed product to represent the company actual processes and to meet in fact the company needs.

Achieving a high quality product or service is the goal of most organizations. It is no longer acceptable to deliver products with low quality and fix the problems and deficiencies after the products were delivered to the customer (Sommerville, 2007).

Quality is a result of processes, people and technology. The relationship between product, quality and each of these factors is complex. Therefore, it is much harder to control the degree of product quality than to control the requirements (Paula Filho, 2009).

When producing software with quality, the real possibility of extracting relevant information from a system is created. This may not only contribute to the decision, but to be a factor of business excellence, enabling new business, retention and survival in an active market. Thus, it is of paramount importance to identify and analyze risks that threaten the success of the project and manage them so that the business objectives may be achieved.

Aiming at quality in the process of software production, risk management has the focus to address the uncertainties inherent to software projects, because many factors that involve technology, people and processes are in conflict and can determine whether the development of the software product will be successful or not.

According to Standish Group (2009), through a study called "Chaos Report", for projects in the area of information technology, the following conclusions were drawn (Table 1):

- 32% of projects finish on time and on budget;
- 44% of projects are challenged;
- 24% are canceled before its deployment.

Table 1. Chaos Report (Standish Group, 2009)

Projects / Year	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16	27	26	28	34	29	35	32
Contested	53	33	46	49	51	53	46	44
Cancelled	31	40	28	23	15	18	19	24
Failed	84	73	74	72	66	71	65	68

As for cost and schedule, the following information was obtained:

- Surplus in original estimated cost in 45% of the projects.
- Surplus in original schedule in 63% of the projects.

Other collected data are:

- 94% of the projects have at least one restart (Standish, 2009);
- 9% of projects in large companies come into operation within initially estimated cost and time.
- In software projects only 67% of originally proposed requirements are delivered in the end.

Despite the "Software Crisis" is not a new problem, even nowadays its impact and its negative effects are faced. The scarce use of methodologies and models of quality in Brazil indicates that this reality has to be modified.

According to the Ministry of Science and Technology (2002), only 11.8% of companies in Brazil have adopted risk management in software projects.

Due to the relevance of the theme and its direct impact on the success in producing software, the number presented by the ministry is alarming because the sample used for the research included both the major software companies and the small and medium enterprises in the country (Nogueira, 2009).

The concerning fact is that small and medium enterprises, which hold 65.1% of the software market in Brazil (MCT, 2002), lack a culture of risk management. Besides contributing to the possibility of failure in current projects, this situation undermines the still promising future opportunities that this sector needs to explore in both domestic and foreign markets.

New research in 2005 and 2008 were made by the Ministry of Science and Technology, but the item risk management was not added to the survey.

3 Risks and Software Engineering

Risk, such as science, was born in the sixteenth century, during the Renaissance. In an attempt to understand the games of chance, Blaise Pascal, in 1654, discovered the "Theory of Probability" and created the "Pascal Triangle", which determines the likelihood of possible outcomes, given a certain number of attempts (Bernstein, 1997).

Risk in the software area was represented in a systematic manner by Barry Boehm in the 80s through the Spiral Model, which has as its principle be iterative and directed to risks, because for each iteration it is performed an analysis of risk (Boehm, 1988).

Risks in software cannot be mere agenda items. They should be the "heart" of the business, as in other areas (Chadbourne, 1999).

Currently, the area that addresses risks in software engineering has evolved from an analysis within the model of development, as proposed by spiral model, to become a management technique that should permeate all the processes of software life cycle.

Risk management is understood as a general procedure for resolution of risk, ie when it is applied in any instance, the possible consequences are all acceptable, and policies to cope with the worst outcome must be defined in the process.

Risk management, in software design domain, is a defined and systematic process with the purpose of treating risk factors in order to mitigate or minimize its effects, producing a quality software product that meets customer needs, within estimated time and costs (Nogueira, 2009).

According to Robert Charette (1989), the definition of risk is:

First, risk affects future events. Present and past are irrelevant, because what is reaped today was planted by our previous actions. The issue is changing our actions today. Can opportunity be created for a different and possibly better situation tomorrow?

Secondly, this means that risk involves change, such as change of thought, opinion, action or places. Thirdly, risk involves choice and the uncertainty that choice entails itself.

Thus, paradoxically, the risk, like death and taxes, is one of the few certainties of life.

In a simplified way, a risk can be thought as a probability that some adverse circumstance will really occur. The risks may threaten the project, the software being developed or the organization.

Sommerville (2007) has described the types of risks that may affect the project and the organizational environment in which software is being built. However, many risks are considered universal and they include the following areas: Technology, personnel, organizational, tools, requirements and estimation.

The estimation of risks involves the following tasks:

- Identification of possible risks to the project;
- Analysis of these risks, evaluating their probability and likely impact;
- Prediction of corrective or preventive countermeasures;
- Prioritization of risks, organizing them according to likelihood and impact.

Risks do not remain constant during the execution of a project. Some disappear, new ones arise, and others suffer changes of probability and impact, therefore changing the priority. Therefore a monitoring report of the project along with an updated table shall be used for monitoring the risks. The estimation table should be reviewed and updated to reflect the modifications until the risks are realized or completely eliminated (Paula Filho, 2009).

The adoption of risk engineering is part of the critical success factors in software projects. The management of risks throughout the life cycle of development is critical to project success (Nogueira, 2009).

Risk management is particularly important for software projects, due to the inherent uncertainties that most projects face (Sommerville, 2007).

Project managers of information systems should regularly assess the risks during the development process to minimize the chances of failure. In particular, the problems of schedule, budget and functionality of the software can not be totally eliminated but they can be controlled through the implementation of preventive actions (Higuera, 1996).

Risk management has six well-defined activities that are: Risk identification, risk analysis, risk planning, risk monitoring, risk control and risk communication (Higuera, 1996).

The activities of risk identification and risk analysis, critical risk assessment, risk mitigation and contingency plans should be made. The methods of risk assessment should be used to demonstrate and evaluate the risks. Constraint policies of the project must also be determined at the time when discussions with all others involved take place. Aspects inherent to risks of software, such as the tendency of professionals to add features that are difficult to measure or even the risks of intangible nature of software, should influence the risk management of project (SWEBOK, 2004).

The Orange Book (2004), originally developed by the British government, now an international reference handbook, details the guidelines for good risk management, involving the following activities: Identifying risks, assessing risks, risk appetite, addressing risks, reviewing and reporting risks, communication and learning.

The ISO 31000 (2009) standard directs the policy for risk management with the following activities: establishing the context, risk identification, risk analysis, risk assessment, risk treatment, communication and consultation and monitoring and review.

4 Roadmap for Risk Management Process

After the literary review, it was possible to identify critical areas in the process of software development.

However, to support risk management in software projects, it is necessary to use a roadmap with activities where the decision maker can use it as an auxiliary instrument in the process of risk management.

Therefore, the following activities make up this roadmap: Communication and consultation; establishing the context; risk identification; risk analysis; risk evaluation; risk treatment and monitoring and review.

These activities are described below according to the complexity of application in accordance with ISO 31000.

4.1 Communication and Consultation

Communication and consultation with external and internal stakeholders should take place during all stages of the risk management process. It should take place in the beginning, with the first meeting of sensitization, during activities and in the end, with the presentation of results.

4.2 Establishing the Context

By establishing the context, the organization articulates its objectives and defines the external and internal parameters to be taken into account when managing risk, and sets the scope and risk criteria for the remaining process.

4.3 Risk Identification

The organization should identify sources of risk, areas of impacts, events (including changes in circumstances) and their causes and their potential consequences. The aim of this step is to generate a comprehensive list of risks based on those events that might create, enhance, prevent, degrade, accelerate or delay the achievement of objectives.

It is important to identify the risks associated to not pursuing an opportunity. Comprehensive identification is critical, because a risk that is not identified at this stage

will not be included in further analysis. It is recommended to use a universal framework with risks common to different designs when it is the first iteration.

4.4 Risk Analysis

Risk analysis involves developing an understanding of the risk. Risk analysis provides an input to risk evaluation and to decisions on whether risks need to be treated, and on the most appropriate risk treatment strategies and methods.

Risk analysis can also provide an input into making decisions where choices must be made and the options involve different types and levels of risk. A framework can be used with the universal risk weights established from expert opinion, especially when you do not have a knowledge base.

4.5 Risk Evaluation

The purpose of risk evaluation is to assist in making decisions, based on the outcomes of risk analysis. It defines which risks need treatment and the priority for treatment implementation.

Risk evaluation involves comparing the level of risk found during the analysis process with risk criteria established when the context was considered. Based on this comparison, the need for treatment can be considered.

4.6 Risk Treatment

Risk treatment involves selecting one or more options for modifying risks, and implementing those options. Once implemented, the provision of treatments or modification of controls must be performed.

Risk treatment involves a cyclical process of: Assessing a risk treatment; deciding whether residual risk levels are tolerable; if not tolerable, generating a new risk treatment; and assessing the effectiveness of that treatment.

4.7 Monitoring and Review

Both monitoring and review should be a planned part of the risk management process and involve regular checking or surveillance. It can be periodic or ad hoc.

Responsibilities for monitoring and review should be clearly defined. The organization's monitoring and review processes should encompass all aspects of the risk management process for the purposes of: Ensuring that controls are effective and efficient in both design and operation; obtaining further information to improve risk assessment; analyzing and learning lessons from events (including near-misses), changes, trends, successes and failures; detecting changes in the external and internal context, including changes to risk criteria and the risk itself which can require revision of risk treatments and priorities; and identifying emerging risks.

5 Conclusion

In this literary review, it was found that the authors recognize the difficulty in the production process of software. It's possible to realize that the scenario of the "software crisis" provides failure to projects. And that the adoption of software engineering and its assumptions are critical to project success. Despite the existence of activities and processes focused on the production of software, its adoption is insufficient, especially in the Brazilian context. However, when teams of software production are guided through a roadmap, it becomes easier to understand "what to do". With the defined scope it is possible to sensitize stakeholders to the adoption of risk management as a common organizational practice. The compliance roadmap in relation to ISO 31000 is essential. As future work, we intend to apply the roadmap on projects that never used the risk management process in software production.

Acknowledgements

This work has been supported by FEDER through *Programa Operacional Fatores de Competitividade – COMPETE* and by *Fundos Nacionais* through *FCT – Fundação para a Ciência e Tecnologia* in the scope of the project: FCOMP-01-0124-FEDER-022674 by Portugal and University Paulista - Software Engineering Research Group by Brazil.

References

1. ISO 31000. (2009). ISO 31000: Risk management – Principles and guidelines: ISO.
2. Bernstein, Peter. (1997). Desafio aos deuses: a fascinante história do risco. RJ: Campus.
3. Blaschek, J. R. (2003). O principal problema dos projetos de software. Rio de Janeiro.
4. Boehm, Barry. (1988). A spiral model of software development and enhancement: IEEE.
5. Chadbourne, B. C. (1999). To the heart of risk management: teaching project teams to combat risk: Pennsylvania.
6. Charette, R. N. (1989). Software Engineering risk analysis and management: McG. Hill.
7. Higuera, R. P.; Haimes, Y. Y. (1996) Software risk management technical report: CMU/SEI 96 TR 012. SEI.
8. Maffeo, Bruno. (1992). Engenharia de Software e Especificação de Sistemas. RJ: Campus.
9. MCT. (2002) Qualidade e Produtividade do Software Brasileiro. Brasília: MCT - Secretaria de Política de Informática.
10. Nogueira, M. (2009). Engenharia de Software. Um Framework para a Gestão de Riscos, Rio de Janeiro: Ciência Moderna.
11. Orange Book (2004). Management of Risk – Principles, HM Treasury, Crown, London.
12. Paula Filho. (2009). Engenharia de Software: fundamentos, métodos e padrões. RJ: LTC.
13. Pressman, R. S. (2006). Engenharia de Software. 6. ed. São Paulo: McGraw-Hill.
14. Sommerville, I. (2007). Engenharia de Software. 8. Ed. São Paulo: Pearson A.Wesley.
15. Standish. (2009). CHAOS Summary 1995...2009, Boston: Standish Group.
16. SWEBOK. (2004). Guide to the software engineering body of knowledge. USA: IEEE Computer Society.