Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 181 (2021) 827–834

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2020

Longevity of risks in software development projects: a comparative analysis with an academic environment

Luís M. Alves[a,*], Gustavo Souza[b], Pedro Ribeiro[c], Ricardo J. Machado[c]

[a]*Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal*
[b]*Universidade do Minho, Campus de Azurém, Alameda da Universidade, 4800-058 Guimarães, Portugal*
[c]*Centro ALGORITMI, Universidade do Minho, Campus de Azurém, Alameda da Universidade, 4800-058 Guimarães, Portugal*

Abstract

The study presented in this paper was developed in an academic environment. All subjects involved in this research are enrolled in the Information Systems Master's Degree of the University of Minho. The main objective of this study is to demonstrate the need for the continuity of studies about the risks presented in software development projects. For this purpose, we analyzed more than four hundred risks and about one hundred problems faced and documented by the working groups. With the collected data we defined a list of twenty risks and conducted a comparative study of these risks with others already formalized in previous industrial studies. The comparison data was analyzed and interpreted, and important conclusions were made.

Peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2020

Keywords: project management; risk management; software projects; academic projects.

* Corresponding author.
E-mail address: lalves@ipb.pt

1877-0509 © 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2020
10.1016/j.procs.2021.01.236

1. Introduction

Over the past two decades, Information Technology (IT) has had a tremendous impact on individuals, organization and society. In fact, it is practically impossible to imagine the current world without the methods, tools and facilities of IT.

Due to this phenomenon, it has generated a lot of interest in several researchers around the world, trying to identify the causes of failures in projects and what various factors that can lead to success [1]. According to [2] IT has spread at a strong pace in organizations, although some of these continue to remain without them it.

With this, due to the speed of updating the technologies involved and also the methodologies inserted in the process addressed, more technologies are experienced and we can access more information in a year than our fathers throughout their entire lifetime, in the past [3].

Subsequently, the work of [4] addresses the term "software crisis", indicating the absence of tools, methods and procedures with the maturity required for successful software development. These problems in software development, over the past few decades, have created difficulties in managing software projects [4]. We can conclude that, it is clear the need to obtain a solid foundation of project management with more targeted attention to the risks they may have.

The study of risks in software development projects is crucial to its success, so there’s a need to learn more about the errors practiced in the management of these projects, errors that are known to many managers and scholars of the area, but still continue to grow.

The *Standish Group* presents each year the CHAOS report, which shows the percentage of success or failure of IT projects [5]. The *Standish Group* definition of project success is based on the triple constraint, which has been the standard for the Project Management Institute (PMI) for several years. Using the triple constraint, the *Standish Group* evaluated projects as successful, challenged or failed. Thus, this institution applies the following definitions:

- *Successful* – A successful project is one that met all three of the triple constraints: schedule, cost, and scope.
- *Challenged* – A challenged project meets two out of three constraints, for example, delivered on time and on budget but not with the desired scope.
- *Failed* – A failed project is one that is canceled before it is completed or completed but not used.

In this report [5], it is also intended to demonstrate the success and failure factors of these same projects. In the last CHAOS report, the top five factors found in successful projects are identified, namely, (i) user involvement; (ii) executive management support; (iii) clear statement of requirements; (iv) proper planning; and (v) realistic expectations. These factors should be put on a checklist for anyone considering an IT project, whether large or small. While risk rises with size and complexity, even simple projects can fail if the participants can’t follow these five principles. Other factors that lead to the success of the projects have also been identified, but research shows that when there is presence of these five factors the probability of success is greater.

By analyzing previous data from CHAOS report, it is possible to create Fig. 1.

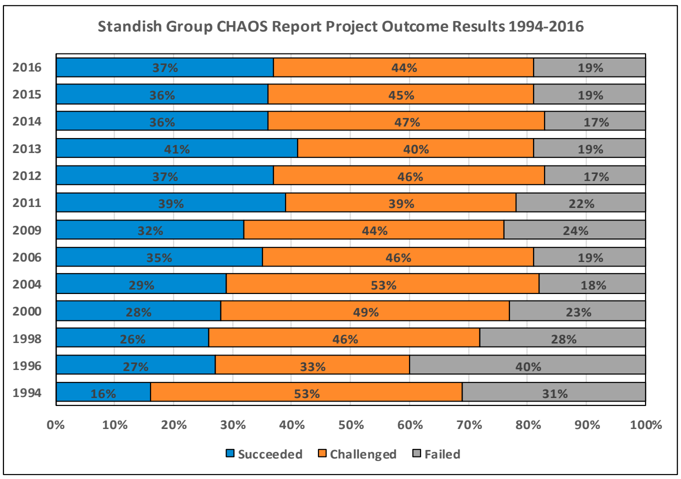

Fig. 1. Standish Group CHAOS Report Project Outcome Results 1994-2016.

It is noted that there is an increase in projects concluded successfully, but there is still a large rate of failed projects. There are studies of this nature that seek to perceive what is still done wrong in the projects. With our study, we intend to understand the level of similarity between the risks occurred in academic projects and the risks occurred and reported in industrial projects in the IT area.

Considering the strong component of Project Management that the course unit of Development of Computer Applications (DCA) provides, this unit has been selected for the comparative analysis of this study. This course unit fits in the 2nd year of the Integrated Masters Course in Engineering and Management of Information Systems of the University of Minho. The members of DCA teams (work groups) perform a software development project for six months, having as a client, a partner company of the University of Minho. The teaching methodology followed in the DCA course unit is the Project-based Learning (PBL). More detailed information can be consulted in [6].

The teams developed a software project of medium complexity, using the Unified Modeling Language (UML) notation encompassed in an iterative and incremental software development process, in this case, the Rational Unified Process (RUP). The teams followed the guidelines established by the RUP reduced model, executing the phases of inception, elaboration and construction according to the best practices suggested by CMMI-DEV (Capability Maturity Model Integration for Development) v1.3 ML2 (Maturity Label 2). This software project was to develop a Web solution using object-oriented technologies (Java or C#) and relational databases (SQL Server or MySQL), to support the information system of one local company that provided all the information about the organization and interacted directly with the teams.

In this paper, a literature review is presented in Section 2. Based on this research, in this section we present three studies of risk lists and the importance and benefits of risk management. Section 3 describes the methodology that we follow to support our research. In section 4 we present the results and discussion of our research, concretely, a comparison of existing literature with the risks identified in DCA and a final list of risks. Finally, in Section 5 we present the conclusions and future work.

2. Literature Review

For Ian Sommerville, the risk is the probability of any adverse situation happening. That author refers that the risk is a measure of the probability that the system will cause an accident. The risk is assessed by considering the hazard probability, the hazard severity, and the probability that the hazard will lead to an accident [7]. McManus refers that during the development of a software project, there are many instances prone to adversity, no project is risk free [8]. This happens because software projects are unpredictable and complex activities.

A risk possesses two characteristics that define it: uncertainty – whether it will happen or not, and loss – if it does happen, it can harm the probability of a project succeeding, partially or as a whole [4, 9-11]. Thus, a risk is considered a negative event, that can happen in a project and can provoke unsatisfactory results [12, 13].

2.1. Literature risks

For this research, we took into consideration risks from three different time periods. Starting with [14], which presented the ten risks that project managers identified as occurring more often during software projects, those are:

- Personnel shortfalls;
- Unrealistic schedules and budgets;
- Development the wrong functions and properties;
- Development of wrong user interface;
- *Gold-plating* (inclusion of functionalities not solicited by the client);
- Continuing stream of requirements changes;
- Shortfalls in externally furnished components;
- Shortfalls in externally performed tasks;
- Real-time performance shortfalls;
- Straining computer-science capabilities.

Afterward, the research made by [15] were analyzed, where a risk approach focused on ERP's (Enterprise Resource

Planning) implementation projects was made, the most common risks according to these authors are:

- Poor project team skills;
- Low top management involvement;
- Ineffective communication system;
- Low key user involvement;
- Complex architecture and high number of implementation modules;
- Bad managerial conduction;
- Ineffective project management techniques;
- Inadequate change management;
- Ineffective consulting services experiences;
- Poor leadership;
- Ineffective strategic thinking and planning strategic;

In the research made by Júnior and Chaves [16], a synthesis of the main studies in the area is carried out and the risks that are still considered important by software project managers are collected, which are:

- Problems with technical artifacts by third-parties;
- Constant changing of the technical requirements;
- Poor development environment acquaintance;
- Technical issues with development;
- System test failure;
- Bad system development management;
- Delivery failure;
- Poor component conception;
- Lack of documentation;
- Incorrect interaction between organization and system processes;
- Poor system mapping.

2.2. The importance and benefits of risk management

Although risk management is one of the greatest needs in project management, it is recognized that little has been done about it [17-20].

In 2012, Kutsch *et al.* [21], demonstrated that many project managers still neglect risk management in the course of their project and presented five key beliefs that can justify this attitude, namely:

- *Legitimacy*: managers believe that by following risk management procedures, they generate acceptance and trust among stakeholders, even if the risk management structure is announced without actually being in use;
- *Value*: they believe that risk management should be proven useful, and when there is no obvious value the managers interest in risk management decreases;
- *Competence*: they believe that by demonstrating to the customer that there is a risk that hinders the success of the project, this may jeopardize the competence of managers in front of customers;
- *Fact*: managers dissociate themselves from risk management when risks are considered fictitious or imaginary;
- *Authority*: managers fail to follow risk management when they felt they did not have the autonomy to act in mitigating risks.

With the application of risk management as a striking and disciplined part of the organizational environment it is possible to perceive various benefits. According to ISO/IEC 31010 [22], the main benefits when performing risk management are:

- Provide information to decision makers;
- Communicating risks and uncertainty;
- Assist in the establishment of priorities;
- Contribute to the prevention of incidents based on post-incident investigation;

- Meet regulatory requirements;
- Understand the risk and its potential impact on the objectives of the project.

3. Methodology

The methodological approach used was the case study. According to Fidel [23], the case study aims to understand the event under analysis and at the same time develop more general theories about the observed event. Questionnaires, interviews, observation, analysis of artefacts or other methods may be carried out.

For the comparative analysis realized in this study, all the projects performed by the teams of the Development of Computer Applications (DCA) unit course in the period 2011/12 up to 2015/16 were analyzed. In this way, we created a database consisting of the risks identified and the main issues faced by the working groups. Twenty-nine works were considered during these five years. Each team (between 12 and 16 elements) produced a list of risks ordered by their degree of seriousness, from 1 to 25. The lists of risks, collected from the working groups, were classified and organized according to the probability and impact of their occurrence and according to the consequences that these risks could have on the project.

In order to follow our approach, for the accomplishment of the treatment and analysis of the results, three steps were accomplished, namely:

1. Individualized analysis of risk exposure;
2. Comparative analysis between list of risks and problems faced;
3. Comparative analysis of the list of risks identified with the literature.

In order to identify the risks most frequently indicated by the working groups, an individual analysis of the risk exposure by impact was carried out. Then, the results obtained in the previous step were compared with the list of problems faced by the working groups during the execution of the projects. Finally, a comparative analysis of the list of risks identified with the risk found in the literature was performed. In this paper, we present with some detail the last step, where the studies were considered in three distinct periods for comparative effect. The first two steps were carried out in [24] and they are out of the scope of this paper.

4. Results and Discussion

After analyzing more than four hundred risks identified and about one hundred problems faced and documented by the working groups, it was possible to highlight the twenty risks mentioned in Table 1. We found that the risks are dispersed in different categories within the development of the project, because there were risks linked to the elements of the teams in a more individual way, specific risks in technical areas and risks related to the clients.

Among the risks presented in Table 1, there were two risks that drew attention during the study, whether due to the type in which they were framed or the way they were denoted out in the works analyzed. These risks were “Changes in requirements by the customer” and “Quality of project documentation and reports”.

About the first risk, the teams must be very attentive to this point because it is one of the main causes of delays and reformulation of task plans. They should always maintain a margin of safety for the planned time and pay great attention to the first meetings to collect the requirements. If possible, they should use audio and video recording.

About the second risk, the teams demonstrated that they ended up missing project submission dates due to the lack of revisions and the need to revalidate some points of the project. This fact seems to demonstrate that the working groups had a need to redo reports instead of just correcting them. A rigorous analysis of the templates provided by the RUP (Rational Unified Process) is essential. These templates help the teams to create a project documentation and reports with good quality.

The Boehm study [14], considered one of the first carried out in the area and where the ten most common risks in software development projects are presented. The study by Aloini, *et al.* [15] for carrying out a review of the literature on risk management in project planning in the area of ERP. Finally, a study by Júnior and Chaves [16] was used, which identified new risks for the management of information technology projects through an exploratory survey with project managers. In order to compare the risks identified in previous studies in the industry with the risks identified by DCA teams in an academic environment, we intend to understand the degree of applicability and correctness between the different environments.

Table 1. Final risk list.

Risk
Delay or non-fulfillment of dates on delivery of artifacts
Lack of effort and commitment of the team members to the project
Quality of project documentation and reports
Workload/hours for some team members
Communications difficulty between team members
Loss of team members
Shortage of time and resources
Lack of knowledge of the tools being used
Inexperience of team members
Changes in requirements by the customer
Complexity of the system functionalities used in the project
Difficulty in communicating and gathering customer requirements
Difficulty in managing subcontracting
Difficulty in managing the evaluations of other unit courses
Problems with software production
Poor knowledge of the business area
Poor quality of system architecture
Failure in artifact planning
Failure in modeling requested requirements
Lack of adequate space for work and meetings

In Table 2, it is possible to visualize the comparisons made of all the studies considered with the data collected within the academic scope in the last five years. Some risks remained present in all studies, such as non-compliance with deliveries, the various changes in requirements, the difficulty of managing third-parties' tasks and the difficulty of properly planning the schedule. While others did not present direct relationships with risks raised in the academy. Thus, it is worth noting that the fact that all risks are not directly correlated may be due to the fact that some risks have a more specific essence while others end up having a more comprehensive definition nature.

In addition, this research was carried out in order to propose a broader approach, among the studies addressed, some are directed to the ERP area and others focused on development, thus making it possible to perform simultaneous combinations between all studies. It is also important to note that all the risks listed in the academy's working groups were identified by the project manager when he became responsible for identifying the risks.

Thus, two risks call attention, the first named as "Poor leadership" may not have been mentioned in the course unit studies because the project manager is responsible for identifying the risks. Certainly, he/she never point to poor leadership as a risk to consider in the project. The second one is the risk of "gold-plating", identified by Boehm [14]. This risk is defined by some researchers as an artifice that managers use to circumvent crises with customers, but that end up generating unnecessary costs to the projects. Because of this, in [25], "gold-plating" was cited by the Computer Emergency Response Team (CERT) as one of the risks that most generate vulnerabilities in the software applications. This risk may not have arisen in any of the works analyzed because they belong to the academic environment and in addition, we should consider the lack of experience of many elements of the working groups with the resolution of problems of this nature and magnitude.

With the analysis carried out along the comparisons made between the studies, it was possible to obtain the risks that presented greater visibility by the managers of software development projects, both in projects developed in an academic environment and not.

Table 2. Comparison of existing literature with the risks identified in DCA.

DCA (2011-2016)	**Boehm (1991)**	**Aloini, Dulmin and Mininno (2007)**	**Junior and Chaves (2014)**
Delay or non-fulfillment of dates on delivery of artifacts	Real-time performance shortfalls	Ineffective strategic thinking and planning strategic	Delivery failure
Lack of effort and commitment of the team members to the project		Low top management involvement	
Quality of project documentation and reports			Lack of documentation
Workload/hours for some team members			
Communications difficulty between team members		Ineffective communication system	
Loss of team members			
Shortage of time and resources			
Lack of knowledge of the tools being used			Technical issues with development (hardware)
Inexperience of team members	Personnel shortfalls	Poor project team skills	
Changes in requirements by the customer	Continuing stream of requirements changes	Inadequate change management	Constant changing technical requisites
Complexity of the system functionalities used in the project		Complex architecture and high number of implementation modules	
Difficulty in communicating and gathering customer requirements		Low key user involvement	Incorrect interaction between organization and system processes
Difficulty in managing subcontracting	Shortfalls in externally performed tasks	Ineffective consulting services experiences	Problems with technical artifacts by third-parties
Difficulty in managing the evaluations of other unit courses			
Problems with software production	Development the wrong functions and properties		Poor component conception
Poor knowledge of the business area			Technical issues with development (business area)
Poor quality of system architecture			System test failure
Failure in artifact planning	Unrealistic schedules and budgets	Ineffective project management techniques	Poor system mapping
Failure in modeling requested requirements	Development of wrong user interface		Bad system development management
Lack of adequate space for work and meetings	Shortfalls in externally furnished components		
	Gold-plating		
	Straining computer-science capabilities		
		Poor leadership	

5. Conclusions and Future Work

In this study it was possible to conclude that in general, over the five school years analyzed, the academic teams have conducted a good risk assessment. One of the reasons for this, it is certainly the fact that teams follow current and rigorous software development practices. The use of RUP in conjunction with CMMI-DEV allows teams to follow development processes close to the references used in the industry.

Also, according to the data gathered in this study, it was possible to notice that the risks identified in the beginning of the 90's, when the propagation of software projects occurred, still remain current. The risks identified in the three selected studies are similar to the risks identified by the teams in an academic environment, although some risks are more specific in terms of granularity.

About future work, we will perform the practical validation of the final list of risks generated in other academic projects in other universities and then in other software projects in real companies. In addition, it would be interesting to study the positive aspect of risks (opportunities) in the academic and non-academic fields, since during the analysis of this study this approach was not considered. It would also be important to apply this study to projects of a different nature, regardless of whether the product is or not a software application, since the vast majority of risks are transversal to different areas.

Acknowledgements

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

[1] K.E. Papke-Shields, C. Beise, J. Quan, Do project managers practice what they preach, and does it matter to project success?, International Journal of Project Management, 28 (2010) 650-662.
[2] R.D. Galliers, A.R. Sutherland, The Evolving Information Systems Strategy, in: R.D. Galliers, D.E. Leidner (Eds.) Strategic Information Management, Butterworth-Heinemann, Oxford, Great Britain, 2003, pp. 33-63.
[3] A.S.G. Miguel, O Risco e a Gestão do Risco em Projectos de Desenvolvimento de Sistemas de Informação, Universidade do Minho, 2002.
[4] R.S. Pressman, Software Engineering: A practitioner's approach, Sixth Edition ed., McGraw-Hill, New York, USA, 2005.
[5] The_Standish_Group_International, Chaos Report 2015, in, The Standish Group International, Inc., 2015, pp. 13.
[6] L.M. Alves, P. Ribeiro, R.J. Machado, Project-Based Learning: An Environment to Prepare IT Students for an Industry Career, in: Overcoming Challenges in Software Engineering Education: Delivering Non-Technical Knowledge and Skills, IGI Global, Hershey, PA, USA, 2014, pp. 230-249.
[7] I. Sommerville, Software Engineering, 9th Ed ed., Addison-Wesley Publishing Company, Boston, Massachusetts, USA, 2011.
[8] J. McManus, Risk Management in Software Development Projects, Elsevier Butterworth-Heinemann, London, United Kingdom, 2004.
[9] A.J. Alencar, E.A. Schmitz, Análise de Risco em Gerência de Projetos, 3ª ed., Brasport Livros e Multimídia Ltda., Rio de Janeiro, Brasil, 2012.
[10] A. Miguel, Gestão Moderna de Projetos - Melhores Técnicas e Práticas, 8ª ed., FCA, Lisboa, Portugal, 2019.
[11] T. Kendrick, Identifying and Managing Project Risk: Essential Tools for Failure-Proofing Your Project, 3 rd ed., AMACOM - America Management Association, New York, USA, 2015.
[12] H. Barki, S. Rivard, J. Talbot, An Integrative Contingency Model of Software Project Risk Management, Journal of Management Information Systems, 17 (2001) 37-69.
[13] H.R. Thiry-Cherques, Modelagem de Projetos, 2ª ed., Atlas, São Paulo, Brasil, 2004.
[14] B.W. Boehm, Software risk management: principles and practices, IEEE Softw., 8 (1991) 32-41.
[15] D. Aloini, R. Dulmin, V. Mininno, Risk management in ERP project introduction: Review of the literature, Information & Management', 44 (2007) 547-567.
[16] I.G. Júnior, M.S. Chaves, Novos Riscos para a Gestão de projetos de Tecnologia da Informação com Equipes locais, Iberoamerican Journal of Project Management (IJoPM), 5 (2014) 16-38.
[17] C.W. Ibbs, Y.H. Kwak, Assessing Project Management Maturity, Project Management Journal, 31 (2000) 32-43.
[18] T. Raz, A.J. Shenhar, D. Dvir, Risk management, project success, and technological uncertainty, R&D Management, 32 (2002) 101-109.
[19] O. Zwikael, S. Globerson, From Critical Success Factors to Critical Success Processes, International Journal of Production Research, 44 (2006) 3433-3449.
[20] O. Zwikael, A. Sadeh, Planning effort as an effective risk management tool, Journal of Operations Management, 25 (2007) 755-767.
[21] E. Kutsch, D. Denyer, M. Hall, E. Lee-Kelley, Does risk matter? Disengagement from risk management practices in information systems projects, European Journal of Information Systems, 22 (2013) 637-649.
[22] International Electrotechnical Commission ISO/IEC 31010:2019, Risk management — Risk assessment techniques, in, International Organization for Standardization (ISO), 2019, pp. 264.
[23] R. Fidel, The Case Study Method: A Case Study, Library and Information Science Research, 6 (1984) 273-288.
[24] G.C. Souza, Gestão de Risco em Projetos Académicos de TI: Estudo de Caso, Universidade do Minho, Guimarães, Portugal, 2016.
[25] A.A.M. Chowdhury, S. Arefeen, Software Risk Management: Importance and Practices, International Journal of Computer and Information Technology, 02 (2011) 49-54.