Software Quality Journal (2019) 27:1149–1174
https://doi.org/10.1007/s11219-018-9427-5

Risk factors in software development projects: a systematic literature review

Júlio Menezes Jr[1] · Cristine Gusmão[2] · Hermano Moura[1]

Published online: 7 November 2018

Abstract

Risks are an inherent part of any software project. The presence of risks in environments of software development projects requires the perception so that the associated factors do not lead projects to failure. The correct identification and monitoring of these factors can be decisive for the success of software development projects and software quality. However, in practice, risk management in software development projects is still often neglected and one of the reasons is due to the lack of knowledge of risk factors that promoted a low perception of them in the environment. This paper aims to identify and to map risk factors in environments of software development projects. We conducted a systematic literature review through a database search, as well as we performed an assessment of quality of the selected studies. All this process was conducted through a research protocol. We identified 41 studies. In these works, we extracted and classified risk factors according to the software development taxonomy developed by Software Engineering Institute (SEI). In total, 148 different risk factors were categorized. The found evidences suggest that risk factors relating to software requirements are the most recurrent and cited. In addition, we highlight that the most mentioned risk factors were the lack of technical skills by the staff. Therefore, the results converged to the need for more studies on these factors as fundamental items for reduction of failure level of a software development project.

Keywords Software risk management · Risk factors · Project management · Systematic literature review

✉ Júlio Menezes, Jr
jvmj@cin.ufpe.br

[1] Center of Informatics (CIn), Federal University of Pernambuco (UFPE), Recife, PE, Brazil

[2] Department of Biomedical Engineering (DEBM) – Center of Technology and Geosciences (CTG), Federal University of Pernambuco (UFPE), Recife, PE, Brazil

1 Introduction

It is understood that project objectives are achieved when they are conducted successfully in terms of schedule and costs, with a good level of client acceptance, which controlled scope changes inside politics and standards of the organization and that can be used as a reference model (Wysocki 2011; Kerzner 2017). We can infer that project management is not a trivial task, given the multiplicity of factors taken into account. An effective application of project management techniques is essential for projects to achieve their objectives.

In software development projects, it is not different. Software project management usually focuses on four pillars: personnel, product, process, and project (Pressman 2005), that is, software engineering activities comprise human factors, associated with the delivery of artifacts, using a set of methods and tools that govern a project plan and must be solid. From this perspective, we can affirm that software projects are essentially risky ventures (Charette 1989; Boehm 1991; De Marco 1997; Oliveira et al. 2012). In this sense, software development projects have specific characteristics that make them complex from the managerial viewpoint. Among these characteristics, three stand out: human interaction, high level of complexity, and product versatility (Sarigiannidis and Chatzoglou 2014; Jorgensen 1999; Subramanian et al. 2007). The most common factors that lead to software projects to fail, according to Charette (2005), are unrealistic goals, wrong estimations, poorly defined system requirements, poor presentation of project status, and unmanaged risks, that is, even its wide application in different areas, software development has a reputation to failure (Savolainen et al. 2012). In this way, it is crucial to systematize strategies that at least seek to identify and to monitor the factors that can lead software development projects to poor performance or failures.

The identification of risks or risk factors is considered the most influent activity of the risk management (De Bakker et al. 2010; López and Salmeron 2012) and widely used in both agile and traditional software development methods (Neves et al. 2014). A risk factor is defined as a condition that can pose a serious threat to the successful completion of a software development project (March & Shapira 1987). The inefficiency in risk identification process in the development of complex systems is considered one of the main causes of project failures (Reeves et al. 2013). Several studies point out that in fact, there is a direct relationship between risk management and success or improvement of the performance of software development projects (Jiang and Klein 2000; Jiang et al. 2001; Raz et al. 2002; Wallace and Keil 2004; Wallace et al. 2004a; De Bakker et al. 2010; Han and Huang 2007). These studies show that at least the risk factors should be identified and well controlled in order for projects to achieve their objectives. Thus, the identification of risk factors plays a crucial role in the success and in the performance of software development projects.

In this context, this paper aims to map risk factors for software development projects through a systematic literature review. The main goal of this work is to find and to categorize evidence of risk factors of software development projects through a comprehensive and rigorous literature review. The usage of a systematic literature review allows extending the reach of results, as well as to bring a deeper insight into the state of art in the context of software development projects. So, the conjecture we want to investigate this systematic literature review is that it is important to get a comprehensive view of risk factors in environments of software development projects.

This work is part of a Ph.D. research whose main objective is to propose and assess an indicator that aims to measure the risk level of a project in the context of multiple software development projects using two sources of information: the risk factors and the project characterization factor. The Ph.D. proposal starts from the premise that the use of indicators can contribute for better perception of risk factors and, as a consequence, better decision-making in order to avoid unsuccessful projects. So, this paper aims to identify and categorize sources of information to measure the risk level of a project, as well as to better support the activities of risk identification, analysis, controlling, and strategies to risk response in environments of software development projects. Therefore, the results of this paper use risk factors as source of information to measure the risk level of a software development project, once the most prevalent categories of risk factors, for example, may suggest higher degree of importance of these categories in the process of risk measurement in the context software development projects.

After this introductory section, this paper is organized as follows: Section 2 presents theoretical foundation about software risk management and risk factors, as well as related works. Section 3 describes in detail the conducted research method. Section 4 presents the results. Section 5 discusses in depth the findings and raises questions and implications for researchers and practitioners. Finally, Section 6 presents conclusions, limitations, and future work.

2 Theoretical foundation

This section provides the corroborating foundations for the data collection performed in the systematic literature review and presents related work.

2.1 Risks and risk management in software development projects

The management of projects is not a trivial task. A set of factors, both quantitative and qualitative, is necessary to guarantee project success or improve the project performance. Undoubtedly, risks practically permeate a whole life cycle of a project, and its biggest sources of information are the uncertainties. The term "risk" derives from the early Italian *risicare*, which means "to dare" (Gerrard and Thompson 2002). As a science, the risk was born in sixteenth century, during the Renaissance, practically laying the foundations of probability theory (Hall 1998).

Some authors, specifically in Software Engineering, focused on the perception that risk measure is associated to adverse factor measures (Hall 1998), using as parameters the probability of loss and the consequences in case of loss, in which the product is called risk exposure. Pfleeger et al. (2001) and Boehm (1989) suggest that three aspects are associated to software risk: (i) the loss associated to event, (ii) the probability of occurrence of the event, and (iii) the degree to which we can change the consequences of the event, being defined as risk exposure, that is the product of the likelihood by the loss.

Risk management is the application of skills, knowledge, tools, and techniques to reduce threats to an acceptable level while maximizing opportunities (Heldman 2010). And the risk management in software projects is the practice of assessment and control of risks that affect projects, processes, and products of software (Hall 1998). An important point to be considered in software projects is the communication, especially of technical risks, that are often known but poorly communicated (Carr et al. 1993). According to Barry Boehm, considered the father of risk management in software projects, "risk management is important specially because it helps people to avoid disasters, rework, and cancellation of projects, and helps to stimulate a situation of success in software projects." (Boehm 1989).

Most of the researches reported in literatures about software risk management focus on the identification and analysis of risk—i.e., the assessment of risks (Boehm 1991; Fairley 1994; Dorofee et al. 1996; Hall 1998; Kontio 2001; Goguen et al. 2002; DoD 2006; Van Loon 2007). However, according to Bannerman (2015), few studies focus on the adoption of risk management in software development projects, since there is an awareness regarding the usage of its practice, but it is poorly applied.

In the last years, several works related to the application of artificial intelligence techniques in risk assessment have appeared, such as grey correlation (Qinghua 2009), probabilistic terms (Fu et al. 2012), fuzzy theory (Tang and Wang 2010; Salmeron and Lopez 2012), Bayesian networks (Fan and Yu 2004), and case-based reasoning (Trigo et al. 2008). Most of these techniques use risk factors as a source of information to predict risks. The area of risk management in software projects, even considered important, still needs several advances, both in research and in practice. Other studies show that the explicit practice of risk management contributes to performance improvement and increases the success of projects (Jiang and Klein 2000; Jiang et al. 2001; Raz et al. 2002; Wallace and Keil 2004; Wallace et al. 2004a; Wallace et al. 2004b; De Bakker et al. 2010; Han and Huang 2007).

2.2 Risk factors in software development projects

Fundamentally, risk factors are the source and cause of each hazard (Silva 2011; Brasiliano 2009). They are the internal and external factors that contribute and/or influence the occurrence of the risk. In this perspective, the risk is the sum of a set of risk factors that if materialized, constitute a hazard. Therefore, risk factors contribute to reducing the abstraction level of a risk. For example, a particular organization develops software without following a development methodology, or a coding standard, much less there are training policies. We can affirm that they are conditions that lead to the rework hazard, that is, the risk can be understood as a set of factors that lead to a certain hazard. This situation is better presented in Fig. 1.

A risk factor alone cannot adequately aggregate value in project management; it only makes sense when there is a clear notion of probability of occurrence and the associated impacts and consequences. For this reason, the use of classification of risks is important for better identification of risks sources. A Risk Breakdown Structure (RBS) is an interesting tool, since it is a risk-oriented grouping, which organizes in a structured way, classifies, and defines the exposure of the identified project risks, providing a hierarchical structure of potential risk factors (Hillson 2002). One known proposal of RBS for software development projects is the risk taxonomy proposed by Carr et al. (1993).

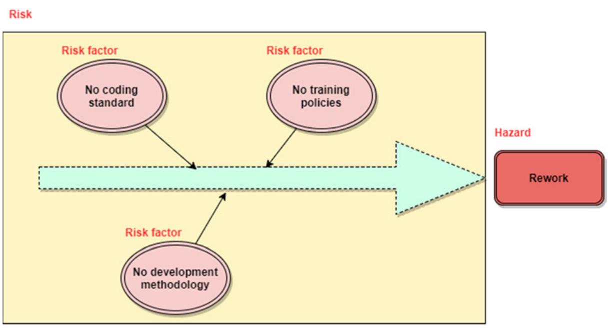

Fig. 1 Risks versus risk factors versus hazards

2.3 Related works

In general, the area of risk management in software development projects still requires greater maturity regarding systematic literature reviews or mapping studies. There is little evidence of application of secondary studies in this area. One of them is the work of Nurdiani et al. (2011), which presents a systematic review with the objective of identifying risk mitigation strategies in global software development. Alam et al. (2012) performed a systematic review seeking to identify adverse factors in knowledge sharing management, in the context of offshore software outsourcing.

Pa and Jnr (2015) searched for activities, processes, frameworks, and models that can support the decision-making process for risk mitigation in software projects. Finally, Khan et al. (2014) identified communications risks in the context of global software development, and Oliveira et al. (2012) performed a systematic mapping study of risk factors in the context of distributed software development.

Deduced from the above, therefore, systematic literature reviews related to the identification and categorization of risk factors in environments of software development projects are not readily available. In other words, no systematic review has been done yet. Therefore, it would be interesting to perform a comprehensive overview of risk factors in this context, and the application of a systematic literature review can be useful and insightful to provide this work.

3 Research method

For conducting this systematic literature review, we used the guidelines provided by Biolchini et al. (2005) and Kitchenham and Charters (2007). The following activities comprised the research method:

1) *Planning the review*: it consists of the definition and assessment of the research protocol that defines all the necessary procedures for conducting the review (next activity). The research protocol in this work is composed by the following elements: (i) research questions (presented at Section 3.1), (ii) search strategy (Section 3.2), (iii) database selection (Section 3.3), (iv) inclusion and exclusion criteria (Section 3.4), and (v) strategy for data extraction (Section 3.5).
2) *Conducting the review*: it is the whole execution of the planning process. It consists on the most exhaustive activity of the systematic literature review, comprising the database search, selection of studies that answer the research questions, and the extraction of data for analysis and posterior synthesis and discussion.
3) *Reporting the review*: it is just the writing up of the systematic literature review for dissemination among the potentially interested parties.

3.1 Research questions

To perform the review, the following research questions (RQ) are defined:

RQ 1. What are the evidence of risk factors in software development projects?
RQ 2. In which class do all the found risk factors fit?

RQ 2.1. What are the rigor values of the reported studies?
RQ 2.2. What are the relevance values of the reported values?

RQ 1 can be considered the main question of the review and simply seeks to map risk factors of software development projects. So, after the selection of studies that answer this research question, the presented risk factor of each study is collected. RQ 2 seeks to categorize the identified risk factors using a taxonomy of risks for software development projects (Carr et al. 1993) and to perform a deeper analysis of the risk factors. Finally, RQ 2.1 and RQ 2.2 analyze respectively, the application of research methods and the applicability in the industry of each study, using the checklist proposed by Ivarsson and Gorschek (2011). Section 3.5 explains how data was collected, aiming to answer each research question.

3.2 Search strategy

The search strategy defines the necessary activities to be performed in order to answer the research questions (Table 1). These are:

To define the search strings, we combined the experience and knowledge of the authors in the domain and dataset of known studies, called initial quasi-gold standard (Initial QGS), as can be seen at Table 2. The initial QGS is a set of known studies by the authors before starting the planning of the systematic literature review and that also answers the research questions. This reduced set of studies is used to evaluate the research protocol (Zhang et al. 2011), once it is expected that the studies that compose the initial QGS are present in the results of the systematic review, providing greater reliability to the research.

This systematic literature review was executed between January 2017 and April 2017. We analyzed studies prior to December 2016 only on digital libraries.

3.3 Database search and duplicate elimination

To perform searches in the selected sources, we defined a set of strings as presented in Table 3. We used these strings to search title, abstract, and keywords fields.

To obtain a reliable result, we performed an analysis of sensitivity, which is defined as the number of relevant reports identified divided by the total number of existing relevant reports (Higgins and Green 2011). Therefore, the sensitivity of the used strings to proceed with the selection of works was calculated by dividing the found Initial QGS by the total number of papers from Initial QGS. In this work, the sensitivity is 83.3% (5 of 6 papers), a value considered satisfactory for the analysis of the studies.

Table 1 Activities of the systematic literature review

Step	Description
1	Database search and extraction of studies using search strings
2	Duplicate elimination
3	Selection of works by applying inclusion-exclusion criteria
4	Extraction of risk factors from selected works
5	Categorization of risk factors
6	Rigor-relevance assessment of the selected works
7	Synthesis

Table 2 Initial quasi-gold standard (Initial QGS)

ID	Reference
QGS 1	Boehm, B. W. (1991). Software risk management: principles and practices. IEEE Software, 8(1), 32–41. https://doi.org/10.1109/52.62930
QGS 2	Barki, H., Rivard, S., & Talbot, J. (1993). Toward an Assessment of Software Development Risk. Journal of Management Information Systems, 10(2), 203–225. https://doi.org/10.1080/07421222.1993.11518006
QGS 3	Ropponen, J., & Lyytinen, K. (2000). Components of software development risk: how to address them? A project manager survey. IEEE Transactions on Software Engineering, 26(2), 98–112. https://doi.org/10.1109/32.841112
QGS 4	Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. Journal of Management Information Systems, 17(4), 5–36. https://doi.org/10.1080/07421222.2001.11045662
QGS 5	Wallace, L., Keil, M., & Rai, A. (2004). How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. Decision Sciences, 35(2), 289–321. https://doi.org/10.1111/j.00117315.2004.02059.x
QGS 6	Takagi, Y., Mizuno, O., & Kikuno, T. (2005). An Empirical Approach to Characterizing Risky Software Projects Based on Logistic Regression Analysis. Empirical Software Engineering, 10(4), 495–515. https://doi.org/10.1007/s10664-005-3864-z

For paper search, we used the following electronic libraries: ACM digital library (advanced search)—http://dl.acm.org, Engineering village (expert search)—https://www.engineeringvillage.com/,IEEExplore (command search)—http://ieeexplore.ieee.org/, Science direct (advanced search)—http://www.sciencedirect.com, and Scopus (document search)—http://www.scopus.com/.

The combination of strings presented in Table 3 was exactly applied at databases Engineering village, Science direct, and Scopus. However, due to database limitations, the strings had to be adjusted to match ACM digital library and IEEExplore. For ACM digital library, we performed a search that matches any of these strings: “risk factor,” “risk component,” “risk list,” “list of risks,” “risk dimension,” “risk taxonomy,” “risk class,” “risk category.” This adaptation was necessary because the database did not provide mechanisms to combine Boolean characters. IEEExplore library has a limitation regarding the maximum number of search terms: 15, so the authors used the following keywords:

```
("Software development" OR "Software project") AND ("Risk management" OR "software risk"
OR "software development risk" OR "software Project risk") AND ("risk factors" OR "risk
  Component" OR "risk List" OR "risk Category" OR "risk items" OR "characterization")
```

To minimize the chances of wrongly eliminating relevant papers, we used strings in ACM digital library and IEEExplore to be as comprehensive as possible. For all databases, we

Table 3 Search strategy

Strings	
Population	“Software project management” OR “Software development project” OR “Software development” OR “Software project”
Intervention	“Risk management” OR “software risk” OR “software development risk” OR “software risk management” OR “software Project risk”
Outcomes	“risk factors” OR “risk Component” OR “risk List” OR “list of risks” OR “risk Dimension” OR “risk Taxonomy” OR “risk Class” OR “risk Category” OR “Ranking” OR “definition” OR “measure” OR “risk items” OR “characterization”
	Search strategy: Population AND Intervention AND Outcomes

looked for the strings at title, abstract, and keywords. After elimination of duplicates, 446 papers were finally added to the initial database search to be analyzed according to inclusion and exclusion criteria by reading title, abstract, and keywords. If there were any doubts, the paper was completely read, seeking for risk factors.

3.4 Inclusion and exclusion criteria

The paper is included if it satisfies two conditions: (i) it discusses project risk management in software development project and (ii) it proposes or maps a list of risk factors for software development projects. Additionally, if at least one of these exclusion criteria is satisfied, the paper is excluded: (i) it is not in English, (ii) it is not a primary study, (iii) it is not available to download, (vi) it presents risk factors whose description do not allow their understanding, and (v) it presents only risk categorization, classification, or taxonomy proposals.

3.5 Data extraction strategy

Data analysis is performed with the selected papers after the application of inclusion-exclusion criteria. What drives data extraction are the RQs, and it is performed using as the basis the full reading of the selected papers. For each selected paper, the following information was extracted:

1. *Risk factors of software development projects*: each identified paper lists risk factors. These factors were extracted and cataloged. They are associated with RQ 1 and RQ 2;
2. *Classification of risk factors found*: after cataloging, the listed factors were classified according to the risk taxonomy proposed by SEI (Carr et al. 1993). This consolidated framework allowed to group the factors in common and to perform the counting of occurrences of each one. The choice of SEI risk taxonomy is due to the fact that it provides an organized way to categorize risk factors, and it is a consolidated report in software risk management. It is associated with RQ 2;
3. *Year*: the publication year. It is associated with RQ 1;
4. *Quality of publications*: h-index of publications. It is a metric of productivity and citation impact of publications and is associated with RQ 1;
5. *Rigor-relevance analysis*: consists in the application of an assessment checklist proposed by Ivarsson and Gorschek (2011), which allows the assessment of the quality of works using two criteria: rigor and relevance. The first one allows assessing the methodological rigor, whereas the second one assesses their relevance for the industry. It is associated with RQs 2.1 and 2.2.

4 Results

This section shows the results found, including the analysis of the selected papers. The database search retrieved 671 studies, and the duplication elimination removed 226. Therefore, 446 studies composed the initial dataset, as can be seen in Table 4. At the end, 41 papers composed the final dataset, and Fig. 2 summarizes the performed work.

After duplicate elimination, we applied the inclusion and exclusion criteria through the reading of titles, abstracts, and keywords, resulting in 63 pre-selected studies. After fully reading these studies,

Table 4 Database search results

Database	Results
ACM digital library	147
Engineering village	164
IEEExplore	65
Science direct	28
Scopus	267
Total	671
Initial dataset	446

40 were considered as the ones that in fact answered the research questions. One study from initial QGS that did not appear in the search results was included for data extraction process, resulting in 41 selected studies. Therefore, 41 studies composed the final dataset and were analyzed by data extraction. Table 13 in the Appendix lists the selected papers listed in chronological order (S1 to S 41). All the processes presented in Fig. 2 were performed between February and April 2017. Next sections present the extracted data as presented in Section 3.5.

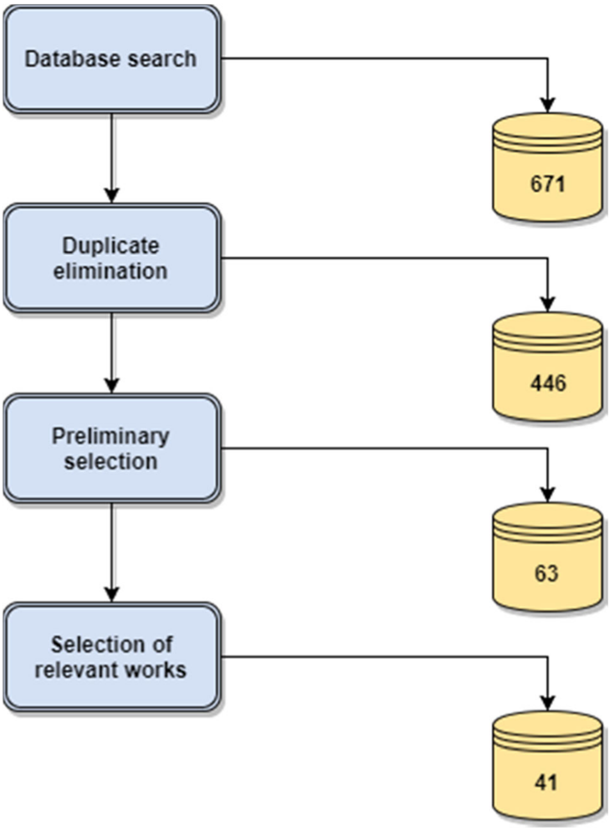

Figure 2 Summarization of the search

4.1 Evidence of risk factors of software development projects and their classification

A total of 148 different risk factors were mapped in this work. It represents an average of approximately 9 occurrences for each risk factor. According to SEI risk taxonomy (Carr et al. 1993), 51 risk factors belong to "Product engineering" class, 41 belong to "Development environment" class, and 56 belong to "Program constraints" class. Table 5 presents the 10 most mentioned risk factors for all classes.

For example, the risk factor "Staff does not have required skills" had 55 occurrences. This means that among the selected studies, we found 55 occurrences of factors related to the same thing so that they were grouped in one single factor. Some examples of occurrence of the factor "Staff does not have required skills" are the following: Personnel shortfalls (S1), Lack of development expertise in team, (S2), Lack of technical expertise (S8), Lack of available skilled personnel (9), and Team members lack specialized skills required by the project (12). It is important to note that there have been occurrences in more than one factor. For example, the Lack of skills and experience (S13) occurrence is considered in 2 factors: "Staff inexperience" and "Personnel shortfalls".

As can be observed, each factor was identified and cataloged according to the number of occurrences of the evidence found. Therefore, all 41 studies present lists of risk factors. These factors were organized according to SEI Taxonomy (Carr et al. 1993), by the identification of classes and respective elements. We can observe a predominance of factors associated with "Product engineering" class, being 5 associated with "Requirement" element. On the other hand, the most mentioned factor is related to lack of project team qualification, which was mentioned 55 times. It is important to highlight that a total of 1414 occurrences were identified, of which 353 (approximately 25%) were mentioned in Table 5.

Table 6 summarizes the number of occurrences of risk factors by class and element. Four hundred eighty-three of the factors are in "Product engineering" class, and within this class, 244 are associated with "Requirements" element. In the class "Development environment," that has 416 occurrences, there is a greater trend to equilibrium, but with a high concentration of factors on the element "Management process." Finally, the class "Program constraints" presents 515 occurrences and privileges the elements "Resources" and "Program interfaces."

Table 5 Top 10 risk factors—all classes

Risk factor	No. of occurrences	Class	Element
1. Staff does not have required skills	55	Program constraints	Resources
2. Requirement ambiguity	44	Product engineering	Requirements
3. Bad commitment of the user/customer	37	Program constraints	Program interfaces
4. Requirement changes	32	Product engineering	Requirements
5. Introduction of new technology	30	Product engineering	Requirements
6. Unstable organizational environment	30	Program constraints	Program interfaces
7. Shortfalls in externally furnished components/bad interfaces	27	Product engineering	Design
8. Technical complexity	25	Product engineering	Requirements
9. No planning or inadequate planning	25	Development environment	Management process
10. Incomplete requirement	24	Product engineering	Requirements
10. Quality of the specifications/documentation	24	Product engineering	Engineering specialties

Table 6 Occurrences of risk factors by class and respective element

Class/element	No. of occurrences	Percent (by class)
Product engineering	483	
Requirements	244	51
Design	77	16
Code and unit test	40	8
Integration and test	62	13
Engineering specialties	60	12
Development environment	416	
Development process	91	22
Development system	47	11
Management process	138	33
Management methods	55	13
Work environment	85	20
Program constraints	515	
Resources	260	50
Contract	23	5
Program interfaces	232	45

Tables 7, 8, and 9, in turn, present the top 10 risk factors of each class. In "Product engineering" class, the issue of bad clarity or ambiguity of requirements is cited as the most recurring factor among the found studies, accounting for 6 of the factors presented in Table 6. The risk factors presented in Table 6 represent 53% of total occurrences of "Engineering product" class.

There is some uniformity in the distribution of risk factors of the class "Development environment," with a greater slope to the factor associated with a bad planning of a project. The risk factors presented in Table 8 representing approximately 54% of the total of occurrences of this class are represented in Table 7.

In the class "Program constraints," issues associated with team qualification are the most relevant risk factors according to evidence found in the research. The factors of low commitment of user or client, as well as negative influences of the organizational environment, are highlighted. As expected, the element "Resources" and "Program interfaces" are the highlighting ones in top 10 presented in Table 8. The risk factors presented in Table 8 represent approximately 51% of total occurrences of "Product constraints" class.

Table 7 Top 10 risks factors—Product engineering class

Risk factor	No. of occurrences	Element
1. Requirement ambiguity	44	Requirements
2. Requirement changes	32	Requirements
3. Introduction of new technology	30	Requirements
4. Shortfalls in externally furnished components/bad interfaces	27	Design
5. Technical complexity	25	Requirements
6. Incomplete requirements	24	Requirements
7. Quality of the specifications/documentation	24	Engineering specialties
8. Lack of internal system Integration	18	Integration and test
9. Developing the wrong functions and properties	17	Requirements
10. Lack of proper tests	15	Integration and test

Table 8 Top 10 risks factors—Development environment class

Risk factor	No. of occurrences	Element
1. No planning or inadequate planning	25	Management process
2. Low commitment of staff	20	Work environment
3. Insufficient discipline and standardization	19	Development process
4. Unclear project objectives	19	Management process
5. Ineffective communications between team members	19	Work environment
6. Scope changes	18	Management process
7. Project progress not monitored closely enough	18	Management methods
8. Development methodology was inappropriate for the project	16	Development process
9. Conflicts between team members	15	Work environment
10. Lack of effective project management methodology	14	Development process
10. Inappropriate CASE tools	14	Development system
10. Project manager's experience	14	Management process
10. Low morale	14	Work environment

4.2 Study distribution by year

Figure 3 illustrates the study distribution by year. As can be seen, there is a greater concentration of recent studies, but we believe that it happens only because the electronic databases are being even more used.

However, the oldest study dates from 1991, written by Barry Boehm, who is considered the father of risk management in software engineering. The results reinforce this statement and demonstrate that risk management in software projects is a relatively new area.

Regarding the risk factors per year of published works, analyzing the top 5 most mentioned risk factors by stages, we observe the following results:

- *1991 to 2000*: (i) Requirement changes, (ii) Shortfalls in externally furnished components/bad interfaces, (iii) Staff does not have required skills, (iv) Technical complexity, (v) Unrealistic schedule;
- *2001 to 2010*: (i) Bad commitment of the user/customer, (ii) Staff does not have required skills, (iii) Shortfalls in externally furnished components/bad interfaces, (iv) Requirement ambiguity, (v) Introduction of new technology;
- *2011 to 2016*: (i) Requirement ambiguity, (ii) Staff does not have required skills, (iii) Quality of the specifications/documentation, (iv) Unstable organizational environment, (v)

Table 9 Top 10 risks factors—Program constraints class

Risk factor	No. of occurrences	Element
1. Staff does not have required skills	55	Resources
2. Bad commitment of the user/customer	37	Program interfaces
3. Unstable organizational environment	30	Program interfaces
4. Bad estimation of resources	22	Resources
5. Unrealistic schedule	21	Resources
6. Staff inexperience	19	Resources
7. High turnover	17	Resources
8. Disagreement with customer	17	Program interfaces
9. Lack of top management commitment/support to the project	16	Program interfaces
10. Unstable budget	15	Resources
10. Unreasonable customers	15	Program interfaces

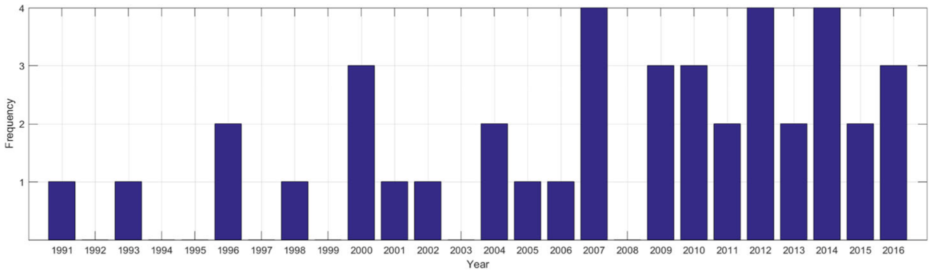

Fig. 3 Study distribution by year

Developing the wrong functions and properties and Lack of internal system Integration (they presented the same number of occurrences).

As can be seen, the factor related to staff skills is the most critical, regardless of the analyzed stage. We also realize that requirement risk factors are always recurrent. Finally, as the stages go by, there is evidence that there is greater emphasis on organizational and environmental aspects as factors that can influence the success or performance of software development projects.

4.3 Quality of publications

Table 10 presents the quality of publications, organized by frequency, h-index, and type (journal or conference). Thirty-five different publications were found, which demonstrates a high variability of studies. The publications are sorted by h-index. Another point is the predominance of studies published in journals, which commonly have a greater impact than conferences. Only 8 conference publications (proceedings) compose the selected studies. The h-index is defined according to Scimago Journal & Country Rank (SJR 2017), which is a known portal for evaluating impact factor of publications. As can be seen, a good part of the studies presents h-index greater than 30, evidencing the high quality of the publications found.

Table 11 presents the top 10 studies that have the highest number of citations according to Google Scholar (2017). It is noteworthy that studies S1, S2, S6, S9, and S11 also belong to Initial QGS. This fact can give a greater reliability to this set, contributing to quality assurance of the found evidence. Only the studies S21 (2009), S28 (2012), S31 (2013), S34 (2014), and S40 (2016) do not present any citation, but, except for S21, all the others are recent.

Finally, Fig. 4 presents the citation graph. The highlighted studies compose the Initial QGS. It is quite noticeable how these studies are strongly related. It occurs, because the most recent works often refer the older ones, with the result that the oldest study (S1) is mentioned by most of the studies. The most cited studies among the final dataset are in this order: S1 (cited by 29 different studies), S2 (16), S5 (15), and S9 (13).

4.4 Rigor-relevance analysis

Aiming to answers to RQ2.1 and RQ 2.2, a study of rigor and relevance analysis was carried out. The purpose was to consider the evaluation from a methodological perspective, but also looking the industrial relevance. Figure 5 summarizes the analysis of rigor (x-axis) and relevance (y-axis). The rigor rubric average was 1.9, whereas relevance rubric average was 3.4.

Table 10 Quality of publications—H-index (SJR 2017)

Publication	No. of studies	H-index	Type
Communications of the ACM	2	157	Journal
IEEE Transactions on Software Engineering	1	128	Journal
Information and Management	1	119	Journal
Journal of Management Information Systems	3	107	Journal
International Conference on Software Engineering	1	89	Conference
IEEE Software	1	84	Journal
Decision Sciences	1	82	Journal
Journal of Systems and Software	1	72	Journal
IEEE Transactions on Engineering Management	1	69	Journal
Information and Software Technology	3	67	Journal
Journal of Information Technology	1	55	Journal
Empirical Software Engineering	1	45	Journal
Journal of Computer Information Systems	1	43	Journal
ACM SIGMIS Database	1	39	Journal
Benchmarking	1	38	Journal
Studies in Computational Intelligence	1	32	Journal
AT&T technical journal	1	30	Journal
Hawaii International Conference on System Sciences	1	28	Conference
International Journal of Information Technology and Decision Making	1	28	Journal
International Journal of Physical Sciences	1	21	Journal
Communications in Computer and Information Science	1	19	Journal
International Symposium on Empirical Software Engineering and Measurement	1	16	Conference
International Journal of Services and Standards	1	16	Journal
Project Management Journal	1	16	Journal
International Review on Computers and Software	1	13	Journal
Journal of Information and Knowledge Management	1	12	Journal
International Journal of Software Engineering and its Applications	1	9	Journal
Journal of Computing and Information Technology	2	4	Journal
Americas Conference on Information Systems	1	3	Conference
Workshop on Embedded Systems Education	1	2	Conference
Malaysian Software Engineering Conference	1	1	Conference
ACM SIGCPR Conference on Computer Personnel Research	1	0	Conference
Procedia Technology	1	0	Journal
International Conference on Computer and Information Sciences	1	0	Conference
International Journal of Hybrid Information Technology	1	0	Journal

Table 11 Top 10 cited studies

Study	No. of citations	Publication	Publication type
S1	1953	IEEE Software	Journal
S9	1180	Journal of Management Information Systems	Journal
S5	873	Communications of the ACM	Journal
S2	861	Journal of Management Information Systems	Journal
S11	464	Decision Sciences	Journal
S6	420	IEEE Transactions on Software Engineering	Journal
S12	417	Information and Management	Journal
S20	189	Communications of the ACM	Journal
S10	170	Information and Software Technology	Journal
S17	114	Journal of Management Information Systems	Journal

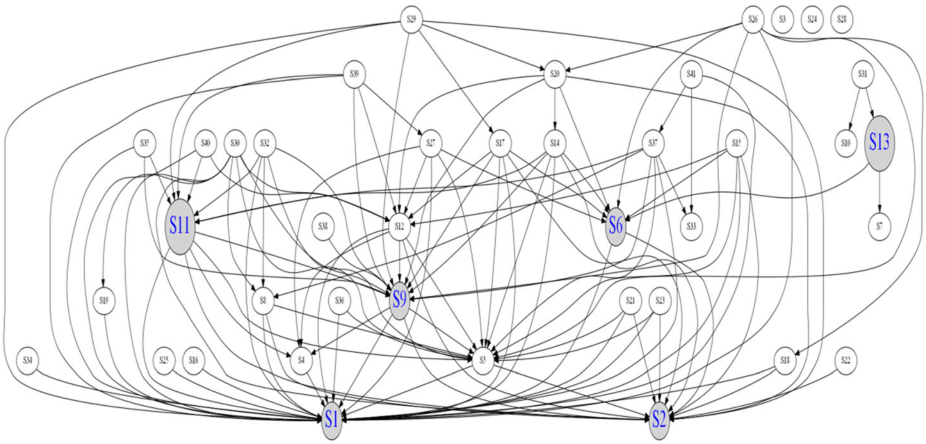

Fig. 4 Citation graph

As can be seen, 8 studies presented the highest rigor and relevance grades (rigor = 3 and relevance = 4), whereas only 1 study presented the lowest grade (rigor and relevance = 0). To facilitate the visualization of classification of these studies, we used the distribution presented by Munir et al. (2016) and Vasconcellos et al. (2017), which describes four possible combinations:

- *High rigor and high relevance*: studies with rigor scores from 2 or higher and relevance scores of 3 or 4;
- *High rigor and low relevance*: studies with rigor scores from 2 or higher and relevance scores from 0 to 2;
- *Low rigor and high relevance*: studies with rigor scores from 0 to 0.5 and relevance scores of 3 or 4;

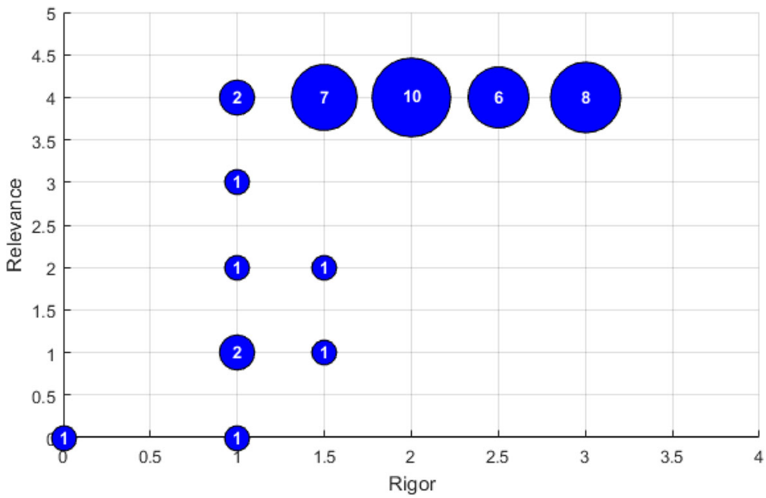

Fig. 5 Rigor-relevance assessment results

Table 12 Rigor-relevance study distribution

(Rigor; relevance) values	Studies	Number of studies
High rigor and high relevance		
(3;4)	S6, S9, S17, S18, S26, S30, S34, S38	8
(2,5;4)	S2, S11, S12, S23, S29, S41	6
(2;4)	S4, S5, S7, S10, S15, S19, S32, S33, S37, S40	10
High rigor and low relevance		
None		0
Low rigor and high relevance		
(1;3)	S3	1
(1;4)	S1, S8	2
(1,5;4)	S16, S20, S21, S28, S31, S35, S39	7
Low rigor and low relevance		
(0;0)	S36	1
(1;0)	S25	1
(1;1)	S13, S27	2
(1;2)	S24	1
(1.5;1)	S14	1
(1.5; 2)	S22	1

- *Low rigor and low relevance*: studies with rigor scores from 0 to 0.5 and relevance scores from 0 to 2.

Table 12 shows the study distribution according to the presented classification. As can be observed, the evidence points to works with high relevance. This occurs because the great majority of the studies were carried out in real environments and/or with industry professionals.

5 Discussions

This section discusses the results of the evidence found. For better reading, it is organized by classes according to SEI Taxonomy (Carr et al. 1993). Finally, a discussion about the quality of the studies is presented.

5.1 Product engineering class

In this class, the element "Requirement" is the one with the highest concentration of identified factors. There are 238 occurrences, distributed in 16 different risks factors. The factor "requirement ambiguity" has the highest number of occurrences and concerns misunderstanding, unclear, inaccurate, and vague requirements.

Still, in this class, we can highlight the high occurrence of factors related to (i) changes in requirements, (ii) technology that was not used before, (iii) technical complexity, and (iv) incompleteness of requirements. According to the evidence found, the importance of risk analysis associated with requirements in software development projects is very clear.

"Design" element, which deals with risks associated with software architecture issues, presents 85 occurrences distributed in 11 different risk factors. The most mentioned one is related to problems with external component and/or bad interfaces. Also, in the line of interfaces, the factor relating to bad interfaces with legacy systems were mentioned, since the evolution of software often makes difficult the integration between systems that simply must not stop.

Regarding the factors that belong to the element “Code and unit test,” we can highlight that the ones that presented greater relevance are (i) bad quality of code and (ii) poor adequacy between the hardware and the software. Regarding the first item, questions associated with inconsistency, poor modularization, lack of coding standards, and difficult to understand the code are mentioned. The second item calls attention to the selection of the target hardware so that the software is appropriate. In this element, we found 40 occurrences, distributed in 7 different risk factors.

The element “Integration and test” presents 54 occurrences, distributed in 8 different risk factors. Of these, 2 stand out. The first one is the bad internal integration of the system, which is related to difficulties in closing the initial release version due to difficulties in integrating system components. The second one is the lack of proper tests, which reinforces the importance of software testing at a similar level of importance in relation to software development processes.

Finally, the element “Engineering specialties” brought 66 occurrences, distributed in 9 different risk factors. It is clear that the most mentioned factor is related to the low quality of documentation and specifications of the development software artifacts. Next, factors related to 2 non-functional requirements (security and usability) are considered important for the good progress of software development projects.

Therefore, the class “Product engineering” presents a good variability of factors that permeate all stages of software development lifecycle. On the other hand, it highlights the importance of software requirements as the most valuable source of information on risk factors. Furthermore, the more correct and objective the requirements are, together with good development practices and testing, using consolidated technologies and with good specification, the lower the chance of software development projects to fail.

5.2 Development environment class

The element “Development process” presents 91 occurrences, distributed in 9 different risk factors. The most mentioned ones are related to lack of development process, low quality, and inappropriate and bad familiarity. Issues related to lack of project and risk management methodologies are also mentioned. Additionally, there are issues of process control, version control, and requirements traceability process. Thus, the importance of minimally specified process is evident to minimize the chances of project failure.

The element “Development system” has a total of 47 occurrences, distributed in 5 different risk factors. The presented factors concern the importance of appropriate development environments to software projects, including software and hardware. Another important factor considering the found evidence is associated to support to team, users, and vendors. The relative low quantity of factors and occurrences in this element shows that these factors are less important in relation to others. It occurs because there nowadays is a maturity regarding the availability of hardware and development environments for most software applications.

The most recurring risk factor in this class belongs to the element “Management process”: no planning or inadequate planning. The findings related to this factor bring information such as unclear milestones, optimistic or pessimistic planning, bad scope, and unclear or bad identification of the most critical activities. The other factors have strong relation to a bad planning because they involve scope changes and unclear project objectives. We also found factors related to low experience in project management, risk management, and leadership. The element “Management process” presents 138 occurrences, distributed in 12 risk factors, and it is the most cited element in the "Development environment" class. In the perspective of risk management, this result makes sense because a well-planned project, with well-trained managers and leaders and with roles clearly defined, is capable to minimize the chances of project failure.

In the element "Management methods" stands out the risk factor related to low control and monitoring of projects. In this sense, the use of metrics and indicators is cited as an important support-tool, because it allows a less subjective assessment of project progress. It is also worth mentioning the existence of permanent education program so that the team is always prepared for the designated roles. The element "Management methods" brought 55 occurrences, distributed in 7 different risk factors.

Finally, the element "Work environment" presents 85 occurrences, distributed in 8 different risk factors. Three factors stand out: (i) low commitment of staff, (ii) bad communication, and (iii) low morale and conflicts between project staff. In addition, we found factors related to cultural differences and language, resistance to changes, and ineffective meetings. It is worthy of note that the low commitment of the team is the second most cited risk factor of the class. Therefore, we can affirm that team motivation and communication are fundamental items for the good progress of software development projects.

5.3 Program constraints class

The element "Resources" plays an important role in the risk identification process. We found 260 occurrences, distributed in 24 different risk factors. The factor with most occurrences of all classes belongs to the following element: "Staff does not have required skills." It is a fact that software development projects still have a very high dependence on technical skills of the team. Added to this, we have other factors, such as low experience, turnover, team and project size relationship, dependence of few people, and low staff availability. Much of this is due to the fact that an organization usually deals with multiple projects simultaneously, and an effective management of people for projects optimizing is necessary. In addition, the issue of estimation quality of schedule and budget also has important role in software project risks. Finally, we found factors related to work environment, especially the issue of lack of disaster prevention. Hence, we can affirm that the element "Resources" requires tactical level of management to aid the mitigation of risks, once this level of management analyze the allocation of resources between multiple projects.

The risk factors related to element "Contract," in turn, have little apparent influence on software development project. Only 23 occurrences were found, distributed in 5 different risk factors. Much of the studies discuss or present few risk factors of this element, possibly because software development environments seek to focus more on technical aspects of management and software development.

Finally, "Program interfaces" element presents 232 occurrences, distributed in 27 different risk factors. It is a very expressive number, which demonstrates its importance and relevance in software development projects. An interesting result points to the importance of the client participation as a fundamental element for reducing project failure rates. Much of the studies treat the customer as the final user or their representatives. Other important factors, given the number of occurrences, are the influence of the organizational environment and policies on projects. Top management support also is placed as one of the most critical success factors, as well as organizational changes and micromanagement.

5.4 Quality of the studies

The results of rigor-relevance analysis demonstrate the good quality of the selected studies. Only five (5) of our review papers used only literature review as a method to raise and/or assess risk factors. Survey with professionals of the industry is the most used research method, being used by 25 studies. Other mentioned methods are Delphi, case studies, interviews, focus groups, and database of past projects.

In this way, we can state that there is evidence that the found risk factors, if combined, are relevant for the industry and practitioners. The high number of occurrence of factors also can indicate saturation of data. On the other hand, within the evidence, there are no related systematic literature reviews or mapping studies. The studies tend to present a list of factors from ad hoc literature review, followed by a specific validation study, making it difficult to generalize.

6 Conclusions

Aiming to map the main risk factors in software development projects, this work performed a systematic literature review. As a result, 41 studies were selected, among which 1414 occurrences of risk factors were identified. After removal of duplication, 148 different risk factors were identified. Next, these factors were categorized according to the classification proposed by SEI Taxonomy (Carr et al., 1993), and finally, as a way of evaluating the quality of the selected studies, we performed a rigor-relevance analysis.

A high number of risk factors were found in the literature. It shows a certain maturity from theoretical viewpoint regarding the main risk factors in software projects. The strong relationship between the studies also points to a saturation of factors.

Analyzing the 11 most mentioned factors, it is clear that requirement risks are the most relevant because they are present on 7 occasions. Ambiguity, changes, new technologies, complexity, and incompleteness are pointed out as the risk factors that deserve more attention in software project management. In addition, the most mentioned risk factor relates to team skills to develop their activities. From these results, it can be inferred that aspects directly related to the product are the major obstacles in guaranteeing a successful software development project.

On the other hand, the most related management factors were bad commitment of the user/customer, unstable organizational environment, and no planning or inadequate planning. The first two are external factors to a project that directly impact it, and the last one has strong relation to the quality of project planning, especially regarding scope.

6.1 Threats to validity

Despite the correct application of research methods to perform the systematic literature review, it is important to highlight the limitations of this research process so that we can identify points of improvements for future work in this area. In general, we can list the following threats to validity:

- Only one author performed the selection of works and rigor-relevance analysis, under the supervision of the others. It is not ideal but follows the recommendations of Kitchenham and Charters guideline (2007), which accepts that PhD works (the case of this study) can be performed by one single researcher since the supervisor checks the consistency of the findings and evaluates the research protocol. The researchers have advanced experience and knowledge about project and risk management, which helps minimize flaws;

- Given the high occurrence of risk factors and the lack of a tool that automates data extraction process, there may have been an influence of the researchers in the identification of risk factors. The use of SEI Risk Taxonomy (Carr et al. 1993) as reference model was very important for reducing errors, once it presents a detailed description of classes, elements, and attributes.
- Bias in the quality of the found evidence is also a limitation. To minimize this, we used recommended guidelines for evidence-based software engineering (Kitchenham and Charters, 2007; Biolchini et al. 2005);
- We did not perform any snowballing search, and it may have biased the work. The authors chose to not perform due to the high amount of information found only with the selection of papers, which indicated data saturation. For this work specifically, the point of saturation is achieved due to a high repetition of information and few novel risk factors in most recent studies.

6.2 Future work

One of the roles of systematic literature reviews is to provide new perspectives and to identify gaps. Therefore, we can enumerate the following research opportunities as future work that would interest software quality and software project management communities:

- Assessment of these factors among researchers and practitioners, seeking to reinforce the indicated results, as well as to identify strengths and weaknesses of these results;
- Perform case studies and quasi-experiments using these factors as a source of information regarding risks. In this sense, the authors are carrying out a case study in two organizations that have software development projects, and one of the objectives of this study consists in evaluating how the mapped factors can be useful in the process of risk identification;
- Another relevant point not addressed in this work is the relationship between risk factors. It is important to mention that a hazard may be caused by one or more risk factors;
- Analyze the influence of software development environment on risk factors;
- In-depth investigation of risk factors related to software requirements;
- Analyzing risk factors as useful information for analysis of projects, an interesting work would be to propose indicators for risk measurement of software development projects based on information of risk factors. For example, requirements-related risks could have higher weight regarding design-related risks. This approach would allow the comparison of risk levels of different projects within the same environment, considering that each project has a set of risk factors and some of them may be present in more than one project at the same time;
- Conduct studies focused on Global Software Development (GDS) and Distributed Software Development (DSD), mapping specific risk factors for this context.
- Finally, we highlight that the most mentioned 11 risk factors correspond to approximately 22% of the total of occurrence of the risk factors found. Analyzing Pareto principle (Ren 2016), a study of analysis of a risk-driven management of software development projects could be an interesting alternative in order to verify if only the analysis of these 11 factors would be enough to minimize the chances of software development project failure in general.

Acknowledgements The authors would like to thank the Brazilian Ministry of Health for the support given to this work.

Appendix

Table 13 Selected studies

S1	Boehm, B. W. (1991). Software risk management: principles and practices. IEEE Software, 8(1), 32–41. https://doi.org/10.1109/52.62930
S2	Barki, H., Rivard, S., & Talbot, J. (1993). Toward an assessment of software development risk. Journal of Management Information Systems, 10(2), 203–225.
S3	Sulgrove, R. N. (1996). Scoping software projects. AT&T Technical Journal, 75(1), 35–45. https://doi.org/10.15325/ATTTJ.1996.6772890
S4	Heemstra, F. J.., & Kusters, R. J.. (1996). Dealing with risk: A practical approach. Journal of Information Technology, 11(4), 333–346. https://doi.org/10.1057/jit.1996.7
S5	Keil, M., Cule, P. E., Lyytinen, K., & Schmidt, R. C. (1998). A Framework for Identifying Software Project Risks. Commun. ACM, 41(11), 76–83. https://doi.org/10.1145/287831.287843
S6	Ropponen, J., & Lyytinen, K. (2000). Components of software development risk: how to address them? A project manager survey. IEEE Transactions on Software Engineering, 26(2), 98–112. https://doi.org/10.1109/32.841112
S7	Mizuno, O., Kikuno, T., Takagi, Y., & Sakamoto, K. (2000). Characterization of risky projects based on project managers evaluation. In Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium (pp. 387–395). https://doi.org/10.1145/337180.337226
S8	Sumner, M. (2000). Risk Factors in Enterprise Wide Information Management Systems Projects. In Proceedings of the 2000 ACM SIGCPR Conference on Computer Personnel Research (pp. 180–187). New York, NY, USA: ACM. https://doi.org/10.1145/333334.333392
S9	Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. Journal of Management Information Systems, 17(4), 5–36.
S10	Procaccino, J. D., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: factors for early prediction of software development success. Information and software technology, 44(1), 53–62.
S11	Wallace, L., Keil, M., & Rai, A. (2004). How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. Decision Sciences, 35(2), 289–321. https://doi.org/10.1111/j.00117315.2004.02059.x
S12	Wallace, L.., Keil, M.., & Rai, A.. (2004). Understanding software project risk: A cluster analysis. Information and Management, 42(1), 115–125. https://doi.org/10.1016/j.im.2003.12.007
S13	Takagi, Y., Mizuno, O. & Kikuno, T.(2005). An empirical approach to characterizing risky software projects based on logistic regression analysis. In Empirical Software Engineering (Vol. 10, pp. 495–515). https://doi.org/10.1007/s10664-005-3864-z
S14	Tiwana, A., & Keil, M. (2006). Functionality Risk in Information Systems Development: An Empirical Investigation. IEEE Transactions on Engineering Management, 53(3), 412–425. https://doi.org/10.1109/TEM.2006.878099
S15	Kim, E. H.., & Park, Y.. (2007). Prediction of IS project escalation based on software development risk management. Journal of Information and Knowledge Management, 6(2), 153–163. https://doi.org/10.1142/S0219649207001688
S16	Bhuta, J., Mallick, S., & Subrahmanya, S. V.. (2007). A Survey of Enterprise Software Development Risks in a Flat World. In First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007) (pp. 476–478). IEEE. https://doi.org/10.1109/ESEM.2007.74
S17	Gemino, A.., Reich, B. H.., & Sauer, C.. (2007). A temporal model of information technology project performance. Journal of Management Information Systems, 24(3), 9–44. https://doi.org/10.2753/MIS0742-1222240301
S18	Costa, H. R.. b, Barros, M. d. O.. c, & Travassos, G. H.. (2007). Evaluating software project portfolio risks. Journal of Systems and Software, 80(1), 16–31. https://doi.org/10.1016/j.jss.2006.03.038
S19	Cerpa, N., & Verner, J. M. (2009). Why did your project fail? Communications of the ACM, 52(12), 130. https://doi.org/10.1145/1610252.1610286
S20	Warkentin, M., Moore, R. S., Bekkering, E., & Johnston, A. C. (2009). Analysis of Systems Development Project Risks: An Integrative Framework. SIGMIS Database, 40(2), 8–27. https://doi.org/10.1145/1531817.1531821
S21	Reed, A. H.., & Knight, L. V.. (2009). Differing impact levels from risk factors on virtual and co-located software development projects. In 15th Americas Conference on Information Systems 2009, AMCIS 2009 (Vol. 2, pp. 817–824).

Table 13 (continued)

S22	Dasgupta, J.., & Mohanty, R. P.. (2010). Estimating operational risk indices for software services outsourcing industry: A case. International Journal of Services and Standards, 6(1), 43–61. https://doi.org/10.1504/IJSS.2010.034459
S23	Reed, A. H.., & Knight, L. V. (2010). Project risk differences between virtual and co-located teams. Journal of Computer Information Systems, 51(1), 19–30. https://doi.org/10.1080/08874417.2010.11645446
S24	Koopman, P. (2010). Risk Areas in Embedded Software Industry Projects. In Proceedings of the 2010 Workshop on Embedded Systems Education (p. 5:1–5:8). New York, NY, USA: ACM. https://doi.org/10.1145/1930277.1930282
S25	Shahzad, B.., Al-Ohali, Y.., & Abdullah, A.. (2011). Trivial model for mitigation of risks in software development life cycle. International Journal of Physical Sciences, 6(8), 2072–2082.
S26	Sharma, A., Sengupta, S., & Gupta, A. (2011). Exploring risk dimensions in the Indian software industry. Project Management Journal, 42(5), 78–91. https://doi.org/10.1002/pmj.20258
S27	López, C., & Salmeron, J. L. (2012). Risks Response Strategies for Supporting Practitioners Decision-Making in Software Projects. Procedia Technology, 5, 437–444. https://doi.org/10.1016/j.protcy.2012.09.048
S28	Zhang, T., & Zhang, Y. (2012). Research on Project Development Key Risk Factors of Small and Medium-Sized Software Enterprises. Communications in Computer and Information Science, 315, 139–146. https://doi.org/10.1007/978-3-642-34240-0_19
S29	Hoermann, S., Aust, M., Schermann, M., & Krcmar, H. (2012). Comparing Risks in Individual Software Development and Standard Software Implementation Projects: A Delphi Study. In 2012 45th Hawaii International Conference on System Sciences (pp. 4884–4893). IEEE. https://doi.org/10.1109/HICSS.2012.168
S30	Verner, J. M.., & Abdullah, L. M.. (2012). Exploratory case study research: Outsourced project failure. Information and Software Technology, 54(8), 866–886. https://doi.org/10.1016/j.infsof.2011.11.001
S31	Tsunoda, M.., Monden, A.., Matsumoto, K.., Hatano, R.., Nakano, T.., & Fukuchi, Y.. (2013). Analyzing risk factors affecting project cost overrun. Studies in Computational Intelligence, 443, 171–184. https://doi.org/10.1007/978-3-642-32172-6-14
S32	Elzamly, A.. b, & Hussin, B.. c. (2013). Managing software project risks (Design Phase) with proposed fuzzy regression analysis techniques with fuzzy concepts. International Review on Computers and Software, 8(11), 2601–2613.
S33	Elzamly, A.. b, & Hussin, B.. (2014). Managing software project risks (analysis phase) with proposed fuzzy regression analysis modeling techniques with fuzzy concepts. Journal of Computing and Information Technology, 22(2), 131–144. https://doi.org/10.2498/cit.1002324
S34	Sambantha Moorthy, J., Bin Ibrahim, S., & Mahrin, M. N. (2014). Identifying usability risk: A survey study. In 2014 8th Malaysian Software Engineering Conference, MySEC 2014 (pp. 148–153). https://doi.org/10.1109/MySec.2014.6986005
S35	Sharif, A. M., & Basri, S. (2014). Risk assessment factors for SME software development companies in Malaysia. In 2014 International Conference on Computer and Information Sciences (ICCOINS) (pp. 1–5). https://doi.org/10.1109/ICCOINS.2014.6868827
S36	Hijazi, H.., Alqrainy, S.., Muaidi, H.., & Khdour, T.. (2014). Identifying causality relation between software projects risk factors. International Journal of Software Engineering and Its Applications, 8(2), 51–58. https://doi.org/10.14257/ijseia.2014.8.2.06
S37	Elzamly, A.. b, & Hussin, B.. (2015). Modeling and evaluating software project risks with quantitative analysis techniques in planning software development. Journal of Computing and Information Technology, 23(2), 123–139. https://doi.org/10.2498/cit.1002457
S38	Shrivastava, S. V, & Rathod, U. (2015). Categorization of risk factors for distributed agile projects. Information and Software Technology, 58, 373–387. https://doi.org/10.1016/j.infsof.2014.07.007
S39	Samantra, C.., Datta, S.., Mahapatra, S. S.., & Debata, B. R.. (2016). Interpretive structural modeling of critical risk factors in software engineering project. Benchmarking, 23(1), 2–24. https://doi.org/10.1108/BIJ-07-2013-0071
S40	Li, J.. b c, Li, M.. d, Wu, D.., Dai, Q.. e, & Song, H.. (2016). A Bayesian Networks-Based Risk Identification Approach for Software Process Risk: The Context of Chinese Trustworthy Software. International Journal of Information Technology and Decision Making, 15(6), 1391–1412. https://doi.org/10.1142/S0219622016500401
S41	Elzamly, A., Hussin, B., & Salleh, N. M. (2016). Top fifty software risk factors and the best thirty risk management techniques in software development lifecycle for successful software projects. International Journal of Hybrid Information Technology, 9(6), 11–32. Retrieved from https://doi.org/10.14257/ijhit.2016.9.6.02

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Alam, A. U., Khan, S. U., & Ali, I. (2012). Knowledge sharing management risks in outsourcing from various continents perspective: a systematic literature review. *International Journal of Digital Content Technology and its Applications, 6*(21), 27–33.

Bannerman, P. L. (2015). A reassessment of risk management in software projects. In: *Handbook on Project Management and scheduling*, vol. 2 (pp. 1119–1134). Springer International Publishing.

Biolchini, J., Mian, P. G., Natali, A. C. C., & Travassos, G. H. (2005). Systematic review in software engineering. In: *System engineering and computer science department COPPE/UFRJ, Technical Report ES*, vol 679(05) (p. 45).

Boehm, B. W. (1989). *Software risk management*. Piscataway: Software risk management.

Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE Software, 8*(1), 32–41. https://doi.org/10.1109/52.62930.

Brasiliano, A. (2009). *Método Brasiliano avançado – Gestão e análise de risco corporativo*. Sicurezza.

Carr, M. J., Konda, S. L., Monarch, I., Ulrich, F. C., & Walker, C. F. (1993). *Taxonomy-based risk identification (No. CMU/SEI-93-TR-06)*. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

Charette, R. N. (1989). *Software engineering risk analysis and management*. New York: Intertext Publications.

Charette, R. N. (2005). Why software fails. *IEEE Spectrum, 42*(9), 42–49.

De Bakker, K., Boonstra, A., & Wortmann, H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management, 28*(5), 493–503.

De Marco, T. (1997). *The deadline: a novel about project management*. Dorset House.

DoD, U. S. (2006). *Risk management guide for DoD acquisition*. USA: Department of Defense.

Dorofee, A. J., Walker, J. A., Alberts, C. J., Higuera, R. P., & Murphy, R. L. (1996). *Continuous risk management guidebook*. Carnegie-Mellon Univ, Pittsburgh.

Fairley, R. (1994). Risk management for software projects. *IEEE Software, 11*(3), 57–67.

Fan, C. F., & Yu, Y. C. (2004). BBN-based software project risk management. *Journal of Systems and Software, 73*(2), 193–203.

Fu, Y., Li, M., & Chen, F. (2012). Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. *International Journal of Project Management, 30*(3), 363–373.

Gerrard, P., & Thompson, N. (2002). *Risk-based E-business testing*. Artech House.

Goguen, A., Stoneburner, G., & Feringa, A. (2002). *Risk management guide for information technology systems and underlying technical models for information technology security.*

Google Scholar citations. (2017). https://scholar.google.com/intl/en/scholar/citations.html. Accessed May 2017.

Hall, E. M. (1998). *Managing risk: methods for software systems development*. Pearson Education.

Han, W. M., & Huang, S. J. (2007). An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software, 80*(1), 42–50.

Heldman, K. (2010). *Project manager's spotlight on risk management*. John Wiley & Sons.

Higgins, J. P., & Green, S. (Eds.). (2011). *Cochrane handbook for systematic reviews of interventions*. http://handbook.cochrane.org/chapter_6/6_4_4_sensitivity_versus_precision.htm. Accessed May 2017.

Hillson, D. (2002). The Risk Breakdown Structure (RBS) as an aid to effective risk management. In: *5th European Project Management conference*. Cannes, France (pp. 1–11).

Ivarsson, M., & Gorschek, T. (2011). A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering, 16*(3), 365–395.

Jiang, J., & Klein, G. (2000). Software development risks to project effectiveness. *The Journal of Systems and Software, 52*(1), 3–10.

Jiang, J., Klein, G., & Discenza, R. (2001). Information systems success as impacted by risks and development strategies. *IEEE Transactions on Engineering Management, 48*(1), 46–55.

Jorgensen, M. (1999). Software quality measurement. *Advances in Engineering Software, 30*(12), 907–912.

Kerzner, H. (2017). *Project management: a systems approach to planning, scheduling, and controlling*. Hoboken: John Wiley & Sons.

Khan, A. A., Basri, S., & Dominic, P. D. D. (2014). Communication risks in GSD during RCM: results from SLR. In: *Computer and Information Sciences (ICCOINS), 2014 International Conference on* (pp. 1–6). IEEE.

Kitchenham, B & Charters, S., 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical report. EBSE.

Kontio, J. (2001). *Software engineering risk management: a method, improvement framework, and empirical evaluation*. Helsinki University of Technology.
López, C., & Salmeron, J. L. (2012). Risks response strategies for supporting practitioners decision-making in software projects. *Procedia Technology, 5*, 437–444.
March, J. G., & Shapira, Z. (1987). Managerial perspectives on risk and risk taking. *Management Science, 33*(11), 1404–1418.
Munir, H., Wnuk, K., & Runeson, P. (2016). Open innovation in software engineering: a systematic mapping study. *Empirical Software Engineering, 21*(2), 684–723.
Neves, S. M., da Silva, C. E. S., Salomon, V. A. P., da Silva, A. F., & Sotomonte, B. E. P. (2014). Risk management in software projects through knowledge management techniques: cases in Brazilian incubated technology-based firms. *International Journal of Project Management, 32*(1), 125–138.
Nurdiani, I., Jabangwe, R., Šmite, D., & Damian, D. (2011). Risk identification and risk mitigation instruments for global software development: systematic review and survey results. In: *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on* (pp. 36–41). IEEE.
Oliveira, K. A., Gusmão, C. M., & de Barros Carvalho Filho, E. C. (2012). Mapeamento de Riscos em Projetos de Desenvolvimento Distribuído de Software. In: *CONTECSI-international conference on information systems and technology management* (vol. 9, no. 1, pp. 3837–3866).
Pa, N. C., & Jnr, B. A. (2015). A review on decision making of risk mitigation for software management. *Journal of Theoretical & Applied Information Technology, 76*(3).
Pfleeger, S. L., Hatton, L., & Howell, C. C. (2001). *Solid software*. Prentice Hall PTR.
Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
Qinghua, P. (2009). A model of risk assessment of software project based on grey theory. In: *Computer Science & Education, 2009. ICCSE'09. 4th International Conference on* (pp. 538–541). IEEE.
Raz, T., Shenhar, A. J., & Dvir, D. (2002). Risk management, project success, and technological uncertainty. *R&D Management, 32*(2), 101–109.
Reeves, J. D., Eveleigh, T., Holzer, T. H., & Sarkani, S. (2013). Identification biases and their impact to space system development project performance. *Engineering Management Journal, 25*(2), 3–12.
Ren, F. (2016) *Understanding Pareto's principle - the 80-20 rule*. https://www.thebalance.com/pareto-s-principle-the-80-20-rule-2275148. Accessed May 2017.
Salmeron, J. L., & Lopez, C. (2012). Forecasting risk impact on ERP maintenance with augmented fuzzy cognitive maps. *IEEE Transactions on Software Engineering, 38*(2), 439–452.
Sarigiannidis, L., & Chatzoglou, P. D. (2014). Quality vs risk: an investigation of their relationship in software development projects. *International Journal of Project Management, 32*(6), 1073–1082.
Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: a systematic literature review. *International Journal of Project Management, 30*(4), 458–469.
Silva, S. (2011). *Proposta de tratamento de fatores de riscos em desenvolvimento de software para uma organização no setor público*. Federal University of Permambuco.
SJR. (2017). *Scimago Journal & Country Rank (SJR)*. http://www.scimagojr.com/aboutus.php. Accessed May 2017.
Subramanian, G. H., Jiang, J. J., & Klein, G. (2007). Software quality and IS project performance improvements from software development process maturity and IS implementation strategies. *Journal of Systems and Software, 80*(4), 616–627.
Tang, A. G., & Wang, R. L. (2010, June). Software project risk assessment model based on fuzzy theory. In: *Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On* (vol. 2, pp. 328–330). IEEE.
Trigo, T. R., Gusmão, C., & Lins, A. (2008). CBR risk – risk identification method using case based reasoning. In: *International Conference on Information Systems and Technology Management* (vol. 5, No. 2008).
Van Loon, H. (2007). A management methodology to reduce risk and improve quality. *IT Professional, 9*(6), 30–35.
Vasconcellos, F. J., Landre, G. B., Cunha, J. A. O., Oliveira, J. L., Ferreira, R. A., & Vincenzi, A. M. (2017). Approaches to strategic alignment of software process improvement: a systematic literature review. *Journal of Systems and Software, 123*, 45–63.
Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM, 47*(4), 68–73.
Wallace, L., Keil, M., & Rai, A. (2004a). Understanding software project risk: a cluster analysis. *Information Management, 42*(1), 115–125.
Wallace, L., Keil, M., & Rai, A. (2004b). How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory model. *Decision Sciences, 35*(2), 289–321.
Wysocki, R. K. (2011). *Effective project management: traditional, agile, extreme*. John Wiley & Sons.
Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology, 53*(6), 625–637.

Júlio Menezes Jr Master in Computer Science by Center of Informatics (CIn), Federal University of Pernambuco (UFPE) (2012), and computer engineer by University of Pernambuco (UPE) (2010). Currently he is PhD student of the post-graduation program in Computer Science (CIn/UFPE) and collaborator of the research group GP2 (CIn/UFPE). He has experience on computer science, with emphasis on software engineering, acting mainly in the following themes: project management, risk management, mobile health, software engineering, and educational technologies.

Cristine Gusmão DSc. Doctor in Computer Science by Center of Informatics (CIn), Federal University of Pernambuco (UFPE) (2007), Master in Computer Science (CIn/UFPE) (2001), and electrical engineer (UFPE) (1993). Currently she is (i) Professor of Federal University of Pernambuco, Department of Biomedical Engineering, with activities in under graduation and post-graduation; (ii) general coordinator of UNA-SUS UFPE; (iii) researcher of *SABER Tecnologias Educacionais e Sociais* (UFPE), GP2 (CIn/UFPE), and *Tecnologia e Inovação em Educação em Saúde* (Federal University of Maranhão – UFMA), and (iv) Permanent member of masters in Biomedical Engineering (UFPE) and collaborator of the Post-graduation program in Computer Science (CIn/UFPE). She has experience on Computer Science, with emphasis on Software Engineering, acting mainly on the following themes: educational and social technologies, agile methods, risk factors in projects, software quality, project management, management of health systems, mobile health, education in health, and business process management.

Hermano Moura PhD Currently, he is a professor of Center of Informatics (CIn), from Federal University of Pernambuco (UFPE). He is electronic engineer (UFPE) (1982), Master in informatics (UFPE) (1989) and PhD in Computer Science by University of Glasgow, Scotland (1993). He has a post-doctoral degree by University of Maryland in College Park, USA (2011). He is PMP certified (2003) by Project Management Institute. He acts in computing, with emphasis on Software Engineering and Information Systems, developing researches and projects, mainly on the following lines: project management, software project management, software development process, IT management, and entrepreneurship in informatics. He has started his activities in informatics in 1979, and currently develops activities of teaching and research, and cooperation projects with industry and university management.