Computer Standards & Interfaces 71 (2020) 103431

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Risk management in the software life cycle: A systematic literature review

Jhon Masso[a,b,*], Francisco J. Pino[c], César Pardo[b], Félix García[a], Mario Piattini[a]

[a] Alarcos Research Group, Institute of Technologies and Information Systems, University of Castilla-La Mancha, Paseo de la Universidad, 4, Ciudad Real 13071, Spain
[b] GTI Research Group, Electronic and Telecommunications Engineering Faculty, University of Cauca, Calle 5 # 4 – 70, Popayán 190002, Colombia
[c] IDIS Research Group, Electronic and Telecommunications Engineering Faculty, University of Cauca, Calle 5 # 4 – 70, Popayán, 190002, Colombia

ARTICLE INFO

Keywords:
Software risk
Risk management activities
ISO 31000
Software life cycle processes
ISO 12207
Systematic literature review

ABSTRACT

Risk management (RM) plays a key role in project management, as it allows identification and prompt management of threats that may arise during project execution. Furthermore, project management within the software industry is evolving rapidly nowadays, a fact that implies new challenges, because the emergence and use of fresh approaches has brought a greater degree of complexity to the RM process. The objective of this paper is to carry out a systematic literature review (SLR) in the field of software risk, in an attempt to characterize and present the state of the art of this field, identifying gaps and opportunities for further research. From the analysis of the results of this SLR it could be observed that interest on the part of the scientific community has turned away from the definition of research work that addressed an integrated risk management process, to pay attention to work that concentrates on specific activities of this process. It was also possible to see that there is a clear lack of scientific rigour as regards the process of validation in the different studies, and a deficiency in the use of standards or of *de facto* models to define these.

1. Introduction

Software is the result of a process that depends on good management in each one of its activities. In this sense, software project risk management is a key element for that management, which is made up of processes, methodologies and tools that are frequently used to address risk in the different phases of the software development life cycle (SDLC) [1]. Risks can lead to organizations' making losses which may have to do with product quality, increased production costs, length of time taken to complete a project, meeting deadlines for the project, and so on [2]. Correct identification and tracking of the different risk factors will therefore help to enhance project success rate and achieve quality software [3]. If the former activities are not carried out, that may in the long term lead to software development firms having a smaller share of the market [2]. Risk in the PMBOK [4] and PRINCE2 [5] is defined as an uncertain event or set of uncertain events which, if they occur, will have a negative or positive effect on one or more of the project objectives. For its part, the ISO 31,000 (*standard that provides principles, framework, and process for risk management, which can be applied to any type of organization*), defines risk as the effect of uncertainty on the achievement of objectives [6]. Risk may sometimes be expressed in terms of the combination of the consequences of an event and the probabilities of its occurring [5,6]. At the same time, it will also depend on how threats and opportunities are perceived, as well as on how great their impact on the objectives is [5]. According to ISO 12,207 (*the standard that defines the software life cycle processes, and which can be adapted by any type of organization that is involved in the acquisition or development of software products and services),* an objective may be associated with different aspects, amongst which the following could be highlighted: financial, health-related, security-related, and environmental [7]. Risk may thus occur at different levels of the organization, and come about as the result of a range of internal or external factors; these factors will influence the probability of risk and its impact on business objectives [8]. In the context of the software life cycle, risk may appear at project level, product level, and process level [7,9]. This is where risk management plays an important role in identifying risk, so that strategies to mitigate it at the corresponding level may be taken, thus reducing possible losses [10].

Risk management is defined as a set of coordinated activities which allow the organization to be directed and controlled as regards risk [6]. This requires the systematic application of principles, approaches and processes to the tasks of identification, risk assessment, planning and implementation of response to risk, as well as to the communication regarding the activities carried out with each one of the stakeholders [5]; the objectives are to identify, direct and eliminate the elements of software risk before they occur and become potential threats to the

* Corresponding author.
E-mail addresses: masso@unicauca.edu.co (J. Masso), fjpino@unicauca.edu.co (F.J. Pino), cpardo@unicauca.edu.co (C. Pardo), felix.garcia@uclm.es (F. García), mario.piattini@uclm.es (M. Piattini).

https://doi.org/10.1016/j.csi.2020.103431
Received 15 March 2019; Received in revised form 3 March 2020; Accepted 3 March 2020
Available online 05 March 2020

success of the operation, or turn out to be sources of delay in development [11].

In a number of studies this management is seen as being amongst the main factors that influence the success of projects [12–15]. That success will depend in large measure on the experience and abilities of the risk managers and the project managers, [13] as well as on factors related to perceptions and expectations of the different interested parties and on the synchronisation of the actions designed to tackle risk [16]. At the same time, poor management or complete lack of management are amongst the main reasons for the failure of projects [15,17,18]. In a recent study carried out by the PMI [19] on professionals involved in project management, it is demonstrated that bad management of risks and opportunities is still one of the main causes for the failure of projects; apart from these reasons, there is the lack of ability on the part of the project managers and the organization to address the challenges that arise. In this report, the PMI also underlines the point that the use of standardized project management practices enables risk to be reduced and better results to be obtained. That is the case as long as these practices are rolled out in the whole organization; this generates the capacity to minimize risks and control the costs, also making it possible to adapt to the changing conditions of the markets and improve value delivery in each of the projects.

Risk management is a complex process which requires skills and experience to carry out decision-making, as well as to interpret information from the projects that will be used to predict future events and their effects on the results of the project [20]. On the other hand, ignoring it may very well produce new risks and additional costs for the project sponsors, which in turn can work against there being a good relationship between the organization and the client/consumer [21]. Risk management therefore requires a comprehensive understanding both of the issues in this field and the tools which assist in the process of managing risk. The standardised practices that allow losses to be minimised and project success to be maximised must also be well understood; these practices should be taken into account right from the very first stages of the project, and kept in steady operation throughout the whole life cycle [22]. In other words, the goal is the identification of the potential threats before they arise; the management activities can therefore be planned and invoked whenever necessary, so that any negative impact on the achievement of the project's objectives can be lessened [23]. It should also be underlined, however, that one of the main issues that arises in this management derives from the subjectivity displayed by each one of the parties interested in the activities of the process. That leads to poor decision-making, which has its consequent effect on whether the project is successful or not [24].

In some cases, the project managers justify not carrying out risk management due to limitations on time or to restrictions on costs; they may not perceive how applying this management might increase the chance of success for the projects [25,26]; it may simply be that the organizations do not have any organizational culture, with highly-qualified and experienced staff, available to them [21]. This means that, as Kutsch et al. [27] affirm, there is a need to carry out studies which would enable researchers to find out how risk management is really perceived by project managers. The aim would be to show whether these implemented practices, while obviously correct, really do help to carry out the risk management activities properly, bearing in mind that this management continues to be affected by the perceptions of project managers, as well as by their prejudices about the issue. These aspects probably jeopardise the effectiveness of risk management practices and influence their results.

The success of projects may be affected by many factors [28,29], amongst which we might find technical factors, factors to do with project management, a lack of support from top management, and a lack of participation by the users. All of these aspects undoubtedly mean that it may be harder to meet deadlines and achieve budget targets; they can also negatively affect the quality of the functional deliveries to final users [28]. That is why success in projects does not depend only on a systematic and rigorous application of risk management practices. There also needs to be active participation on the part of all stakeholders; this will help to improve communication and perception of risk, make risk management more acceptable, and increase the effectiveness of corrective actions [16]. However, despite the contributions made by researchers and professionals through the definition of strategies and recommendations (e.g. models, methods, standards and so on), based on theoretical knowledge and practical experience, the rate of success in the development of software projects is still low [30].

It should also be said that the software industry is having to deal with a globalised market at the present time. This brings with it new opportunities, as well as demands that need to be met when taking each project right through to completion. In the quest to provide support for this trend, Global Software Development (GSD) has become established as one of the paradigms that is receiving the greatest amount of interest on the part of companies from all over the world [31]. This is in spite of the host of challenges presented (especially those to do with temporal distance across multiple time zones, as well as geographical and sociocultural distance and difference; these create problems that are not found in traditional software development projects [32]). GSD is therefore a complex task from the technological and organisational points of view, and there is a whole range of new challenges for project managers when attempting to guarantee the success of their projects [33]. In addition to this, the industry has shown strong and clear interest in the use of agile strategies, which enable the project and the software process to be well-directed, while also allowing there to be better management of the teams involved, and a more appropriate prioritisation of the ever-changing requirements to be carried out. The basic purpose in the use of these strategies is always to reduce risk and achieve success in software projects. Currently, agile approaches have become more popular as companies have sought to achieve greater speed in their software development and in their ability to take on the challenge of change that firms face in their present-day dynamic environments [34].

All that has been said above shows that the software industry is extremely keen to address risks and opportunities; it is not overly concerned about what particular management styles and development paradigms are being used to carry out software projects. The aim of this article, therefore, is to present the results of a study that identifies and analyses the state of the art of publications in the field of software risks. A systematic literature review (SLR) was used to carry out this study; its scope was the context of the software life cycle. All these components will allow existing gaps to be identified, and will make it possible to offer future research guidelines to researchers and software engineering professionals on issues related to risk management in software projects, processes and products. The rest of the article is set out as follows. In Section 2 the related work is presented. The methodology used in the research is explained in Section 3. The results of the SLR are shown and discussed in Section 4. Lastly, the conclusions drawn from the study are given in Section 5, along with a description of future work proposed.

2. Related work

Having carried out an analysis of the literature, it was possible to find different types of studies that tackled issues related to software risk in different fields within software engineering. Some of these studies are SLR, focusing on Global Software Development (GSD) or Distributed Software Development (DSD), and they have been carried out to: (i) Identify risk in different GSD contexts, along with the mitigation strategies involved; these latter are elements that enable professionals engaged in GSD to understand the different risks that hinder success of projects, and they serve as tools to test whether the approach can be used to identify and mitigate risk [35–37]. (ii) identify aspects or characteristics that have an impact on risk assessment and on management strategies in order to propose a decision support system (DSS)

to help professionals in their decision-making during risk management [38], and (iii) that establishes an integrated framework for risk assessment that would support organisations as they assess capabilities at the beginning of their efforts in GSD-related issues [31].

Liu et al. [12] published an SLR focusing on software process simulation modelling (SPSM) in risk management, whose final goal was to: (i) Identify in which risk management activities this topic has been employed most, (ii) see the types of risk where SPSM has been applied most widely, (iii) establish which SPSM paradigms are used most in risk management, (iv) identify which SPSM tools have been used to support risk management and (v) analyse the implementation in practice of SPSM approaches and models in risk management.

In addition, two studies were identified which present results of systematic mapping reviews (SMR) focusing on (1) determining the different risk factors that have an impact on the development of software product lines (SPL), as well as identifying gaps that exist in this knowledge area with respect to risk management practices and strategies [39], and (ii) the identification, classification and comparison of risks in software projects based on off-the-shelf (OTS) components, from the point of view of software development, and also from the perspective of acquisition. Supported by a survey carried out on developers belonging to firms in Indonesia, the study enabled the occurrence of risk to be monitored [40].

On the other hand, authors such as Teklemariam et al. [41], Arnuphaptrairong [42] and Kajko-Mattsson et al. [43] carried out empirical studies by means of surveys, focusing on the identification and analysis of risk management practices in software companies. These studies all come to the same conclusion, which is that there is limited competence and knowledge on the part of project managers when it comes to attempts to perform risk management properly. In addition to this, there is inconsistency and heterogeneity in the practices carried out in the industry regarding the definitions which some international standards and models put forward in addressing risk management.

Another relevant study is the one carried out by Roy et al. [10]; they focused on the review of different risk management models and practices, and their aim was to identify which stages in the software development life cycle (SDLC) these models and practices give support to. The authors also establish the main risk factors and types of technical risks that occur in the different stages of the SDLC (requirements, planning, design, building of the code/implementation, deployment and maintenance), identifying, moreover, what risks are associated with the scheduling or calendar, the costs, the quality and the business. For their part, Elzamly et al. [44] published a literature review whose purpose was to identify which quantitative and intelligent models are used for risk management, apart from the techniques associated with each one of these models for risk management in software development projects. The researchers conclude that in the future it would be possible to work to design hybrid models that combine both types of models, incorporating different artificial intelligence techniques to mitigate risk in cloud computing and in banking transactions.

In a nutshell, it is evident that the work mentioned above has focused mainly on (i) Risk identification [31,35–37], as well as on strategies for risk mitigation in GSD [31]; (ii) identification of the challenges that have an impact on risk assessment and the establishing of what particular risk management strategies are used in DSD in order to support decisions (DSS) [38]; (iii) identification and listing of the risks in software product lines (SPL), while also establishing what their practices and management strategies are [39]; (iv) identification and classification of risks in software projects based on components (Off-the-Shelf, OTS) [40]; (v) identification of the use of modelling for the simulation of software processes (SPSM) [12]; (vi) identification and analysis of risk management practices at the level of the literature [10,44], as well as at the software company level [41–43].

The present SLR is different from those already in existence, since it does not focus on identifying risk in specific areas of the software development, nor does it seek to identify the particular risk management practices used in specific areas of software engineering or in the software industry. This SLR in the risk management field in the context of the software life cycle aims to identify the state of the art and main features of publications as regards: i) The coverage of the different activities in the risk management process and the software life cycle processes, ii) the application domain of the different research studies iii) the use of other techniques, models and tools belonging to other areas of knowledge and which set out to help improve the activities in the software management process, or to make them more agile. Finally on this point, this SLR offers a much clearer vision of the field in question, so that new research can be undertaken to strengthen risk management practices in the software industry and make it possible to improve the success rate of the projects being carried out.

3. Research methodology

This study has followed the approach suggested by Kitchenham y Charters [45] in conducting the SLR; those authors propose a process that makes it possible to identify, analyse and interpret the relevant studies that are available in a particular knowledge area with respect to a research question. The systematic review process is described in three main phases:

- Phase 1: Planning the review, where the aim is to produce a review protocol that will allow the review to be guided.
- Phase 2: Conducting the review, the goal here is to identify and choose the relevant studies, to carry out the quality assessment of the studies found, extract and synthesise the information.
- Phase 3: Reporting the review, which has to do with creating a review document that gives an account of the results obtained [45–47].

3.1. Research questions

The main research question (RQ) that will guide this SLR is:

"*What is the state of the art of publications in the field of risk management in the software life cycle?*"

The objective of this SLR is to offer an answer to the main RQ, referring to the identification and characterisation of the state of the art of the relevant publications with respect to the different strategies (approaches, models, methodologies, frameworks, etc.) which address the risk management process in the context of the software life cycle. Given the complexity of this task, and in an effort to provide a more specific vision of the most relevant aspects of the primary studies, six research questions were designed, aiming to help enable that first RQ to be answered. Table 1 shows the research questions, along with what motivated each one.

3.2. Data sources and search strategy

In order to carry out an automated search of primary studies we selected the Scopus database, because this includes some of most important journals, conferences or workshops in different research fields. To this end, we designed a search string following the recommendations described by Brereton et al. [46] for incorporating logical connectors into the chain, in which it is suggested that the logical operator OR is used for alternative terms, and that the operator AND be employed to link the main terms. The search strategy is described in Table 2.

The search string defined using the terms in Table 2 is:

("Software Risk" AND ("Framework" OR "Model" OR "Methodology" OR "Standard" OR "Guide" OR "Guideline" OR "Technology" OR "Method" OR "Ontology"))

Lastly, the search string was adjusted to meet the specific requirements of the Scopus data base, to which we applied the following filters

Table 1
Research questions.

Research Questions	Motivation
RQ1.What strategies have been defined for risk management in the software life cycle?	Determine what types of strategies (approaches, models, methods, methodologies, frameworks, etc.) have been established to carry out risk management in the context of the software life cycle.
RQ2. What risk management activities have been covered by the research studies?	Identify what risk management process activities have been covered in the different primary studies as regards the activities established in the ISO 31,000 international standard for the risk management process.
RQ3. What software life cycle processes have been covered by the research studies?	Determine which software life cycle processes have been covered in the different primary studies, with reference to those set out in the ISO 12,207 international standard.
RQ4. What is the application domain of the research studies? What particular knowledge areas, techniques or models or tools do they involve?	Identify the domain or field of application (projects, software development, software life cycle, etc.) of the different primary studies. In addition, identify the different techniques, models, or tools from other areas of knowledge that they include for the purpose of helping to improve the risk management activities, or to make them more agile.
RQ5. How do the respective researchers frame their research studies?	Identify how the researchers have framed each of the pieces of research work (as approach, model, method, methodology, framework, etc.).
RQ6. What methods of scientific validation have been used in the research studies?	Establish the particular type of empirical method that has been employed to validate the different research studies.

Table 2
Search string terms.

Main terms	Alternative terms	
Software Risk	"Software Risk"	AND
Framework	("Framework" OR "Model" OR "Methodology" OR "Standard" OR "Guide" OR "Guideline" OR "Technology" OR "Method" OR "Ontology")	

in order to set limits on the results, and obtain the best ones possible: i) Subject area: "Computer Science", ii) Document type: "Conference Paper AND Article AND Article in Press" and iii) Language: "English". Our search was, moreover, restricted in such a way as to analyse only those studies published between 2000 and July 2018.

3.3. Selection of studies

The selection of papers was carried out in three phases, following the guidelines provided by Kitchenham y Charters [45]:

- Phase 1: Selection of potential studies by means of the application of the search string, carrying out the identification and review of each one of the studies by title, abstract and keywords.
- Phase 2: Selection of the primary studies from amongst the candidate papers that were obtained in Phase 1, carrying out the full text analysis and the application of inclusion and exclusion criteria.
- Phase 3: Application of quality assessment to the primary studies selected in Phase 2.

The inclusion and exclusion criteria that were established in the review protocol were:

Inclusion criteria (IC):

- IC1: Articles written in English and which address risk management in the software life cycle context.
- IC2: Complete articles published or sent to a prestigious journal or congress or workshop and with peer-review, between 2000 and July 2018.

Exclusion criteria (EC):

EC1: Articles related to risk management in areas which are not specific to software engineering (i.e. information management, cybersecurity, IT services and platforms, etc.).
EC2: Articles whose contribution has nothing to do with risk management in the software life cycle, or in which this topic is dealt with only superficially.
EC3: Duplicate articles (always giving preference to the most comprehensive and most recent article).
EC4: Articles which present systematic reviews, systematic mappings and meta-analysis on risk management.
EC5: Articles that present an informal literature survey on the implementation of risk management, the use of techniques and tools on the part of the software industry (i.e., a literature survey that does not set out clearly what the research questions and objectives are; or those articles which do not specify what search process is used, or how information is extracted, for instance).
EC6: Articles which address only types of risk.
EC7: Articles related to the teaching of, and education in, risk management.
EC8: Opinion articles related to risk management, or articles available only in PowerPoint presentations or abstracts.

3.4. Quality assessment

Aiming to measure the quality of each one of the papers selected, and to obtain the best results for future research, we produced our own assessment protocol, based on five questions which were adapted from [48–50]. A scoring system was established for each of the questions, to be applied to each article; the studies could obtain a score between 0 and 5. In addition, the questions are associated with an evaluation goal related to the following aspects: (i) Relevance to what has been reported in the primary studies, according to the objectives of this SLR (criteria 1 and 2) and (ii) credibility of the results as regards the assessment methods and scientific dissemination (criteria 3 to 5).

Quality criteria (QC):

QC1: Does the paper contain a detailed description regarding to which activity/activities the risk management offers support? (Yes +1 /No +0).
QC2: Does the study describe to which process/processes of the software life cycle it can be applied? (Si +1 / No +0).
QC3: Is the paper validated as regards the type of proposal it defends? The possible answers are: Validated empirically by means of a case study, survey, or experiment (+1) / Unvalidated (+0).
QC4: Does the study present a validity plan? The possible answers are: It has validity as regards its construction, internal validity, reliability, limitations, and biases (+1) / partially (+0.5) / No type of validity (+0).
QC5: Has the paper been published in a well-known journal or conference? (With reference to the journals the JCR index [51] was used, and for conferences the CORE ranking [52] was employed). The possible answers are: Q1 or A* (+2) / Q2 or A (+1.5) / Q3 or B

Table 3
Consolidated summary of the studies taken into account in the systematic review.

Consolidated									
Data Base	Scopus								
Years covered	2000 - 2018								
Returned Results	187								
Phase 1	71								
Phase 2	IC Results	EC Results							
	IC1 and IC2	EC1	EC2	EC3	EC4	EC5	EC6	EC7	EC8
	45	1	2	12	4	2	3	1	1

(+1) / Q4 or C (+0.5) / Unranked (+0) (Adapted from [31]).

With reference to the quality assessment of the primary studies, it is important to clarify that it is a process that may be full of biases; in an effort to reduce these, it was decided that the application of the quality criteria should be carried out by the principal investigator, and that the results should be shared with the other researchers and checked by them, in the quest to resolve any disagreement that might arise. As regards the scores obtained for each of the studies, these were not used to exclude the papers from the SLR; they do, however enable us to determine which studies are the most relevant and representative for consideration in future research. Finally, Table 4 presents a complete statement of the results obtained for each of the primary studies, while Section 4.1.6 carries out a general analysis with respect to assessment or experimental validation.

4. Results

After applying the search string described above, 187 papers were found by the bibliographic data base. In Table 3 a consolidated summary of the studies analysed and chosen in the systematic review is presented. The list of studies that came about as a result of this search was reduced to 71 articles by the end of Phase 1. A careful analysis and use of inclusion and exclusion criteria in each one of the articles of Phase 2 enabled us to choose 45 articles (see results in Table 3). In Table 4, we list each of the articles in our study with the letter "P", and a number for the paper; for example; [P1]. This convention is used to distinguish the primary studies from the other references used in the paper. In Phase 3, all the quality criteria were applied to each of the papers and, as was explained in Section 3.4, this quality assessment was not used to reduce the number of articles, but rather to identify the most important ones for future research. Table 4 displays the quality assessment scores for each of the papers chosen.

In Fig. 1, the network of co-occurrence of keywords in the primary studies, built using the VOSviewer [98,99] software tool, is shown. All of the keywords that were supplied by each of the authors in the different primary studies, along with those used in each of the publications for their indexation (*Index Keywords*), were employed in the creation of this network. It is made up of 43 keywords, which were chosen if they occurred at least 3 times. It can be observed in this figure that the knowledge domains connected with risk management that are of most interest to the scientific community are those related to the risk assessment process, software design, perception of risk, and risk analysis. It can also be seen that a frequently-occurring application domain in the studies is that of software projects, whether at project level, management level, or in terms of making known the range of risks that may directly affect software projects. Lastly on this point, it is worth highlighting the integration of techniques from other knowledge areas to the risk management process; those techniques take on the task of strengthening and improving this process.

4.1. Results of the research questions

This section reports the results of the research in relation to each of the primary studies, findings which were obtained using the classification criteria designed for each of the research questions described previously.

4.1.1. RQ1: What strategies have been defined for risk management in the software life cycle?

As regards the different strategies aiming to support the risk management process, it was established that these do so in a range of application domains. For example, in the development of software for medical devices, the framework for risk management known as RiskUse [57] was found, as was the model of capacity for risk management designed by McCaffrey [87] based on the area of risk management processes as defined in CMMI-DEV v1.2 [100] and on different international regulations regarding control of medical devices. In relation to software projects, the model proposed by Islam et al. [64] for goal-driven software development risk management appeared; it is one that is integrated explicitly into the requirements engineering stage. This is also the case with the model and prototype tool for software risk management, designed by Keshlaf y Hashim, and known as SoftRisk [97]. For agile projects, a lightweight approach supported by a prototype tool was discovered, in the work of Odzaly et al. [69]; its aim is to reduce human effort in risk management tasks by using software agents to carry out the identification, assessment and monitoring of risk. Elzamly et al. [65] propose a methodological framework for the software life cycle; it is based on statistical techniques and on extracting information that offers support in software risk management. An analysis of the studies mentioned showed that only the paper by McCaffrey et al. [87] is supported by a reference model that is used on a worldwide scale, namely CMMI-DEV v1.2 [100], and it is in harmony with international regulations on the control of software in medical devices. The remaining papers lean on other research studies which are not, however, standardised.

Table 5 displays the primary studies that support the risk management process, along with the bases used to support their production.

4.1.2. RQ2: What risk management activities have been covered by the research studies?

In an effort to identify what risk management activities have been covered by the research studies, the ISO 31,000 [6] was employed, aiming to classify the information with respect to the activities of a widely-known and accepted risk management process. The ISO 31,000 [6] provides the generic principles and guidelines for this process, and those concepts are applicable to any organization, of no matter what type. The activities established for this process are: "communication and consultation", "scope, context, criteria", "risk identification", "risk analysis", "risk evaluation", "risk treatment", "monitoring and review" and "recording and reporting" [6].

Fig. 2 presents the distribution of the primary studies in terms of the particular activities in the risk management process to which they give cover. Similarly, Table 6 provides a detailed analysis about which activities in the risk management process are covered by each of the primary studies, as well as their classification according to the types of strategies which have been associated with the terms used by each of the researchers in the description or definition of their research study

Table 4
Primary studies of the SLR and their quality assessment scores.

Primary studies	Authors and title	Scores (max.6)	References
[P1]	Arun Kumar Sangaiah, Oluwarotimi Williams Samuel, Xiong Li, Mohamed Abdel-Basset, and Haoxiang Wang. 2017. Towards an efficient risk assessment in software projects-Fuzzy reinforcement paradigm.	4	[53]
[P2]	Jianping Li, Minglu Li, Dengsheng Wu, Qianzhi Dai, and Hao Song. 2016. A Bayesian Networks-Based Risk Identification Approach for Software Process Risk: The Context of Chinese Trustworthy Software.	4	[54]
[P3]	Afriyanti Dwi Kartika and Kridanto Surendro. 2016. A fuzzy-based methodology to assess software usability risk.	2.5	[55]
[P4]	Josua Johan Pandapotan Sipayung and Jaka Sembiring. 2016. Risk assessment model of application development using Bayesian Network and Boehm's Software Risk Principles.	2	[56]
[P5]	Christin Lindholm. 2015. Involving user perspective in a software risk management process.	5	[57]
[P6]	Shou Yu Lee and Yihao Li. 2015. DRS: A Developer Risk Metric for Better Predicting Software Fault-Proneness.	3	[58]
[P7]	Ching Pao Chang. 2015. Software Risk modelling by Clustering Project Metrics.	3.5	[59]
[P8]	Chandan Kumar and Dilip Kumar Yadav. 2015. A Probabilistic Software Risk Assessment and Estimation Model for Software Projects.	3	[60]
[P9]	Changkyun Jeon, Neunghoe Kim, and Hoh Peter. 2015. Probabilistic Approach to Predicting Risk in Software Projects Using Software Repository Data.	3.5	[61]
[P10]	Mukesh Vijay Goyal, Shashank Mouli Satapathy, and Santanu Kumar Rath. 2015. Software project risk assessment based on cost drivers and Neuro-Fuzzy technique.	2	[62]
[P11]	Shruti Patil and Roshani Ade. 2015. A software project risk analysis tool using software development goal modelling approach.	3.5	[63]
[P12]	Shareeful Islam, Haralambos Mouratidis, and Edgar R. Weippl. 2014. An empirical study on the implementation and evaluation of a goal-driven software development risk management model.	5	[64]
[P13]	Abdelrafe Elzamly and Burairah Hussin. 2014. An enhancement of framework software risk management methodology for successful software development.	2	[65]
[P14]	Jun Liu and Jianzhong Qiao. 2014. A grey-based rough set approach for software risk prediction: A case study.	4	[66]
[P15]	Lei Bai and Fuling Li. 2014. The model of project risk assessment based on BP neural network algorithm.	3	[67]
[P16]	Kulbhushan Bansal and Harish Mittal. 2014. Analysis and evaluation of software aggregative risk using soft computing techniques.	2	[68]
[P17]	Edzreena Edza Odzaly, Des Greer, and Darryl Stewart. 2018. Agile risk management using software agents.	5	[69]
[P18]	Christian Haisjackl, Michael Felderer, and Ruth Breu. 2013. RisCal - A risk estimation tool for software engineering purposes.	3.5	[70]
[P19]	Yong Hu, Xiangzhou Zhang, E. W.T. Ngai, Ruichu Cai, and Mei Liu. 2013. Software project risk analysis using Bayesian networks with causality constraints.	2	[71]
[P20]	S. Laqrichi, D. Gourc, and F. Marmier. 2013. Toward an effort estimation model for software projects integrating risk.	4	[72]
[P21]	Ajay Jaiswal and Meena Sharma. 2013. Expert webest tool: A web based application, estimate the cost and risk of software project using function points.	2.5	[73]
[P22]	Yu Wang, Shun Fu, and Teng Zhang. 2012. Ranking software risks based on historical data.	2.5	[74]
[P23]	Masood Uzzafer. 2011. A novel risk assessment model for software projects.	2	[75]
[P24]	Tetyana Bragina and Galina Tabunshchyk. 2011. Fuzzy model for the software projects design risk analysis.	2	[76]
[P25]	Francisco Reyes, NarcisoCerpa, Alfredo Candia-Véjar, and Matthew Bardeen. 2011. The optimization of success probability for software projects using genetic algorithms.	3	[77]
[P26]	D Wu, H Song, M Li, C Cai, and J Li. 2010. modelling risk factors dependence using Copula method for assessing software schedule risk.	3	[78]
[P27]	Mohd. Sadiq, M W Ahmad, Md.K.I. Rahmani, and S Jung. 2010. Software Risk Assessment And Evaluation Process (SRAEP) using model based approach.	2	[79]
[P28]	M Sadiq, A Rahman, S Ahmad, M Asim, and J Ahmad. 2010. EsrcTool: A tool to estimate the software risk and cost.	2	[80]
[P29]	M Uzzafer. 2010. A financial tool for software risk measurement.	2	[81]
[P30]	A Hosseingholizadeh and A Abhari. 2010. A new compound metric for software risk assessment.	4	[82]
[P31]	L Minglu, L Jianping, S Hao, and W Dengsheng. 2009. Risk management in the trustworthy software process: A novel risk and trustworthiness measurement model framework.	3	[83]
[P32]	P Cao and F Chen. 2009. A risk control optimization model for software project.	2	[84]
[P33]	J Gao, M Shah, M Shah, D Vyas, P Pattabhiraman, K Dandapani, and E Bari. 2009. Systematic risk assessment and cost estimation for software problems.	3	[85]
[P34]	W E Wong and Y Qi. 2009. BP neural network-based effective fault localization.	4	[86]
[P35]	F Mc Caffery, J Burton, and I Richardson. 2009. Risk management capability model for the development of medical device software.	3	[87]
[P36]	D Gupta and M Sadiq. 2008. Software risk assessment and estimation model.	2	[88]
[P37]	R Hewett and A Thipse. 2007. Building business considerations into enterprise application designs.	2	[89]
[P38]	Y Takagi, O Mizuno, and T Kikuno. 2005. An empirical approach to characterizing risky software projects based on logistic regression analysis.	5.5	[90]
[P39]	X Liu, G Kane, and M Bambroo. 2003. An Intelligent Early Warning System for Software Quality Improvement and Project Management.	2	[91]
[P40]	X Ruzhi, L Qian, and X Jing. 2003. CMM-based software risk control optimization.	2	[92]
[P41]	D E Neumann. 2002. An enhanced neural network technique for software risk analysis.	4	[93]
[P42]	S M Yacoub and H Ammar. 2002. A methodology for architecture-level reliability risk analysis.	5	[94]
[P43]	D X Houston, G T Mackulak, and J S Collofello. 2001. Stochastic simulation of risk factor potential effects for software development risk management.	3	[95]
[P44]	D Gotterbarn. 2001. Enhancing risk analysis using software development impact statements.	2	[96]
[P45]	A Keshlaf and K Hashim. 2000. A model and prototype tool to manage software risks.	2	[97]

(e.g. approach, methodology, method, framework, and so on). It is also important to clarify that a primary study may cover more than one activity. In that sense, the total amount of the distribution in Fig. 2 is greater than the amount of primary studies found. As may be observed in the results of this classification, the activities of greatest interest in the software risk field are risk analysis, risk evaluation, and risk identification (the ISO 31,000 [6] regards the set of these activities as a fundamental part of the risk assessment process). In addition, this risk assessment process will only be carried out in a systematic way if the methods and techniques used to carry it out are well-chosen. The results of the process will be the key components employed in making decisions about what approaches may be considered most suitable for the

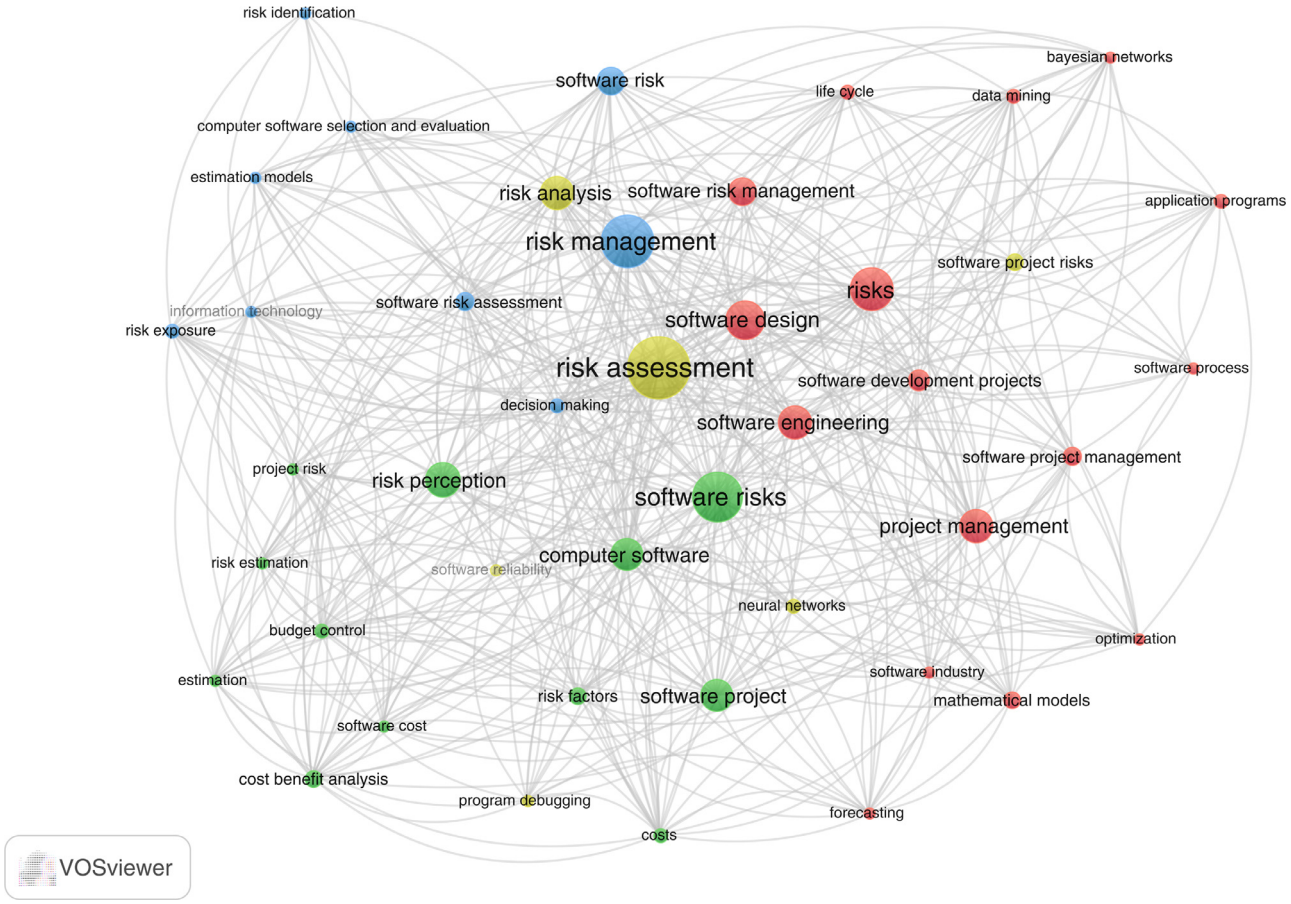

Fig. 1. Network of co-occurring keywords in the primary studies.

Table 5
Software risk management studies and the bases they are built on.

#	Study	Basis
1	RiskUse [57]	Lindholm C et al. [101], Vilbergsdottir S.G. [102]
2	Islam et al. [64]	Van Lamsweerde, A. [103]
3	Elzamly et al. [65]	No evidence
4	Odzaly et al. [69]	Islam, S. [104]
5	McCaffery et al. [87]	CMMI 1.2 [100], ISO 14,971 [105], Set of FDA guidelines [106-111], GAMP 4 [112], IEC: 62,304 [113], 6061–1–4 [114]
6	SoftRisk [97]	Keshlaf, A. y Hashim, K. [115]

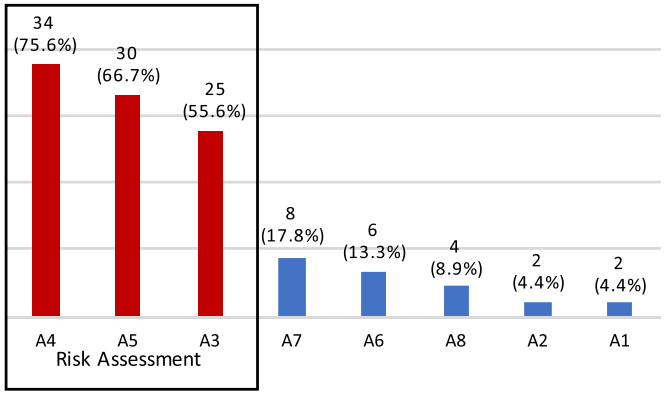

Fig. 2. Distribution of the primary studies in relation to the risk management activities defined by the ISO 31,000.
Acronyms and Abbreviations - A1: Communication and consultation, A2: Scope, context, criteria, A3: Risk identification, A4: Risk analysis, A5: Risk evaluation, A6: Risk treatment, A7: Monitoring and review, A8: Recording and reporting.

treatment of risk [116].

It was noteworthy that, in spite of the fact that there are several studies that cover the different activities in the risk management process, there are in fact very few papers that give a detailed definition of the purpose of the activities in each of their studies. In the following paragraphs there is a description of each of the activities in the risk management process, as set out in the ISO 31,000 (using the same terms used in Fig. 2); each of the primary studies is then organised in relation to the activities established by that standard.

- **A1 - Communication and consultation:** This refers to the continuous and reiterative processes that an organization carries out in its quest to supply, share, and obtain information by means of meetings or informed dialogues with the stakeholders on a given issue related to risk management before decisions are made, or in an effort to seek guidance and direction on that particular topic. In other words, the communication process has the task of producing awareness of risk in such a way that it is understood by each of the stakeholders; this guarantees proper treatment of the risk involved. The consultation process looks to promote opportune feedback, and to obtain information that will give support to the decision-making process with regard to the risk management process cycle. Lindholm [57], in her preparation stage of the risk management process, places great emphasis on this activity by establishing the risk management team; this should be made up of: (i) Developers who have some knowledge of the context of the system to be developed, (ii) users to represent other user-groups; they will form part of the

Table 6
Distribution of the primary studies in the risk management activities and the types of strategies used by each.

Primary studies	Risk Management Activities								Types of Strategies						
	A1	A2	A3	A4	A5	A6	A7	A8	T1	T2	T3	T4	T5	T6	T7
[P1]					✓				✓	✓					
[P2]			✓						✓						
[P3]			✓	✓	✓							✓	✓	✓	
[P4]			✓	✓	✓								✓		
[P5]	✓	✓	✓	✓	✓		✓	✓		✓					
[P6]					✓									✓	
[P7]			✓						✓		✓				
[P8]			✓	✓	✓							✓	✓		
[P9]			✓	✓	✓								✓		
[P10]			✓	✓	✓								✓		
[P11]			✓	✓					✓						✓
[P12]		✓	✓	✓	✓	✓				✓			✓		
[P13]	✓		✓	✓	✓	✓	✓	✓		✓		✓			
[P14]				✓	✓						✓				
[P15]			✓	✓	✓								✓		
[P16]					✓							✓	✓		
[P17]			✓	✓	✓		✓	✓	✓						✓
[P18]			✓	✓	✓								✓		✓
[P19]				✓						✓		✓	✓	✓	
[P20]			✓	✓	✓				✓				✓		
[P21]			✓	✓	✓		✓								✓
[P22]				✓							✓			✓	
[P23]			✓	✓	✓								✓		
[P24]				✓	✓								✓		
[P25]			✓								✓		✓	✓	
[P26]					✓								✓		
[P27]			✓	✓	✓								✓		
[P28]				✓										✓	✓
[P29]				✓										✓	
[P30]				✓					✓					✓	
[P31]							✓			✓			✓	✓	
[P32]							✓		✓				✓		
[P33]			✓	✓	✓	✓			✓		✓				✓
[P34]				✓							✓				
[P35]		✓	✓	✓	✓	✓							✓		
[P36]			✓	✓	✓								✓		
[P37]				✓	✓				✓						
[P38]					✓				✓						
[P39]			✓	✓	✓				✓					✓	✓
[P40]					✓		✓				✓				
[P41]				✓					✓						
[P42]			✓	✓	✓							✓		✓	
[P43]				✓		✓							✓		
[P44]				✓											✓
[P45]			✓	✓	✓	✓	✓	✓					✓		✓

Acronyms and Abbreviations - A1: Communication and consultation, A2: Establishing the context, A3: Risk identification, A4: Risk analysis, A5: Risk evaluation, A6: Risk treatment, A7: Monitoring and review, A8: Register of the process for the risk management T1: Approach, T2: Framework, T3: Method, T4: Methodology, T5: Model, T6: Measures and T7: Tool.

risk management process, (iii) the risk manager, as the person responsible for the risk management process, and who is in charge of presiding over the different meetings related to the process, and (iv) an assistant risk manager, who will be responsible for the documentation in each of the meetings.

- **A2 - Scope, context, criteria:** This activity establishes the internal and external parameters that should be considered when managing a risk, and when establishing the scope of the risk criteria that will make it possible to define the policies for risk management; the aim is to carry out a risk assessment that is effective, alongside treatment that deals with risk appropriately. In her preparation stage for the risk management process, Lindholm [57] proposes that preparations for the whole process be carried out, according to the necessary requirements and the intentions of use. In addition, having an understanding of the context of the system is recommended, since this will make it easier to set up the risk team, making sure that each of the competences needed is catered for.
- **A3 - Risk identification:** This is the process that makes it possible to find, recognise, and describe the different risks, the sources of the risks, and the events, in addition to identifying each of the causes and their potential consequences. Sipayung et al. [56] define risk identification as an early stage in risk assessment which often allows the risks involved in the development projects for applications to be pinpointed. The authors use a set of risks identified in software projects as a guide in this stage; these are listed in [117]. Haisjackl et al. [70], for their part, state that this stage should make it possible to identify the different risk elements, as well as to then store them on a list of risk elements, since they will serve to establish the criteria and measurements that will enable us to determine the probability of risk, the factors of impact and of evolution of risk. This stage also provides a procedure that allows the risk exposure value to be calculated. In similar vein, Jaiswal et al. [73] affirm that it is at this stage that a list should be produced, containing the specific elements of risk that exist in the project and that could endanger its success. (ii) Jaiswal contributes by recommending the following techniques to support this stage: examination of the project guidelines, creation of hypotheses or suppositions, analysis and checklists. In their framework, Islam et al. [64] establish a layer that they call

"obstacles", the purpose of which is to (i) identify the possible risk factors in the software development, and (ii) determine the lack of fulfilment of the project goals which these risk factors produce. Elzamly et al. [65] put forward the idea of a risk identification stage, composed of three stages, which are (i) making a risk plan; its aim is to establish the responsibilities of all those involved in the software projects (e.g. project leads, members of software development team, stakeholders, and so on). It also produces the organizational scheme, which should follow an iterative approach that makes it possible to manage risk, (ii) the risk identification stage, which enables a general vision of risk to be obtained, while also identifying the software risk factors, and the management strategies; in addition, it clarifies the effects risk has on measurement–on the quantity and quality techniques, for example—and (iii) risk prioritization, a stage which allows the degree of risk to be categorised, as determined by the range, probability, and impact of the particular risk. Lastly, Keshlaf y Hashim [97] regard risk identification as a stage that should enable all the potential risk in projects to be identified; it can be carried out from two different perspectives: (i) using the data on risk that can be employed in any type of software development project, and (ii) for data on risk in specific projects that may affect the projects that are planned.

- **A4 - Risk analysis:** This is the process that enables there to be an understanding of the nature of risk, while also establishing the level of risk (expressed in terms of the consequences and causes of the risk involved). Moreover, it takes on the task of providing the bases for risk evaluation, serving as input for decision-making as regards the treatment, the strategies and the methods that are the most suitable for dealing with each risk as corresponds; it does the same for risk estimation. Lindholm [57], in her discussion on risk analysis, sees this as the risk associated with each dangerous situation identified; it is estimated using a scale of risk severity and risk probability, where the risk value is defined by multiplying the severity of the risk by the probability value. Elzamly et al. [65], in their contribution, state that risk analysis contributes to the analysis of risk probability and its consequences with reference to those risks that have been identified. They also assert that in risk analysis it is possible to carry out an estimation of impact, exposure, and relationships between risks, as well as analysing the options for risk mitigation and the actual mitigation strategies used. Haisjackl et al. [70] establish that in risk analysis exposure to risk can be calculated by applying some metrics. These measures can be applied manually; it is recommended that the stakeholders should take part in this process. When they are applied automatically, this will be by means of the RisCal tool; this incorporates a statistical tool like Sonar, which makes it possible to carry out an analysis of the source code of an informatics application. Jaiswal et al. [73] state that in risk analysis the magnitude and probability of loss for each one of the elements of risk identified can be evaluated, using performance models, network analysis and cost models, and so on.
- **A5 - Risk evaluation:** This is the process of comparison of the results of the risk analysis with each of the criteria or terms of reference, allowing there to be a prioritisation of risk, and enabling decisions to be taken as regards the execution of treatment of risk. Lindholm [57] asserts that risk evaluation is the decision-making on each of the risks in relation to their being reduced. These decisions are made through criteria that are based on control measures, which can be debated and then deployed in order to reduce the risk. Should the control measures allow new risks to be found, these should be documented and analysed, so that their size may be established. Elzamly et al. [65] say that the purpose of risk evaluation is to determine the levels of the identified information risks, so that project managers can carry out different risk control strategies, while also preventing the risks from becoming reasons for delays in completion of projects.
- **A6 - Risk Treatment:** This is seen as the process that enables risk to be modified by the choice of plans to treat it, along with the implementation of these. This process is meant to remove the risk, or to change the probability and/or its consequences. Islam et al. [64] assert that the treatment of risk focuses on the control actions that make it possible to counter the risks, so that the objectives can be achieved. Its main goal is also to obtain control of risk as soon as possible, preferably from the requirements engineering stage onwards. Treatment of risk makes it possible for there to be modelling, analysis, reasoning, and tracking of the control actions that have been adopted for the mitigation of risk and to satisfy the project objectives. Elzamly et al. [65] focus on risk treatment from the viewpoint of four strategies that make it possible to respond to risk; these are: (i) Avoidance of risk, so that the appropriate action can be taken for the probability of risk to drop to zero, (ii) transfer of risk, aiming to share the risk management with other members of the team, (iii) mitigation of risk, which sets out to reduce the probability of risk, or its impact, or both, and (iv) acceptance of risk, which allows us to analyse and understand the consequences and impact of the occurrence of risk, or of exposure to it. Keshlaf y Hashim [97] make reference to this activity of risk treatment in a stage of the model known as control; this is based on the severity of the risk, and on the execution of techniques that are suitable for reducing that risk. A plan for mitigation, contingency, or crisis, can be considered at this stage. They also recommend that re-estimation, re-evaluation and documentation of the action carried out on the risk should be performed after the application of the reduction techniques.
- **A7 - Monitoring and review:** The processes are designed to guarantee the efficiency of the risk treatment strategies and their management plans, as well as to analyse and assimilate the lessons learnt about these treatments so that risk assessment can be improved by registering it and then publishing it (internally and externally). Lindholm [57] states that in this activity the control measures that will be applied to risk should be discussed and debated. In addition, she suggests that the risk should be analysed once again, using the scales that have been defined, and that it will be possible to include new problems, as well as to discuss and identify residual risk. In risk monitoring the tasks of verification and validation of the risk control activities should be performed, thereby updating the risk values, and taking decisions about whether new risk control measures should be implemented. Elzamly et al. [65] assert that this activity is the action of controlling the handling of risk reduction, according to the control measures and the levels of risk, along with the decision-making actions regarding rolling out contingency plans, tracking them, and seeing that the risk mitigation is finally achieved satisfactorily. Jaiswal et al. [73] establish that risk monitoring should mean keeping track of the programme of the project, and lead towards risk resolution and proper follow-up of corrective action.
- **A8 - Recording and reporting:** This is the mechanism that offers the bases that make it possible to carry out the improvement of the risk management process, and of the methods and tools that are used in the process. The records should be produced in the different activities of the risk management process, so that a culture of continuous learning is generated. The recording and reporting must therefore ensure that it can be proved that the different management actions to do with the risk have actually been completed. They should also allow the lessons learnt to be consolidated; the goal is for them to be applied in fostering ongoing improvement of the process and of the risk management system. Lindholm [57] considers the documentation process in the preparation stage of risk management; this is a process that consists in producing all the documentation regarding: (i) The risk management plan, which depends on the whole process of assessment and acceptance of risk, (ii) the risk meetings and (iii) the report on the entire risk management process. Elzamly et al. [65], for their part, have affirmed that this stage is fundamental for the success of the risk management, since it enables there to be documentation of the strategies

followed to solve the problems and to mitigate risk with respect to the successes and the mistakes that occur during the management process. Keshlaf y Hashim [97] contribute by stating in their documentation stage that this should support the whole process of the documentation of the information related to risk, so that it may be used later in following up the situations of the projects, as well as in statistical analysis and in predicting future risk.

4.1.3. RQ3: What software life cycle processes have been covered by the research studies?

To establish what software life cycle processes have been covered by the research studies, the ISO 12,207 [7] was used; this establishes a set of processes, activities and tasks involved in a common framework for the software life cycle; these are applied from the acquisition and configuration of the services of the system until the end of its use. This standard has 29 processes, which have been grouped into four process categories; these are: "agreement processes", "organizational project-enabling processes", "technical management processes", and "technical processes" [7].

As regards the distribution of the primary studies in terms of the software life cycle processes, we could see that the processes focused on by the majority of the pieces of research work are those grouped together in the category of technical processes; the aim of these is to provide support for the transformation of software system requirements in a product; that transformation should, amongst other things, comply with the specifications of the users with regard to the type of service that the product will offer. In Fig. 3, the distribution of each of the primary studies in each of these processes can be seen. It is important to appreciate that throughout the software product life cycle (which is made up of four fundamental stages: software development, software operation, software maintenance and software disposal), risk management has been given most attention during the development of the software. This is seen clearly in the fact that the authors involved have tackled the issue of risk management in processes such as: implementation, stakeholder needs and requirements definition, system/software requirements definition, verification, design definition, system analysis, architecture definition, and validation. There are only a few papers, however, which have sought to address the other fundamental stages of the product life cycle; i.e., maintenance, operation and disposal. This shows very clearly where the interest on the part of the scientific community lies; they are concerned with helping to reduce and mitigate risk, right from the time the product is conceived of and through the development stages of the software development life cycle, by managing each of the technical and process risks. The goal is to minimise the impact on the product and achieve success in its development.

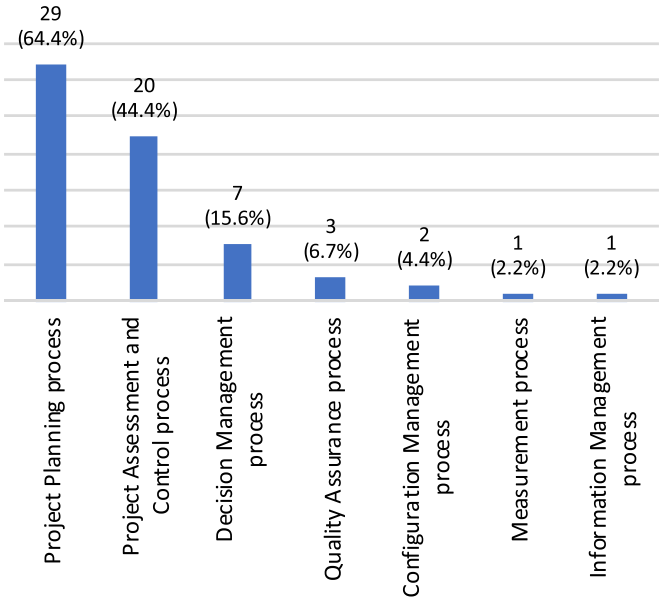

Fig. 4. Distribution of the primary studies in the technical management processes.

Another category of process covered by the primary studies is the one that includes processes related to technical management. These types of processes deal with the technical management of projects, as well as with decision-making. Fig. 4 displays the individual distribution of each of the primary studies in relation to these processes, where it is possible to see that the studies cover risk management in the context of the software project, as well as to its assessment and control, and to decision management. This is very plain to see, since the great majority of the primary studies address the issue of risk management in the

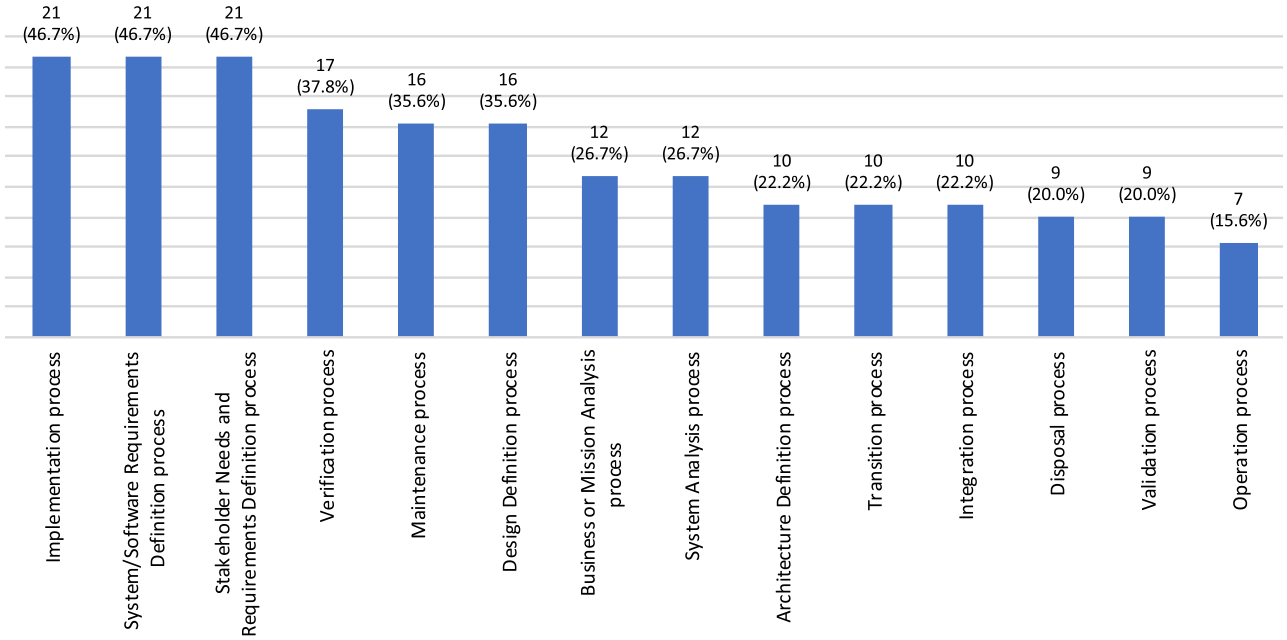

Fig. 3. Distribution of the primary studies in the technical processes.

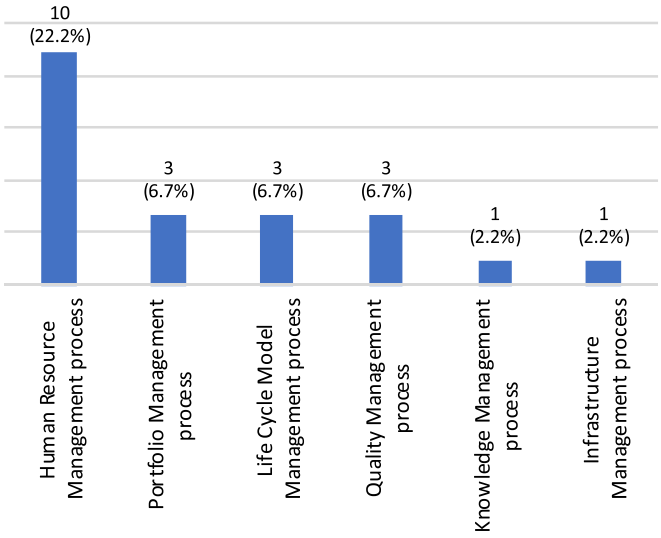

Fig. 5. Distribution of the primary studies in the organizational project-enabling processes.

context of software projects. As may be seen in Table 6, some of the pieces of research work make use of other knowledge areas to strengthen and improve the risk management. In addition, there is a clear lack of studies that enable risk management to be supported in aspects such as the quality assurance of software projects and products that would guarantee that the product fulfils the technical requirements established by the client effectively. There is also a scarcity of papers that support the measurement process using pieces of research work which provide organisations with mechanisms making it possible to measure the performance and efficiency of the risk management; we notice that same deficiency as regards the amount of studies that would assist in estimating risk at project, process and product level, offering support for decision-making throughout the risk management process, as well as in improving practices in this area.

The process categories treated least by the primary studies are the organisational ones; they are the organizational project-enabling processes that facilitate the agreement process (see Fig. 5). Of all the processes that form part of these two categories, we could see that the human resources management process is the one that was involved most frequently in the studies that had been chosen. The agreement processes are included least often in the primary studies, since only one piece of work contains these. This may be due to the fact that over the last decade organisations have dedicated their efforts to improving the quality of software process, as well as to assuring the quality of the products; they have, however, tended to ignore the organisational processes. It was also clear that the primary studies did not take into account agile approaches in project management and software development.

Finally, of all the primary studies analysed, it was evident that only the one put forward by Minglu et al. [83] focuses primarily on the software life cycle processes. In that study, a reliable integrated process model is presented; it is made up of three main components; risk management of software processes, management of the software development life cycle, and the supervision of the reliability of the deliverables. The software processes established for this process are: acquisition, supply, development, operation and maintenance, all of which were defined in compliance with the ISO 12,207 [118] standard.

4.1.4. RQ4: What is the application domain of the research studies? What particular knowledge areas, techniques or models or tools do they involve?

In Table 7, there is a general classification of each of the studies according to the application domain, knowledge areas, and techniques or algorithms that they have to do with. It is also important to clarify that a primary study may cover more than one application domain and involve different knowledge areas. As may be observed in the table, most of the primary studies address software risk management in the context of software project management and software development. It is therefore clear that there are very few studies that look at the management of risk in specific domains of software application development, and the only case is in fact in relation to the development of medical devices, with two papers. As regards risk management in agile projects, only one piece of research work was found, made up of a risk management model and a prototype tool.

It should also be noted that 29 papers, that is to say 64.4% of the studies analysed, incorporate other areas of knowledge to give support to their initiatives. Of these, 15 (51.7%) studies use artificial intelligence techniques, such as fuzzy logic, Bayesian networks, artificial neural networks, software agents and so on, in their quest to assist and support decision-making in risk management activities. There are 7 (24%) pieces of work which make use of statistical techniques to support the risk assessment process activities; some of the most widely-used are the Monte Carlo method, simple and multiple linear regression, and analysis of principal components, amongst others. It should also be remembered that this SLR does not aim to verify the effectiveness of the techniques or algorithms in the process or activities of risk management; it sets out rather to make known which of these has/have been incorporated into the different studies.

Due to the large number of primary studies analysed in this paper, we report the incorporation of some of these knowledge areas and their techniques in the risk assessment process, since that process has been taken into account by 46.7% (see Fig. 2) of the primary studies selected.

Kartika et al. [55] propose a usability risk assessment model, based on fuzzy logic; their model makes it possible to help the development team to identify, analyse and evaluate the different risks that may occur in the various stages of software development. In similar vein, a fuzzy-logic-based methodology has been proposed to measure the magnitude of the risk through an assessment of the risk factors, the likelihood and severity of risk. Goyal et al. [62] propose a model that integrates neural networks and fuzzy logic capability based on the COCOMO Expert modelling the effort to improve the precision of the risk assessment process. Liu et al. [91] present a proposal for an early-alert software system based on fuzzy logic; it enables the detection of risk at early stages in software development and makes an intelligent assessment of risk possible, by making use of a fuzzy logic inference engine.

Sipayung et al. [56] contribute by presenting a model that enables calculation of the probability and impact of risk in application development projects. This probability is calculated using Bayesian networks. For their part, Kumar et al. [60] put forward a probabilistic software risk estimation model, which makes use of Bayesian networks, and focuses on the main risk indicators in development projects. Bai et al. [67] propose a model based on back-propagation neural networks, making it possible for prediction of potential risk in software projects to be more effective. It also includes an indexation system as a risk evaluation tool; it consists of four main risk categories (requirement risk, management risk, technical risk and environmental risk).

Sadiq et al. [79] propose an evaluation and prioritisation process that makes use of a Software Fault Tree Approach – SFTA) for risk identification and analysis. It also performs a prioritisation of risk, employing Risk Reduction Leverage – RRL), so as to carry out a measurement of each of these.

Yacoub et al. [94] present a methodology for evaluating and analysing reliability risk at the software architecture level. This risk evaluation is conducted by means of a heuristic technique based on two dynamic metrics (dynamic coupling and dynamic complexity), as well as on the analysis of risk severity, which is carried out using the Failure Model and Effect Analysis (FMEA). When the factors of severity and complexity are combined, the heuristic risk factors can be developed for the components and connectors of the architecture. The risk analysis is done using a model and application of an algorithm which makes it

Table 7
Application domains of the primary studies, along with the knowledge areas and techniques that are involved in them.

Primary studies	Application Domain	Knowledge area	Technique/Algorithm
[P1]	Software Projects	Artificial Intelligence	Decision- making technique, Fuzzy Logic, DEMATEL, FMCDM and TODIM Method
[P4]			Bayesian Networks
[P8]			Bayesian networks or Bayesian belief network (BBN).
[P10]			Artificial Neural Networks, Fuzzy logic and Fuzzy Ex-COM (Fuzzy Expert COCOMO)
[P15]			Backpropagation Algorithm for neural networks
[P19]			Bayesian networks, V-structure discovery algorithm and Cheng's BN learning algorithm
[P24]			Mamdani fuzzy interference algorithm and Fuzzy logic
[P25]			Genetic algorithms, Bayesian Belief Network (BBN), Naïve Bayesian Classifier (NBC) and Selective Bayesian Classifier (SBC)
[P39]			Fuzzy logic
[P41]			Artificial Neural Networks (ANN), Backpropagation and Adaline
		Statistics	Principal Component Analysis (PCA)
[P9]			Markov chain
[P20]			Pearson's Correlation, One-way ANOVA for analysis of k fold cross-validation approach, Multi linear regression, Ordinal regression, Non-linear regression, MMRE, PRED, MdMRE and R2
[P23]			Beta distribution and Monte Carlo Simulation
[P38]			Logistic Regression Model
[P26]			Monte Carlo Method
		Probability	Copula Method
[P14]		Computational Intelligence	Grey System Theory and Rough Set Theory
[P7]		Data Mining	Clustering
[P29]		Finance	Value-at Risk measure and Expected Shortfall
[P40]		Algorithmics	Dynamic Programming
[P32]		Metaheuristics	Particle Swarm Optimization
[P12]		No evidence	
[P21]			
[P27]			
[P28]			
[P30]			
[P36]			
[P43]			
[P45]			
[P11]	Software development	No evidence	
[P33]			
[P44]			
[P13]		Statistics Data Mining	No evidence
[P42]		Data Structures	Component Dependency Graphs
[P6]		Machine Learning	Logistic Regression (LR), Radial Basis Function (RBF), Decision Tree (DT) and Random Forest (RF)
[P3]		Marketing	Design-Reality Gap Model
		Artificial Intelligence	Analytic Hierarchy Process (AHP) and Fuzzy logic decision-making technique
[P16]			Fuzzy logic
[P2]	Software Processes		Bayesian networks
[P31]		No evidence	
[P17]	Agile Projects	Artificial Intelligence	Software Agents
[P34]	Program debugging - Fault localization		Backpropagation (BP) Neural Network
		Machine Learning	Supervised learning algorithm
[P18]	Risk-based Testing and Release Planning	No evidence	
[P37]	Risk-based software design		
[P22]	Risk classification activities	Mathematics	No evidence
[P5]	Software Development of medical devices	No evidence	
[P35]			

possible to aggregate the risk factors to the different components and connectors of the architecture.

Elzamly et al. [65], in their methodological framework for risk management, propose the use of quantitative, qualitative and mining techniques for the activities in the risk assessment processing the quest to conduct good risk management in the software development life cycle. Uzzafer et al. [75] present an innovative software risk evaluation model that quantitatively and qualitatively classifies the impact of risk events as dependant or independent, and also treats the impact of risk events according to dependence/independence statistics.

4.1.5. RQ5: How do the respective researchers frame their research studies?

As may be observed in Fig. 6 and Table 6 (in the section on types of strategies), most of the primary studies have been defined and classified as “model”, a term which, just as in the case of the other types (approaches, methodologies, methods and frameworks), is associated with how the researchers have named and thus classified their research studies. These may address one or more of the activities of the risk management process, while a small number of the studies are related to supporting the management process. On the other hand, although it is true that many studies aim to improve the risk management activities and make them more agile by including new knowledge areas, they are in fact isolated pieces of work that show little evidence of having used a widely-recognised standard, model or *de facto* proposal in their definition. Proof of this is the fact that only 15 (33.3%) of the primary studies have actually done so. The types of support used most widely for the definition of these studies are: the software risk principles of B. Boehm [11], risk identification based on a taxonomy from SEI [119] the Software Risk Assessment and Estimation model SRAEM [88], and the proposal by Wallace et al. [120].

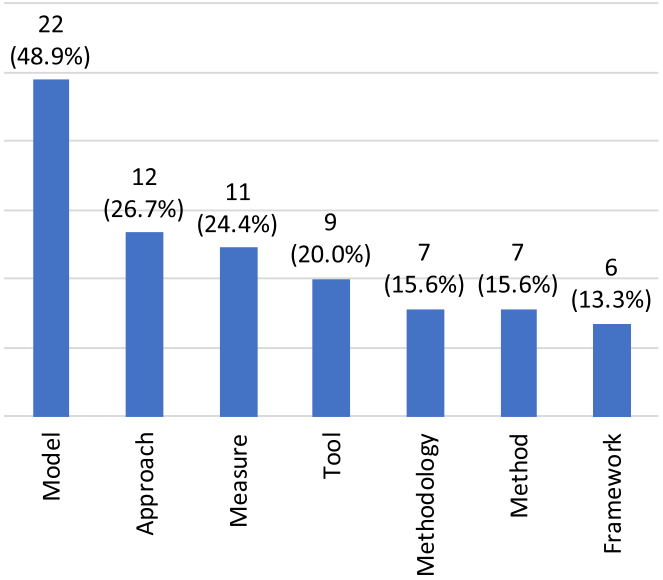

Fig. 6. Types of Strategies.

We also report that in the primary studies there were some studies found which involve technological tools that aim to support the process or the risk management activities. The main characteristics of the tools found in the primary studies are set out in Table 8. Finally, it should be highlighted that there are many more tools available to purchase, but these are outside the scope of this SLR.

4.1.6. RQ6: What methods of scientific validation have been used in the research studies?

As far as this classification is concerned, it was seen that fewer than half of the primary studies, i.e., 46.7%, provide any type of experimental evaluation or validation. Table 9 shows in greater detail the distribution of each of the primary studies with respect to the evaluation methods used to validate their proposals. It can be seen in this table that 4 primary studies carried out their evaluation by means of experiments, and 17 did so using case studies. amongst these studies, it was observed that articles [57,64], and [69] had a validation plan, which allows us to infer that this is a research field where various advances have been made, but where there needs to be greater rigour and adherence to formal guidelines; this would be achieved if correct evaluation were carried out to verify the contribution and suitability of each of the papers.

Table 9
Distribution of the primary studies according to their evaluation methods.

Evaluation	Studies
Experiment	[P7],[P8],[P15],[P38]
Case studies	[P1],[P2],[P5],[P6],[P9],[P11],[P12],[P14],[P17],[P18],[P20], [P26],[P30],[P31],[P33],[P34],[P42]

5. Limitations of this study

One of the main limitations that might have been present in our study is that other relevant articles that could exist in the literature may have been missed, since we use only the Scopus data base as our search engine. Although it certainly is one of the largest and most comprehensive data bases in existence at present, given the enormous amount of magazine articles and conference papers indexed there, it is likely that Scopus may not contain all the literature that is to be found in the field of risk management. We may also have left out some potential articles when carrying out Phases 1 and 2 of the selection process set out in Section 3.3. However, in Phase 1 there was a real effort on our part to try to verify other sections of the articles, such as their introduction or conclusion, in those cases where the title or key words or summary did not provide clarity as regards the objective or context of the research article; with respect to Phase 2, whenever there were problems in deciding whether to include or exclude an article, we discussed the particular case with the other reviewers, endeavouring to come to a fair consensus and thereby minimise any element of bias.

To carry out the quality assessment of the primary studies, we made use of our own protocol; this allowed us to obtain results which were not actually taken into account when we were attempting to choose the highest quality and most relevant articles, since it was quite likely that, this being a rather subjective activity, errors would have appeared which would have meant losing information that would been valuable for analysis in the SLR, so we decided to leave these results for possible consideration in future research.

In addition, many of the scientific articles that included terms such as *software risk* or *software life cycle* were excluded; this is because most of these papers addressed the identification of risk factors, defects, or threats in computer applications, or showed results regarding the perceptions of project managers, or presented the results of non-structured literature reviews and/or summaries. These articles were therefore not taken into account, since they did not fit into the objectives of the SLR.

As regards the classification of the articles in the software life cycle processes, this was an onerous task. Firstly, because the ISO 12,207 is made up of 29 processes (see Section 4.1.3) and secondly, because a

Table 8
Tools.

#	Tool	Characterise
1	Sketch the risk [63]	Modelling tool designed to identify risk events and avoid risk factors in software development [63].
2	Odzaly et al. [69]	A prototype tool for agile risk management, within the context of agile projects; it makes use of software agents to support risk identification, evaluation, and monitoring [69].
3	RisCal [70]	This is a generic tool that supports all the risk estimation activities. The activities in this estimation are risk identification, risk analysis, and risk prioritisation. These are meant to support and improve the estimation process of specific risk in software engineering, as well as in risk-based tests and in planning of releases [70].
4	Expert webest tool [73]	This web-based tool makes it possible to estimate the cost of software by calculating the function points and risk estimation based on an assessment of risk in software projects: its main elements are the identification, analysis and prioritization of risk. In addition, the tool enables risk to be calculated in the different stages of the project as these progress from one stage to another [73].
5	EsrcTool [80]	This is a tool that makes it possible to estimate software risk, along with the cost of the software, using the function point approach [80].
6	Gao et al. [85]	As a prototype tool for the evaluation of risk built for eBay Inc., it helps engineers to conduct an error-based analysis of risk in order to liberate a software product. It also allows there to be an estimation of the costs and impact of the project that are related to errors during the product life cycle [85].
7	Liu et al. [91]	Proposal for an early-alert software system based on fuzzy logic; it uses a set of integrated metrics to detect risk at very early stages of the software development [91].
8	D. Gotterbarn [96]	This tool makes it possible to illustrate the process and to conduct an analysis and monitoring of the impact of risk on the project [96].
9	SoftRisk [97]	A prototype tool for software risk management [97].

large amount of the studies analysed did not provide an explicit description of which processes of the software life cycle were covered by means of the research work; this situation complicated the classification process.

It was also possible to establish that researchers used the terms *approach, methodology, method* and *framework* indiscriminately in their work. Moreover, they used more than one of these terms in the description of that work. This created difficulties in classifying these studies according to the types of strategy represented; we found it complicated to decide exactly which of the terms they had employed should be the main one to use when categorising the studies. This being so, in the quest to minimise the risk of making mistakes in this classification, it was decided to distribute the primary studies on the basis of each of the terms used by the researchers in their papers. We therefore encourage researchers to be more rigorous in their description and definition of their articles, in order to avoid ambiguity or misinterpretation. Furthermore, we foresee that part of future work will be to analyse in greater depth whether or not these studies really comply with the types of strategies used to name or classify them.

Furthermore, it was shown that 64.4% of the total amount of the primary studies incorporate other knowledge areas by means of techniques or algorithms to support their initiatives in the risk management field. One limitation that was detected, however, was the lack of information regarding the scientific validation, as only 46.7% of all the primary studies present one of the empirical methods described in Section 4.1.6. and presented in Table 9. In regard to this situation, we acknowledge that this may create misunderstandings with respect to the scientific relevance of the scientific papers chosen, and regarding the extraction of information from them. In our case, however, this information is valuable, since it enabled us to obtain an overview of the state of the art, also allowing us to see gaps and trends in the field of risk management.

6. Discussion, conclusions and future work

This article has presented a systematic review which aimed to identify, analyse and describe the characteristics of the state of the art with respect to software risks in the software life cycle. After conducting a search of potential studies, and having selected the primary studies, the different types of strategies that addressed the risk management process were identified. The studies were also classified according to the risk management activities to which they offer cover, using the ISO 31,000 standard. It was shown that the activities that have been dealt with most by the authors are: (i) Risk identification, (ii) risk analysis, and (iii) risk evaluation. These activities, when taken together, make up the risk assessment process, which offers the key elements for decisions to be made on the action needed with respect to the treatment of risk.

In addition, we sought to analyse the application domain of the different studies, and it was thus possible to show that a large percentage of these are directed at risk management in: (i) The context of management related to software projects and (ii) in the technical context, related to the stages of the software development life cycle. At the same time, it was observed that 64.4% of the studies analysed involve other knowledge areas, especially those of artificial intelligence and statistical techniques, all of which enable decision-making in the risk management activities to be assisted and supported.

Although it is true that this SLR does not set out to compare the ISO 31,000 with the approaches analysed in the literature, and with those implemented by the software industry, this standard can be shown to be one which, conceptually speaking, has practices that are similar to the risk management processes proposed in PMBOK, PRINCE2, CMMI, etc. [121–124]. The ISO 31,000 could help organisations to implement integrated risk management. By "integrated risk management" we mean management that may be considered by all the levels in the organisation and in which the same principles, objectives and processes for addressing risk are shared by all involved [125,126]. Moreover, it is a standard which, according to the particular needs of the organisation, can be adapted and combined with other approaches/standards that deal especially with risk. The aim would be to improve this management, making it more efficient and effective [6]. This is possible thanks to the fact that the ISO 31,000 offers the general guidelines for carrying out strategic and overall risk management; other approaches, for their part, provide the specific recommendations for addressing risk according to its particular context.

We should also highlight that we were able to identify on which process in the software life cycle the different studies focus. The ISO 12,207 standard was used for that purpose; it was found that the processes which deal most with the issue of risk management are those which are grouped in the categories of technical processes and of technical management. The technical processes which were taken into account most frequently by the authors in the different studies are: (i) Implementation (ii) definition of the system/software requirements and (iii) definition of the requirements and needs of the stakeholders. In the technical management processes, however, those seen most are (i) planning of projects, and (ii) control and evaluation of projects.

As far as the type of strategy that the different primary studies represent, it was seen that the vast majority are models; it was apparent that just like other types (approaches, methodologies, methods and frameworks), they are generally designed to support one, or several, activities of the risk management process, apart from including techniques from other areas of knowledge to improve the implementation of these activities. At the same time, there was a complete lack of internationally-recognised standards or *de facto* models that support the definition of these pieces of research work. It was also shown that one of the weak points as regards the methodological types (model, approaches, methodologies, methods and frameworks) is that no author states exactly what the features of the methodological element they are proposing are. This means that in the field of software engineering it is necessary to consider or analyse an ontology or knowledge area that enables each of these elements to be defined. At the moment, these concepts are being used rather indiscriminately; some authors may call their work an approach, while in the description of the strategy in the document it may very well be called a model, or a framework, or something else.

All these advances show that there is great interest on the part of the scientific community as regards contributing to the strengthening of this field of research. Nevertheless, there was a lack of scientific rigour as far as the validation of the different pieces of research work is concerned, since only 46.7% of the studies analysed have dealt with this aspect by means of case studies and experiments. What is more, it is seen that in this field, as in others in the software engineering field, the term *case study* is used when what is really being referred to are application experiences or applications in practice. It should be said, however, that most of the studies lack any protocol that would lead to, amongst other aspects, a systematic execution of the proposal in practice "out in the field", along with a replica, a research question, a unit of analysis, subjects of analysis, and a validation plan. Greater adherence to formal guidelines is therefore needed on the part of the scientific community when its members are publishing contributions and results about a given issue in a research area.

Another aspect that deserves attention is related to the support tools for the risk management process; as witnessed in this SLR, these are few and far between, and most of those that do exist are prototypes. This opens the way for new research to be undertaken, seeking to identify the gaps and propose new informatics solutions that incorporate other knowledge areas. These would set out to achieve automisation, and to minimise human effort in the activities in the risk management process, as well as in the decision-making process.

The analysis of the study led us also to conclude that it is certainly the case that attention is now focusing on carrying out research work that will address one or several activities in the risk management

process. It has been shown, nonetheless, that there are very few pieces of work seeking to support the management process; hardly any of them integrate international regulations which provide specific requirements that particular types of software applications should fulfil. Nor do we see that they support other management styles or styles of software development, such as agile approaches. According to Albadarneh [127], such has been the eagerness of companies to achieve excellent time-to-market results and to get their products into the market before their competitors launch theirs, that they have resorted to agile approaches to manage their projects and reduce development times. However, this is a style of work in which risk management is not tackled by means of an explicit management approach, but rather by treating risk in the same way as in traditional projects, using practices that belong to the context of agile development. In that sense, an important gap in the software risk field becomes apparent, since the agile approaches in project management and in software development have not been considered fully, or in such a way that software risk can be tackled from the point of view of an agile risk management process. That is why, though there have been important contributions made in research into software risk in other contexts, risk management in agile software development is still an area that demands greater attention. It is thus vital to search for new strategies that support risk management in agile methodologies [127]. It is also important to allow some agile methodologies to incorporate coherent risk management approaches in the practices that are widely-used in the field of risk. This would produce not only proper supervision and responsibility in risk management; it would also generate value for these methodologies [128].

As future work, it will be important to complement the results obtained in this SLR with a survey with which to discover the state of risk management in software companies. As a result, a research opportunity which will be exploited is in relation to the harmonisation of multiple models of risk management. In our opinion, this harmonisation could help to improve the risk management process, by defining a framework that would permit the integration of the best existing practices that comply with widely-recognised international standards. This framework will seek to; (i) manage all the risk involved in the different levels and contexts in the software life cycle processes in an integrated way, using an agile approach, (ii) guide the software development firms in their implementation and deployment of the risk management process, and (iii) offer mechanisms that would make it possible to monitor the process and ensure its ongoing improvement. Finally, to accomplish this ultimate goal, it will be necessary to carry out research that would enable the identification and analysis of tools and techniques capable of supporting the risk management process. These tools and techniques would have to be written up on and reported in the literature and in the software industry, so that they could be evaluated and incorporated into our framework.

Acknowledgements

Jhon Masso, Francisco Pino and César Pardo would like to thank the Universidad del Cauca, where all three work as assistant professor, full professor, and associate professor, respectively, for its contribution to this work.

Professors Félix García and Mario Piattini are grateful for funding from the BIZDEVOPS-Global (ref. RTI2018-098309-B-C31) project, financed by the Spanish Ministry of Economy, Industry and Competition (MINECO), as well as from the European Fund for Regional Development (EFRD), G3Soft (Engineering of Models for Governance and Management of Global Software Development), and GEMA (Generation and Evaluation of Models for dAta Quality), financed by the Education and Science Council of the Castilla-La Mancha Regional Government (Spain).

References

[1] S. Zardari, Software risk management, 2009 International Conference on Information Management and Engineering, ICIME 2009, Karachi, Pakistan, Department of Computer Science and IT, NED University of Engineering and Technology, 2009, pp. 375–379, , https://doi.org/10.1109/ICIME.2009.138.
[2] M.F. Rabbi, K.O.B. Mannan, A review of software risk management for selection of best tools and techniques, 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2008 in Conjunction with 2nd International Workshop on Advanced Internet Technology and Applications, AITA 2008, Blekinge Institute of Technology, 2008, pp. 773–778, , https://doi.org/10.1109/SNPD.2008.127.
[3] J. Menezes J., C. Gusmão, H. Moura, Risk factors in software development projects: a systematic literature review, Softw. Qual. J. 27 (2019) 1149–1174, https://doi.org/10.1007/s11219-018-9427-5.
[4] PMI, A guide to the project management body of knowledge (PMBOK® guide), sixth ed., Project Management Institute, Inc., Newtown Square, PA USA, 2017. https://bit.ly/2gDuS9V.
[5] AXELOS, Managing Successful Projects With PRINCE2®, sixth ed., AXELOS, 2017, https://bit.ly/1buzMiJ.
[6] ISO, ISO 31000: risk management – Guidelines, Geneva, Switzerland, 2018. https://bit.ly/3cmnZUF.
[7] ISO, ISO/IEC/IEEE 12207: Systems and Software Engineering – Software life Cycle Processes, International Organization for Standardization, International Electrotechnical Commission and Institute of Electrical and Electronics Engineers, 2017, https://bit.ly/32L62e3.
[8] ISACA., COBIT® 5 for risk, ISACA, rolling meadows, Illinois EE.UU., 2013. https://bit.ly/2ZI0M7I.
[9] ISO, ISO/IEC 16085: systems and software engineering — life cycle processes — risk management, Geneva, Switzerland, 2006. https://bit.ly/38jRJOs.
[10] B. Roy, R. Dasgupta, N. Chaki, A study on software risk management strategies and mapping with SDLC, in: R. Chaki, A. Cortesi, K. Saeed, N. Chaki (Eds.), Advanced Computing and Systems for Security. Advances in Intelligent Systems and Computing, Springer Verlag, 2016, pp. 121–138, , https://doi.org/10.1007/978-81-322-2653-6_9.
[11] B.W. Boehm, Software risk management: principles and practices, IEEE Software 8 (1991) 32–41, https://doi.org/10.1109/52.62930.
[12] D. Liu, Q. Wang, J. Xiao, The role of software process simulation modeling in software risk management: a systematic review, 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, Beijing 100190, China, Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, 2009, pp. 302–311, , https://doi.org/10.1109/ESEM.2009.5315982.
[13] R. Rabechini Junior, M.M. de Carvalho, Understanding the impact of project risk management on project performance: an empirical study, J. Technol. Manage. Innov. 8 (2013) 64–78, https://doi.org/10.4067/S0718-27242013000300006.
[14] A.H. Reed, M. Angolia, Risk management usage and impact on information systems project success, Int. J. Inf. Technol. Project Manage. 9 (2018) 1–19, https://doi.org/10.4018/IJITPM.2018040101.
[15] A. Yahya, Y. Jusoh, M.A. Jabar, N. Mohd, The critical success factors (CSFs) for it projects, journal of telecommunication, Electron. Comput. Eng. (JTEC) 9 (3–3) (2017).
[16] K. De Bakker, A. Boonstra, H. Wortmann, Risk management affecting is/it project success through communicative action, Project Manage. J. 42 (2011) 75–90, https://doi.org/10.1002/pmj.20242.
[17] R.N. Charette, Why software fails [software failure], IEEE Spectr 42 (2005) 42–49, https://doi.org/10.1109/MSPEC.2005.1502528.
[18] J. Stewart, Top 10 reasons why projects fail, Project Manage. Articles. (2018), https://goo.gl/fgNgNk (accessed November 30, 2019).
[19] PMI, Pulse of the profession 2018, Newtown Square, PA, 2018. https://bit.ly/2o92lea.
[20] M. El-Masri, S. Rivard, Towards a design theory for software project risk management systems, International Conference on Information Systems, ICIS 2012, HEC Montréal, 3000, Côte-Sainte-Catherine, Montréal (QC) H3T 2A7, Canada, 2012, pp. 2328–2338 https://bit.ly/2RpUV4g.
[21] K.B. Ratsiepe, R. Yazdanifard, Poor risk management as one of the major reasons causing failure of project management, International Conference on Management and Service Science, MASS 2011, Cyberjaya, Malaysia, Faculty of Information Communication and Technology, Limkokwing University of Creative Technology, 2011, , https://doi.org/10.1109/ICMSS.2011.5999104.
[22] S.M. Avdoshin, E.Y. Pesotskaya, Software risk management, Software Engineering Conference in Russia (CEE-SECR), 2011 7th Central and Eastern European, 2011, pp. 1–6, , https://doi.org/10.1109/CEE-SECR.2011.6188471.
[23] A.A.M. Chowdhury, S. Arefeen, Software risk management: importance and practices, International Journal of Computer and Information Technology (IJCIT), 2011, pp. 49–54 https://bit.ly/3akusxo.
[24] L.F. Sanz, P.B. Silva, Risk management in software development projects in Spain: a state of art [Gestión de riesgos en proyectos de desarrollo de software en España: estudio de la situación], Revista Facultad de Ingenieria (2014) 233–243 https://bit.ly/2YcWo0J.
[25] D. Pimchangthong, V. Boonjing, Effects of risk management practice on the success of it project, in: W.A. Halicka, K. Nazarko L (Eds.), Procedia Engineering, Elsevier Ltd, 2017, pp. 579–586, , https://doi.org/10.1016/j.proeng.2017.03.158.
[26] E. Kutsch, M. Hall, The rational choice of not applying project risk management in information technology projects, Project Manage. J. 40 (2009) 72–81, https://doi.org/10.1002/pmj.20112.
[27] E. Kutsch, M. Hall, Deliberate ignorance in project risk management, Int. J. Project Manage. 28 (2010) 245–255, https://doi.org/10.1016/j.ijproman.2009.05.003.
[28] H.A.H. Gondal, S.M.U. Din, S. Fayyaz, M.D. Zeb, B. Nadeem, Preeminent risk factor affecting software development, 2018 International Conference on

Advancements in Computational Sciences, ICACS 2018, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1–7, , https://doi.org/10.1109/ICACS.2018.8333492.
[29] K. de Bakker, A. Boonstra, H. Wortmann, Does risk management contribute to it project success? a meta-analysis of empirical evidence, Int. J. Project Manage. 28 (2010) 493–503 https://doi.org/10.1016/j.ijproman.2009.07.002.
[30] U.I. Janjua, J. Jaafar, I.B.A. Aziz, Integration of supportive processes with elementary processes for making current practices of software project risk management more effective, 2015 International Symposium on Mathematical Sciences and Computing Research (ISMSC), Institute of Electrical and Electronics Engineers Inc., 2016, pp. 292–297, , https://doi.org/10.1109/ISMSC.2015.7594068.
[31] S.Y. Chadli, A. Idri, Identifying and mitigating risks of software project management in global software development, Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, New York, NY, USA, ACM, 2017, pp. 12–22, , https://doi.org/10.1145/3143434.3143453.
[32] H. Olsson, E. Ó Conchúir, P. Ågerfalk, B. Fitzgerald, Global software development challenges: a case study on temporal, Geograph. Socio-Cultural Distance (2006), https://doi.org/10.1109/ICGSE.2006.261210.
[33] S.Y. Chadli, A. Idri, J.L. Fernández-Alemán, J.N. Ros, Frameworks for risk management in gsd projects: a survey, 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), 2015, pp. 1–6, , https://doi.org/10.1109/SITA.2015.7358381.
[34] A. Elbanna, S. Sarker, The risks of agile software development: learning from adopters, IEEE Softw. 33 (2016) 72–79, https://doi.org/10.1109/MS.2015.150.
[35] J.M. Verner, O.P. Brereton, B.A. Kitchenham, M. Turner, M. Niazi, Risks and risk mitigation in global software development: a tertiary study, Inf. Softw. Technol. 56 (2014) 54–78 https://doi.org/10.1016/j.infsof.2013.06.005.
[36] I. Nurdiani, R. Jabangwe, D. Šmite, D. Damian, Risk identification and risk mitigation instruments for global software development: systematic review and survey results, 2011 IEEE Sixth International Conference on Global Software Engineering Workshop, 2011, pp. 36–41, , https://doi.org/10.1109/ICGSE-W.2011.16.
[37] S.Y. Chadli, A. Idri, J.L. Fernández-Alemán, J.N. Ros, A. Toval, Identifying risks of software project management in global software development: an integrative framework, 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 2016, pp. 1–7, , https://doi.org/10.1109/AICCSA.2016.7945664.
[38] A. Aslam, N. Ahmad, T. Saba, A.S. Almazyad, A. Rehman, A. Anjum, A. Khan, Decision support system for risk assessment and management strategies in distributed software development, IEEE Access 5 (2017) 20349–20373, https://doi.org/10.1109/ACCESS.2017.2757605.
[39] L.L. Lobato, T.J. Bittar, P.A.D.A.M.S. Neto, I.D.O.C. Machado, E.S. De Almeida, S.R.D.E.L. Meira, Risk management in software product line engineering: a mapping study, Int. J. Softw. Eng. Knowl. Eng. 23 (2013) 523–558, https://doi.org/10.1142/S0218194013500150.
[40] D.S. Kusumo, M. Staples, L. Zhu, H. Zhang, R. Jeffery, Risks of off-the-shelf-based software acquisition and development: a systematic mapping study and a survey, 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), 2012, pp. 233–242, , https://doi.org/10.1049/ic.2012.0031.
[41] M.A. Teklemariam, E. Mnkandla, Software project risk management practice in Ethiopia, Electron. J. Inf. Syst. Dev. Countries 79 (2017) 1–14.
[42] T. Arnuphaptrairong, Software risk management practice: evidence from Thai software firms, International MultiConference of Engineers and Computer Scientists, IMECS 2014, Newswood Limited, Hong Kong, 2014.
[43] M. Kajko-Mattsson, J. Nyfjord, State of software risk management practice, IAENG Int. J. Comput. Sci. 35 (2008), https://bit.ly/2Igvkr0.
[44] A. Elzamly, B. Hussin, Quantitative and intelligent risk models in risk management for constructing software development projects: a review, Int. J. Softw. Eng. Appl. 10 (2016) 9–20, https://doi.org/10.14257/ijseia.2016.10.2.02.
[45] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, technical report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.
[46] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J. Syst. Softw. 80 (2007) 571–583, https://doi.org/10.1016/j.jss.2006.07.009.
[47] B. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software Engineering and Systematic Reviews, CRC Press, Boca Raton, 2015.
[48] H. Zhang, B. Kitchenham, D. Pfahl, Reflections on 10 years of software process simulation modeling: a systematic review bt - Making Globally distributed software development a success story, in: Q. Wang, D. Pfahl, D.M. Raffo (Eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2008: pp. 345–356.
[49] T. Dyb\ra, T. Dings\oyr, Strength of evidence in systematic reviews in software engineering, Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, New York, NY, USA, Association for Computing Machinery, 2008, pp. 178–187, , https://doi.org/10.1145/1414004.1414034.
[50] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Inf. Softw. Technol. 50 (2008) 833–859 https://doi.org/10.1016/j.infsof.2008.01.006.
[51] JCR, Journal citations report, (2018). https://bit.ly/2uNFfBb.
[52] CORE, Computing research & education, (2018). https://bit.ly/2wkeTXZ.
[53] A.K. Sangaiah, O.W. Samuel, X. Li, M. Abdel-Basset, H. Wang, Towards an efficient risk assessment in software projects-Fuzzy reinforcement paradigm, (2017). doi:10.1016/j.compeleceng.2017.07.022.
[54] J. Li, M. Li, D. Wu, Q. Dai, H. Song, A bayesian networks-based risk identification approach for software process risk: the context of chinese trustworthy software, Int. J. Inf. Technol. Decis. Mak. 15 (2016) 1391–1412, https://doi.org/10.1142/S0219622016500401.
[55] A.D. Kartika, K. Surendro, A fuzzy-based methodology to assess software usability risk, Information and Communication Technology (ICoICT), 2016 4th International Conference On, Institute of Electrical and Electronics Engineers Inc., 2016, pp. 1–5, , https://doi.org/10.1109/ICoICT.2016.7571930.
[56] J.J.P. Sipayung, J. Sembiring, Risk assessment model of application development using Bayesian network and Boehm's software risk principles, 2015 International Conference on Information Technology Systems and Innovation (ICITSI), Institute of Electrical and Electronics Engineers Inc., 2016, pp. 1–5, , https://doi.org/10.1109/ICITSI.2015.7437722.
[57] C. Lindholm, Involving user perspective in a software risk management process, J. Softw.: Evol. Process 27 (2015) 953–975, https://doi.org/10.1002/smr.1753.
[58] S.Y. Lee, Y. Li, DRS: a developer risk metric for better predicting software fault-proneness, 2015 S International Conference on Trustworthy Systems and Their Applications, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 120–127, , https://doi.org/10.1109/TSA.2015.27.
[59] C.P. Chang, Software risk modeling by clustering project metrics, Int. J. Softw. Eng. Knowl. Eng. 25 (2015) 1053–1076, https://doi.org/10.1142/S0218194015500175.
[60] C. Kumar, D.K. Yadav, A probabilistic software risk assessment and estimation model for software projects, Procedia Computer Science, Elsevier, 2015, pp. 353–361, https://doi.org/10.1016/j.procs.2015.06.041.
[61] C. Jeon, N. Kim, H.P. In, Probabilistic approach to predicting risk in software projects using software repository data, Int. J. Softw. Eng. Knowl. Eng. 25 (2015) 1017–1032, https://doi.org/10.1142/S0218194015500151.
[62] M.V. Goyal, S.M. Satapathy, S.K. Rath, Software project risk assessment based on cost drivers and neuro-fuzzy technique, International Conference on Computing, Communication & Automation, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 823–827, , https://doi.org/10.1109/CCAA.2015.7148487.
[63] S. Patil, R. Ade, A software project risk analysis tool using software development goal modeling approach, in: J.K. Mandal, S.C. Satapathy, M. Kumar Sanyal, P.P. Sarkar, A. Mukhopadhyay (Eds.), Information Systems Design and Intelligent Applications, Advances in Intelligent Systems and Computing 340, Springer, India, New Delhi, 2015, pp. 767–777, , https://doi.org/10.1007/978-81-322-2247-7_78.
[64] S. Islam, H. Mouratidis, E.R. Weippl, An empirical study on the implementation and evaluation of a goal-driven software development risk management model, Inf. Softw. Technol. 56 (2014) 117–133, https://doi.org/10.1016/J.INFSOF.2013.06.003.
[65] A. Elzamly, B. Hussin, An enhancement of framework software risk management methodology for successful software development, J. Theor. Appl. Inf. Technol. 62 (2014) 410–423.
[66] J. Liu, J. Qiao, A grey-based rough set approach for software risk prediction: a case study, 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, IEEE Computer Society, 2014, pp. 1147–1151, , https://doi.org/10.1109/HPCC.and.EUC.2013.162.
[67] L. Bai, F. Li, The model of project risk assessment based on BP neural network algorithm, Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, Institute of Electrical and Electronics Engineers Inc., 2014, pp. 326–329, , https://doi.org/10.1109/ICCSNT.2013.6967122.
[68] J.H. Iversen, L. Mathiassen, P.A. Nielsen, Managing risk in software process improvement: an action research approach, MIS Quarterly: Manage. Inf. Syst. 28 (2004) 395–434 https://goo.gl/poEwZM.
[69] E.E. Odzaly, D. Greer, D. Stewart, Agile risk management using software agents, J. Ambient Intell. Humaniz Comput. 9 (2018) 823–841, https://doi.org/10.1007/s12652-017-0488-2.
[70] C. Haisjackl, M. Felderer, R. Breu, RisCal - A risk estimation tool for software engineering purposes, 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, 2013, pp. 292–299, , https://doi.org/10.1109/SEAA.2013.10.
[71] Y. Hu, X. Zhang, E.W.T. Ngai, R. Cai, M. Liu, Software project risk analysis using Bayesian networks with causality constraints, Decis. Support Syst. 56 (2013) 439–449, https://doi.org/10.1016/j.dss.2012.11.001.
[72] S. Laqrichi, D. Gourc, F. Marmier, Toward an effort estimation model for software projects integrating risk, 22nd International Conference on Production Research, Iguassu Falls, International Foundation for Production Research (IFPR), 2013.
[73] A. Jaiswal, M. Sharma, Expert webest tool: a web based application, estimate the cost and risk of software project using function points, in: N. Meghanathan, D. Nagamalai, N. Chaki (Eds.), Advances in Computing and Information Technology. Advances in Intelligent Systems and Computing, Springer Verlag, 2013, pp. 77–86, , https://doi.org/10.1007/978-3-642-31552-7_9.
[74] Y. Wang, S. Fu, T. Zhang, Ranking software risks based on historical data, in: D. Jin, S. Lin (Eds.), Advances in Computer Science and Information Engineering. Advances in Intelligent and Soft Computing, 2012, pp. 393–398, , https://doi.org/10.1007/978-3-642-30223-7_61.
[75] M. Uzzafer, A novel risk assessment model for software projects, Computer and Management (CAMAN), 2011 International Conference On, 2011, pp. 1–5, , https://doi.org/10.1109/CAMAN.2011.5778729.
[76] T. Bragina, G. Tabunshchyk, Fuzzy model for the software projects design risk analysis, 2011 11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), 2011, pp. 335–341.
[77] F. Reyes, N. Cerpa, A. Candia-Véjar, M. Bardeen, The optimization of success probability for software projects using genetic algorithms, J. Syst. Softw. 84 (2011) 775–785, https://doi.org/10.1016/j.jss.2010.12.036.
[78] D. Wu, H. Song, M. Li, C. Cai, J. Li, Modeling risk factors dependence using Copula method for assessing software schedule risk, 2nd International Conference on Software Engineering and Data Mining, SEDM 2010, Beijing, 100190, China, Institute of Policy and Management, Chinese Academy of Sciences, 2010, pp. 571–574 https://goo.gl/8so1H3.
[79] M. Sadiq, M.W. Ahmad, M.K.I. Rahmani, S. Jung, Software risk assessment and evaluation process (SRAEP) using model based approach, 2010 International

Conference on Networking and Information Technology, ICNIT 2010, India, Computer Engineering Laboratory, University M. Tech. Scholars, 2010, pp. 171–177, , https://doi.org/10.1109/ICNIT.2010.5508535.
[80] M. Sadiq, A. Rahman, S. Ahmad, M. Asim, J. Ahmad, EsrcTool: a tool to estimate the software risk and cost, 2nd International Conference on Computer Research and Development, ICCRD 2010, New Delhi-25, India, University Polytechnic, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), 2010, pp. 886–890, , https://doi.org/10.1109/ICCRD.2010.29.
[81] M. Uzzafer, A financial tool for software risk measurement, 2010 International Conference in Information Science and Applications, ICISA 2010, Kuala Lumpur Campus, Malaysia, Department of Computer Science, University of Nottingham, 2010, , https://doi.org/10.1109/ICISA.2010.5480537.
[82] A. Hosseingholizadeh, A. Abhari, A new compound metric for software risk assessment, in: R. Lee, O. Ormandjieva, A. Abran, C. Constantinides (Eds.), Software Engineering Research, Management and Applications 2010. Studies in Computational Intelligence, Springer, Berlin, Heidelberg, 2010, pp. 115–131, , https://doi.org/10.1007/978-3-642-13273-5_8.
[83] L. Minglu, L. Jianping, S. Hao, W. Dengsheng, Risk management in the trustworthy software process: a novel risk and trustworthiness measurement model framework, NCM 2009 - 5th International Joint Conference on Int. Conf. on Networked Computing, Int. Conf. on Advanced Information Management and Service, and Int. Conf. on Digital Content, Multimedia Technology and Its Applications, Beijing 100190, China, Institute of Policy and Management, Chinese Academy of Sciences, 2009, pp. 214–219, , https://doi.org/10.1109/NCM.2009.283.
[84] P. Cao, F. Chen, A risk control optimization model for software project, 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009, Fuzhou, China, College of Public Administration, Fuzhou University, 2009, , https://doi.org/10.1109/CISE.2009.5362886.
[85] J. Gao, M. Shah, M. Shah, D. Vyas, P. Pattabhiraman, K. Dandapani, E. Bari, Systematic risk assessment and cost estimation for software problems, 21st International Conference on Software Engineering and Knowledge Engineering, SEKE 2009, San Jose State University, United States, 2009, pp. 103–109 https://goo.gl/Co7vA7.
[86] W.E. Wong, Y. Qi, BP neural network-based effective fault localization, Int. J. Softw. Eng. Knowl. Eng. 19 (2009) 573–597, https://doi.org/10.1142/S021819400900426X.
[87] F. Mc Caffery, J. Burton, I. Richardson, Risk management capability model for the development of medical device software, Softw. Qual. J. 18 (2009) 81–107, https://doi.org/10.1007/s11219-009-9086-7.
[88] D. Gupta, M. Sadiq, Software risk assessment and estimation model, International Conference on Computer Science and Information Technology, ICCSIT 2008, Bawana Road, Delhi-110042, India, Department of Computer Engineering, Delhi College of Engineering, 2008, pp. 963–967, , https://doi.org/10.1109/ICCSIT.2008.184.
[89] R. Hewett, A. Thipse, Building business considerations into enterprise application designs, 19th International Conference on Software Engineering and Knowledge Engineering, SEKE 2007, United States, Department of Computer Science, Texas Tech University, 2007, pp. 513–518 https://bit.ly/2x5ntds.
[90] Y. Takagi, O. Mizuno, T. Kikuno, An empirical approach to characterizing risky software projects based on logistic regression analysis, Empir. Softw. Eng. 10 (2005) 495–515, https://doi.org/10.1007/s10664-005-3864-z.
[91] X. Liu, G. Kane, M. Bambroo, An intelligent early warning system for software quality improvement and project management, Proceedings: 15th IEEE International Conference on Tools with Artificial Intelligence, United States, Dept. of Comp. Sci., Univ. of Missouri-Rolla, 2003, pp. 32–38 https://bit.ly/2TytdE8.
[92] X. Ruzhi, L. Qian, X. Jing, CMM-based software risk control optimization, in: S. W.W., M. A.M. (Eds.), IEEE International Conference on Information Reuse and Integration, IRI 2003, Fudan University, 220 Handan Road, Shanghai, China, Institute of Electrical and Electronics Engineers Inc., Department of Computing and Information Technology, 2003, pp. 499–503, , https://doi.org/10.1109/IRI.2003.1251457.
[93] D.E. Neumann, An enhanced neural network technique for software risk analysis, IEEE Trans. Softw. Eng. 28 (2002) 904–912, https://doi.org/10.1109/TSE.2002.1033229.
[94] S.M. Yacoub, H.H. Ammar, T. Robinson, Methodology for architectural-level risk assessment using dynamic metrics, 11th International Symposium on Software Reliability Engineering (ISSRE 2000), West Virginia Univ, Morgantown, United States, IEEE, 2000, pp. 210–221 https://bit.ly/2KtoTUF.
[95] D.X. Houston, G.T. Mackulak, J.S. Collofello, Stochastic simulation of risk factor potential effects for software development risk management, J. Syst. Softw. 59 (2001) 247–257, https://doi.org/10.1016/S0164-1212(01)00066-8.
[96] D. Gotterbarn, Enhancing risk analysis using software development impact statements, 26th Annual NASA Goddard Software Engineering Workshop, IEEE/NASA SEW 2001, Johnson City, TN, United States, Institute of Electrical and Electronics Engineers Inc., Software Eng. Ethics Res. Inst., East Tennessee State Univ., 2001, pp. 43–51, , https://doi.org/10.1109/SEW.2001.992654.
[97] A.A. Keshlaf, K. Hashim, A model and prototype tool to manage software risks, in: C. T.Y., T. T.H. (Eds.), 1st Asia-Pacific Conference on Quality Software, APAQS 2000, P.O.Box 3633, Tripoli, Libyan Arab Jamahiriya, Institute of Electrical and Electronics Engineers Inc., Software Engineering Research Laboratory, Information Department, Industrial Research Center, 2000, pp. 297–305, , https://doi.org/10.1109/APAQ.2000.883803.
[98] N.J. Van Eck, L. Waltman, Software survey: vOSviewer, a computer program for bibliometric mapping, Scientometrics 84 (2010) 523–538, https://doi.org/10.1007/s11192-009-0146-3.
[99] N.J. Van Eck, L. Waltman, Y. Ding, R. Rousseau, D. Wolfram (Eds.), Springer International Publishing, Cham, 2014, pp. 285–320, , https://doi.org/10.1007/978-3-319-10377-8_13.
[100] SEI, Capability maturity model for software - CMMI for development v 1.2. technical report CMU/SEI-2006-TR-008, Pittsburg, Pennsylvania, USA, 2006. https://bit.ly/38i2G3l.
[101] C. Lindholm, J.P. Notander, M. Höst, A case study on software risk analysis and planning in medical device development, Softw. Qual. J. 22 (2014) 469–497, https://doi.org/10.1007/s11219-013-9222-2.
[102] S.G. Vilbergsdóttir, E.T. Hvannberg, E.L.-C. Law, Classification of usability problems (CUP) scheme: augmentation and exploitation, Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles, New York, NY, USA, ACM, 2006, pp. 281–290, , https://doi.org/10.1145/1182475.1182505.
[103] A. Van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications, 1st ed., Wiley Publishing, 2009.
[104] S. Islam, Software development risk management model - A goal driven approach, ESEC/FSE Doctoral Symposium'09 - Proceedings of the Doctoral Symposium for ESEC/FSE, Amsterdam, 2009, pp. 5–8, , https://doi.org/10.1145/1595782.1595785.
[105] ISO, ANSI/AAMI/ISO:14971, 2007, medical devices—Application of risk management to medical devices, 2007. https://bit.ly/2VEZXhJ.
[106] FDA's, Mission statement, 2007. https://bit.ly/2VCuFI2.
[107] FDA/CDRH, Code of federal regulations 21 CFR part 820, 2006. https://bit.ly/2PHUwKU.
[108] FDA/CDRH, Guidance for the content of premarket submissions for software contained in medical devices, 11 de Mayo, 2005.
[109] FDA/CDRH, Guidance for off-the-shelf software use in medical devices, 9 de Septiembre, 1999.
[110] FDA/CDRH, General principles of software validation; final guidance for industry and FDA staff, 11 de Enero, 2002.
[111] FDA/CDRH, Guidance for industry and FDA premarket and design control reviewers—medical device use-safety: incorporating human factors engineering into risk management, 18 de Julio, 2000.
[112] ISPE, GAMP guide for validation of automated systems. gamp 4, Diciembre, 2001.
[113] IEC, AAMI/IEC 62304:2006, medical device software - Software Life cycle processes, 19 de Julio, 2006. https://bit.ly/3ckv6gh.
[114] BS/EN, BS en 60601-1-4:2000, medical electrical equipment, part 1. general requirements for safety, 2000. https://bit.ly/39iGbwp.
[115] A. Keshlaf, K. Hashim, Practical system to evaluate and manage the risks in software, The Proceedings of the Third Arab Congress of Electronics, Telecommunications and Computing Arabelectronics'2000, Tunis, 2000, pp. 18–24.
[116] ISO, ISO/IEC 31010: risk management - Risk assessment techniques, Geneva, Switzerland, 2009.
[117] P. Sonchan, S. Ramingwong, Top twenty risks in software projects: a content analysis and Delphi study, 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014, pp. 1–6, , https://doi.org/10.1109/ECTICon.2014.6839820.
[118] ISO, ISO/IEC 12207: Systems and Software Engineering – Software life Cycle Processes, International Organization for Standardization and International Electrotechnical Commission, 2008.
[119] M. Carr, S. Konda, I. Monarch, C. Walker, F.C. Ulrich, Taxonomy-Based risk identification. CMU/SEI-93-TR-006., Pittsburgh, Pennsylvania, 1993. https://goo.gl/uR1FGv.
[120] L. Wallace, M. Keil, A. Rai, How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory model*, Decis. Sci. 35 (2004) 289–321, https://doi.org/10.1111/j.00117315.2004.02059.x.
[121] P.L. Bannerman, C. Schwindt, J. Zimmermann (Eds.), Springer International Publishing, Cham, 2015, pp. 1119–1134, , https://doi.org/10.1007/978-3-319-05915-0_20.
[122] A. Yamami, S. Ahriz, K. Mansouri, M. Qbadou, E. Illoussamen, Representing it projects risk management best practices as a metamodel, engineering, Technol. Appl. Sci. Res. 7 (2017) 2062.
[123] P. Rehacek, Risk management standards for project management, Int. J. Adv. Appl. Sci. 4 (2017) 1–13, https://doi.org/10.21833/ijaas.2017.06.001.
[124] A.J.G. Silvius, Integrating sustainability into project risk management, Global Business Expansion: Concepts, Methodologies, Tools, and Appl. 2018, pp. 330–352, , https://doi.org/10.4018/978-1-5225-5481-3.ch017.
[125] L.K. Meulbroek, A senior manager's guide to integrated risk management, J. Appl. Corp. Finance 14 (2002) 56–70, https://doi.org/10.1111/j.1745-6622.2002.tb00449.x.
[126] D. Hillson, Integrated risk management as a framework for organisational success, PMI® Global Congress 2006, North America, Seattle, WA. Newtown Square, PA, Project Management Institute, 2006shorturl.at/hkFLZ.
[127] A. Albadarneh, I. Albadarneh, A. Qusef, Risk management in agile software development: a comparative study, 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Institute of Electrical and Electronics Engineers Inc., 2015, , https://doi.org/10.1109/AEECT.2015.7360573.
[128] Moran Alan, Risk management in agile projects, Project Manage.: Methodol. Assoc. Risk 2 (2016) 1–4 https://goo.gl/YiLnNo.