The Journal of Systems and Software 125 (2017) 68–92

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Integration between requirements engineering and safety analysis: A systematic literature review

Jéssyka Vilela[a,*], Jaelson Castro[a], Luiz Eduardo G. Martins[b], Tony Gorschek[c]

[a] Centro de Informática, Universidade Federal de Pernambuco, Recife-PE, Brazil
[b] Departamento de Ciência e Tecnologia, Universidade Federal de São Paulo, São José dos Campos, Brazil
[c] Blekinge Institute of Technology (BTH), Sweden

ARTICLE INFO

Article history:
Received 27 June 2016
Revised 18 October 2016
Accepted 21 November 2016
Available online 22 November 2016

Keywords:
Safety-critical systems
Requirements engineering
Safety analysis
Integration
Communication
Systematic literature review

ABSTRACT

Context: Safety-Critical Systems (SCS) require more sophisticated requirements engineering (RE) approaches as inadequate, incomplete or misunderstood requirements have been recognized as a major cause in many accidents and safety-related catastrophes. **Objective:** In order to cope with the complexity of specifying SCS by RE, we investigate the approaches proposed to improve the communication or integration between RE and safety engineering in SCS development. We analyze the activities that should be performed by RE during safety analysis, the hazard/safety techniques it could use, the relationships between safety information that it should specify, the tools to support safety analysis as well as integration benefits between these areas. **Method:** We use a Systematic Literature Review (SLR) as the basis for our work. **Results:** We developed four taxonomies to help RE during specification of SCS that classify: techniques used in (1) hazard analysis; (2) safety analysis; (3) safety-related information and (4) a detailed set of information regarding hazards specification. **Conclusions:** This paper is a step towards developing a body of knowledge in safety concerns necessary to RE in the specification of SCS that is derived from a large-scale SLR. We believe the results will benefit both researchers and practitioners.

1. Introduction

Safety-critical systems are those systems composed by a set of hardware, software, process, data and people (Du et al., 2014) whose failure could result in accidents that cause damage to the environment, financial losses, injury to people and even the loss of lives (Leveson, 2011; Hatcliff et al., 2014).

The use of software in SCS, in particular in control systems, has increased to such an extent that failures in the software can impair system safety (Saeed et al., 1995). Investigations into the causes of accidents indicate that more rigor is required in setting the requirements and specification of safety-related systems (Simpson and Stoker, 2002) since inadequate or misunderstood requirements (Ratan et al., 1996) have been recognized as the major cause (not coding or implementation Guillerm et al., 2010) of a significant proportion of accidents (Simpson and Stoker, 2002) and safety-related catastrophes (Leveson, 2002).

Therefore, safety cannot be assured without its already being part of the RE process and the system design (Leveson, 2011; Wilikens et al., 1997). The system requirements definition corresponds to the activity that focus on what the system should do and it may contain elements of design (Ivarsson and Gorschek, 2011). It is most efficient to consider safety concerns as early as possible during software development in order to ensure that safety problems do not propagate through subsequent phases of development (Saeed et al., 1995; Wilikens et al., 1997).

According to Leveson (2011), separating safety engineering from the RE process is almost guaranteed to make the effort and resources expended a poor investment and possibly the development of a system that is not safe (Leveson, 2011). Safety engineering is effective when it participates in and provides input to the RE process and to the system design, not when it focuses on making arguments about the artifacts created after the major safety-related decisions have been made.

In the context of SCS development, a tighter integration of safety engineering concerns into the RE process is desired by academia and industry (Leveson, 2011; Lutz, 2000; Martins and Gorschek, 2016; Heimdahl, 2007; Sikora et al., 2012; Hatcliff et al., 2014). A systematic integration of safety engineering (Navarro et al., 2006; El Ariss et al., 2011) with the requirements engineering and a common nomenclature (Zoughbi et al., 2011) are required in order to satisfy system correctness and safety require-

* Corresponding author.
E-mail addresses: jffv@cin.ufpe.br (J. Vilela), jbc@cin.ufpe.br (J. Castro), legmartins@unifesp.br (L.E.G. Martins), tony.gorschek@bth.se (T. Gorschek).

http://dx.doi.org/10.1016/j.jss.2016.11.031
0164-1212/

ments (Thramboulidis and Scholz, 2010) as well as to provide a framework for effective cooperation between experts (Scholz and Thramboulidis, 2013; Zoughbi et al., 2011).

Catastrophic safety incidents continue to occur despite advances in the understanding of the underlying causes over the years (Mannan et al., 2015). Hence, in order to cope with the complexity of performing hazard/safety analysis and specifying SCS, we are investigating, through a SLR, the approaches proposed to improve the requirements communication and integration between RE and Safety Engineering in the development of SCS.

In this paper, we analyze the activities that should be performed by requirements engineering during safety analysis, the hazard/safety techniques they could use, the relationships between safety information that they should specify, the tools that can be used to support safety analysis as well as the benefits of the integration between RE and safety engineering.

The results of this paper are a step towards developing a body of knowledge in safety concerns necessary to be handled to requirements engineers in the specification of SCS that were derived from a large-scale rigorous literature review. As part of our work, we developed four taxonomies to help the requirements engineers in the specification of SCS that classify (1) the techniques used in the hazard analysis; (2) the techniques used in the safety analysis; (3) the safety-related information and (4) a detailed set of information regarding the specification of hazards.

This paper is organized as follows. Section 2 presents background and related work. The research methodology adopted to conduct the SLR is presented in Section 3. The results and the analysis related to our research questions are presented in Section 4. Our conclusions are presented in Section 5.

2. Background and related work

Safety-critical systems are those systems whose failure could result in harm (generally meaning injury or death) (Hatcliff et al., 2014). Loss may involve human death and injury, but it may also involve other major losses, including mission, equipment, financial, and information losses (Leveson, 2011).

During the development of SCS, safety engineers typically review the requirements documents in early development stages in order to perform safety analysis. Such reviews are periodically repeated throughout the entire development process in order to align the safety analysis with requirements changes. As a major result of the safety analysis, safety engineers define many concepts that will be discussed in Section 4.6 such as, for example, hazard, accident, safety requirement, functional safety requirement.

When developing SCS, the requirements engineering activities and process are critical to avoid the introduction of defects and misunderstandings among engineers and developers (Leveson, 2011; Firesmith, 2004; Pernstål et al., 2015). An elaborated requirements engineering (RE) approach is crucial in order to meet time, cost, and quality goals in safety-critical systems development (Sikora et al., 2012; Hatcliff et al., 2014). Therefore, the integration between RE and safety engineering is desired by academia and industry (Leveson, 2011; Lutz, 2000; Martins and Gorschek, 2016; Heimdahl, 2007; Sikora et al., 2012; Hatcliff et al., 2014).

However, requirements engineers, traditionally, are not well familiar with system safety analysis processes which are performed by safety engineers. One reason is the gap that exists among the traditional development processes, methodologies, notations and tools used in safety engineering (Scholz and Thramboulidis, 2013). Furthermore, according to Levesson (Leveson, 1995), for practical reasons, training a software engineer in system safety may be more successful than training a safety engineer in software engineering. This work is a step in this direction to reduce the gap between RE and safety engineering.

2.1. Related work

Catastrophic accidents continue to occur despite advances in the understanding of the underlying causes over the years (Mannan et al., 2015). This may constitute an evidence that the traditional development process of SCS must be changed in order to consider the safety concerns early in the RE process. Therefore, the system safety analysis is the first phase to identify software safety requirements necessary to support the development of the software requirements specification (Medikonda and Panchumarthy, 2009).

In this context, the integration between RE and safety engineering is well desired by academia and industry. We noticed a tendency in RE and safety research in trying to join these two areas by different research lines.

From the point of view of RE, the model-driven paradigm has been used to some works to reduce the gap between these two areas by the use of shared models that includes safety characteristics (Leveson, 2002; Murali et al., 2015) or system engineering best practices (Guillerm et al., 2010). In this context, UML (Beckers et al., 2013; Briones et al., 2007; Zoughbi et al., 2011) or SysML (Scholz and Thramboulidis, 2013; Biggs et al., 2016) profiles have been proposed.

From the safety engineering, safety analysis tools (Ratan et al., 1996), methodologies (Kelly and Weaver, 2004) and metamodels (OMG, 2016; Panesar-Walawege et al., 2010; de la Vara and Panesar-Walawege, 2013; de la Vara et al., 2016) for the construction of assurance safety cases have been developed. However, many of these studies are partially or not compliant with safety standards and it is unclear whether these approaches can cope with the complexity of the large-scale development of software-intensive systems, taking inter-departmental and multi-disciplinary aspects into account (Pernstål et al., 2015). Hence, further research is necessary to investigate the extent that these approaches integrate these engineers as well as their compliance with safety standards.

Another research line in the safety area is improving the construction of assurance safety cases. They should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context (Kelly and Weaver, 2004). Hence, such cases are developed aiming to build arguments regarding the safety and security properties of systems (OMG, 2016).

Safety arguments are most typically communicated in existing safety cases through free text. The GSN (Goal Structuring Notation) is a graphical argumentation technique (Kelly and Weaver, 2004) proposed to improve the structure, rigor, and clarity of safety arguments. In this notation, the elements (goal, solution, strategy, context, and undeveloped goal) are linked together in a network described as a “goal structure”. Such technique can be used to assist the maintenance, construction, reuse and assessment of safety cases.

The Structured Assurance Case Metamodel (SACM) defines a metamodel for representing structured assurance cases. It is a combination of Argumentation Metamodel (ARM) and Software Assurance Evidence Metamodel (SAEM) documents. The metamodels present concepts for communicating the way in which evidence artefacts are collected by various participants using elements that identify the main elements that determine the evidence collection process: techniques, resources, activities, artefacts, and participants.

There are also the works of Panesar-Walawege et al. (2010); de la Vara and Panesar-Walawege (2013) as well as de la Vara et al. (2016) that propose safety taxonomies from an analysis of safety standards. Panesar-Walawege et al. (2010) present a conceptual model to characterize the evidence for arguing about software

safety. Their model captures both the information requirements for demonstrating compliance with IEC 61508 and the traceability links necessary to create a seamless chain of evidence.

The SafetyMet model of de la Vara and Panesar-Walawege (2013) distinguishes among safety compliance metamodels and safety compliance models as well as between safety standard-related information and project-specific information. In another work, de la Vara et al. (2016) describe a metamodel that supports the specification of safety compliance needs for most critical computer-based and software-intensive systems. Their metamodels describe the actions required to a system to be compliant to safety standards; however, they do not describe in details the content of safety requirements specifications as we do in our work.

The above works are focused on the construction of safety cases. Although they are related to our paper, we are more concerned with defining the content of safety requirements specifications since requirements engineers do not have all the required knowledge regarding the specification of safety-critical systems, and the terms and concepts of SCS have been misunderstood and used inconsistently.

Hence, the GSN technique, the SACM metamodel and the works (Panesar-Walawege et al., 2010; de la Vara and Panesar-Walawege, 2013) as well as de la Vara et al. (2016) are concerned with collecting and relating documents that demonstrate that the safety analysis was conducted properly and/or safety standards were followed. One example of such documents is the safety requirements specification whose content we describe in our work (see Section 4.6). Understanding the concepts that should be specified by requirements engineers in safety requirements specifications, as we want to close this gap in this paper, contributes to better elaborating assurance cases.

Few systematic literature reviews (SLR) about the development of safety-critical systems are found in the literature as for example the works of Martins and Gorschek (2016); Gadelha Queiroz and Vaccare Braga (2014); Nair et al. (2014), and Vilela et al. (Under Submission).

The work of Martins and Gorschek (2016) focuses on requirements engineering for safety-critical systems. Their work investigated which approaches have been proposed to elicit, model, specify and validate safety requirements in the context of safety-critical systems, as well as to what extent such approaches have been validated in industrial settings. Moreover, they analyzed the usability and usefulness of the reported approaches, and to what extent they enabled requirements communication among the development project/team actors in the safety-critical systems domain. However, their work does not investigate deeply the integration of requirements and safety engineers. In fact, in this work, we restricted our search string to capture the results related to communication and integration. Moreover, we performed an analysis of the activities that should be performed by requirements engineers during safety analysis, the hazard/safety techniques they could use, the tools that can be used to support safety analysis as mostly the relationships between safety information that requirements engineers should specify through taxonomies not contemplated in Martins and Gorschek (2016).

A systematic literature review was conducted by Gadelha Queiroz and Vaccare Braga (2014) aiming to identify approaches, methods and methodologies whose goal was the combination of product line engineering and model-driven engineering for the development of safety-critical embedded systems. They also analyzed the existence of empirical studies that demonstrate the application of these techniques in this type of development.

Nair et al. (2014) synthesised the existing knowledge in the academic literature about safety evidence through a SLR concentrating on three facets: the information that constitutes evidence; structuring of evidence; and evidence assessment. Their work as well as the GSN technique, the SACM metamodel and the works (Panesar-Walawege et al., 2010; de la Vara and Panesar-Walawege, 2013; de la Vara et al., 2016) are concerned with demonstrating the evidence of the system safety by means of constructing assurance safety cases and not the integration of RE and safety engineering neither the content of safety requirements specifications.

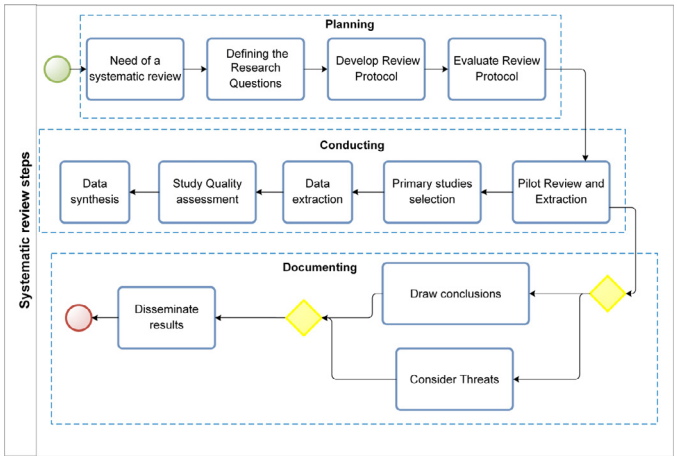

Fig. 1. Systematic review steps adapted from Martins and Gorschek (2016) and Kitchenham and Charters (2007).

In Vilela et al. (Under Submission), we addressed the challenges and needs involved in the communication in the requirements engineering process among engineers when developing safety-critical systems, the application context, the research type, the evaluation methods, the type of contribution, the domain, the requirements activity, the languages and tools used to specify the requirements. Furthermore, we also analysed the information described in the requirements specification, the stakeholders involved, the communication format, and the compliance with safety standards. In this work, we investigate the integration between RE and safety engineering by analyzing the activities that should be performed by requirements engineering during safety analysis, the hazard/safety techniques they could use, the relationships between safety information that they should specify, the tools that can be used to support safety analysis as well as the benefits of the integration between RE and safety engineering.

The above works are complementary since they explore different aspects in the development of safety-critical systems; nevertheless, they do not perform an extensive identification and mapping of the state-of-the-art on the integration between RE and Safety Engineering aiming to decrease the gap between these two areas.

Moreover, the many challenges associated with requirements engineering as well as with requirements communication in safety-critical systems development reported in many studies (Leveson, 2011; Lutz, 2000; Martins and Gorschek, 2016; Heimdahl, 2007; Sikora et al., 2012; Hatcliff et al., 2014) are still valid today.

3. Research methodology

We adopted the empirical method through a systematic literature review to conduct our research. The methodology to conduct the SLR (Fig. 1) was based on the guidelines and the systematic review protocol template proposed by Kitchenham and Charters (2007). According to these guidelines, the SLR process includes several activities, which can be grouped in three main phases: planning of the SLR, conducting the SLR and reporting the SLR.

The SLR was motivated (Step 1, Fig. 1) by the tighter integration of safety engineering concerns into the RE process de-

Table 1
Research questions and motivations.

Research Question	Description and Motivation
RQ1. What are the approaches proposed to improve the integration and communication between RE and safety engineering in the requirements engineering process of safety-critical systems?	The purpose of this question is to identify and analyze the approaches proposed to improve the integration and communication between RE and safety engineering.
RQ1.1. What are the activities can be performed by requirements engineers as a part of safety analysis in the approaches that integrate requirements and safety engineering?	This question intends to detect which activities (actions, tasks) are proposed by approaches that integrate requirements and safety engineering to be conducted requirements engineers during the safety analysis.
RQ1.2. What are the techniques can be used by requirements engineers during safety analysis in the approaches that integrate requirements and safety engineering?	This question aims to identify the techniques (systematic procedures, methods, formulas, routines by which a task is accomplished) can be used by requirements engineers in the approaches that integrate requirements and safety engineering for performing the safety analysis of the systems. This information will be used to develop two taxonomies to classify the techniques used in hazard/safety analysis.
RQ1.3. What data/information artifacts can be created by requirements engineers in the analysis and specification of SCS in the approaches that integrate requirements and safety engineering?	The aim of this question is to identify the various pieces of safety-related information (data, concepts, knowledge, facts) can be created by requirements engineers in the approaches that integrate requirements and safety engineering to document the safety concerns during the specification of SCS. The data/information obtained in this research question are used to develop two taxonomies regarding safety requirements classification.
RQ1.4. What are the tools used by the approaches that integrate requirements and safety engineering in safety analysis?	This question maps the Computer-Aided Software Engineering (CASE) tools used in the approaches that integrate requirements and safety engineering in the analysis of the safety requirements specifications of safety-critical systems.
RQ1.5. What are the benefits of the approaches that integrate requirements and safety engineering identified in RQ1?	The purpose of this question is to analyze the benefits of the approaches (selected in RQ1) for integration and communication between RE and safety engineering extracted from the selected studies.
RQ2. What challenges/problems are identified in research literature relating to SCS and RE?	This question aims to identify works needed in this area.

sired by academia and industry as reported in many studies (Lutz, 2000; Martins and Gorschek, 2016; Leveson, 2011; Heimdahl, 2007; Sikora et al., 2012; Hatcliff et al., 2014). The gap that exists between the traditional development processes, methodologies, notations and tools and the ones used in safety engineering also contributes to the need of this SLR.

In order to determine if a SLR about integration and communication between RE and safety engineering had already been performed, we searched the ACM, Springer, IEEE and Google Scholar digital libraries (performed in September, 2015). None of the retrieved studies were directly related to the objectives expressed in the research questions (Step 2, Defining Research Questions). In fact, few systematic literature reviews about the development of safety-critical systems are found in the literature.

3.1. Research questions

This systematic review's purpose is to analyze the approaches proposed to improve the integration and communication between RE and safety engineering as well as to understand which information regarding safety requirements should be specified by requirements engineers in order to reduce the gap between these two areas. Thus, we intend to answer the research questions described in Table 1.

3.2. Search strategy

The search strategy included an automatic search, using a string validated by experts on the RE and safety-critical areas and a manual inclusion of papers well-known about requirements communication.

The development of our review protocol (Step 3, Fig. 1) followed the PICOC (Population, Intervention, Comparison, Outcome, Context) criteria as suggested by Kitchenham and Charters (2007) as well as Petticrew and Roberts (2008):

- ***Population***: peer-reviewed publications reporting approaches to improve the integration and communication between RE and safety engineering;
- the aim of the ***intervention*** was to collect empirical evidence in relation to approaches proposed to improve the integration and communication among requirements and safety engineers during the development of SCS.
- ***Comparison***: it does not apply.
- ***Outcomes***: activities that should be performed by requirements engineering during safety analysis, the hazard/safety techniques they could use, the relationships between safety information that they should specify, the tools that can be used to support safety analysis as well as the benefits of the integration between RE and safety engineering. The activities, techniques and safety information should be performed, used or specified in the RE process. These activities and techniques have better results by RE and safety engineering working jointly.
- ***Context***: any context in which engineers in the RE process or safety analysis create or modify the specifications of safety-critical systems.

Moreover, the ***selected resources*** chosen were Science Direct, SpringerLink, ACM Digital Library, IEEE Xplore, Scopus, and Compendex; and the ***search method*** consisted of web search in digital libraries.

Our search string was specified considering the main terms of the phenomena under investigation (safety-critical systems, requirements engineering, safety requirements, and integration/communication).

We conducted pilot searches to refine the search string in an iterative way. We excluded keywords whose inclusion did not return additional papers in the automatic searches. After several iterations, we defined the following search string used to search within keywords, title, abstract and full text of the publications:

(1) *("safety critical system" OR "safety critical systems" OR "safety-critical system" OR "safety-critical systems") AND*
(2) *("requirements engineering" OR "requirements engineer" OR "requirements team" OR "requirements specification") AND*
(3) *("safety requirements" OR "safety engineering" OR "safety engineer" OR "safety team" OR "safety analysis" OR "safety specification") AND*

Table 2
Inclusion/exclusion criteria.

#	Inclusion Criterion
1	Primary studies
2	Studies that address in the objectives the integration and communication between RE and safety engineering
3	Study published in any year until September 2015
4	Studies that relate Requirements and Safety
5	Studies that relate Design and Safety
#	**Exclusion Criterion**
1	Secondary studies
2	Short-papers ($\leq$ 3 pages)
3	Duplicated studies (only one copy of each study was included)
4	Non English written papers
5	Studies clearly irrelevant to the research, taking into account the research questions
6	Gray literature
7	Redundant paper of same authorship
8	Publications whose text was not available (through search engines or by contacting the authors)
9	Studies whose focus was not the integration and communication between RE and safety engineering or safety requirements specification (they addresses specific issues of safety-critical systems such as safety/hazard analysis, risk assessment/management, safety assurance or evidence, dependability/reliability, security, RE activities, traceability, software product lines, safety standards, design/architecture, human computer interaction concerns or human factors or operator behavior, robots development, and agile development)

(4) *("communication" OR "integration" OR "interaction" OR "collaboration" OR "alignment" OR "understanding" OR "relationship" OR "share" OR "sharing" OR "combination" OR "interrelation" OR "interplay" OR "interdependency")*

Keywords related to safety-critical systems are presented in the first group of terms. The second one concerns to the requirements engineering, and the third group to the specification of safety requirements. Finally, the last group of terms are related to integration and communication. In this paper, we want to collect information that should be shared by RE and safety engineering. Therefore, such information should be collected and specified early in the development process by requirements engineers to reduce the causes of accidents related to inadequate, incomplete or misunderstood requirements. If we had focused on words in the safety area, we would have extracted information that is beyond the scope and competence of requirements engineers, that is, more focused on tasks and competences of design engineers and safety engineers for example. Furthermore, we adapted the string for each search engine to consider their peculiarities.

We adopted the StArt (State of the Art through Systematic Reviews) tool (LAPES, 2014) to support the protocol definition and SLR conduction due to the positive results in the execution of SLRs reported in Hernandes et al. (2012).

3.3. Inclusion and exclusion criteria

The summarized inclusion and exclusion criteria are presented in Table 2.

We were interested only in primary studies, published in any year until September 2015, that present some contribution on the requirements communication of safety-critical systems or relate requirements and safety or relate design and safety.

Our protocol was validated (Step 4, Fig. 1) by professionals of requirements engineering and safety-critical systems areas.

3.4. Procedure for studies selection

Our procedure for studies selection, presented in Fig. 2, consisted in four main steps. Besides the input and output of each step, this figure also shows two frames containing the exclusion criteria which were exclusively applied to the studies in Steps 3 and 4.

In Step 1, the studies were obtained from electronic databases using the search string. Springer returned 411 titles, IEEE Xplore 151, Science Direct 111, Scopus 159, Engineering Village (Compendex) 9 and ACM 193 search results. The search results (1034) were downloaded and were entered into and organized with the aid of StArt tool. Moreover, we included 3 papers manually.

Out of 1037 search results, 761 were unique (step 2, Fig. 2). Afterwards, reading the title and the abstract of the papers, we excluded 591 studies, based on the 16 exclusion criteria (step 3), as indicated on the left side of Fig. 2. If there was insufficient data, the paper was left for the next step. After finishing the Step 3, 170 papers remained in the selection process.

After reading and analyzing 170 papers left for the full-text reading (step 4), we obtained 57 relevant papers. In this step, the papers were excluded according to the same 16 exclusion criteria (-111 papers) considered in the previous step, as also indicated in the right frame of Fig. 2.

We excluded many studies from this SLR that address mainly hazard analysis, safety assurance, security or dependability. Hence, we selected papers that only propose the integration and communication between RE and safety engineering.

3.5. Data extraction and synthesis

We prepared digital forms to accurately record any information needed to answer the research questions. We extracted the data described in Table 3 from each of the 57 primary studies included in this systematic review. As well as the selection process, the data extraction was fully aided by the StArt tool.

During the synthesis phase, we normalized the terms describing the same phenomenon and we continued to use the most common term. We built four taxonomies using these terms: (1) the techniques that can be used by requirements engineers in the hazard analysis; (2) the techniques that can be used by requirements engineers in the safety analysis; (3) the relationships between safety-related information that requirements engineers should specify; (4) a detailed set of information regarding the specification of hazards by requirements engineers.

3.6. Quality assessment

The quality assessment is critical in a SLR to investigate whether quality differences provide an explanation for differences in study results (Kitchenham and Charters, 2007).

Following the guidelines of Kitchenham and Charters (2007), we considered that quality relates to the extent to which the study minimises bias and maximises internal and external validity.

The quality assessment (QA) of selected studies in both parts of our SLR was achieved by a scoring technique to evaluate the credibility, completeness and relevance of the selected studies. All papers were evaluated against a set of 20 quality criteria. The assessment instrument used is presented in Table 4.

Our primary studies were of different types, hence in order to assess their quality, we classified the 57 studies into five different categories - Evaluation Research Papers (EVA), Validation Research Papers (VAL), Solution Proposal Papers (SOL), Experience Papers (EXP), and Opinion Papers (OP). Then, we used a set of quality assessment questions for each category (see Table 4) as suggested by Kitchenham and Charters (2007); Tiwari and Gupta (2015), as well as Wieringa et al. (2006).

Each quality assessment question is judged against three possible answers: "Yes" (score = 1), "Partially" (score = 0.5) or "No" (score = 0). Consequently, the quality score for a particular study

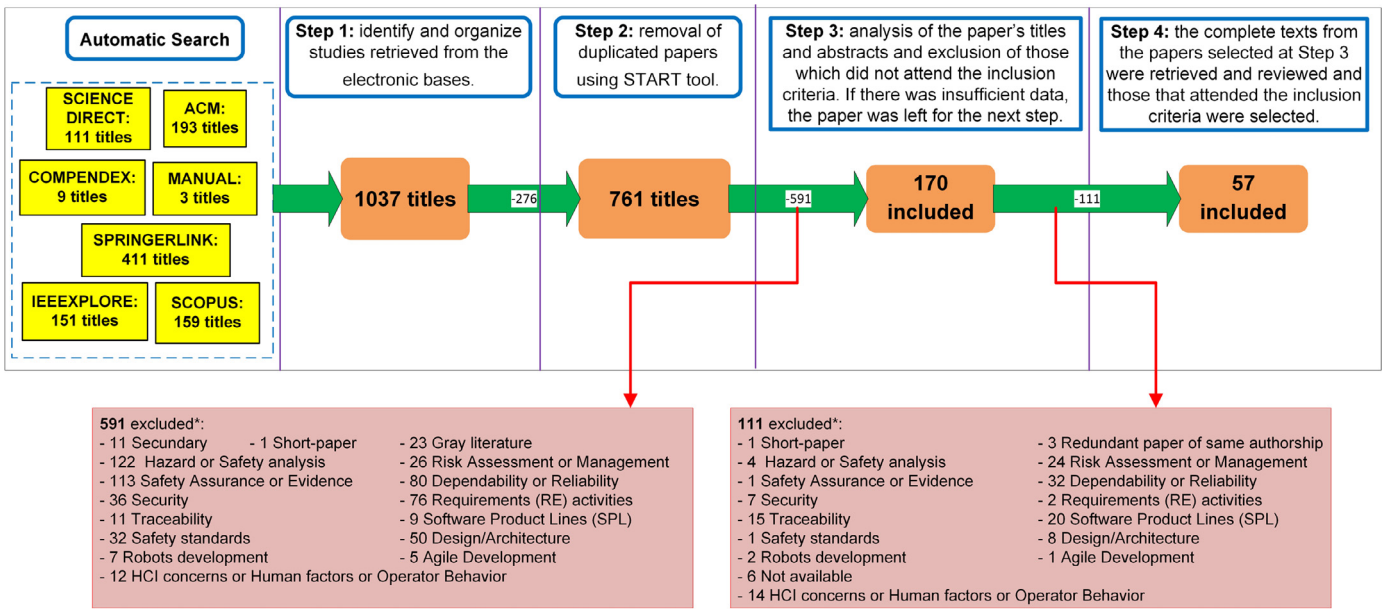

Fig. 2. Paper selection flowchart.

Table 3
Extraction form.

#	*Study Data*	*Description*	Relevant RQ
1	Study identifier	Unique id for the study	Study overview
2	Authors, Year, Title, Country		Study overview
3	Article source	ACM, Springer, IEEE, Science Direct, Scopus, El Compendex	Study overview
4	Type of article	Journal, conference, symposium, workshop, book chapter	Study overview
5	Application context	Industrial, academic, both	Study overview
6	Research Type (based on Wieringa et al., 2006)	Validation research, evaluation research, solution proposal, philosophical papers, experience papers	Study overview
7	Evaluation method (based on Easterbrook et al., 2008)	Controlled experiment, case study, survey, ethnography, action research, illustrative scenario, not applicable	Study overview
8	Safety Activities	What are the activities can be performed by requirements engineers as a part of safety analysis in the approaches that integrate requirements and safety engineering?	RQ1.1
9	Safety Techniques	What are the techniques can be used by requirements engineers in safety analysis in the approaches that integrate requirements and safety engineering?	RQ1.2
10	Safety Information	What data/information artifacts should be created by requirements engineers in the analysis and specification of SCS in the approaches that integrate requirements and safety engineering?	RQ1.3
11	Safety Tools	What are the tools used by the approaches that integrate requirements and safety engineering in safety analysis?	RQ1.4
12	Benefits	What are the benefits of the approaches that integrate requirements and safety engineering identified in RQ1?	RQ1.5
13	Challenges/Problems issues	What challenges/problems are identified in research literature relating to SCS and RE?	RQ2

is computed by taking the sum of the scores of the answers to the questions related to its research type. The quality scores of the selected studies are presented in Table A.11 (see Appendix).

3.7. Threats to validity

We used the categorization of threats presented by Wohlin et al. (2000), which includes four types of validity threats, namely, conclusion, internal, construct, and external validity threats.

Construct validity: Construct validity is related to generalization of the result to the concept or theory behind the study execution (Wohlin et al., 2000). Aiming to minimize threats of this nature, we used many synonyms for the main constructs in this review: "safety-critical systems", "requirements engineering", "safety requirements", and "communication". During the synthesis phase, we normalized the terms describing the same phenomenon and we continued to use the most common term (Gasparic and Janes, 2016). We built four taxonomies using these terms: (1) the techniques that can be used by requirements engineers in the hazard analysis; (2) the techniques that can be used by requirements engineers in the safety analysis; (3) the relationships between safety-related information that requirements engineers should specify; (4) a detailed set of information regarding the specification of hazards by requirements engineers.

Internal validity threats are related to possible wrong conclusion about causal relationships between treatment and outcome (Wohlin et al., 2000). The primary objective of conducting a SLR is to minimize internal validity threats in the research. We tried to mitigate threats due to personal bias on study understanding by conducting the selection process iteratively. Moreover, the first author is a PhD student in Requirements Engineering, and the other three authors are experienced researchers with expertise in Requirements Engineering, Software Engineering or Safety-Critical Systems.

External validity is concerned with establishing the generalizability of the SLR results, which is related to the degree to which the primary studies are representative for the review topic. In the case of a literature review, the external validity depends on the

Table 4
Study quality assessment criteria.

Questions	Eva	Val	Sol	Exp	Op
Q1. Is there a clear statement of the goals of the research (Dermeval et al., 2016)?	x	x	x	x	
Q2. Is the proposed technique clearly described (Dermeval et al., 2016)?			x		
Q3. Is there an adequate description of the context (industry, laboratory setting, products used and so on) in which the research was carried out (Dermeval et al., 2016)?	x	x			
Q4. Were treatments randomly allocated (Kitchenham and Charters, 2007)?	x				
Q5. Is the sample representative of the population to which the results will generalise (Kitchenham and Charters, 2007)?	x	x			
Q6. Was there any control group present with which the treatments can be compared, if applicable (Tiwari and Gupta, 2015)?	x				
Q7. If there is a control group, are participants similar to the treatment group participants in terms of variables that may affect study outcomes (Kitchenham and Charters, 2007)?	x				
Q8. Was the data analysis sufficiently rigorous (Tiwari and Gupta, 2015)?	x	x			
Q9. Is there a discussion about the results of the study (Dermeval et al., 2016)?	x	x	x		
Q10. Are the limitations of this study explicitly discussed (Dermeval et al., 2016)?	x	x	x		
Q11. Are the lessons learned interesting (Tiwari and Gupta, 2015)?				x	
Q12. Is the article relevant for practitioners (Tiwari and Gupta, 2015)?	x	x	x	x	
Q13. Is there sufficient discussion of related work (Tiwari and Gupta, 2015)? (Are competing techniques discussed and compared with the present technique?)	x	x	x		
Q14. Are the study participants or observational units adequately described (Kitchenham and Charters, 2007)? For example, Software Engineering experience, type (student, practitioner, consultant), nationality, task experience and other relevant variables.	x	x			
Q15. What evidence is there of attention to ethical issues (Kitchenham and Charters, 2007)?	x	x			
Q16. Is the study significantly increase the knowledge about integration and communication between RE and safety engineering research (Tiwari and Gupta, 2015)?	x	x	x	x	
Q17. Is the stated position sound (Wieringa et al., 2006)?					x
Q18. Is it likely to provoke discussion (Wieringa et al., 2006)?					x
Q19. How well has diversity of perspective and context been explored (Kitchenham and Charters, 2007)?					x
Q20. How clear are the assumptions/theoretical perspectives/values that have shaped the form and opinions described (Kitchenham and Charters, 2007)?					x

identified literature: if the identified literature is not externally valid, neither is the synthesis of its content (Gasparic and Janes, 2016). By the choice of our exclusion criteria, we excluded gray literature papers. In order to mitigate external threats, our search process was defined after several trial searches and validated with the consensus of the authors.

Conclusion validity: The used methodology of Kitchenham and Charters (2007) already assumes that not all relevant primary studies that exist can be identified. To mitigate this threat, the research protocol was carefully designed and validated by the authors to minimize the risk of exclusion of relevant studies. Besides, we used many synonyms for the constructs of this paper to improve high coverage of possibly important studies from automatic search. Furthermore, we did not conduct a complementary manual search since the main venues about requirements engineering and safety-critical systems are indexed by the search engines adopted in our protocol. Therefore, we only added 3 well-known studies (Pernstål et al., 2015; Fricker et al., 2010; 2008) on requirements communication that were not captured by the search string. It is worth highlighting that we did not restrict the time period of published studies for this SLR aiming to obtain the maximum coverage possible. As mentioned in Section 2, for the best of our knowledge, this is the first SLR with a specific focus on integration and communication between RE and safety engineering.

4. Results and analysis

This section describes the results of our study; we discuss the answers of each research question separately. Our selection process resulted in 57 studies that met the inclusion criteria and we extracted the data following the extraction form described in Section 3.5. Before presenting the results and analysis for each research question, we depict the quality assessment results and give an overview of the general characteristics of the studies.

4.1. Quality assessment results

The quality assessment helped to increase the reliability of the conclusions obtained in this work and in ascertaining the credibility and coherent synthesis of results (Dermeval et al., 2016).

We present the results of the quality assessment of the included studies in Table A.11 (see Appendix) according to the assessment questions described in Table 4. These 20 criteria provided a measure of the extent to which we could be confident that a particular selected study could make a valuable contribution to our review. The overall quality of the selected studies is reasonable since the mean of quality was 82.37%.

4.2. Overview of the studies

The selected studies were published between 1994 and 2015. In Fig. 3, we present the number of studies by year of publication. We can notice an increasing number of publications in the context of this review from 2007.

After analyzing the temporal view of the studies, we can conclude that the number of studies about integration and communication between RE and safety engineering is little through the years. Although the apparent increasing of the number of studies on this topic from 2007, this result corroborates with the statement that the integration of safety analysis and requirements engineering has been somewhat neglected (Broomfield and Chung, 1997).

We categorized the application context of the studies as *industrial, academic* or *both*. The studies that make explicit that they were performed in a real company or some authors works in the industry (we use their affiliations to obtain these information) we classified them as *industrial*. On the other hand, we classified the studies in *both* category, the studies conducted in/by industries or some authors are affiliated to the academia.

The results show that 27 studies (47%) belong to the *academic* context. 10 studies (17.54%) were conducted in *industrial* settings and 20 studies (35.09%) belong to the *both* category. These results

Table 5
Research types of the selected studies.

Research Type	Studies	Count	%
Solution Proposal	(Kaiser et al., 2010; Saeed et al., 1995; David et al., 2010; Mostert and von Solms, 1994; Lutz, 1993; Ratan et al., 1996; Thramboulidis and Scholz, 2010; Black and Koopman, 2008; Navarro et al., 2006; Kim and Chung, 2005; Mannering et al., 2008; Medikonda and Panchumarthy, 2009; Wu and Kelly, 2007; Nejati et al., 2012; Martin-Guillerez et al., 2010; Leveson, 2002; Hansen et al., 1998; Scholz and Thramboulidis, 2013; Markovski and van de Mortel-Fronczak, 2012; Beckers et al., 2013; Arogundade et al., 2012; El Ariss et al., 2011; Guiochet et al., 2010; Chandrasekaran et al., 2009; Briones et al., 2007; Broomfield and Chung, 1997; Górski and Wardziński, 1996; Du et al., 2014; Zoughbi et al., 2011; Jrjens, 2003; Simpson and Stoker, 2002; Biggs et al., 2016; Lu and Halang, 2007; Mustafiz and Kienzle, 2009; Ekberg et al., 2014; Guillerm et al., 2010; Rafeh, 2013; Chen et al., 2011; Tschrtz and Schedl, 2010; Elliott et al., 1995; Croll et al., 1997; Cant et al., 2006; Murali et al., 2015; Pernstål et al., 2015; Fricker et al., 2010; 2008)	46	85.19%
Evaluation Research	(Galvao Martins and De Oliveira, 2014; Stålhane and Sindre, 2014; 2007; Mustafiz and Kienzle, 2009; Paige et al., 2008; Jurkiewicz et al., 2015; Stålhane et al., 2010)	7	12.96%
Validation Research	(Nejati et al., 2012; Hatcliff et al., 2014; Pernstål et al., 2015; Fricker et al., 2010)	4	7.41%
Opinion Papers	(Heimdahl, 2007; Sikora et al., 2012)	2	3.7%
Experience Papers	(Wilikens et al., 1997; Schedl and Winkelbauer, 2008)	2	3.7%

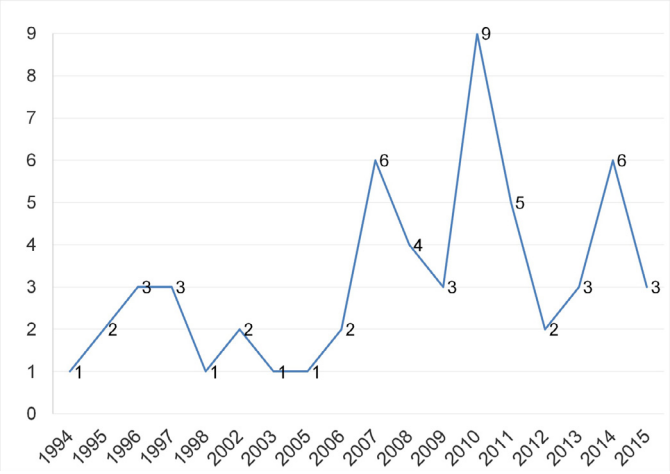

Fig. 3. Temporal view of the studies.

shows that more than half of the studies were classified as the industrial context. This may indicate that the studies are attempting to solve challenges faced by the industries. Besides, it also suggests that there is some approximation between industries and universities.

The selected studies were classified according to the applied research types defined by Wieringa et al. (2006), as can be seen in Table 5.

The most adopted research type is *Solution Proposal* with 85.19% (46 studies) followed by *Evaluation Research* with 12.96% (7 studies), *Validation Research* with 7.41% (4 studies), and *Experience Papers* and *Opinion Papers* tied with 3.7% (2 studies) each. None of the selected studies belongs to *Philosophical Papers* category of research types.

We propose our classification of the evaluation method based on the categories (*controlled experiment, case study, survey, ethnography* and *action research*) defined by Easterbrook et al. (2008). In addition, we adopted two extra categories used in a previous work (Dermeval et al., 2016): *illustrative scenario* and *not applicable*. The first category is used to classify papers that only evaluate their contributions using small examples. The second extra category refers to the papers that do not contain any kind of evaluation method in the study.

Despite more than half of the selected studies were classified in the *industry context*, 84.21% of the studies were not evaluated empirically. 39 studies (68.42%) were evaluated only using small examples and 9 studies (15.79%) did not mention any kind of evaluation method or it does not apply since it is an opinion paper. Only 22.81% were evaluated empirically where 5 studies (8.77%) presented a controlled experiment and 8 studies (14.04%) adopted the case study strategy.

We noticed that there are few real experiments even though there is some empirical studies published on integration and communication between RE and safety engineering. In addition, we find that a lot of what is labelled as case study is really a proof of concept discussions related to simple examples. Hence, we classified them in the illustrative scenario category.

These findings reveal the need of applying the approaches in practice with real users in order to assess to what extent they contribute to integration and communication between RE and safety engineering. However, there are many difficulties faced when conducting controlled software engineering experiments in realistic environments that we are aware of. The absence of professionals as subjects in (software engineering) experiments is directly related to the high costs and large organizational effort spent in the conduction of such experiments as recognized by many authors, such as Basili et al. (1986); Fenton (1993) as well as Sjoberg et al. (2002).

The distribution of the selected studies according to the countries of the authors' affiliation was also analyzed. The studies belong to many countries being *United Kingdom, United States of America (USA), France, Germany, Norway, Sweden*, and *China* the most common.

North America, Europe, and Asia continents have well established companies in the development of safety-critical systems such as in the avionics and automotive domains. This may be a reason for the high number of selected papers in above countries since the difficulties of having access of real data and professionals as well as conducting realistic studies decrease. Other countries have few contributions on integration and communication between RE and safety engineering such as *Australia, Austria, Brazil, Canada, Denmark, Greece, India, Iran, Italy, Japan, Korea, Nigeria, Poland, South Africa, Spain, Switzerland*, and *The Netherlands*.

In the next sections, we present and discuss the results of each research question.

4.3. RQ1: what are the approaches proposed to improve the integration and communication between RE and safety engineering in the requirements engineering process of safety-critical systems?

The purpose of this question is to identify and analyze the approaches proposed to improve integration and communication between RE and safety engineering. This research question was divided into five sub research questions (RQ1.1 to RQ1.5) aiming to analyze many aspects of the topic. In each one of these research questions, we provided a detailed discussion about our results. Our

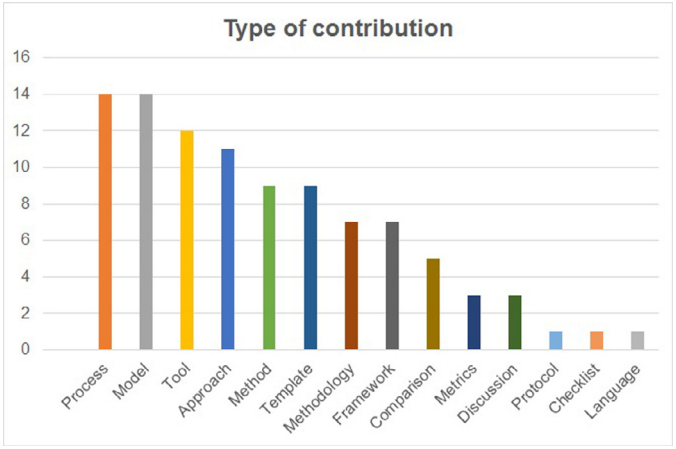

Fig. 4. Types of contributions on integration and communication between RE and safety engineering.

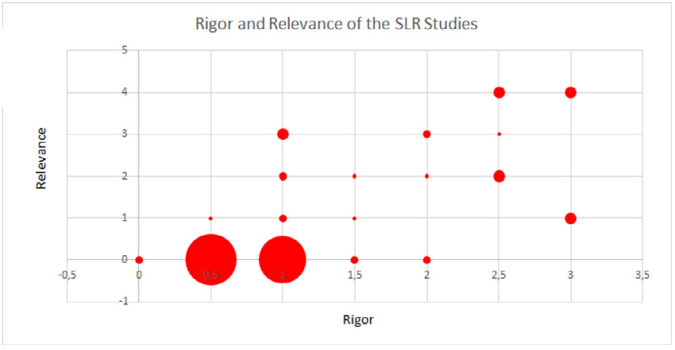

Fig. 5. Rigor and relevance of the approaches.

SLR returned 57 studies presented in Table A.11 whose results will be discussed in Sections 4.4–4.8.

We analyzed the types of existing contributions based on the classification presented in the work of Petersen et al. (2008) that includes the Approach, Framework, Method, Tool, Process, Model, Methodology, Template, Comparison, Metrics, Protocol, Checklist, Language, and Discussion categories (see Fig. 4).

Note that this analysis allows a study to be included in more than one category. The predominant contributions that we identified were *Process* and *Model*, followed by *Tool, Approach, Method*, and *Template*. These types of contributions show that there is a tendency of using common models among requirements, design and safety teams. This contribution was adopted by Wu and Kelly (2007); Nejati et al. (2012); Markovski and van de Mortel-Fronczak (2012); Arogundade et al. (2012); El Ariss et al. (2011); Stålhane and Sindre (2007); Ekberg et al. (2014); Chen et al. (2011); Murali et al. (2015) to improve integration and communication between RE and safety engineering. UML profiles were proposed by the works of Beckers et al. (2013); Zoughbi et al. (2011) as well as Lu and Halang (2007). The SysML language has been used by the works of Scholz and Thramboulidis (2013) as well as Biggs et al. (2016) as an approach to integrate safety engineering with an SysML-based development process.

An approach for safety management which can be used in different phases of software development before implementation and disposal phase is described in Rafeh (2013). In the proposed approach, safety begins from requirements as the infrastructure of design and continues through other phases of software production.

The shortcomings of the existing safety analysis techniques in software safety analysis were investigated by the works of Galvao Martins and De Oliveira (2014) as well as Du et al. (2014). The former described a case study adopting a protocol to help requirements engineers to derive safety functional requirements from Fault Tree Analysis. The latter proposed a safety requirement elicitation technique combined with scenario to refine the system-level safety analysis into software behaviors in specific scenarios.

The variety of requirements specification languages motivated some works such as Stålhane and Sindre (2014) as well as Jurkiewicz et al. (2015) to perform experiments to identify which are the ones that best support non-experts in identifying hazards of safety-critical systems.

Finally, the work of Pernstål et al. (2015) presented a lightweight RE framework, demonstrated and evaluated its industrial applicability in response to the needs of a Swedish automotive company for improving specific problems in inter-departmental requirements coordination and communication in large-scale development of software-intensive systems.

We performed a rigor and relevance analysis of the approaches following the model proposed by Ivarsson and Gorschek (2011) for evaluating the rigor and industrial relevance of technology evaluations in software engineering.

The rigor is not the actual rigor of studies, e.g. use of a correct analysis method, that is considered in the model, but rather the extent to which aspects related to rigor are presented (Ivarsson and Gorschek, 2011). According to the model, the rigor is evaluated through three aspects: *Context described, Study design described*, and *Validity discussed*. All these aspects are scored with the same three score levels in a three point scale: 0 (weak), 0.5 (medium), and 1 (strong) description.

The relevance is evaluated by analyzing four aspects: *Subjects* that participated in the studies; (2) the *Context* in which the studies were performed; (3) the *Research Method* adopted in the studies; and (4) the *Scale* used in the studies evaluation. If the aspect contributes to industrial relevance it receives the score 1, otherwise, it receives 0. Therefore, the maximum value for rigor an approach can have is three, while relevance has a maximum of four (Ivarsson and Gorschek, 2011).

The results of rigor and relevance analysis of the approaches described in the selected studies are depicted in Fig. 5. The figure shows that the majority of evaluations end up in the lower left quadrant of the bubble chart, indicating a lack of both rigor and relevance. Only 2 approaches have zero rigor and relevance, on the other hand, 13 studies (22.81%) have 0.5 rigor and 0 relevance. Moreover, 17 studies (29.82%) have 1 rigor and 0 relevance. This means that more than half of the studies (52.63%) of all approaches included in this review did not provide a description, or provided a weak one, of the context, study design or validity threats.

31 studies (54.39%) have 0 relevance. This means that they are examples of application of a proposal done by either students or researchers in academia in toy examples. This combination of low rigor and relevance is disappointing from a technology transfer perspective, as these evaluations have less potential for actually influencing practice (Ivarsson and Gorschek, 2011).

12 studies (21.05%) have the highest rigor (2.5 or 3) and relevance (3 or 4), but only three studies have the maximum rigor and relevance. These well classified studies proposed different types of contributions: framework, method, tool, process, model, template, comparison with other approaches, protocol or discussed challenges in the integration and communication between RE and safety engineering. All these papers were published in the last ten years (2006–2016) and classified as industrial or both (academic and industrial) context. Only 2 papers were written and published in the academic context. These results means that the year of publication might indicate that the quality of the studies changed over time and the relevance is affected by the affiliation of the re-

Table 6
Average number of rigor and relevance per type of contribution.

	Rigor				Relevance				
Type	C	SD	V	Sum - Rigor	CO	RM	U	S	Sum - Relevance
Approach	0.36	0.23	0.50	1.09	0.18	0.09	0.09	0.00	0.36
Framework	0.64	0.36	0.64	1.64	0.29	0.29	0.29	0.29	1.14
Method	0.44	0.33	0.50	1.28	0.33	0.33	0.22	0.22	1.11
Tool	0.42	0.25	0.54	1.21	0.33	0.17	0.17	0.17	0.83
Process	0.61	0.39	0.50	1.50	0.50	0.29	0.50	0.43	1.71
Model	0.43	0.32	0.57	1.32	0.36	0.14	0.21	0.14	0.86
Methodology	0.36	0.07	0.57	1	0.14	0	0.14	0	0.29
Template	0.56	0.22	0.56	1.33	0.33	0.11	0.22	0.22	0.89
Comparison	1	1	0.90	2.90	1	1	0.40	0.40	2.80
Metrics	0.33	0	0.33	0.67	0.33	0	0.33	0.33	1
Protocol	1	1	0.5	2.5	1	1	1	1	4
Checklist	0.5	0	0.5	1	0	0	0	0	0
Language	0	0	0.5	0.5	0	0	0	0	0
Discussion	0.50	0.33	0.50	1.33	0.33	0.33	0.33	0.33	1.33

searchers and the context in which the study is conducted. Furthermore, they also show that the levels of validation are increasing.

We present in Table 6 the average of each aspect regarding rigor (Context described (C), Study design described (SD), and Validity discussed (V)) and relevance (Context (CO), Research method (RM), User/Subject (U), and Scale (S)) per type of contribution. We also present the average of the sum of the aspects.

The results of Table 6 show that the studies that proposed *Metrics* and *Language* have the smallest number of rigor meaning that they are poor described. These papers discussed some limitations but they did not present the study design or research methodology adopted.

The types of contribution that have the worst relevance are *Approach, Methodology, Checklist* and *Language*. The scale of all papers in these categories do not contribute to relevance. This means that the evaluation is performed using applications of unrealistic size (down-scaled industrial) or toy examples.

4.4. RQ1.1: what are the activities can be performed by requirements engineers as a part of safety analysis in the approaches that integrate requirements and safety engineering?

This question intends to detect which activities can be conducted by requirements engineers during the safety analysis. In Table 7, we list the safety activities proposed by the selected studies that should have be conducted by RE and safety engineering working jointly to get better results during the development of safety-critical systems.

4.4.1. RQ1.1: analysis and discussion

The life cycle appropriate for SCS and embedded systems is very different than one appropriate for developing office software (Leveson, 2002). Accordingly, some activities should be performed aiming to ensure that the system is safe.

The results confirmed that there is no unified vocabulary among the approaches as well as they are not in compliance with safety standards. This lack of unified terminology hampers exchanging information between stakeholders contributing to a poor requirements analysis and specifications.

Several of these activities have the same purpose, for example *Safety analysis, Assessing Safety, Safety verification*, and *Safety Assessment*. Risk analysis is another activity with many variations: *Risk assessment, Risk identification, Risk evaluation*, and *Risk management*.

Many approaches do not explicitly mention which activities should be undertaken by requirements team or they generalize all activities as *Safety analysis*. Therefore, safety analysis is the activity reported in 37 studies (64.91%) as shown in Table 7.

Table 7
Activities that should be performed in safety analysis.

Safety Activity	Count	%
Safety analysis	31	54.39%
Assessing Safety	2	3.51%
Safety verification	2	3.51%
Safety Assessment	2	3.51%
Hazard analysis	24	42.11%
Hazard Identification	6	10.53%
Risk analysis	9	15.79%
Risk assessment	5	8.77%
Risk identification	2	3.51%
Risk evaluation	1	1.75%
Risk management	1	1.75%
Dependability analysis	3	5.26%
Safety requirements specification	3	5.26%
It does not cite	3	5.26%
Reliability analysis	2	3.51%
Simulation	2	3.51%
Deviation analysis	2	3.51%
Verification of the completeness of requirements criteria	2	3.51%
Safety case generation	2	3.51%
Cause-consequence analysis	1	1.75%
Vulnerability analysis	1	1.75%
Robustness analysis	1	1.75%
Mode Confusion Analysis	1	1.75%
Human Error Analysis	1	1.75%
Timing and other analysis	1	1.75%
Operational Analysis	1	1.75%
Performance Monitoring	1	1.75%
Periodic Audits	1	1.75%
Incident and accident analysis	1	1.75%
Change Analysis	1	1.75%
Definition of System Level Requirements	1	1.75%
Definition of Safety Measures	1	1.75%
Definition of 1st Level System Architecture	1	1.75%
Refinement of Architecture	1	1.75%
System use modeling & task analysis	1	1.75%
Common cause, common mode and zonal analysis	1	1.75%

The second activity most referenced by the studies is the *Hazard analysis* that also has a variation *Hazard Identification*. This activity, cited in 30 studies (52.63%), consists in examining the system specification to identify potentially dangerous situations that may lead to an accident. When these situations are found, they should be adequately handled by specifying safety requirements (appropriate ways to eliminate or control the hazards).

Risk analysis is another activity well cited in the approaches (18 studies - 31.58%). This analysis comprises the evaluation of the risks, the likelihood of an injury or illness occurring, and the severity associated with the hazards.

Table 8
Techniques that should be used in the safety analysis by RE and safety teams.

Technique	Class.	Count	%
Fault Tree Analysis (FTA)	D	18	31.58%
Preliminary Hazard Analysis (PHA)	I	18	31.58%
It does not cite		15	26.32%
HAZOPS (Hazard and Operability Studies)	Both	9	15.79%
Risk analysis (RA)	G	8	14.04%
Code hazard analysis (CoHA)	I	8	14.04%
System Hazard Analysis (SHA)	I	6	10.53%
Preliminary System Safety Assessment (PSSA)	I	6	10.53%
Deductive safety technique	D	5	8.77%
Failure Modes and Effects Analysis (FMEA)	I	5	8.77%
Misuse case (MUC)	G	5	8.77%
Guide-words	Both	5	8.77%
System safety analysis (SSA)	I	5	8.77%
Functional Hazard Analysis (FuHA)	I	4	7.02%
Inductive safety technique	I	4	7.02%
Scenario-based analysis	G	3	5.26%
Cause-consequence analysis (Cause-ConA)	Both	3	5.26%
Failure Modes Effects and Criticality Analysis (FMECA)	I	2	3.51%
Forward simulation (ForSim)	I	2	3.51%
Mind storms and historical information	G	2	3.51%
Interface analysis and human error analysis	G	2	3.51%
Deviation Analysis (DevA)	I	2	3.51%
Preliminary controller task analysis (PTA)	G	1	1.75%
Software Hazard Analysis (SwHA)	I	1	1.75%
Safety Requirements/Criteria Analysis (SRCA)	G	1	1.75%
Requirement Risk Assessment (RRAM)	D	1	1.75%
Risk Modes and Effect Analysis (RMEA)	I	1	1.75%
Event Tree Analysis (ETA)	I	1	1.75%
Indirect Control Path Analysis (ICPA)	G	1	1.75%
Preliminary Safety Analysis (PSA)	I	1	1.75%
Software safety design analysis (SSDA)	I	1	1.75%

Many activities were cited in the selected studies and we noticed that there is no consensus about which activities are essential in the development of a safety-critical system. The definition of the activities depends on the domain, culture and size of the company as well as the knowledge and experience of the requirements and safety engineers. Moreover, few studies consider the guidelines and restrictions imposed by the safety standards.

4.5. RQ1.2: what are the techniques can be used by requirements engineers during safety analysis in the approaches that integrate requirements and safety engineering?

The safety analysis in the development of safety-critical systems requires the use of a technique to help finding the hazards of the system. Therefore, this question aims to identify the techniques can be used by requirements engineers in the approaches for performing the hazard and safety analysis of the systems. These techniques are listed in the selected studies as having better results if RE and safety engineering work jointly. Accordingly, we did not investigate in this paper the techniques for safety analysis that are used only by the safety engineering team.

The techniques obtained from the extracted data, the distribution of selected studies over the techniques including their count (i.e., the number of selected studies from each source), and the percentage of selected studies are presented in Table 8.

These techniques should be used in safety analysis to discover the hazards of a system, their causes and consequences. Such techniques can be classified according to the forms of logic and reasoning during the hazard analysis in Deductive (D), Inductive (I) or Both. The Table 8 also shows the General (G) category in which the techniques do not have the aim of discover hazards but another aspect of SCS.

Besides the classification according to the type of hazard analysis, the techniques can be classified following the analytical styles that have two perspectives: qualitative and quantitative analysis.

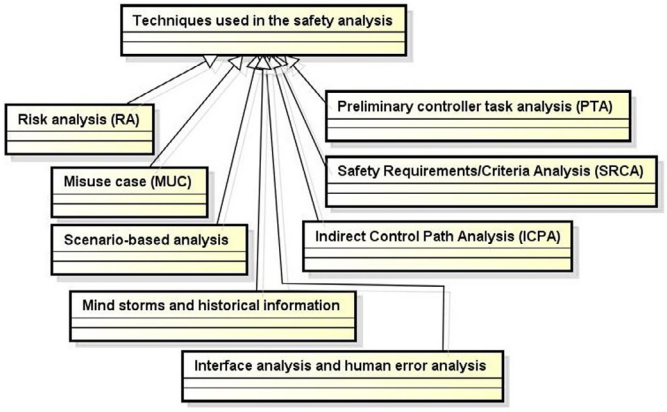

Fig. 6. Taxonomy of general techniques used in the safety analysis according to the selected studies.

4.5.1. RQ1.2: analysis and discussion

In Fig. 6, we present a taxonomy we created to classify the techniques used in the safety analysis but their main goal is not to discover hazards but some other aspect of SCS, for instance risk, safety criteria, usability of interface. On the other hand, the techniques used in the hazard analysis are listed in Fig. 7.

These techniques were cited by the selected studies as techniques that requirements engineers could use to perform hazard/safety analysis during the RE process. Accordingly, these taxonomies contribute to reduce the gap between RE and safety engineering and they allow practitioners to compare such techniques with the ones used in their companies.

The taxonomies of Figs. 6 and 7 provide an intuitive and yet comprehensive way to present and summarise the techniques used in hazard/safety analysis. Furthermore, a taxonomy is an effective means for communicating the results in a more structured manner (Nair et al., 2014). Experts in RE and safety engineering reviewed and provided feedback on the extracted safety techniques.

Some papers adopt techniques for safety analysis that their main objective is not discovering hazards but analyze other aspects of the SCS such as risk, safety criteria, usability of interface. Such techniques are exhibited in the taxonomy of Fig. 6 and comprise *Mind storms and historical information, Preliminary controller task analysis (PTA), Risk analysis (RA), Interface analysis and human error analysis, Indirect Control Path Analysis (ICPA), Safety Requirements/Criteria Analysis (SRCA), Misuse cases (MUC)*, and *Scenario-based analysis*. They are cited 23 times in the selected studies corresponding to 15.75%.

On the other hand, 15 studies (26.32%) did not cite any specific technique used in the safety analysis. They argue that this analysis should be conducted but they did not make any reference.

In the taxonomy of Fig. 7, we list the techniques developed to perform hazard analysis classified in categories or analytical styles. They can be *Inductive (Forward), Deductive (Backward)* or *Both*.

Inductive (forward) analysis is an approach for analysing causal relations that starts with a set of particular facts and reasons to the more general. When employed for safety analysis, inductive analysis starts with a set of failure events and proceeds forward, seeking possible consequences (i.e. hazards) resulting from the events (Saeed et al., 1995). This analytical style is cited 67 times (45.89%) in the studies.

The purpose of a forward search is to look at the effect on the system state of both (1) an initiating event and (2) later events that are not necessarily caused by the initiating event. In fact, causal independence is often assumed (Leveson, 1995). Tracing an event forward can generate a large number of states, and the problem of determining all reachable states from an initial state may be un-

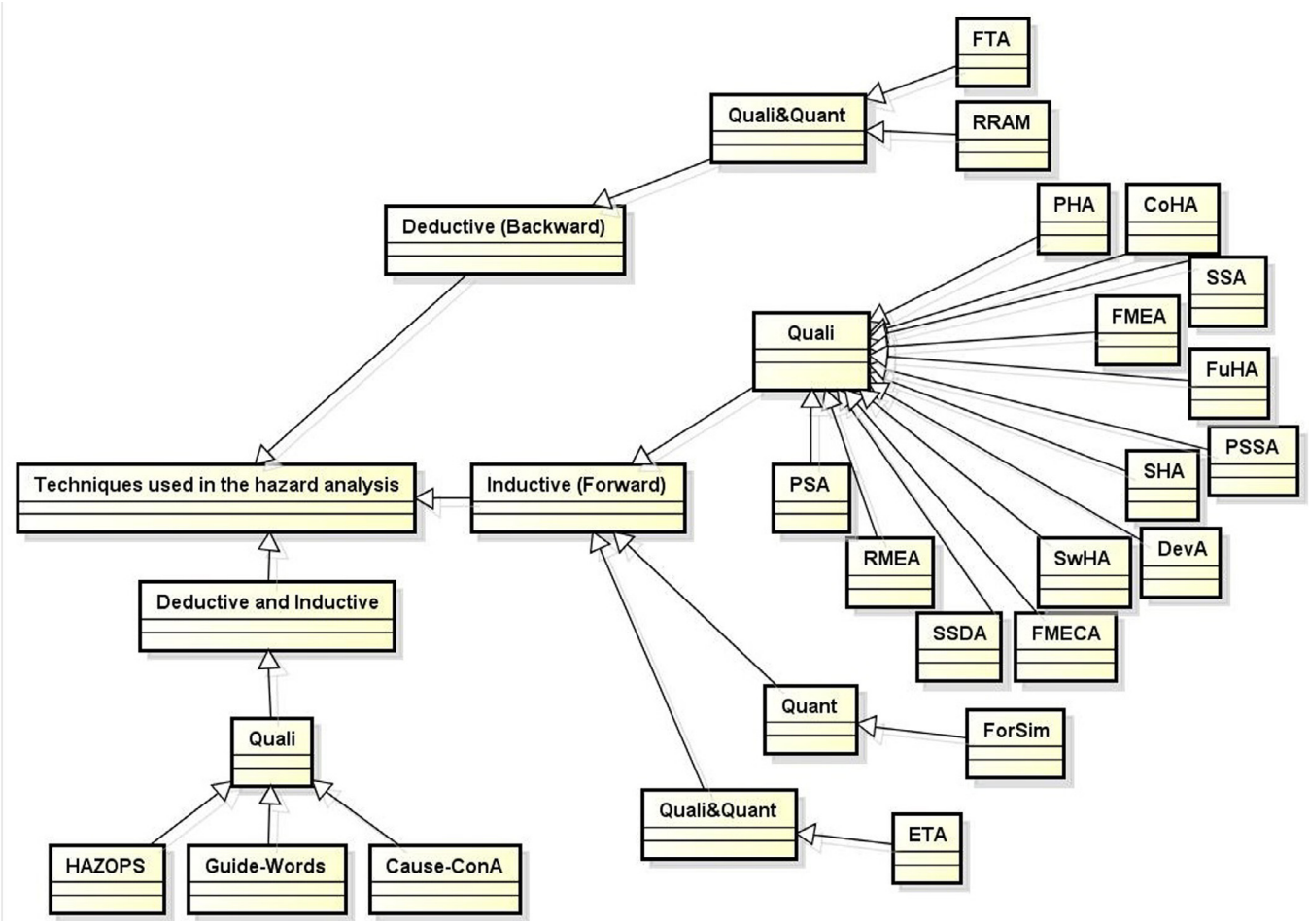

Fig. 7. Taxonomy of techniques used in the hazard analysis according to the selected studies.

solvable using a reasonable set of resources. For this reason, forward analysis is often limited to only a small set of temporally ordered events (Leveson, 1995).

Deductive (backward) analysis is an approach for analysing causal relations that starts with a general fact and reasons towards the more particular. When employed for safety analysis, deductive analysis starts with a hazard and proceeds backwards, seeking possible failures that can lead to the specific hazard (Saeed et al., 1995). This analytical style is cited 24 times (16.44%) in the studies.

Backward search methods fit well with chain-of-event accident models, where the goal is to determine the paths (set of states or events in temporal ordering) that can lead to a particular hazard or accident. They are useful in accident investigations and in eliminating hazards by installing controls to eliminate predecessor events (Leveson, 1995).

Furthermore, some techniques can be classified in *both* analytical styles. These techniques are cited 17 times (11.64%). Both inductive and deductive analysis can be employed during the safety analysis and the degree of application of one style of analysis versus the other varies according to the level of abstraction being considered and the representation technique (Saeed et al., 1995).

The analytical styles can consider two perspectives: *qualitative analysis* and *quantitative analysis*. The *qualitative analysis* is conducted by examining the causal relations between events and states in sequences connecting failures of components to hazard states of the system (Saeed et al., 1995). In the *quantitative* safety analysis, probabilities (or probability density functions) are assigned to the events in the chain and an overall likelihood of a loss is calculated (Leveson, 2011).

Nancy Leveson in Leveson (1995) argues that even if quantitative methods are used, qualitative analysis must precede them - hazards and their causal factors must be identified before numerical values can be assigned to them. Thus, the quality of the quantitative analysis depends on how good the qualitative one was.

The majority of the studies adopts the *qualitative analysis*. Such analysis is indicated at higher levels of abstraction, since the description of the requirements specifications is more general. Quantitative analysis is more appropriate at lower levels of abstraction, since the information on the elements of a requirements specification and their inter-relationships is more concrete and the stakeholders have a better understanding of the system's requirements.

In the work of Martins and Gorschek (2016), FTA and HAZOP were the most reported approaches in the studies aimed at supporting a variety of different actors in the safety-critical system engineering process. These results are similar to ours since they are well-known techniques for safety analysis. These results show that the nature of safety is continuing to be widely misunderstood and there is seemingly a widespread ignorance of established safety techniques in particular with software engineering practitioners (Heimdahl, 2007). For example, safety is often confused with reliability, thus leading to resources being spent on improving component reliability rather than designing safety into the system (Heimdahl, 2007).

Although safety and reliability engineering are, in practice, intertwined in safety-critical system development, and techniques

for reasoning about hazards, their causes and their impact overlap with techniques for reasoning about failures and their impact, safety is not equivalent to reliability or reliability implies safety (Hatcliff et al., 2014). In safety engineering, hazards are the basic unit of management - one tries to determine all of the hazards that are theoretically possible, and then, design a system where the hazards are, if not impossible, then at least very unlikely. On the other hand, in reliability engineering, failures are the basic unit of management - one tries to design the system so that failures are unlikely or using techniques that help ensure that the system will perform its intended function in the presence of faults/failures (Hatcliff et al., 2014).

Heimdahl (2007) also says that the ability to demonstrate (certify) that safety requirements have been met is currently inadequate. Moreover, reliance on models and automated tools in software development, for example, formal modeling, automated verification, code generation, and automated testing, promises to increase productivity and reduce the very high costs associated with software development for critical systems. Such issue can be addressed largely through education and training of the software and safety engineering professionals.

Therefore, many different hazard analysis techniques have been proposed and are used, but all have serious limitations and only a few are useful for software (Leveson, 1995). But whether these techniques or more ad hoc ones are used, it is necessary to identify the software behaviors that can contribute to system hazards (Leveson, 2011; 1995).

4.6. RQ1.3: what data/information artifacts can be created by requirements engineers in the analysis and specification of SCS in the approaches that integrate requirements and safety engineering?

The aim of this question is to identify the various pieces of information used to define safety requirements in the specifications. The results of the conceptual analysis of the information extracted from the studies were used to develop taxonomies of safety-related information that are presented in Figs. 8 and 9. The relationships among these information provide a structured representation of such concepts.

The safety information involved in safety analysis are the *Input, Hazard*, and *Output* (Saeed et al., 1995) as presented in Fig. 8. The safety analysis has as *Input* (Saeed et al., 1995): *mission requirement, domain model, standard and guideline, requirements specification, safety integrity level* (Lu and Halang, 2007), *assumption* (Kaiser et al., 2010; Nejati et al., 2012; Wilikens et al., 1997), and *criteria*.

The *Output* includes the *Safety Goal* as well as *Artefact* generated in the analysis, which can be *Safety Specification* or *Safety Specification Graph (SSG)* (Saeed et al., 1995). The *SSG* is an information model to record the results of the requirements and safety analysis, and their interrelationships. *Evidence* (Saeed et al., 1995; Elliott et al., 1995) is an output whose aim is to demonstrate that the *Hazard* was properly treated. The *Evidence* is composed by *File, Argument*, and *Safety Case* (Lu and Halang, 2007).

Safety Requirement, which constitutes another type of *Output* of safety analysis, is typically of the form of a quality criterion (a system-specific statement about the existence of a sub-factor of safety) combined with a minimum or maximum required threshold along some quality measure. It directly specifies how safe the system must be (Medikonda and Panchumarthy, 2009).

The *Safety Requirement* has the following attributes (Wilikens et al., 1997): *Name, Description, satisfyStatus, validationStatus, InputFlow, Outputflow*, and a *Responsible*. Moreover, it can be of three types (Medikonda and Panchumarthy, 2009): *Safety-significant requirement, Pure safety requirement*, and *Safety constraint* as shown in Fig. 8.

Safety-significant requirement is a normal functional, data, interface, and non-safety quality requirement that is relevant to the achievement of the safety requirements. In other words, a *safety-significant requirement* can lead to hazards and accidents when not implemented correctly (Medikonda and Panchumarthy, 2009).

Pure safety requirement is a requirement that describe what actions and/or constraints should or should not be performed to maintain the system in a safe state (Medikonda and Panchumarthy, 2009). Finally, *Safety constraint* is an architecture or design constraint mandating the use of specific safety mechanism or safeguards (Medikonda and Panchumarthy, 2009). Moreover, *Safety Requirement* and *Hazard* are related to *Functional Requirement* (Paige et al., 2008).

Fig. 9 depicts the information related to *Hazard*. It is a system state that might, under certain environmental or operational conditions (*Context* Wu and Kelly, 2007; Biggs et al., 2016), lead to a *Mishap* or cause a *Harm* (Medikonda and Panchumarthy, 2009; Biggs et al., 2016).

Hazard has four attributes (see Fig. 9): *Description* (Beckers et al., 2013; Wilikens et al., 1997), *Severity Level, Probability*, and *Recommendation* (Thramboulidis and Scholz, 2010). The *Severity Level* (Leveson, 1995; Thramboulidis and Scholz, 2010) can be *Catastrophic* (may cause death or system loss), *Critical* (may cause severe injury, severe occupational illness, or major system damage), *Marginal* (may cause minor injury, minor occupational illness or minor system damage), and *Negligible* (will not result in injury, occupational illness, or system damage).

The *Probability* of a hazard (Wilikens et al., 1997) can be *Frequent* (likely to occur frequently to an individual item, continuously experienced throughout the fleet or inventory); *Probable* (will occur several times during the life of an individual item, frequently throughout the fleet or inventory); *Occasional* (likely to occur sometime during the life of an individual item, several times throughout the fleet or inventory); *Remote* (unlikely to occur but possible during the life of an individual item; unlikely but reasonable expected to occur in a fleet or inventory); and *Implausible* (extremely unlikely to occur to an individual item; possible for a fleet or inventory).

Recommendation can be classified in the following categories (Guiochet et al., 2010): *modification of allowed use, modification of the specification*, and *modification of the design* as depicted in Fig. 9.

Risk is associated with *Hazard* and it is a combination of consequence (severity hazard) and likelihood of the hazard (risk = probability hazard x severity hazard) (Thramboulidis and Scholz, 2010; Simpson and Stoker, 2002).

Some strategies should be defined to minimize the consequence or probability of the hazard. A *Strategy* has the following attributes: *Strategy Type, Description, Responsible*, and *Refinement*.

The *Strategy Type* (Kaiser et al., 2010) can be *Reaction* or *Detection* through mechanisms (*Timeout, Value outside of a valid scope*) (Kim and Chung, 2005). The strategy of *Reaction* can be *Arrest, Mitigation, Recovery*, and *Analysis*.

Hazard also has at least one *Cause* (Thramboulidis and Scholz, 2010). It occurs due to *Environmental Hazard, Procedural Hazard, Interface Hazard, Human Factor* or *System Cause* (Medikonda and Panchumarthy, 2009). The last one can be *Failure* or *System MisBehavior* (Scholz and Thramboulidis, 2013).

Failure is an event where a system or subsystem component does not exhibit the expected external behavior. *Failure* has a *Probability* (Paige et al., 2008) and it is related to *Hardware* (*Electronic* or *Mechanical*) or *Software* (Galvao Martins and De Oliveira, 2014).

Hazard also has *Effect or consequence* in five levels of *Impact* (Zoughbi et al., 2011): *Catastrophic, Hazardous/Severe-Major, Major, Minor* or *No Effect*. This *Effect or consequence* can be a *Mishap* (Medikonda and Panchumarthy, 2009) or a *Harm. Mishap* can

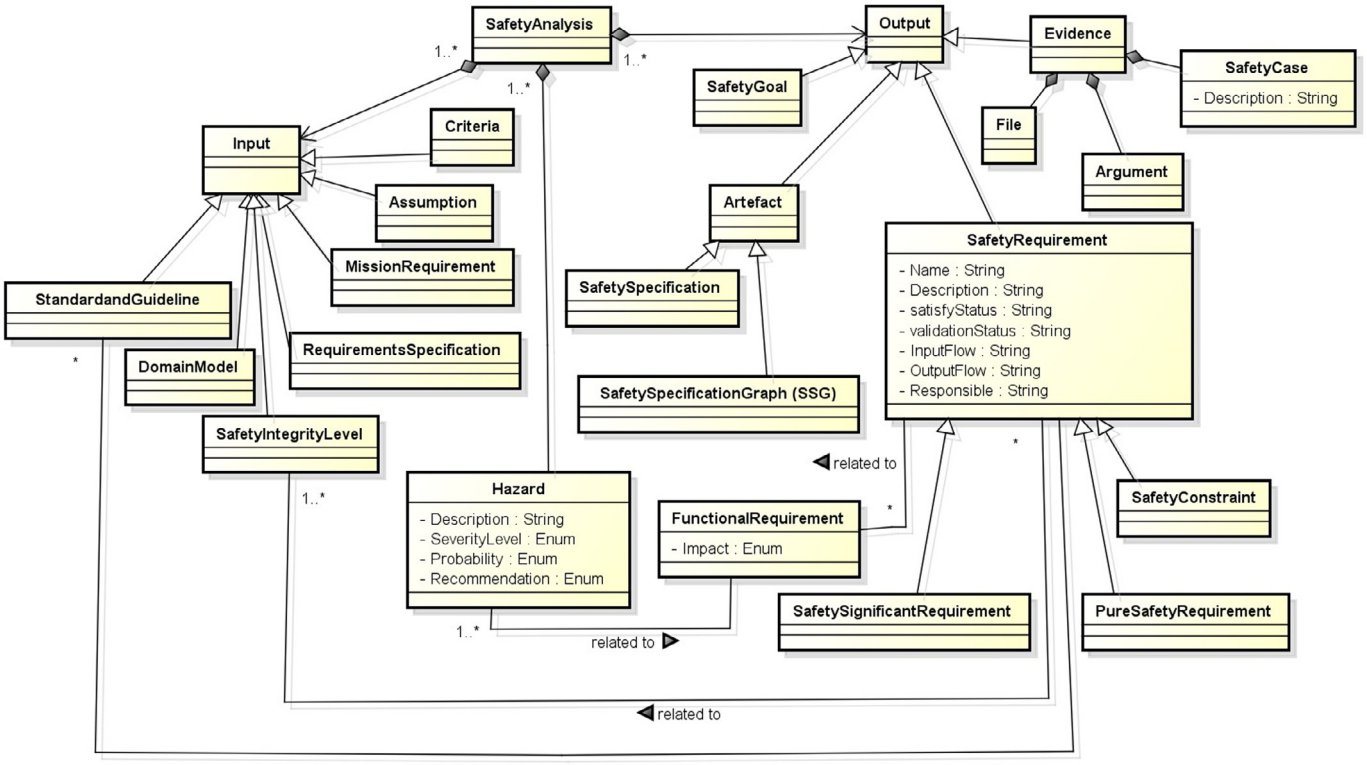

Fig. 8. Safety information taxonomy according to the selected studies.

be a *Accident* (Górski and Wardziński, 1996) or a *Safety Incident* (Broomfield and Chung, 1997).

An *Accident* is an undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss. On the other hand, a *Safety Incident* (Broomfield and Chung, 1997) is an event that involves no loss (or only minor loss) but with the potential for loss under different circumstances (Medikonda and Panchumarthy, 2009).

A *Harm* has a *Type* which occur to (Biggs et al., 2016) *People, Property, Environment* or *Service* (Mustafiz and Kienzle, 2009). Each type is an *Asset* of a system.

Harm to *People* (Human beings, roles played or organizations) can be *Death (Loss), Injury* (Guiochet et al., 2010), *Illness, Kidnap, Hardship*, or *Corruption (bribery or extortion)*.

Harm to *Property* can be *Destruction, Damage, Corruption, Theft, Unauthorized access* or *Unauthorized disclosure*. A *Property* has two attributes *PropertyType* and *PropertyOwner*. A *PropertyType* can be *Tangible* or *Not Tangible* and the *PropertyOwner* can correspond to *Private Property, Public Property* or *Commercial Property*.

Harm to *Environment* can be *Destruction, Loss of Use* or *Damage*. Finally, *Harm* to *Service* can be *Corruption, Unauthorized usage (theft), Accidental loss of service, Denial of service (DOS)* or *Repudiation of transaction*.

4.6.1. RQ1.3: analysis and discussion

Terms related to system safety are not used consistently in the selected studies. Differences exist among countries and industries. The confusion is compounded by the use of the same terms, but with different definitions, by engineering, computer science and natural language (Leveson, 1995).

To promote the effective integration of safety analysis and requirements analysis, a common formal basis should be provided for the results of these techniques (Simpson and Stoker, 2002). The taxonomies presented in our SLR provide a more solid background for requirements engineers during safety analysis at system level. Our goal is to create an agreed-upon vocabulary and semantic structure containing all the relevant concepts, their relations and axioms within safety domain for the purpose of exchanging information and facilitating reasoning.

With the taxonomies, presented in Figs. 8 and 9, we aim to capture in particular information that is shared by RE and safety engineering. The structuring of concepts that belong to different areas is a challenging task since we have to consider the non-standardization of nomenclature, synonyms, redundancies and the relationships between the various pieces of information. Although we expect that the taxonomy can be refined and extended since we consider only the data extracted from the selected studies, we believe that the taxonomies contribute to a better integration and communication between RE and safety engineering.

Besides the communication among requirements and safety engineers, there is also the communication with certification authorities. The certification process of a system requires demonstrating that appropriate safety standard was followed during the development process.

Many standards require that a safety assessment be performed when developing or modifying a safety-critical system (Zoughbi et al., 2011). However, the differences among standards make it hard to translate evidence of compliance among them (Hatcliff et al., 2014). In this context, the proposed taxonomies although did not considered the constraints proposed by the safety standards, they may help understanding and applying the safety standards and regulations since they provide the information that both engineering groups have to exchange for system safety analysis, during design and in the preparation of reports for certification. This common basis contributes to ensure that sufficient correct evidence has been collected to satisfy the relevant standards for certifying a system and thus improve the certification process (Biggs et al., 2016; Hatcliff et al., 2014).

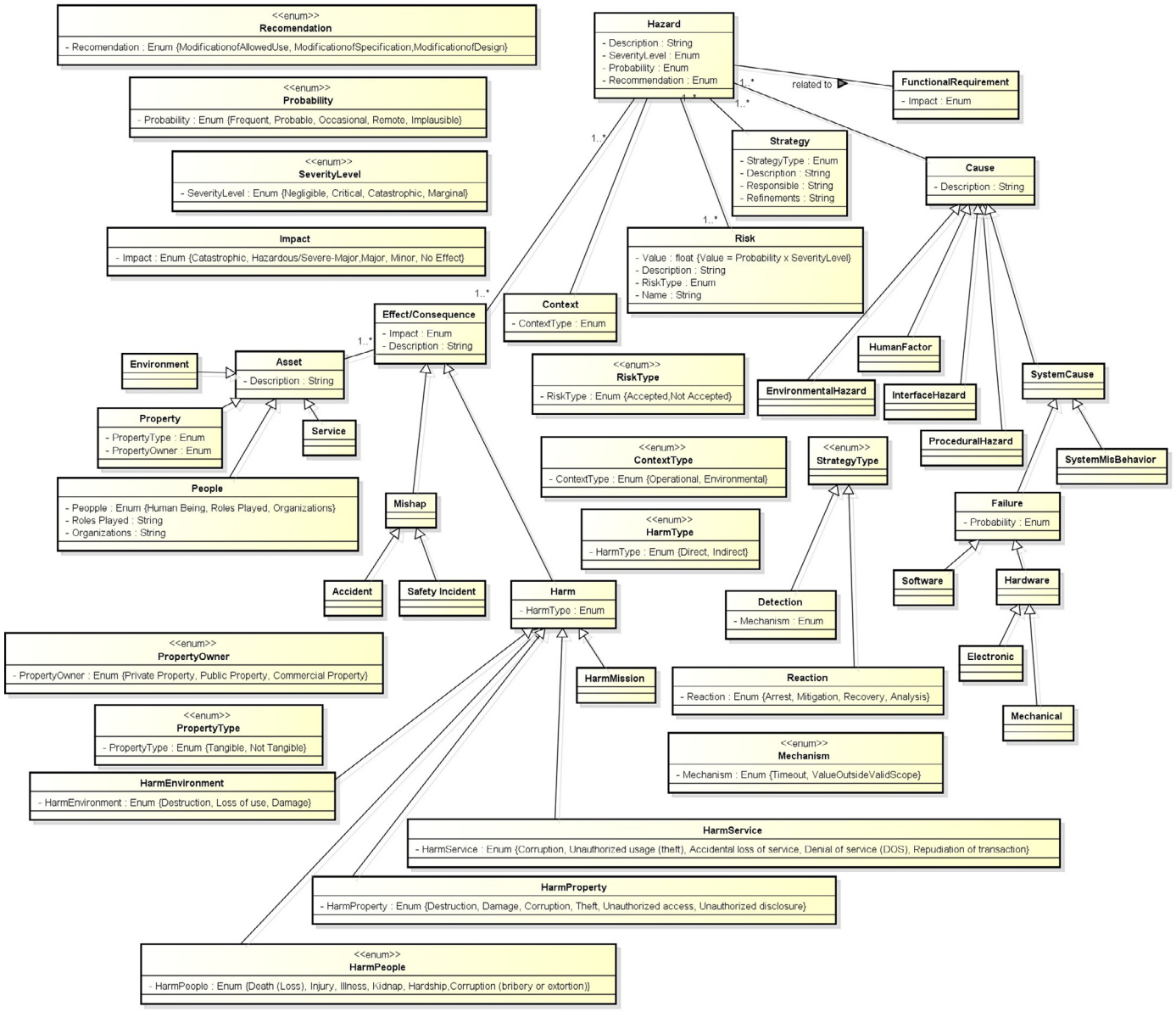

Fig. 9. Hazard information taxonomy according to the selected studies.

The taxonomies representing the safety-related information proposed in this paper can contribute to researchers and practitioners to elicit (Maiden et al., 1999; Firesmith, 2005), organize, and manage safety requirements. Moreover, taxonomies can be used in many purposes (Firesmith, 2005): training; improve communications through the use of common terms and concepts of the taxonomy and thereby improving the collaboration (Unterkalmsteiner et al., 2014); verify requirements completeness in systems specification against the terms presented in the taxonomy (Maiden et al., 1999); structure the table of contents of the requirements specification related to safety requirements; as a checklist for evaluating the safety-related requirements in a requirements repository; improve the traceability between the requirements and the safety-related information.

4.7. RQ1.4: what are the tools used by the approaches that integrate requirements and safety engineering in safety analysis?

This question maps the tools shared by requirements and safety engineers to develop the requirements specification of safety-critical systems. In Fig. 10, we present the list of tools referenced in the approaches. The list is composed by the following tools: Sparx Systems Enterprise Architect, DOVE (Design Oriented Verification and Evaluation) tool, HIVE (Hierarchical Verification Environment) tool, Isabelle theorem prover, ISPRA-FTA, UWG3, Adelard's ASCE, Rodin platform, UM4PF, SafeSlice, Aralia Sim Tree, BPA DAS, KB3, Jagrif, Netica, AToM, ERRSYSL, APIS IQ-RM tool, Doors and tools developed by the approach (i.e. a proposed one).

4.7.1. RQ1.4: analysis and discussion

The analysis of Fig. 10 shows that 66.67% of the studies *did not cite* any tool support for the safety analysis. This outcome indicates the need of development of tools capable of integrating the requirements specification, safety analysis, and system management maintaining traceability links among all artefacts, models, and the information necessary for the development and certification of SCS.

The proposal of *new tools* is presented in 14.04% of the approaches (8 studies). It is too soon to affirm that there is a tendency in the academia and the market regarding to the develop-

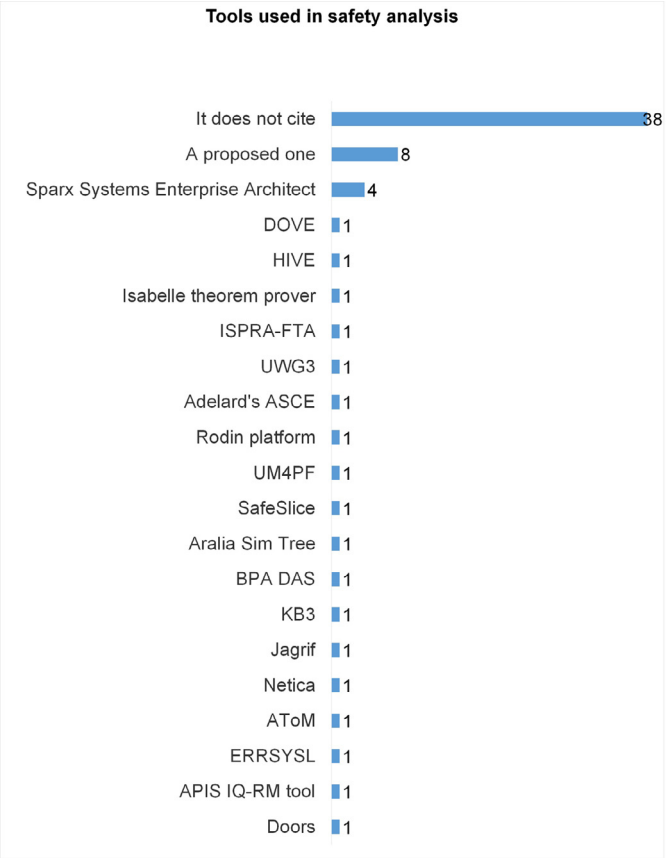

Fig. 10. Tools used in safety analysis.

ment of tools for improving the safety analysis as required by the industry.

The *Sparx Systems Enterprise Architect* is used in 4 approaches (7.02%). This tool provides full life cycle modeling for: Business and IT systems, Software and Systems Engineering, and Real-time and embedded development (Sparx, 2016).

The remaining tools are cited in only one paper each: DOVE (Design Oriented Verification and Evaluation) tool (Cant et al., 2006), HIVE (Hierarchical Verification Environment) tool (Cant et al., 2006), Isabelle theorem prover (Cant et al., 2006), ISPRA-FTA (Wilikens et al., 1997), UWG3 (Paige et al., 2008), Adelard's ASCE (Paige et al., 2008), Rodin platform (Stålhane et al., 2010), UM4PF (Beckers et al., 2013), SafeSlice (Nejati et al., 2012), Aralia Sim Tree (David et al., 2010), BPA DAS (David et al., 2010), KB3 (David et al., 2010), Jagrif (David et al., 2010), Netica (Wu and Kelly, 2007), AToM (Mustafiz and Kienzle, 2009), ERRSYSL (Schedl and Winkelbauer, 2008), APIS IQ-RM tool (Kaiser et al., 2010), and Doors (Kaiser et al., 2010).

The lack of (commercial) tool support that would allow integrating requirements models in a seamless development process as well as the insufficient guidance, pointed by Sikora et al. (2012), lead to uncertainty about how models should be used in the RE process.

The results about the tools cited in the selected studies demonstrate the absence of information on tools in the academic literature. Accordingly, we cannot draw conclusions about tool use with only the extracted information. One possible reason of these few number of tools can be the fact that regulatory agencies have been quite reluctant to qualify any tools for use on critical systems projects and they are actively debating how to address the issue (Heimdahl, 2007). In this context, Heimdahl (2007) complements stating that it is "highly unlikely that the researchers be able to provide the level of confidence necessary to trust a specific tool as a development tool in SCS development. Moreover, there are others aspects involved such as the tool evolution (maintenance, upgrades, and migration to new platforms)".

According to Hatcliff et al. (2014), researchers are not encouraged to engineer their tools to support certification (Heimdahl, 2007; Hatcliff et al., 2014). Moreover, the certification requirement vary significantly across safety standards for different domains. This requires tool vendors to develop different versions of their tool and qualification kits for each standard supported (Hatcliff et al., 2014). Finally, another problem pointed out by Hatcliff et al. (2014) is that most of the tools are not standalone and rely on other tools and libraries that have their own independent life.

Table 9
Benefits of the approaches for integration between RE and safety engineering.

Benefit	Count	%
B1: Reduction of errors in requirements specifications (increases quality).	25	43.86%
B2: It improves system safety.	17	29.82%
B3: It improves the analysis during overall system design.	8	14.04%
B4: Reduction of the software cost.	8	14.04%
B5: Models contributes to a precise (unambiguous) communication.	5	8.77%
B6: Bridge the existing gap between the disciplines and provide a framework for effective cooperation between experts.	4	7.02%
B7: Improves the traceability among requirements, design and safety requirements.	4	7.02%
B8: Better information presentation and increased information consistency.	3	5.26%
B9: Reduction of the workload on safety engineers.	3	5.26%
B10: Make appropriate design decisions and adaptation of the design to meet the safety requirements.	3	5.26%
B11: It contributes to have the same vocabulary.	3	5.26%
B12: Structuring the analysis in different steps on different levels.	3	5.26%
B13: Reduction of safety-related interface faults.	2	3.51%
B14: Reduction of the time in safety analysis.	2	3.51%
B15: It increases the confidence in the overall system development process.	2	3.51%
B16: Reduction of the number of iterations between system engineers and safety engineers.	1	1.75%
B17: It allows exhaustive and detailed user feedback and make possible to discover and then specify the complete system behavior.	1	1.75%

4.8. RQ1.5: what are the benefits of the approaches that integrate requirements and safety engineering identified in RQ1?

In Table 9, we present the benefits of the approaches for the integration and communication between requirements engineering and safety engineering from the extracted data of the selected studies.

4.8.1. RQ1.5: analysis and discussion

The use of software in safety-critical systems, in particular in control systems, has increased to such an extent that failures in the software can impair system safety. In this context, analysis of software-related errors in computer based safety-critical systems have shown that mistakes made during the requirements analysis phase can easily introduce faults which subsequently lead to accidents (Saeed et al., 1995). In the next sections, we discuss the benefits of the approaches to the integration and communication between RE and safety engineering.

4.8.2. B1: reduction of errors in requirements specifications (increases quality)

A tendency we observed in the direction of decreasing the ambiguity and inconsistency of natural language specifications is to use common models for requirements specification and safety/hazard analysis shared by the requirements and safety engineers. This contributes partially to improve the process of exchanging information, increasing completeness, and correctness of the requirements specifications (Fricker et al., 2008) since there are other aspects involved such as the engineers experience and knowledge. The proposals are concerned with improving the quality of the specification of safety-critical systems as mentioned in 25 studies (43.86%) (Saeed et al., 1995; Mostert and von Solms, 1994; Lutz, 1993; Black and Koopman, 2008; Galvao Martins and De Oliveira, 2014; Medikonda and Panchumarthy, 2009; Wu and Kelly, 2007; Martin-Guillerez et al., 2010; Hansen et al., 1998; Markovski and van de Mortel-Fronczak, 2012; Arogundade et al., 2012; El Ariss et al., 2011; Jrjens, 2003; Biggs et al., 2016; Stålhane and Sindre, 2007; Mustafiz and Kienzle, 2009; Ekberg et al., 2014; Wilikens et al., 1997; Paige et al., 2008; Schedl and Winkelbauer, 2008; Rafeh, 2013; Tschrtz and Schedl, 2010; Murali et al., 2015; Pernstål et al., 2015; Fricker et al., 2010).

4.8.3. B2: it improves system safety

Addressing the safety concerns early in software development contributes to ensure that safety problems do not propagate through subsequent phases of development (Saeed et al., 1995): the less time a misunderstanding has to unfold, the lower its impact (Glinz and Fricker, 2015). According to 17 approaches (29.82%), conducting the safety analysis in the requirements phase allows hazards be identified and addressed early. This benefit was explicit discussed in Saeed et al. (1995); Lutz (1993); Black and Koopman (2008); Kim and Chung (2005); Medikonda and Panchumarthy (2009); Arogundade et al. (2012); Chandrasekaran et al. (2009); Górski and Wardziński (1996); Zoughbi et al. (2011); Simpson and Stoker (2002); Lu and Halang (2007); Rafeh (2013); Chen et al. (2011); Tschrtz and Schedl (2010); Elliott et al. (1995); Jurkiewicz et al. (2015); Stålhane et al. (2010).

4.8.4. B3: improving the analysis during overall system design

This benefit is presented in 8 (Kaiser et al., 2010; Ratan et al., 1996; Kim and Chung, 2005; Martin-Guillerez et al., 2010; Zoughbi et al., 2011; Tschrtz and Schedl, 2010; Fricker et al., 2010; 2008) approaches (14.04%). Identifying and assessing hazards is not enough to make a system safe; the information obtained in the hazard analysis needs to be used in the design (Leveson, 1995) and implementation. In this context, some approaches aim to improve the integration between requirements and safety engineering by conducting the design and the safety analysis concurrently, thereby making it possible to let safety analysis results influence the system design. In order to do this, safety analysis and the system design may use the same system safety specification.

4.8.5. B4: reduction of the software cost

The late identification of problems in the safety specification leaves a number of potential risks that must be accounted for at huge cost at a later stage (Mostert and von Solms, 1994). The cost is reduced since the hazards are discovered in the requirements phase where the cost of fixing an error is cheaper than in the later stages of the software development. Hence, safety must be designed into a system. This benefit was pointed by the studies (Kaiser et al., 2010; Kim and Chung, 2005; Stålhane and Sindre, 2014; Zoughbi et al., 2011; Jrjens, 2003; Biggs et al., 2016; Wilikens et al., 1997; Pernstål et al., 2015) as a benefit of the integration between requirements and safety engineering.

4.8.6. B5: models contributes to a precise (unambiguous) communication

The use of common models among requirements, design and safety teams is the approach adopted by Wu and Kelly (2007); Nejati et al. (2012); Markovski and van de Mortel-Fronczak (2012); Arogundade et al. (2012); El Ariss et al. (2011); Stålhane and Sindre (2007); Ekberg et al. (2014); Chen et al. (2011); Murali et al. (2015) to improve the integration between requirements and safety engineering. The benefits of model orientation compass its structuring effect, the creation of shared meaning around the models, the elimination of error and ambiguity within the models (Whitehead, 2007). Model-based specifications are consistent and less ambiguous than informal specification documents, forcing the stakeholders to clarify all aspects of the system early in the design process (Markovski and van de Mortel-Fronczak, 2012).

4.8.7. B6: bridge the existing gap between the disciplines and provide a framework for effective cooperation between experts

By working on with common artifacts and complementing them by a well-defined process of safety analysis, the requirements, including safety ones, can be refined stepwise and both system developers and safety engineers can leverage synergetic effects in their workflow. Moreover, it also increases the efficiency in coordination and management of members of the development teams, including safety engineers (Scholz and Thramboulidis, 2013). This benefit is mentioned in 4 studies (Kaiser et al., 2010; Thramboulidis and Scholz, 2010; Scholz and Thramboulidis, 2013; Guillerm et al., 2010).

4.8.8. B7: improves the traceability among requirements, design and safety requirements

The traceability links among requirements, design, safety requirements or implementation allow changes to be made much more quickly and easily paying off the effort of them as mentioned in Navarro et al. (2006); Mannering et al. (2008); Nejati et al. (2012); Guillerm et al. (2010). According to Leveson (2002), it could be prohibitively expensive, for example, to generate a new hazard and safety assessment for each system change that is proposed. Being able to trace a particular design feature or implementation item to the original hazard analysis allows decisions to be made about whether and how that feature or code can be changed. The same is true for changes that affect operator activities and basic task allocation and usability principles. In some regulated industries, traceability is required by the certification authorities. Moreover, the creation of traceability links between the concepts involved in safety analysis facilitate the comprehension and/or the impacts analysis (Guillerm et al., 2010).

4.8.9. B8: better information presentation and increased information consistency

Modelling languages promote better information presentation and increased information consistency (Biggs et al., 2016; Fricker et al., 2010). Briones et al. (2007) complement this point of view by affirming that working with the same models improves the integration and communication between RE and safety engineering among stakeholders by having the same vocabulary but also avoids the need to keep model consistency.

4.8.10. B9: reduction of the workload on safety engineers

This benefit was pointed out by 3 studies (Mannering et al., 2008; Nejati et al., 2012; Biggs et al., 2016). Safety is not only the responsibility of a couple of safety specialists in a separated department, writing extensive safety documentation (Schedl and Winkelbauer, 2008). Hence, it is necessary to make people aware of the fact that safety is everybody's responsibility to improve the quality of the safety specifications. Considering safety concerns as

early as possible, i.e. during requirements phase contributes to reduce the timespan between causing and detecting misunderstandings. Moreover, an efficient integration and communication among RE and safety engineering could reduce the number of iterations between requirements engineers and safety engineers, and reduce the workload on safety engineers, without impacting the quality of system designs (Biggs et al., 2016).

4.8.11. B10: make appropriate design decisions and adaptation of the design to meet the safety requirements

It is the benefit of the approaches of Martin-Guillerez et al. (2010); Scholz and Thramboulidis (2013), as well as in Biggs et al. (2016). They argue that conducting safety analysis in the early phases of system development it is possible to make appropriate design decisions and increase the effectiveness of the development process.

4.8.12. B11: it contributes to have the same vocabulary

The same vocabulary is an important issue for the integration of requirements and safety engineering since it contributes for a better exchanging information. This benefit was mentioned by Briones et al. (2007); Du et al. (2014) as well as Fricker et al. (2010). This common basis increases the elicitation techniques for SCS (Du et al., 2014) and it enables a deep requirements understanding and prepare for projects (Fricker et al., 2010).

4.8.13. B12: structuring the analysis in different steps on different levels

In the development of a safety-critical system, it is necessary to make trade-offs between the level of detail of system specifications needed for safety analysis and the cost of elaborating them. Safety requirements defined on different levels of abstraction contributes to a better integration between the requirements and safety engineering (Beckers et al., 2013; Guiochet et al., 2010; Croll et al., 1997). These different levels of abstraction improve the quality of system specification and increase the shared understanding.

4.8.14. B13: reduction of safety-related interface faults

A better integration between requirements and safety engineering decreases the miscommunication between development teams which is the primary cause of safety-related interface faults, misunderstood requirements, and misunderstood interfaces (Lutz, 1993; Black and Koopman, 2008). Hence, in order to manage safety at the subsystem level prior to system integration, system safety requirements must be clearly defined for subsystems (Black and Koopman, 2008).

4.8.15. B14: reduction of the time in safety analysis

When requirements engineers consider safety concerns early in the system development process, it is possible to detect hazards in advance of the safety lifecycle. The detection of hazards in the beginning of the development process (i.e. RE phase) reduces the time in safety analysis in comparison they being discovered in the final stages where the analysis should consider the entire system and all of its components.

4.8.16. B15: it increases the confidence in the overall system development process

Appropriate design decisions are performed when the groups work together increasing the effectiveness of the development process (Scholz and Thramboulidis, 2013). The teams should collaboratively produce artifacts, safety analysis and the evidence that the safety concerns were taken care of. The role of the certification authority is to review the evidence provided by the teams and to make recommendations for acceptance of the system in the light of operational requirements.

4.8.17. B16: reduction of the number of iterations between system engineers and safety engineers

The integration and communication between RE and safety engineering could reduce the number of iterations among system engineers (Biggs et al., 2016) since if the adequate training and sharing the same vocabulary, the safety analysis can be performed in a more efficient way. Therefore, the reduction of the number of interactions in this context may even can improve the quality of system designs.

4.8.18. B17: it allows exhaustive and detailed user feedback and make possible to discover and then specify the complete system behavior

Thinking about behaviour or events that affect the reliability or safety of the system has to start at the requirements phase, because it is up to the stakeholders of the system to decide how they expect the system to react to exceptional situations. Only with exhaustive and detailed user feedback it is possible to discover and then specify the complete system behavior in a subsequent analysis phase, and decide on the need for employing fault masking and fault tolerance techniques for achieving run-time dependability during design (Mustafiz and Kienzle, 2009).

4.9. RQ2: what challenges/problems are identified in research literature relating to SCS and RE?

This question aims to identify works needed in this area. These challenges/problems were extracted from the selected studies and they are presented in Table 10.

4.9.1. RQ2: analysis and discussion

Many studies presented their proposals, discussed some benefits but they did not explicitly discussed challenges/problems in integration between requirements engineering and safety engineering as well as in requirements communication. This corresponds to 64.91% of the studies (37 studies).

The most cited challenges/problems are *Analysis of scalability of the technique about integration and communication between RE and safety engineering in real case studies (O1)* and *Conduction of more empirical studies about integration and communication between RE and safety engineering (O2)*. They were referenced in 4 studies (7.02%) each and are the consequence of the low number of proposals evaluated in the industrial context. These results show the need of applying the proposal in practice with real users in order to evaluate the extension of the contributions.

Develop safety analysis tools integrated with requirements specification (O3) is a concern mentioned in 3 studies (5.26%). Considering that 66.67% of the studies did not cite any kind of tool support (see Section 4.7), this outcome might indicate the need of development of tools capable of integrating the requirements specification, safety analysis, and system management maintaining traceability links among all artefacts, models, and the information necessary for the development and certification of SCS. This challenge about *Maintaining the traceability among (safety) requirements, architecture and implementation along with system development and evolution (O4)* is pointed out by 2 approaches (3.51%).

The *Creation of formal guidelines to help requirements engineers to derive and communicate safety functional requirements from safety analysis (O5)* is discussed in 2 studies (3.51%). According to Broomfield and Chung (1997), the creation of generic tasks from safety requirements is done in an ad hoc fashion.

How to *Integrate formal description techniques with safety requirements specifications (O6)* is described by 2 studies (3.51%). The capability of formal languages in system analysis might be used to improve safety analysis and verification of safety requirements specifications.

Table 10
Challenges/problems in the integration and communication between RE and safety engineering.

Challenge/Problem	Studies	Count	%
It does not cite.		37	64.91%
O1: Analysis of scalability of the technique about integration and communication between RE and safety engineering in real case studies.	(Ratan et al., 1996; Black and Koopman, 2008; Navarro et al., 2006; Stålhane and Sindre, 2014)	4	7.02%
O2: Conduction of more empirical studies about integration and communication between RE and safety engineering.	(Saeed et al., 1995; Galvao Martins and De Oliveira, 2014; Mannering et al., 2008; Stålhane and Sindre, 2014)	4	7.02%
O3: Develop safety analysis tools integrated with requirements specification.	(Navarro et al., 2006; El Ariss et al., 2011; Jrjens, 2003)	3	5.26%
O4: Maintaining the traceability among (safety) requirements, architecture and implementation along with system development and evolution.	(Kaiser et al., 2010; Chen et al., 2011)	2	3.51%
O5: Creation of formal guidelines to help requirements engineers to derive and communicate safety functional requirements from safety analysis.	(Galvao Martins and De Oliveira, 2014; Broomfield and Chung, 1997)	2	3.51%
O6: Integrate formal description techniques with safety requirements specifications.	(Kim and Chung, 2005; Mannering et al., 2008)	2	3.51%
O7: Improve the completeness of requirements specification for safety analysis.	(Sikora et al., 2012; Hatcliff et al., 2014)	2	3.51%
O8: Different standards in varying depth of compliance to be fulfilled can be bewildering to the stakeholders and a significant barrier to communication.	(Sikora et al., 2012; Hatcliff et al., 2014)	2	3.51%
O9: Lack of experience of different stakeholders in safety engineering and the application domain (gaps in assumed knowledge, vocabulary and understanding) hampers exchanging information.	(Heimdahl, 2007; Hatcliff et al., 2014)	2	3.51%
O10: Requirements documentation tends to become large, ambiguous, inconsistent, and often lack clear structure affecting the process of exchanging information.	(Heimdahl, 2007; Hatcliff et al., 2014)	2	3.51%
O11: Decide and communicate which safety subgoals are “best”.	(Black and Koopman, 2008)	1	1.75%
O12: Devise safety analysis techniques based on novel abstraction notions, that are appropriate for communication between application and software domains.	(Saeed et al., 1995)	1	1.75%
O13: How safety checklists can be employed during the requirements phase to predict which factors in a particular system are likely to cause subsequent safety-related software errors.	(Lutz, 1993)	1	1.75%
O14: Extending safety concepts in UML diagrams to improve exchanging safety information.	(Zoughbi et al., 2011)	1	1.75%
O15: Evaluation of the time and cost of implementing an approach related to integration and communication between RE and safety engineering.	(Medikonda and Panchumarthy, 2009)	1	1.75%
O16: Adapt the integration and communication between RE and safety engineering proposal to the needs of any project size and of complexity.	(Paige et al., 2008)	1	1.75%
O17: Ensuring the correctness, completeness and consistency of safety requirements, analysis results and the subsequent design solutions contributing to a better communication process.	(Chen et al., 2011)	1	1.75%
O18: Mastering, during design phase, the complexity of the combination of various technologies.	(Heimdahl, 2007)	1	1.75%
O19: Available safety analysis techniques are not adequate to establish explicit shared understanding among stakeholders and perform requirements validation and verification.	(Heimdahl, 2007)	1	1.75%
O20: Support for defining requirements across different abstraction layers to improve shared understanding.	(Sikora et al., 2012)	1	1.75%

Improve the completeness of requirements specification for safety analysis (O7) is necessary for the identification of all hazards and the associated safety requirements, identification of all necessary interfaces to the system, and specified behaviour for the complete input domain for each interface; and automated simulation of the specified behavior (Hatcliff et al., 2014).

Different standards in varying depth of compliance to be fulfilled can be bewildering to the stakeholders and a significant barrier to communication (O8). Safety-critical systems should be submitted to certification by regulatory agencies, therefore, the regulator will typically require detailed, explicit specifications, thus leaving little room for alternative forms such as implicit shared understanding (Glinz and Fricker, 2015). The presence of so many standards, and the differences among them, can difficult the education of new practitioners and researchers.

The knowledge about the domain of the system to be built enables software engineers to better understand the stakeholders' needs, thus fostering shared understanding. Domain knowledge reduces the probability that software engineers misinterpret specifications or fill gaps in the specification in an unintended way (Glinz and Fricker, 2015). In this context, the *lack of experience of different stakeholders in safety engineering and the application domain (gaps in assumed knowledge, vocabulary and understanding) hampers exchanging information (O9).*

Some studies mention that *Requirements documentation tends to become large, ambiguous, inconsistent, and often lack clear structure affecting the process of exchanging information (O10).* This constitutes a challenge since the system specification is written in natural language and, generally, in an unstructured way. The ambiguity and inconsistency affect the process of exchanging information, decreasing completeness, and correctness of the requirements specifications (Fricker et al., 2008).

In many cases, there are many goals and subgoals that should be satisfied by the system. However, they can be conflicting and it is necessary guidance for choosing among them as well as to *decide and communicate which safety subgoals are “best” (O11).*

Devise safety analysis techniques based on novel abstraction notions, that are appropriate for communication between application and software domains (O12) is pointed out by Saeed et al. (1995). The use of abstract layers contributes to increase the safety analysis in all stages of the SCS development and reduce the workload of safety and requirements engineers.

Safety checklists were proposed for using during the analysis of software requirements in different safety-critical and embedded systems. Nevertheless, it is necessary guidelines to *How safety checklists can be employed during the requirements phase to predict which factors in a particular system are likely to cause subsequent safety-related software errors (O13).*

System safety requirements should be specified through different abstraction layers and different diagrams. Therefore, *extending safety concepts in UML behavioral diagrams to improve exchanging safety information (O14)* is necessary to analyze the safety behavioral aspects of the systems (e.g., sequences of events and reactions).

The few number of case studies about the technology transfer related to integration and communication between RE and safety engineering makes the *Evaluation of the time and cost of implementing an approach related to integration and communication between RE and safety engineering (O15)* a problem. It is necessary to conduct more empirical studies, so the community be able to use these empirical data and to propose solutions according to need of the industry.

Investigations are also necessary to propose mechanisms to *Adapt the integration and communication between RE and safety engineering proposal to the needs of any project size and of complexity (O16)*. A possible solution may be the definition of core parts of the proposals that can remain unchanged, and some guidelines that will allow engineers to tailor the proposal to their needs (Paige et al., 2008).

Difficulties are also described in *Ensuring the correctness, completeness and consistency of safety requirements, analysis results and the subsequent design solutions contributing to a better communication process (O17)*. Considering that many artifacts are generated with different functionalities in development of SCS, specially in the RE phase and safety analysis, the traceability of all these artifacts is a substantial challenge.

In the development of SCS, many technologies are used to handle the problems and technology decisions are made by engineers mostly unaware of the challenges they will face, the technologies readily available. Hence, *mastering, during design phase, the complexity of the combination of various technologies (O18)* is an important problem to integration and communication between RE and safety engineering.

Available safety analysis techniques are not adequate to establish explicit shared understanding among stakeholders and perform requirements validation and verification (O19). Due to the implicit shared understanding in the system's specifications, too much information about the system is hidden. It might be useful to document what requirements and design decisions have not been documented in detail due to reliance on implicit shared understanding so that this information is not lost when the developed system evolves.

RE approaches do not always account for complex systems since the use of abstraction layers is often inconsistent, and sophisticated approaches and guidance are urgently needed to manage this complexity (Sikora et al., 2012). Accordingly, the *support for defining requirements across different abstraction layers to improve shared understanding (O20)* is an obstacle for the specification of embedded systems (Sikora et al., 2012) since there are many stakeholders involved.

The above open issues may be useful in different contexts. For example, a newcomer (e.g. new student research) will be able to identify new research opportunities and they can become the subject of new research projects.

5. Conclusions

Investigations into the causes of accidents suggest that more rigour is required in setting the requirements and specification of safety-related systems. In this context, safety engineering is effective when it participates in and provides input to the RE process and to the system design. Accordingly, the main objective of this paper is to synthesise the existing knowledge about the integration between RE and safety engineering. Different aspects of such integration was analyzed such as activities, hazard/safety techniques, relationships between safety information, tools to support safety analysis as well as the benefits of the integration.

Our SLR draws on 57 studies, selected out of 1037, through a multi-stage process. A key feature of the review is that it does not restrict itself to a particular domain or safety standard. This broad scope in the search gives us deeper insights on the state-of-the-art about the content of the integration of RE and safety engineering.

The most relevant findings from this review and their implications for further research are as follows.

Non-standardization of nomenclature. There is a gap that exists between the traditional development processes, methodologies, notations and tools and the ones used in safety engineering. This gap makes the integration and communication between RE and safety engineering a difficult and challenging task. Hence, we believe a first step to this direction is the definition of a common nomenclature in order to satisfy system correctness and safety requirements and to provide a framework for effective cooperation between experts. Moreover, it is also necessary to look over safety standards to compare the nomenclature use in practice and what is required by the standards.

Need of improving the completeness of requirements specification for safety analysis. Providing the document details assists in getting complete and consistent documents. Absolute completeness may be unnecessary or uneconomical for many situations (Glinz and Fricker, 2015; Leveson, 1995). Hence, the requirements specification must simply be suitable to specify safe behavior in all circumstances in which the system is to operate (Leveson, 1995). Therefore, RE approaches for SCS need to provide a significantly improved account of safety engineering concerns (Sikora et al., 2012). We noticed that UML profiles and modeling languages have been used aiming to improve these specifications.

Compliance with safety standards. Different standards have to be fulfilled by the companies. Nevertheless, RE approaches do not provide explicit guidelines whether they comply with specific safety integrity levels or how the approach should be tailored to achieve compliance (Sikora et al., 2012). Hence, standardization can be seen as a key coordination mechanism, enabling organizations to deal with interdependencies in development work (Pernstål et al., 2015).

Need of improving safety analysis techniques. Available safety analysis techniques are not adequate to establish explicit shared understanding among stakeholders and perform requirements validation and verification. Due to the implicit shared understanding in the system's specifications, too much information about the system is hidden. It might be useful to document which requirements and design decisions have not been documented in detail due to reliance on implicit shared understanding so that this information is not lost when the developed system evolves. Furthermore, it is necessary to improve the safety analysis techniques to handle with implicit shared understanding to allow the safety engineers to perform an adequate safety analysis.

Need of developing and maintaining traceability mechanisms for safety requirements specification. Consistency checking in large safety requirements specification demands different approaches than uncovering ambiguity or checking for testability (Sikora et al., 2012). Furthermore, engineers consider tedious and error prone to deal with large bodies of natural language requirements. Since checking the consistency of the natural language requirements specification must be done manually by means of inspections, this leads to an enormous effort (Sikora et al., 2012).

Need of integration tools. Many artifacts are generated by the requirements and safety teams during the development of SCS. The results of our SLR presented some tools that support safety analysis. Although 66.67% of the studies did not cite any kind of tool support, 27 tools were mentioned in the safety analysis. This out-

come suggests the need of development of tools capable of integrating the requirements specification, safety analysis, and system management maintaining traceability links among all artefacts, models, and the information necessary for the development and certification of SCS.

Need of more integration between researchers and practitioners. The few number of real experiments on integration and communication between RE and safety engineering reveal the need of applying the approaches in practice with real users in order to assess to what extent they contribute to integration and communication between RE and safety engineering. This fact is corroborated by the rigor and industrial relevance we performed. The results of our analysis showed that more than half of the studies have 0 relevance meaning that they are examples of application of a proposal done by either students or researchers in academia in toy examples. Such results highlight the need of more integration between researchers and practitioners in order to improve the relevance of the research.

In the next section, we suggest further research on the integration between requirements engineering and safety engineering in the development of SCS.

5.1. Further research

The results of this SLR showed that although there are some approaches to improve the integration and communication between RE and safety engineering of safety-critical systems, many problems still remain since many studies do not support the real needs of the industry. Therefore, this SLR has generated some research directions to be explored:

(1) What is the core set of activities to be performed by requirements engineers in safety analysis (**RQ1.1**)?
(2) How to evaluate the implicit shared understanding in the safety analysis techniques (**RQ1.2**)?
(3) What is the core set of safety-related information that should be specified by requirements engineers in the development of safety-critical systems (**RQ1.3**)?
(4) In what extent are the tools used in the requirements specification capable of improving integration and communication between RE and safety engineering and safety analysis (**RQ1.4**)?
(5) How to measure the costs and benefits of improving the integration and communication between RE and safety engineering in safety-critical systems (**RQ2**)?

Acknowledgements

This work was partially supported by FACEPE (Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco) and by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.

Appendix A. List of papers and quality scores

Table A.11
List of papers included in the review along with their quality scores and number of citations.

ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Total Score	Qual.	Citations
(Kaiser et al., 2010)	1	1							0.5	0.5		1	0			1					5	71.4%	6
(Saeed et al., 1995)	1	1							0.5	0.5		1	0			1					5	71.4%	14
(David et al., 2010)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	78
(Mostert and von Solms, 1994)	1	1							0.5	0.5		1	0			1					5	71.4%	9
(Lutz, 1993)	1	1							0.5	0.5		1	1			1					6.0	85.7%	151
(Ratan et al., 1996)	1	1							0.5	0.5		1	0			1					5	71.4%	24
(Thramboulidis and Scholz, 2010)	1	1							0.5	0.5		1	1			1					6.0	85.7%	17
(Black and Koopman, 2008)	1	1							1	0.5		1	1			1					6.5	92.9%	2
(Navarro et al., 2006)	1	1							0.5	0.5		1	1			1					6.0	85.7%	3
(Galvao Martins and De Oliveira, 2014)	1		1	0	1	0	1	1	1	0.5		1	1	1	0.5	1					11	78.57%	4
(Kim and Chung, 2005)	1	1							0.5	0.5		1	0			1					5	71.4%	14
(Mannering et al., 2008)	1	1							1	0.5		1	1			1					6.5	92.9%	16
(Medikonda and Panchumarthy, 2009)	1	1							0.5	0.5		1	0			1					5	71.4%	10
(Wu and Kelly, 2007)	1	1							1	0.5		1	1			1					6.5	92.9%	12
(Nejati et al., 2012)	1	1	1		1			1	1	1		1	1	1	0.5	1					11.5	95.83%	26
(Martin-Guillerez et al., 2010)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	21
(Leveson, 2002)	1	1							0	0.5		1	0			1					4.5	64.3%	12
(Stålhane and Sindre, 2014)	1		1	1	0	1	1	1	1	1		1	1	1	0.5	0.5					12	85.71%	0
(Hansen et al., 1998)	1	1							0.5	0.5		1	1			1					6.0	85.7%	137
(Scholz and Thramboulidis, 2013)	1	1							0.5	0.5		1	1			1					6.0	85.7%	1
(Markovski and van de Mortel-Fronczak, 2012)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	0
(Beckers et al., 2013)	1	1							0.5	0.5		1	1			1					6.0	85.7%	5
(Arogundade et al., 2012)	1	1							0.5	0.5		1	1			1					6.0	85.7%	1
(El Ariss et al., 2011)	1	1							0.5	0.5		1	1			1					6.0	85.7%	22
(Guiochet et al., 2010)	1	1							0.5	0.5		1	1			1					6.0	85.7%	20

(*continued on next page*)

Table A.11 (*continued*)

ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Total Score	Qual.	Citations
(Chandrasekaran et al., 2009)	1	1							0.5	0.5		1	0			1					5	71.4%	1
(Briones et al., 2007)	1	1							1	0.5		1	1			1					6.5	92.9%.	4
(Broomfield and Chung, 1997)	1	1							0.5	0.5		1	0			1					5	71.4%	14
(Górski and Wardziński, 1996)	1	1							0.5	0		1	0			1					4.5	64.3%	16
(Du et al., 2014)	1	1							0.5	0		1	0			1					4.5	64.3%	1
(Zoughbi et al., 2011)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	29
(Jrjens, 2003)	1	1							0.5	0.5		1	1			1					6.0	85.7%	52
(Simpson and Stoker, 2002)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%.	0
(Biggs et al., 2016)	1	1							1	1		1	1			1					7	100%	3
(Lu and Halang, 2007)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	18
(Stålhane and Sindre, 2007)	1		1	1	0	1	1	1	1	1		1	1	1	0.5	0.5					12	85.71%	22
(Mustafiz and Kienzle, 2009)	1	1	1		0.5			0.5	1	0.5		1	1	1	0	1					9.5	79.17%	17
(Ekberg et al., 2014)	1	1							0.5	0		1	0			1					4.5	64.3%	0
(Wilikens et al., 1997)	1										1	1				0.5					3.5	87.5%	4
(Paige et al., 2008)	1		1	0	1	0	1	0.5	1	0		1	1	1	0	1					9.5	67.86%	8
(Guillerm et al., 2010)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	8
(Schedl and Winkelbauer, 2008)	1										0.5	1				0.5					3	75%	1
(Rafeh, 2013)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	3
(Chen et al., 2011)	1	1							0.5	0.5		1	1			1					6.0	85.7%	3
(Tschrtz and Schedl, 2010)	1	1							0.5	0.5		1	0			1					5	71.4%	1
(Elliott et al., 1995)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	3
(Croll et al., 1997)	1	1							0.5	0.5		1	0.5			1					5.5	78.6%	6
(Cant et al., 2006)	1	1							1	0.5		1	0			1					5.5	78.6%	3
(Jurkiewicz et al., 2015)	1		1	1	1	1	1	1	1	1		1	1	1	0.5	1					13.5	96.43%	4
(Stålhane et al., 2010)	1		1	1	0	1	1	1	1	1		1	1	1	0.5	0.5					12	85.71%	10
(Murali et al., 2015)	1	1							1	1		1	1			1					7	100%	0
(Pernstål et al., 2015)	1	1	1		1			1	1	1		1	1	1	0.5	1					11.5	95.83%	0
(Fricker et al., 2010)	1	1	1		1			0.5	1	1		1	1	0.5	0.5	1					10.5	87.5%	52
(Fricker et al., 2008)	1	1							1	1		1	1			1					7	100%	24
(Heimdahl, 2007)																	1	1	1	1	4	100%	44
(Sikora et al., 2012)	1	1	1		1			1	1	1		1	1	1	0.5	1					10.5	95.45%	34
(Hatcliff et al., 2014)																	1	1	1	1	4	100%	12
Average																						82.37%	17.58

References

Arogundade, O., Akinwale, A., Jin, Z., Yang, X., 2012. A unified use-misuse case model for capturing and analysing safety and security requirements. Privacy Solutions Secur. FrameworksInf. Prot. 202.

Basili, V.R., Selby, R.W., Hutchens, D., 1986. Experimentation in software engineering. IEEE Trans. Softw. Eng. SE-12 (7), 733–743.

Beckers, K., Heisel, M., Frese, T., Hatebur, D., 2013. A structured and model-based hazard analysis and risk assessment method for automotive systems. In: Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on. IEEE, pp. 238–247.

Biggs, G., Sakamoto, T., Kotoku, T., 2016. A profile and tool for modelling safety information with design information in sysml. Softw. Syst. Model. 15 (1), 147–178.

Black, J., Koopman, P., 2008. Indirect control path analysis and goal coverage strategies for elaborating system safety goals in composite systems. In: Dependable Computing, 2008. PRDC '08. 14th IEEE Pacific Rim International Symposium on, pp. 184–191.

Briones, J.F., De Miguel, M.Á., Silva, J.P., Alonso, A., 2007. Application of safety analyses in model driven development. In: Software Technologies for Embedded and Ubiquitous Systems. Springer, pp. 93–104.

Broomfield, E., Chung, P., 1997. Safety assessment and the software requirements specification. Reliab. Eng. Syst. Saf. 55 (3), 295–309.

Cant, T., Mahony, B., McCarthy, J., Vu, L., 2006. Hierarchical verification environment. In: Proceedings of the 10th Australian Workshop on Safety Critical Systems and Software - Volume 55. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 47–57.

Chandrasekaran, S., Madhumathy, T., Aparna, M., Shilpa Jain, R., 2009. A safety enhancement model of software system for railways. In: Systems Safety 2009. Incorporating the SaRS Annual Conference, 4th IET International Conference on, pp. 1–6.

Chen, D., Johansson, R., Lønn, H., Blom, H., Walker, M., Papadopoulos, Y., Torchiaro, S., Tagliabo, F., Sandberg, A., 2011. Integrated safety and architecture modeling for automotive embedded systems*. e & i Elektrotechnik und Informationstechnik 128 (6), 196–202.

Croll, P., Chambers, C., Bowell, M., Chung, P., 1997. Towards safer industrial computer controlled systems. In: Safe Comp 97. Springer, pp. 321–331.

David, P., Idasiak, V., Kratz, F., 2010. Reliability study of complex physical systems using sysml. Reliab. Eng. Syst. Saf. 95 (4), 431–450.

Dermeval, D., Vilela, J., Bittencourt, I., Castro, J., Isotani, S., Brito, P., Silva, A., 2016. Applications of ontologies in requirements engineering: a systematic review of the literature. Requirements Eng. 21 (4), 1–33.

Du, J., Wang, J., Feng, X., 2014. A safety requirement elicitation technique of safety-critical system based on scenario. In: Huang, D.-S., Bevilacqua, V., Premaratne, P. (Eds.), Intelligent Computing Theory. Springer International Publishing, pp. 127–136.

Easterbrook, S., Singer, J., Storey, M.-A., Damian, D., 2008. Selecting empirical methods for software engineering research. In: Shull, F., Singer, J., Sjberg, D. (Eds.), Guide to Advanced Empirical Software Engineering. Springer London, pp. 285–311.

Ekberg, J., Ingelsson, U., Lønn, H., Skoog, M., Søderberg, J., 2014. Collaborative development of safety-critical automotive systems: exchange, views and metrics. In: Computer Safety, Reliability, and Security. Springer, pp. 55–62.

El Ariss, O., Xu, D., Wong, W., 2011. Integrating safety analysis with functional modeling. IEEE Trans. Syst. Man Cybern. Part A 41 (4), 610–624.

Elliott, J., Brooks, S., Hughes, P., Kanuritch, N., 1995. A framework for enhancing the safety process for advanced robotic applications. In: Achievement and Assurance of Safety. Springer, pp. 131–152.

Fenton, N., 1993. How effective are software engineering methods? J. Syst. Softw. 22 (2), 141–146.

Firesmith, D., 2004. Engineering safety requirements, safety constraints, and safety–critical requirements. J. Object Technol. 3 (3), 27–42.

Firesmith, D.G., 2005. A taxonomy of security-related requirements. International Workshop on High Assurance Systems (RHAS'05). Citeseer.

Fricker, S., Gorschek, T., Byman, C., Schmidle, A., 2010. Handshaking with implementation proposals: negotiating requirements understanding. IEEE Softw. (2) 72–80.

Fricker, S., Gorschek, T., Glinz, M., 2008. Goal-oriented requirements communication in new product development. In: Second International Workshop on Software Product Management (IWSPM), pp. 27–34.

Gadelha Queiroz, P.G., Vaccare Braga, R.T., 2014. Development of critical embedded systems using model-driven and product lines techniques: a systematic review. In: Software Components, Architectures and Reuse (SBCARS), 2014 Eighth Brazilian Symposium on. IEEE, pp. 74–83.
Galvao Martins, L., De Oliveira, T., 2014. A case study using a protocol to derive safety functional requirements from fault tree analysis. In: Requirements Engineering Conference (RE), 2014 IEEE 22nd International, pp. 412–419.
Górski, J., Wardziński, A., 1996. Deriving real-time requirements for software from safety analysis. In: Real-Time Systems, 1996., Proceedings of the Eighth Euromicro Workshop on. IEEE, pp. 9–14.
Gasparic, M., Janes, A., 2016. What recommendation systems for software engineering recommend: a systematic literature review. J. Syst. Softw. 113, 101–113.
Glinz, M., Fricker, S.A., 2015. On shared understanding in software engineering: an essay. Comput. Sci.-Res. Dev. 30 (3–4), 363–376.
Guillerm, R., Demmou, H., Sadou, N., 2010. Information model for model driven safety requirements management of complex systems. In: Complex Systems Design & Management. Springer, pp. 99–111.
Guiochet, J., Martin-Guillerez, D., Powell, D., 2010. Experience with model-based user-centered risk assessment for service robots. In: High-Assurance Systems Engineering (HASE), 2010 IEEE 12th International Symposium on. IEEE, pp. 104–113.
Hansen, K.M., Ravn, A.P., Stavridou, V., 1998. From safety analysis to software requirements. Softw. Eng. IEEE Trans. 24 (7), 573–584.
Hatcliff, J., Wassyng, A., Kelly, T., Comar, C., Jones, P., 2014. Certifiably safe software-dependent systems: challenges and directions. In: Proceedings of the on Future of Software Engineering. ACM, pp. 182–200.
Heimdahl, M.P.E., 2007. Safety and software intensive systems: challenges old and new. In: Future of Software Engineering. IEEE Computer Society, pp. 137–152.
Hernandes, E.M., Zamboni, A., Fabbri, S., Thommazo, A.D., 2012. Using gqm and tam to evaluate start - a tool that supports systematic review. CLEI Electron. J. 15 (1), 3–3.
Ivarsson, M., Gorschek, T., 2011. A method for evaluating rigor and industrial relevance of technology evaluations. Empirical Softw. Eng. 16 (3), 365–395.
Jrjens, J., 2003. Developing safety-critical systems with uml. In: Stevens, P., Whittle, J., Booch, G. (Eds.), UML 2003 - The Unified Modeling Language. Modeling Languages and Applications. Springer Berlin Heidelberg, pp. 360–372.
Jurkiewicz, J., Nawrocki, J., Ochodek, M., Gowacki, T., 2015. Hazop-based identification of events in use cases. Empirical Softw. Eng. 20 (1), 82–109.
Kaiser, B., Klaas, V., Schulz, S., Herbst, C., Lascych, P., 2010. Integrating System Modelling with Safety Activities. Springer.
Kim, H.-K., Chung, Y.-K., 2005. Automatic translation form requirements model into use cases modeling on uml. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Lagan, A., Lee, H., Mun, Y., Taniar, D., Tan, C. (Eds.), Computational Science and Its Applications ICCSA 2005. Springer Berlin Heidelberg, pp. 769–777.
Kelly, T., Weaver, R., 2004. The goal structuring notation-a safety argument notation. In: Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases. Citeseer.
Kitchenham, B., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001. Keele University and Durham University Joint Report.
LAPES, 2014. Start ?- state of the art through systematic review tool. Available in http://lapes.dc.ufscar.br/tools/start_tool. Accessed in October, 2013.
Leveson, N., 2011. Engineering a Safer World: Systems Thinking Applied to Safety. Mit Press.
Leveson, N.G., 1995. Safeware: System Safety and Computers. ACM.
Leveson, N.G., 2002. An approach to designing safe embedded software. In: Embedded Software. Springer, pp. 15–29.
Lu, S., Halang, W.A., 2007. A uml profile to model safety-critical embedded real-time control systems. In: Contributions to Ubiquitous Computing. Springer, pp. 197–218.
Lutz, R.R., 1993. Targeting safety-related errors during software requirements analysis. In: Proceedings of the 1st ACM SIGSOFT Symposium on Foundations of Software Engineering. ACM, New York, NY, USA, pp. 99–106.
Lutz, R.R., 2000. Software engineering for safety: a roadmap. In: Proceedings of the Conference on The Future of Software Engineering. ACM, pp. 213–226.
Maiden, N., Minocha, S., Sutcliffe, A., Manuel, D., Ryan, M., 1999. A co-operative scenario based approach to acquisition and validation of system requirements: how exceptions can help!. Interact. Comput. 11 (6), 645–664.
Mannan, M.S., Sachdeva, S., Chen, H., Reyes-Valdes, O., Liu, Y., Laboureur, D.M., 2015. Trends and challenges in process safety. AIChE J. 61 (11), 3558.
Mannering, D., Hall, J., Rapanotti, L., 2008. Safety process improvement with pose and alloy. In: Redmill, F., Anderson, T. (Eds.), Improvements in System Safety. Springer London, pp. 25–41.
Markovski, J., van de Mortel-Fronczak, J., 2012. Modeling for safety in a synthesis–centric systems engineering framework. In: Computer Safety, Reliability, and Security. Springer, pp. 36–49.
Martins, L.E.G., Gorschek, T., 2016. Requirements engineering for safety-critical systems: a systematic literature review. Inf. Softw. Technol. 75, 71–89.
Martin-Guillerez, D., Guiochet, J., Powell, D., Zanon, C., 2010. A uml-based method for risk analysis of human-robot interactions. In: Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems. ACM, pp. 32–41.
Medikonda, B.S., Panchumarthy, S.R., 2009. A framework for software safety in safety-critical systems. SIGSOFT Softw. Eng. Notes 34 (2), 1–9.
Mostert, D., von Solms, S.H., 1994. A methodology to include computer security, safety and resilience requirements as part of the user requirement. Comput. Secur. 13 (4), 349–364.
Murali, R., Ireland, A., Grov, G., 2015. A rigorous approach to combining use case modelling and accident scenarios. In: NASA Formal Methods. Springer, pp. 263–278.
Mustafiz, S., Kienzle, J., 2009. Drep: a requirements engineering process for dependable reactive systems. In: Methods, Models and Tools for Fault Tolerance. Springer, pp. 220–250.
Nair, S., De La Vara, J.L., Sabetzadeh, M., Briand, L., 2014. An extended systematic literature review on provision of evidence for safety certification. Inf. Softw. Technol. 56 (7), 689–717.
Navarro, E., Sanchez, P., Letelier, P., Pastor, J., Ramos, I., 2006. A goal-oriented approach for safety requirements specification. In: Engineering of Computer Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on, pp. 8pp.–326.
Nejati, S., Sabetzadeh, M., Falessi, D., Briand, L., Coq, T., 2012. A sysml-based approach to traceability management and design slicing in support of safety certification: framework, tool support, and case studies. Inf. Softw. Technol. 54 (6), 569–590.
OMG, 2016. Structured assurance case metamodel. URL http://www.omg.org/spec/SACM.
Paige, R.F., Charalambous, R., Ge, X., Brooke, P.J., 2008. Towards agile engineering of high-integrity systems. In: Computer Safety, Reliability, and Security. Springer, pp. 30–43.
Panesar-Walawege, R.K., Sabetzadeh, M., Briand, L., Coq, T., 2010. Characterizing the chain of evidence for software safety cases: a conceptual model based on the iec 61508 standard. In: 2010 Third International Conference on Software Testing, Verification and Validation. IEEE, pp. 335–344.
Pernstål, J., Gorschek, T., Feldt, R., Florén, D., 2015. Requirements communication and balancing in large-scale software-intensive product development. Inf. Softw. Technol. 67, 44–64.
Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. British Computer Society, Swinton, UK, UK, pp. 68–77.
Petticrew, M., Roberts, H., 2008. Systematic Reviews in the Social Sciences: A Practical Guide. John Wiley & Sons.
Rafeh, R., 2013. A proposed approach for safety management in medical software design. J. Med. Syst. 37 (1), 1–5.
Ratan, V., Partridge, K., Reese, J., Leveson, N., 1996. Safety analysis tools for requirements specifications. In: Computer Assurance, 1996. COMPASS '96, Systems Integrity. Software Safety. Process Security. Proceedings of the Eleventh Annual Conference on, pp. 149–160.
Saeed, A., de Lemos, R., Anderson, T., 1995. On the safety analysis of requirements specifications for safety-critical software. {ISA} Trans. 34 (3), 283–295.
Schedl, G., Winkelbauer, W., 2008. Practical ways of improving product safety in industry. In: Improvements in System Safety. Springer, pp. 177–193.
Scholz, S., Thramboulidis, K., 2013. Integration of model-based engineering with system safety analysis. Int. J. Ind. Syst. Eng. 15 (2), 193–215.
Sikora, E., Tenbergen, B., Pohl, K., 2012. Industry needs and research directions in requirements engineering for embedded systems. Requirements Eng. 17 (1), 57–78.
Simpson, A., Stoker, J., 2002. Will it be safe?an approach to engineering safety requirements. In: Components of System Safety. Springer, pp. 140–164.
Sjoberg, D.I., Anda, B., Arisholm, E., Dyba, T., Jorgensen, M., Karahasanovic, A., Koren, E., Vokac, M., 2002. Conducting realistic experiments in software engineering. In: Proceedings of 2002 International Symposium in Empirical Software Engineering, pp. 17–26.
Sparx, 2016. Enterprise architect. URL http://www.sparxsystems.com.au/
Stålhane, T., Sindre, G., 2007. A comparison of two approaches to safety analysis based on use cases. In: Conceptual Modeling-ER 2007. Springer, pp. 423–437.
Stålhane, T., Sindre, G., 2014. An experimental comparison of system diagrams and textual use cases for the identification of safety hazards. Int. J. Inf. Syst.Model. Des. (IJISMD) 5 (1), 1–24.
Stålhane, T., Sindre, G., Du Bousquet, L., 2010. Comparing safety analysis based on sequence diagrams and textual use cases. In: Advanced Information Systems Engineering. Springer, pp. 165–179.
Thramboulidis, K., Scholz, S., 2010. Integrating the 3+ 1 sysml view model with safety engineering. In: Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on. IEEE, pp. 1–8.
Tiwari, S., Gupta, A., 2015. A systematic literature review of use case specifications research. Inf. Softw. Technol. 67, 128–158.
Tschrtz, H., Schedl, G., 2010. An integrated project management life cycle supporting system safety. In: Dale, C., Anderson, T. (Eds.), Making Systems Safer. Springer London, pp. 71–83.
Unterkalmsteiner, M., Feldt, R., Gorschek, T., 2014. A taxonomy for requirements engineering and software test alignment. ACM Trans. Softw. Eng.Methodol. (TOSEM) 23 (2), 16.
de la Vara, J.L., Panesar-Walawege, R.K., 2013. Safetymet: a metamodel for safety standards. In: International Conference on Model Driven Engineering Languages and Systems. Springer, pp. 69–86.
de la Vara, J.L., Ruiz, A., Attwood, K., Espinoza, H., Panesar-Walawege, R.K., López, Á., del Río, I., Kelly, T., 2016. Model-based specification of safety compliance needs for critical systems: a holistic generic metamodel. Inf. Softw. Technol. 72, 16–30.

Vilela, J., Castro, J., Martins, L. E. G., Gorschek, T., Requirements communication in safety-critical systems: a systematic literature review. Under Submission. For a copy: jffv@cin.ufpe.br.

Whitehead, J., 2007. Collaboration in software engineering: a roadmap.. FOSE 7 (2007), 214–225.

Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Eng. 11 (1), 102–107.

Wilikens, M., Masera, M., Vallero, D., 1997. Integration of safety requirements in the initial phases of the project lifecycle of hardware/software systems. In: Safe Comp 97. Springer, pp. 83–97.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell, MA, USA.

Wu, W., Kelly, T., 2007. Towards evidence-based architectural design for safety–critical software applications. In: Architecting Dependable Systems IV. Springer, pp. 383–408.

Zoughbi, G., Briand, L., Labiche, Y., 2011. Modeling safety and airworthiness (rtca do-178b) information: conceptual model and uml profile. Softw. Syst. Model. 10 (3), 337–367.

Jéssyka Vilela received a B.S. degree in Computer Engineering from Universidade Federal do Vale do São Francisco (UNIVASF), Brazil in 2012. She completed her Master in Computer Science at the Universidade Federal de Pernambuco (UFPE), Brazil in 2015. Jéssyka is currently a Ph.D. student at UFPE and her research interests include Software Engineering, Requirements Engineering, Software Architecture, Safety-Critical Systems and Context-Sensitive Systems.

Jaelson Castro is a Full Professor at Universidade Federal de Pernambuco, Brazil, where he leads the Requirements Engineering Laboratory (LER). He holds a Ph.D. in Computing from Imperial College, London, UK. His research interests include requirements engineering, adaptive systems, model-driven development and robotics. He serves on the editorial board of the Requirements Engineering Journal and the Journal of Software Engineering Research and Development and acted as Editor-in-Chief of the Journal of the Brazilian Computer Society - JBCS.

Luiz Martins is a professor of software engineering at Federal University of São Paulo (UNIFESP). His research interests include requirements engineering, embedded systems, safetycritical systems and model-driven software development. He has published more than 30 papers in these areas. Martins has a Ph.D. in Electrical Engineering from State University of Campinas (UNICAMP).

Tony Gorschek is a professor of software engineering at Blekinge Institute of Technology, Sweden. He has 10 years of experience working with SW intensive product development in domains ranging from automotive to telecom, working as CTO, chief architect and developer. His research interests include requirements engineering, technology and product management, lean product development, quality assurance, and innovation. Gorschek has a Ph.D. in software engineering from BTH. He's a member of the IEEE and the ACM (visit www.gorschek.com).