# Requirements Communication in Safety-Critical Systems: an extended literature overview

**Jéssyka Vilela** [1], **Jaelson Castro** [1], **Luiz Martins** [2], **Tony Gorschek** [3], **Camilo Almendra** [1,4]

[1]Universidade Federal de Pernambuco (UFPE), Brazil

[2]Universidade Federal de São Paulo (UNIFESP), Brazil

[3]Blekinge Institute of Technology (BTH), Sweden

[4]Universidade Federal do Ceará (UFC), Brazil

{jffv,jbc}@cin.ufpe.br, legmartins@unifesp.br, tony.gorschek@bth.se, camilo.almendra@ufc.br

***Abstract. Context:*** *Safety-critical systems (SCS) must be carefully planned since inadequate or misunderstood requirements have been recognized as the major cause of safety-related catastrophes.* ***Objective****: We investigate the integration and requirements communication in the requirements engineering (RE) process among different parties when developing SCS.* ***Method****: We used a Systematic Mapping Study as the basis for our work.* ***Results****: We analyze several aspects such as challenges, domain, requirements activity, languages, tools, stakeholders involved, communication format, and safety standards.* ***Conclusions****: This information contributes to setting up possible collaborative networks and as a reference when developing new research projects.*

## 1. Introduction

Safety-critical systems are mainly controlled by software nowadays [Sikora et al. 2012][Hatcliff et al. 2014]. New generations of medical devices, means of transportation (aircraft, automated trains and cars), nuclear power generating stations, banking and investment systems, as well as a growing number of automated systems rely on software to enable new functions, provide pre-existing functions more efficiently, and reduce time to service a user need as well as the effort and competence required by people providing services.

There are many cases in the literature [Leveson 2011] where inadequate or misunderstood requirements were the major cause (not coding or implementation) of a significant proportion of accidents and safety-related catastrophes. Therefore, these systems must be carefully specified, demanding more rigorous RE approaches [Leveson 2011].

RE focuses on good specification practices but has yet to find working solutions for effective requirements communication. Furthermore, the competences of requirements engineers and safety engineers normally work independently of each other and have inherently different tools and engineering practices - resulting in a lack of coordination that can compromise the quality of safety analysis and safety specifications [Vilela et al. 2017].

In this work, we investigate the approaches proposed to improve the integration of requirements communication in the RE process among different parties when developing

SCS. We adopted the systematic mapping study as a research method. We believe the results of such novel study will benefit both researchers and practitioners. The review will provide researchers with important research gaps regarding the requirements communication between safety and RE. For the industrial readership, the review will provide practitioners with useful information about the state-of-the-art and advances so far. This information contributes to setting up possible collaborative networks and as a reference when developing new research projects.

This paper is organized as follows. Section 2 presents background and related work. The research methodology adopted to conduct the mapping study 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

SCS are those software or system operations that, if not performed, performed out of sequence, or performed incorrectly could result in improper control functions, or lack of control functions required for proper system operation. Such problems can directly or indirectly cause or allow a hazardous condition to exist [Leveson 2011].

In order to set the scope and make clear the adopted definition of requirements communication used in this mapping study, and to ensure consistency throughout this paper, we discuss this concept in the next section.

### 2.1. Requirements Communication

Requirements communication is a traversal process of exchanging information [Glinz and Fricker 2015] about the requirements among all stakeholders [Bjarnason et al. 2011] involved in the system lifecycle. This concept does not only comprise the communication itself but the specification and analysis of all artifacts involved in the RE process. Since changes occur throughout the project, requirements communication must also continue during the entire life cycle [Bjarnason et al. 2011].

This process aims to achieve a shared understanding [Glinz and Fricker 2015] of the system's requirements to increase completeness and correctness of the requirements specifications. It encompasses all the activities needed to inform the stakeholders of the content, meaning and status of requirements. The elicited requirements need to be communicated, and changes to those requirements negotiated and communicated among all affected roles, e.g. requirements engineers, developers, and testers [Bjarnason et al. 2011].

### 2.2. Related Work

The communication of requirements among different parties in the development of SCS is critical for the quality of the system. This occurs since requirements should be understood in the same way by different roles in the development. We argue that the requirements engineers and safety engineers should collaborate, exchange information and work jointly and in an iterative way. However, they usually work independently of each other and have inherently different tools and engineering practices - resulting in a lack of coordination that can compromise the quality of safety analysis [Wang et al. 2018], and therefore, the quality of safety specifications.

Communication problems in software development were investigated by some authors such as Brady et al. [Brady et al. 2007], Pernstal [Pernstal 2015], Rasmussen and Lundell [Rasmussen and Lundell 2012], Wang et al. [Wang et al. 2018] as well as Nakamura et al. [Nakamura et al. 2016]. Although these works explore several challenges related to the integration of RE and safety, little has been done to date to perform an extensive identification and mapping, in a comprehensive manner, the state-of-the-art on the communication of requirements among different parties in the development activities/process when developing SCS. Hence, to the best of our knowledge, this is the first mapping study with such specific focus. In the next section, we detail our research protocol.

## 3. Research Methodology

In this section, we present the design and the execution of the mapping study. The research methodology used was based on the guidelines and template proposed by Kitchenham and Charters [Kitchenham and Charters 2007].

The focus of this review is the integration between RE and safety engineering and the requirements communication among different parties during the RE process. We included only English primary studies, published in any year until February 2018, that address in their objectives the communication in the RE process among different parties when developing SCS, related Requirements and Safety in the context of RE process, or covered Design in the relationship with Requirements and Safety.

We excluded Secondary studies, Short-papers ($\leq$ 3 pages), Duplicated studies, Studies clearly irrelevant to the research, taking into account the research questions, Gray literature, Redundant paper of same authorship, Publications whose text was not available (through search engines or by contacting the authors), and Studies whose focus was not the communication in the RE among different parties when developing SCS or safety requirements specification.

Our study was guided by the research questions presented in Table 1. The search strategy included two types of search to find studies relevant to the scope of the review. The first type was an automatic search, using a string validated by experts on RE and SCS. The second strategy was the manual inclusion of papers well-known about requirements communication.

We developed a review protocol in which the main elements are as follows: the ***se-***

**Tabela 1. Research questions.**

| **Research Question** |
|---|
| RQ1. What challenges have been identified pertaining to the communication among engineers during the RE process when developing SCS? |
| RQ2. Which approaches have been proposed to improve the communication in the RE process among engineers when developing SCS? |
| RQ2.1. What are the types of these approaches? |
| RQ2.2. For which domains were these approaches proposed? |
| RQ2.3. What RE activities were supported by these approaches? |
| RQ2.4. Which requirements specification languages are used by these approaches? |
| RQ2.5. Which tools are used for the requirements specification? |
| RQ2.6. For which stakeholder were they proposed? |
| RQ2.7. What are the communication formats used? |
| RQ2.8. For what safety standards have the approaches been proposed? |

***lected resources*** chosen were Science Direct, SpringerLink, ACM Digital Library, IEEE Xplore, Scopus, and Compendex; the ***search method*** used web search engines; the ***population*** was composed of peer-reviewed publications reporting approaches to improve the communication in the RE process among parties when developing SCS; the aim of the ***intervention*** was to collect empirical evidence in relation to the research questions.

We developed the search string by specifying the main terms of the phenomena under investigation (SCS and requirements communication). After several iterations, we defined the search string below 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*
**(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")*

Figure 1 depicts the steps of the selection process showing the number of studies in each step. The data were extracted from the 60 primary studies using an extraction form fully aided by the StArt tool.

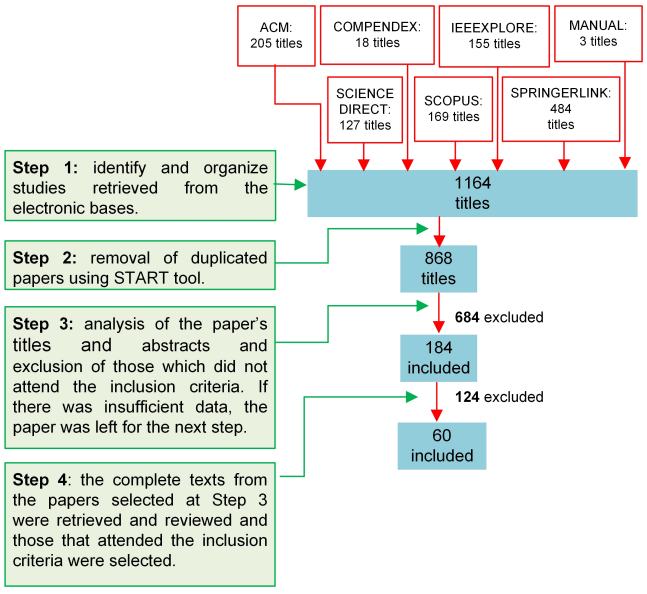


**Figura 1. Paper selection flowchart.**

The quality assessment (QA) of selected studies was achieved by a scoring technique to evaluate selected studies in terms of credibility, completeness and relevance. All papers were evaluated against a set of 20 quality criteria whose assessment instrument we developed and used in a previous work [Vilela et al. 2017] and described in the supplementary material.

## 4. Results and Analysis

A total of 60 studies satisfied the inclusion criteria and their data were extracted. The quality scores of the selected studies are presented in Table S1 on supplementary material [1]. The mean of quality was 83.12%, hence, the overall quality of the selected studies is acceptable.

### 4.1. Overview of the Studies

The reviewed papers were published between 1994 and February 2018. From a temporal point of view (Figure 2), we can notice that the number of studies about requirements communication in SCS is low over the years. Despite the apparently increasing number of studies on this topic (peak in 2009-2011), this result corroborates the statement that the collaboration of safety analysis and RE has been somewhat neglected [Leveson 2011]. It is also worth noting that, as the search process of this review was performed in February 2018, a slight decrease in the number of publications would be expected in 2018 because some papers might have been in press.

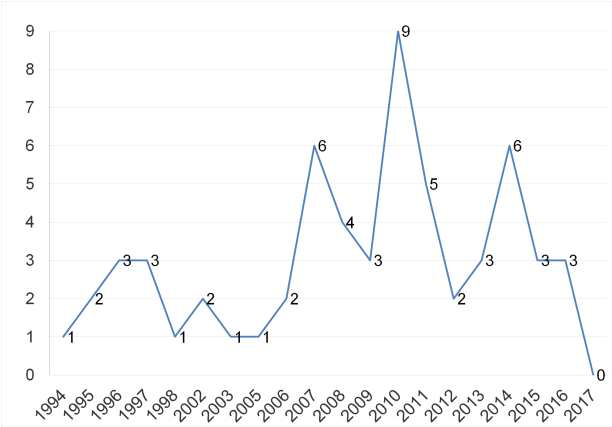


**Figura 2. Temporal view of the studies.**

Figure 3 presents a bubble plot distributed over three dimensions regarding three characteristics of the studies: evaluation method, research type and application context (academic, industrial or both). The left part in this figure denotes the relationship between the research type of the studies and their evaluation method. The number in a bubble represents the number of studies that present both characteristics. On the other hand, in the right part of this figure, the number in a bubble represents the number on a specific research type in a certain application context.

The results of each research question are presented and discussed in the next sections.

### 4.2. RQ1: What challenges have been identified pertaining to the communication among engineers during the RE process when developing SCS?

The selected studies point out many challenges as listed in Section 2 supplement material. In this section, we also discuss the details the elicited challenges.

[1]Available at: `www.cin.ufpe.br/~jffv/papers/wer2019`

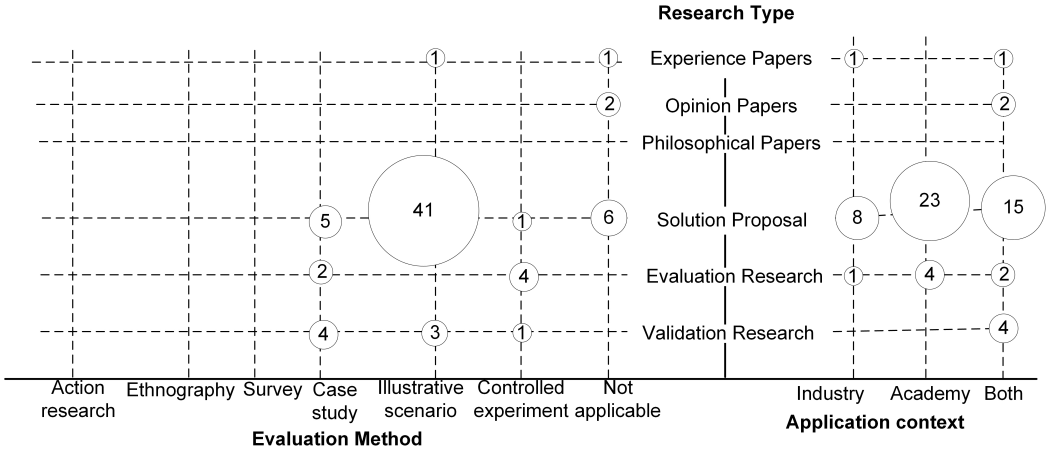


**Figura 3. Bubble plot with application context, research type and evaluation method dimensions.**

Many challenges of requirements communication are related to the concept of shared understanding [Glinz and Fricker 2015]. Shared understanding among a group of people has two facets: explicit shared understanding is about interpreting explicit specifications, such as requirements, design documents, and manuals, in the same way by all group members. Implicit shared understanding denotes the common understanding of non-specified knowledge, assumptions, opinions, and values. The shared context provided by implicit shared understanding reduces the need for explicit communication and, at the same time, lowers the risk of misunderstandings.

### 4.3. RQ2: Which approaches have been proposed to improve the communication in the RE process among engineers when developing safety-critical systems?

This research question was divided into eight sub research questions (RQ2.1 to RQ2.8) aiming to analyze many aspects of requirements communication of SCS.

### 4.4. RQ2.1: What are the types of these approaches?

The contribution types are reported considering the classification presented in the selected studies. The final list of contribution types are presented in Figure 4. Note that, similarly to other research questions, this question also allows a study to be included in more than one category.

The results suggest that the majority of studies are concerned with proposing some kind of *Model*, *Tool*, *Process*, *Approach*, *Method*, and *Template*. These different types of contributions may be an indication that not all artifacts types are equally suited for all activities in software and RE. Moreover, several persons with various roles and different requests use artifacts based on their individual work throughout the project [Liskin 2015]. This high number of approaches in these categories may also suggest that how requirements engineers should perform the safety activities is still an open research question.

### 4.5. RQ2.2: For which domains were these approaches proposed?

Figure 5 shows the distribution of the studies by application domain. 78.3% of the studies were classified as domain-independent, the remainder of the studies were developed in

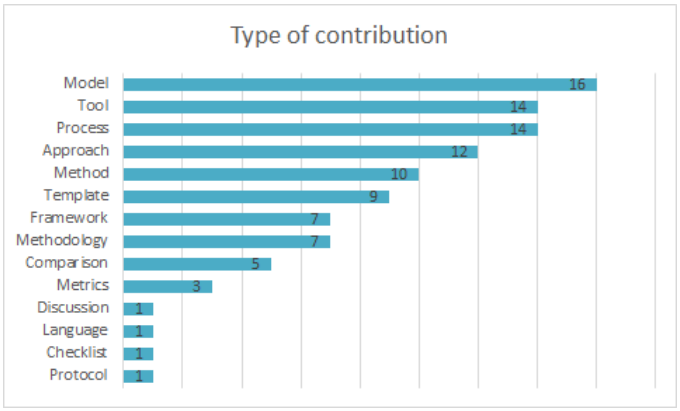


**Figura 4. Type of contributions on requirements communication of SCS.**

the following application domains: robotics, automotive, avionics, medical, railway, and mechatronics.

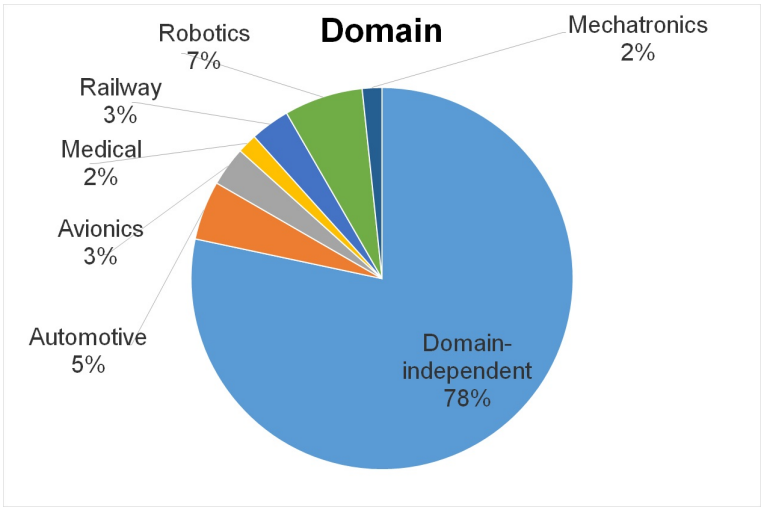


**Figura 5. Application Domain.**

In 47 studies (78.3%), the authors mentioned or gave some indication that the proposed approaches could be used in different safety-critical domains. We adopted the strategy used by Martins and Gorschek [Martins and Gorschek 2016] of classifying them as domain-independent. From these results, we can suggest that there is some tendency of improving requirements communication with a common body of knowledge in safety analysis. In a previous work [Vilela et al. 2018], we have proposed a domain independent metamodel called Safe-RE which is a Safety Requirements Metamodel Based on Industry Safety Standards.

Considering specific application domains, robotics was the domain where more approaches for requirements communication were proposed (7%). It was a surprise since we are expecting more contributions in traditional fields such as avionics or automotive. This outcome suggests that safety is a concern in different areas and requires the collaboration of multiple professionals [Leveson 2011][Martins and Gorschek 2016][Sikora et al. 2012][Hatcliff et al. 2014].

Analyzing the publication year of the 13 domain-specific approaches, we noticed that 69.23% are recent contributions (published in 2010 or after). This may suggest that the model-driven architecture and the domain specific languages as well as the frameworks for model-driven development might be influencing the approaches for considering domain-specific concerns.

Many standards require that a safety analysis is performed when developing or modifying a SCS. Hence, another factor influencing these results may be for what safety standards have the approaches been proposed (see RQ2.8) since many standards are domain-specific.

### 4.6. RQ2.3: What RE activities were supported by these approaches?

We categorized these activities according to the main steps of a RE process: elicitation, analysis and negotiation, specification, validation and management[2] (see Table 2). In summary, the results suggest that all RE activities are covered by the studies.

**Tabela 2. RE activities supported by the approaches in requirements communication.**

| Activity | Count | % |
|---|---|---|
| Elicitation | 13 | 21.67% |
| Analysis and Negotiation | 31 | 51.67% |
| Specification | 38 | 63.33% |
| Validation | 28 | 46.67% |
| Management | 12 | 20% |

In summary, the results suggest that the studies cover all RE activities. The *Specification* activity is addressed by more than 60% of the studies. In fact, to some extent, this result was expected, since SCS are submitted to certification processes, and many of them must have to be compliant with some safety standard.

*Analysis and Negotiation* activities also encompass a high number of studies (51.67%). This result is also interesting because we argue that RE and safety engineers should collaborate in this activity to conduct better safety analysis and better safety requirements specifications. Furthermore, 28 studies (46.67%of papers included) covered both Specification and Analysis and Negotiation activities in the same paper, indicating that almost all studies, except [Chandrasekaran et al. 2009, Stålhane and Sindre 2007, Stålhane et al. 2010], that covered the Analysis and Negotiation are also concerned with the documentation of such analysis.

A significant number of studies addressed the Validation activity. This supports the need for certification that requires the construction of safety cases. Some studies (20%) covered the Management activity, indicating that it is important to manage the requirements and apply traceability mechanisms to improve the requirements communication.

There are also studies that aim to provide guidance for Elicitation activities. For instance, there are approaches that aim to represent safety knowledge to drive elicitation tasks. It is worth noting that the majority of the papers which addressed Elicitation activity also included Analysis and Negotiation and Specification activities.

[2]In this paper, we considered requirements management as a requirements activity.

**Tabela 3. Requirements specification languages per domain.**

| RE language | Count |
|---|---|
| Natural Language | 27 |
| Use Cases Description | 20 |
| UML | 16 |
| Block diagram design language, and State machine design language, SysML | 8 (each) |
| Context Diagram | 6 |
| Logics, and Formal methods | 5 (each) |
| Mathematical notations | 3 |
| Problem Frames, and Event Time Diagram (ETD) | 2(each) |
| KAOS, RSML language, SpecTRM- RL modeling language, System Diagrams, HIVE requirements language, Goal Model of ATRIUM, Alloy, VDM++, Structured Analysis and Design Technique (SADT), Event-B, EAST-ADL, Bayesian Belief Networks (BBNs), ALTARICA, AUTOSAR, User Requirements Notation (URN), Requirements Definition and Analysis Language (RDAL), and Architecture Analysis and Design Language (AADL) | 1 (each) |

A research direction to enable continuous communication could be by making sure cross-competence teams work jointly on the most dominant group of activities such as elicitation and specification or specification and validation. However, assuring these competencies is a challenge to managers since other variables can influence the collaborative work.

Finally, among all 60 studies, only five papers address all activities of the RE process and only one study [Heimdahl 2007] proposed an entire RE process. This may be an indication that a holistic approach to improving the requirements communication that supports all activities of the RE process is needed. This may be one of the reasons for so many problems and challenges faced (see RQ1) in the development of SCS.

### 4.7. RQ2.4: Which requirements specification languages are used by these approaches?

The languages used by the studies to specify the requirements are listed in Table 3. Furthermore, five papers did not cite any language. We identified a great variety of requirements specification languages adopted by the approaches. *Textual requirements*, written in natural language, are the most frequent type of RE specification language addressed by the studies. This category included studies on all analyzed domains, but the majority of studies are domain-independent as can be seen in Table 3. It also encompasses studies that specify requirements documents using a requirements document template. In the supplement material (Section 3.2), we discuss the pros and cons of the identified languages.In the supplement material (Section 3.2), we discuss the pros and cons of the identified languages.

Although *Textual requirements* category is the dominant documentation style for requirements, it does not imply that natural language is considered a satisfactory specification technique [Sikora et al. 2012]. The qualitative study conducted by Sikora et al. [Sikora et al. 2012] showed that many embedded systems practitioners are dissatisfied with the use of natural language for requirements specification. In their study, practitioners revealed that they consider tedious and error-prone to deal with large bodies of natural language requirements.

For example, Sikora et al. [Sikora et al. 2012] frequently noted that checking the consistency of the natural language requirements specification must be done manually by means of inspections, which leads to an enormous effort. Furthermore, the requirements

specification must precisely define the physical process to be controlled as well as the requirements for the controller since embedded systems perform control tasks. Therefore, according to the practitioners, there is a need for approaches that support the specification of controller requirements because the use of natural language allows too many interpretations, whereas controller specifications are written as pseudo-code preempt the controller design [Sikora et al. 2012].

The natural language is wonderfully expressive, but frequently ambiguous [Whitehead 2007]. Furthermore, natural language is, generally, unsuitable to the capture of the rigorous arguments needed for a safety case. Recent work on Goal Structuring Notation (GSN) [Heimdahl 2007] is a direction that promises to assist in the construction of rigorous arguments. However, it should be not the only specification language used in the development of large systems such as the SCS.

The *Use Case Description* category is also popular in the selected studies. It encompasses 17 papers and is adopted mainly in domain-independent approaches. The *UML* category includes studies that adopted any diagram defined by this language, for instance, use case diagrams, activity diagrams, sequence diagrams, and class diagrams (statecharts were included in the state machine design language).

*UML* is a language used by many academic and industrial works. *UML* is also mainly adopted in domain-independent approaches. However, it imposes several constraints for modeling systems that are not limited only to software [Scholz and Thramboulidis 2013]. Being a software-oriented language, representing other aspects of systems (e.g., hardware, mechanics, and electronics), is more complicated than in a system-oriented language such as SysML. The development of SCS requires the modeling of the entire system, not just their software-related aspects, as requested by many international safety standards [Biggs et al. 2014].

*Block diagram design language* category is another language frequently used since many studies target at increasing the reliability of system components or the dependability rather than the safety of a system. However, although reliability is important for achieving safety, making a system more reliable is not sufficient if it has unsafe functions [Medikonda and Panchumarthy 2009]; and threats to safety are not limited to failing components. According to Biggs et al. [Biggs et al. 2014], when targeting dependability, it is important to model properties such as performance, possible faults, and maintenance, but they do not provide for important safety information such as hazards (other than faults) and potential harm.

*State machine design language* category is adopted by eight studies and is only adopted in domain-independent approaches. This language can model the behavior of a large number of problems through states, the possible input events, and the possible actions or output events that result from state transition.

The *Systems Modeling Language (SysML)* is used in seven studies in domain-independent approaches and one study in the Mechatronics domain. This outcome is expected since this language is a general-purpose visual modeling language for systems engineering applications. SysML is defined as a dialect of the UML standard and supports the specification, analysis, design, verification, and validation of a broad range of systems and systems-of-systems. This system may include hardware, software, information, pro-

cesses, personnel, and facilities.

In the study of Sikora et al. [Sikora et al. 2012], UML/SysML models are the most commonly used types of models in RE for embedded systems. They are used to describe structural aspects of the system and its environments, such as a power plant, a pumping station, a vehicle, or an airplane. Although GSN has been used for the construction of safety cases in some segments, the authors [Sikora et al. 2012] describe that goal models are rarely used.

An interesting question pointed out by Sikora et al. [Sikora et al. 2012] is that even when a model type has a potential benefit for the development project, it tends to be used less frequently, if its use is not mandatory, e.g., by project constraints or intra-organizational regulations.

Considering specific-domain approaches, we noticed that in the automotive and mechatronics domains, there is a variety of RE languages adopted. This may suggest that there is no consensus, and the researchers are trying to find which language is the most adequate for this domain.

These results seem to indicate the prevalence in the selected studies of informal and semi-formal approaches over formal ones in order to document and communicate requirements, design decisions, and relevant information among the project teams and actors in the development and certification of SCS as also concluded by [Martins and Gorschek 2016].

Many specification languages were cited by only one selected study. This suggests that many of them are not acceptable to all stakeholders involved in theRE process and, as a consequence, such languages not get used. According to Heimdahl [Heimdahl 2007], without a modeling language acceptable to all stakeholders, the language will not get used, and all research into formal techniques will not make them into software engineering practice. This result shows that a good specification language for the SCS domain should have a well-defined graphic notation to avoid misunderstandings as well as formal reasoning, preferably embedded in the language, to improve the consistency checks and the quality of the requirements specifications.

### 4.8. RQ2.5: Which tools are used for the requirements specification?

We believe that requirements communication is also improved by the use of shared tools, hence this question maps the ones used to develop the requirements specification of SCS. Table 4 lists the tools mentioned more than once in the selected studies.

**Tabela 4. Tools used in the requirements specification.**

| Tool | Count | % |
|---|---|---|
| It does not cite | 36 | 60% |
| A proposed one | 10 | 16.67% |
| Sparx Systems Enterprise Architect | 6 | 10% |
| ARTi-SAN Studio, DOORS, SystemWeaver, mCRL2, Rodin platform | 2 (each) | 3.33% |
| IBM Rational Software Architect, IBM Rational Rhapsody, IBM Rational Harmony for Embedded Real-Time Development tool, HIVE (Hierarchical Verification Environment) tool, Siemens Teamcenter Systems Engineering and Requirements Management, Elektra, Spreadsheet tool, Visio, SafeSlice, EATOP, Artop, Supremica, TCT, NBC, UPPAAL, UML4PF, Papyrus UML, ERRSYS, SRSV, OSATE, and jUCMNav | 1 (each) | 1.67% |

The majority of the studies (36 studies - 60%) did not mention

any tool for requirements specification. On the other hand, some approaches used or recommended different tools such as Markovski and Mortel-Fronczak [Markovski and van de Mortel-Fronczak 2012] as well as Pernstal et al. [Pernstål et al. 2015]. This lack of tools is a substantial issue since they can contribute to the requirements communication and should consider safety concerns to improve shared understanding.

Other works (ten studies - 16.67%) report that they developed a tool to support their proposals but they did not present their names. The results might indicate that the tools are not adapted for SCS or to enable communication in large teams. Most tools are expensive per license and this forces companies to buy few licenses, limiting access to the central repository and thus hindering communication. Perhaps, the use of no tool (or using internal ones like excel) is a reaction towards the expensive licenses.

Table 4 present the tools cited in the included studies. However, many of them did not explicitly discuss how the tools support the communication throughout the RE activities (*elicitation, analysis, specification, validation, management*). As we described in Section 4.10, some studies adopt analysis tools as a form to improve the requirements communication. This category includes tools shared by stakeholders involved in the requirements specification and safety analysis as well as tools that support some kind of safety analysis.

### 4.9. RQ2.6: For which stakeholder were they proposed?

The stakeholders mentioned in the selected studies are listed in Table 5. The majority of the approaches were designed to be used by safety engineers and developers.

**Tabela 5. Stakeholders involved in the approaches.**

| Stakeholder | Count | % |
|---|---|---|
| Safety Engineer | 29 | 48.33% |
| Developer | 23 | 38.33% |
| Software Engineer | 19 | 31.67% |
| Requirements Engineer | 18 | 30% |
| Design Engineer | 10 | 16.67% |
| Architect | 9 | 15% |
| Customer | 7 | 11.67% |
| System Engineer | 6 | 10% |
| Certification authorities and Project manager | 3 (each) | 5% |
| Human factors expert, Manufacturing (MAN), and Product development (PD) | 2 (each) | 3.33% |
| Supplier, Test engineer, Quality Manager, Cognitive engineer, Operator, Constraints Engineer, Domain Engineer, and Reliability Engineer | 1 (each) | 1.67% |

The results presented in Table 5 suggest that, as expected, *safety engineers* are the stakeholders for which most studies have been proposed. The next most cited stakeholders in the selected studies were *Developer*, *Software Engineer*, and *Requirements Engineer*. This outcome might indicate that there is some confusion in the selected studies, perhaps not in the industry, of their roles and the division of attributions is not clearly defined.

These results show that the proposed approaches embrace different types of stakeholders involved in the development of SCS. Such approaches could help them in different but complementary activities, such as safety analysis and safety requirements specification as also concluded by Martins and Gorschek [Martins and Gorschek 2016]. Furthermore, the main responsibility of safety analysis is from safety engineers. However, it is advisable when it is shared with requirements engineers and stakeholders.

Moreover, there is a tendency of sharing the responsibility of safety analysis conduction by all these stakeholders mentioned above.

### 4.10. RQ2.7: What are the communication formats used?

We based our analysis on the work of Jim Whitehead [Whitehead 2007] that classifies the collaboration tools as *Model-based, artifacts-based, Process support, Awareness*, and *Collaboration infrastructure* in a roadmap about collaboration in software engineering. We complemented such classification with *Analysis tools*, and *Face-to-face verbal communication* categories according to the formats presented in the selected studies. Table 6 lists the communication formats used in the approaches.

**Tabela 6. Communication format used in the approaches.**

| Communication Format | Count | % |
|---|---|---|
| Model-based collaboration | 42 | 70% |
| Process support | 26 | 43.33% |
| Artifacts-based | 21 | 35% |
| Analysis tools | 19 | 31.67% |
| Face-to-face verbal communication | 4 | 6.67% |
| Collaboration infrastructure | 3 | 5% |
| Awareness | 2 | 3.33% |

The *model-based collaboration* was used by 70% of the selected studies (42 studies). Hence, there is a tendency, in the selected studies, of using models to improve the requirements communication in SCS. Model-based specifications are consistent and less ambiguous than informal specification documents, forcing the stakeholders to make clear all aspects of the system early in the design process. Therefore, models provide a shared meaning that engineers use when coordinating their work, as when stakeholders consult a requirements specification to determine how to design a portion of the system or to perform the safety analysis.

The *Process support* was the second communication format most adopted by the approaches (26 studies - 43.33%). It consists of collaborating through on a predefined structure for the sequence of steps to be performed, the roles stakeholders must fulfill, and the artifacts that must be created. This communication format serves to reduce the amount of coordination required to initiate a project and to define the typical sequence of steps that should be followed in the development and the roles and artifacts that should be produced.

As we already discussed in Section 4.4, the development of a safety-critical system involves the creation of multiple *artifacts*. We considered as an artifact all documents that were not based on models. Each type has its own semantics and creating it is an inherently collaborative activity. Several stakeholders contribute to each of these artifacts, working to understand what each other has done, eliminate errors, and add their contributions [Whitehead 2007]. This communication format was used by 21 studies (35%).

*Analysis tools* were adopted as a communication format by 19 studies (31.67%). This category comprehends tools shared by stakeholders involved in the requirements specification and safety analysis as well as tools that support some kind of safety analysis.

The *Face-to-face verbal communication*, used by [Paige et al. 2008][Pernstål et al. 2015][Fricker et al. 2010][Fricker et al. 2008] (four

studies - 6.67%), includes meetings, informal conversations in hallways, doorways, and offices. However, the conversations are not formally structured, despite being the concern the development of a formal system, a piece of software, [Whitehead 2007].

The *Collaboration infrastructure* category comprises the technologies developed for the integration of software tools. Main forms of tool integration include data integration, ensuring that tools can exchange data, and control integration, ensuring that tools are aware of the activities of other tools, and can take action based on that knowledge [Whitehead 2007]. Only the studies [Beckers et al. 2013][Briones et al. 2007][Heimdahl 2007] (three studies - 5%) explicitly discussed this communication format.

The *Awareness* communication format was only explicitly used in [Medikonda and Panchumarthy 2009][Schedl and Winkelbauer 2008] (two studies - 3.33%). It consists of providing information about the current activities of other stakeholders. By increasing this awareness of the activities, the stakeholders are able to perform coordination activities sooner, and potentially avoid conflicts [Whitehead 2007]. Furthermore, it is important to highlight that the other communication formats also contribute to improving the awareness but it was not the specific objective of the studies.

### 4.11. RQ2.8: For what safety standards have the approaches been proposed?

The safety standards presented in the approaches are exhibited in Table 7. Table 7 shows that the great majority of the approaches of requirements communication (46 studies - 76.67%) *does do not follow* any safety standard.

**Tabela 7. Safety standards adopted.**

| Safety Standard | Year | Domain | Count | % |
|---|---|---|---|---|
| No | | | 46 | 76.67% |
| IEC 61508 | 2010 | Generic | 4 | 6.67% |
| ISO 26262 | 2011 | Automotive | 3 | 5% |
| DO-178B | 1992 | Avionics | 1 | 1.67% |
| ISO/IEC 15504 | 2003 | Generic | 1 | 1.67% |
| ISO 12207 | 1995 | Generic | 1 | 1.67% |
| ISO 12100 | 2010 | Machinery | 1 | 1.67% |
| IEC/SC65A | 1992 | Generic | 1 | 1.67% |
| Australian Defence Standard Def (Aust) 5679 | 1998 | Generic | 1 | 1.67% |
| ANSI/RIA R15.06-1999 | 1999 | Robotics | 1 | 1.67% |
| ISO/IEC 9126 | 2001 | Generic | 1 | 1.67% |
| IEC 1508 | 1995 | Generic | 1 | 1.67% |
| IEC 61499 | 2011 | Generic | 1 | 1.67% |
| IEC 61131 | 1993 | Generic | 1 | 1.67% |
| EIA-632 | 1994 | Generic | 1 | 1.67% |

This finding is worrying since SCS should be certified by regulatory bodies and this requires submitting relevant system safety information to appropriate authorities [Zoughbi et al. 2011]. Hence, the communication with certification authorities with regards to system certification requires providing proof that appropriate standards were followed during the development process [Zoughbi et al. 2011].

From the approaches which based their concepts in part on the definitions given by the international standards for safety, we identified fourteen standards (see Table 7). The date of the safety standard release varies between 1992 and 2011. 64.29% of the followed safety standards are developed for general purposes such as defining the safety life cycle,

the requirements for evaluation of the software development process, the terminology and guidelines.

## 5. Conclusions

The RE activities are critical to avoid the introduction of defects and misunderstandings among engineers and developers when developing SCS. Communication among workgroups that develop interdependent pieces of a system is crucial for a successful outcome of software development projects [Pernstal 2015]. This is an important question in the development of SCS considering that many safety problems occur due to errors and misunderstandings in safety requirements specifications.

Our mapping study draws on 60 studies, selected out of 1164, through a multistage process. An important 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 into the state-of-the-art about how the requirements communication is conducted in the RE process. Currently, we are working on the analysis of safety standards and the comparison of the results of the state-of-art with the state-of-practice is in progress.

### 5.1. Threats to validity

We adopted the classification of threats to validity well adopted in the literature which corresponds to Internal, External, Construct and Conclusion categories. *Construct validity*: For all concepts, we used many synonyms to ensure high coverage of potentially-relevant studies from a database search. *Internal validity*: In order to minimize selection and extraction mistakes, the selection process was performed in an iterative way. It is also worth noting that the all authors are lecturers and experienced researchers with expertise in RE, Software Engineering or SCS. *External validity*: In order to mitigate external threats, the search process was defined after several trial searches and validated with the consensus of the authors. *Conclusion validity*: The research protocol was carefully designed and discussed by the authors to minimize the risk of exclusion of relevant studies. It is worth highlighting that we did not restrict the time period of published studies to obtain the maximum coverage possible.

The results of this mapping study showed that although there are some approaches to improve the requirements communication of SCS, several problems still remain since many studies do not support the real needs of the industry. Therefore, this mapping study has generated several promising research directions:

**(1)** How safety analysis techniques can be improved to evaluate shared understanding (**RQ2.1**)?
**(2)** To what extent do the domain-independent approaches cover the needs of domain-specific critical systems (**RQ2.2**)?
**(3)** Why the approaches do not cover the entire RE process? (**RQ2.3**)
**(4)** To what extent do the tools used in the requirements specification are capable of improving requirements communication (**RQ2.5**)?
**(5)** Which is the most effective communication format in requirements communication of safety-critical systems (**RQ2.7**)?
**(6)** Why do the approaches not follow the guidelines of safety standards (**RQ2.8**)?

## 6. ACKNOWLEDGMENTS

We would like to acknowledge that this work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) from Brazil, and by KKS foundation through the S.E.R.T. Research Profile project at the Software Engineering Research Lab, Blekinge Institute of Technology, Sweden.

## Referências

Bjarnason, E., Wnuk, K., and Regnell, B. (2011). Requirements are slipping through the gaps? a case study on causes & effects of communication gaps in large-scale software development. In *International Requirements Engineering Conference (RE)*, pages 37–46. IEEE.

Brady, A., Seigel, M., Vosecky, T., and Wallace, C. (2007). Addressing communication issues in software development: A case study approach. In *Software Engineering Education & Training, 2007. CSEET'07. 20th Conference on*, pages 301–308. IEEE.

Glinz, M. and Fricker, S. A. (2015). On shared understanding in software engineering: an essay. *Computer Science-Research and Development*, 30(3-4):363–376.

Hatcliff, J., Wassyng, A., Kelly, T., Comar, C., and Jones, P. (2014). Certifiably safe software-dependent systems: challenges and directions. In *Proceedings of the on Future of Software Engineering*, pages 182–200. ACM.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Leveson, N. (2011). *Engineering a safer world: Systems thinking applied to safety*. Mit Press.

Liskin, O. (2015). How artifacts support and impede requirements communication. In *Requirements Engineering: Foundation for Software Quality*, pages 132–147. Springer.

Martins, L. E. G. and Gorschek, T. (2016). Requirements engineering for safety-critical systems: A systematic literature review. *Information and Software Technology*, 75:71–89.

Nakamura, H., Umeki, H., and Kato, T. (2016). Importance of communication and knowledge of disasters in community-based disaster-prevention meetings. *Safety Science*.

Pernstal, J. (2015). *Towards Managing the Interaction between Manufacturing and Development Organizations in Automotive Software Development*. PhD thesis, Department of Computer Science and Engineering, CHALMERS UNIVERSITY OF TECHNOLOGY.

Rasmussen, J. and Lundell, Å. K. (2012). Understanding "communication gaps"among personnel in high-risk workplaces from a dialogical perspective. *Safety science*, 50(1):39–47.

Sikora, E., Tenbergen, B., and Pohl, K. (2012). Industry needs and research directions in requirements engineering for embedded systems. *Requirements Engineering*, 17(1):57–78.

Vilela, J., Castro, J., Martins, L. E. G., and Gorschek, T. (2017). Integration between requirements engineering and safety analysis: A systematic literature review. *Journal of Systems and Software*, 125:68–92.

Vilela, J., Castro, J., Martins, L. E. G., and Gorschek, T. (2018). Safe-re: a safety requirements metamodel based on industry safety standards. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, pages 196–201. ACM.

Wang, Y., Graziotin, D., Kriso, S., and Wagner, S. (2018). Communication channels in safety analysis: An industrial exploratory case study. *arXiv preprint arXiv:1804.08909*.

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

## Systematic Literature Review References

Beckers, K., Heisel, M., Frese, T., and 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*, pages 238–247. IEEE.

Biggs, G., Sakamoto, T., and Kotoku, T. (2014). A profile and tool for modelling safety information with design information in sysml. *Software & Systems Modeling*, pages 1–32.

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

Chandrasekaran, S., Madhumathy, T., Aparna, M., and 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*, pages 1–6.

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

Fricker, S., Gorschek, T., and Glinz, M. (2008). Goal-oriented requirements communication in new product development. In *Software Product Management, 2008. IWSPM '08. Second International Workshop on*, pages 27–34.

Hatcliff, J., Wassyng, A., Kelly, T., Comar, C., and Jones, P. (2014). Certifiably safe software-dependent systems: challenges and directions. In *Proceedings of the on Future of Software Engineering*, pages 182–200. ACM.

Heimdahl, M. P. E. (2007). Safety and software intensive systems: Challenges old and new. In *Future of Software Engineering*, pages 137–152. IEEE Computer Society.

Markovski, J. and van de Mortel-Fronczak, J. (2012). Modeling for safety in a synthesis-centric systems engineering framework. In *Computer Safety, Reliability, and Security*, pages 36–49. Springer.

Medikonda, B. S. and Panchumarthy, S. R. (2009). A framework for software safety in safety-critical systems. *SIGSOFT Softw. Eng. Notes*, 34(2):1–9.

Paige, R. F., Charalambous, R., Ge, X., and Brooke, P. J. (2008). Towards agile engineering of high-integrity systems. In *Computer Safety, Reliability, and Security*, pages 30–43. Springer.

Pernstål, J., Gorschek, T., Feldt, R., and Florén, D. (2015). Requirements communication and balancing in large-scale software-intensive product development. *Information and Software Technology*, 67:44–64.

Schedl, G. and Winkelbauer, W. (2008). Practical ways of improving product safety in industry. In *Improvements In system Safety*, pages 177–193. Springer.

Scholz, S. and Thramboulidis, K. (2013). Integration of model-based engineering with system safety analysis. *International Journal of Industrial and Systems Engineering*, 15(2):193–215.

Sikora, E., Tenbergen, B., and Pohl, K. (2012). Industry needs and research directions in requirements engineering for embedded systems. *Requirements Engineering*, 17(1):57–78.

Stålhane, T. and Sindre, G. (2007). A comparison of two approaches to safety analysis based on use cases. In *Conceptual Modeling-ER 2007*, pages 423–437. Springer.

Stålhane, T., Sindre, G., and Du Bousquet, L. (2010). Comparing safety analysis based on sequence diagrams and textual use cases. In *Advanced Information Systems Engineering*, pages 165–179. Springer.

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