Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 30, Num. 02 (2023) 75-88

RESEARCH ARTICLE

Provision and Collection of Safety Evidence: A Systematic Literature Review

Fornecimento e Coleta de Evidências de Segurança: Uma Revisão Sistemática da Literatura

Lilandra Maria de Oliveira[1]*, Luiz Eduardo Galvão Martins[1], Johnny Cardoso Marques[2]

Abstract: Safety-Critical Systems (SCS) are becoming more and more present in modern societies' daily lives, increasing people's dependence on them. Current SCS are firmly based on computational technology; possible failures in the operation of these systems can lead to accidents and endanger human life, as well as damage the environment and property. SCS are present in many areas such as avionics, automotive systems, industrial plants (chemical, oil & gas, and nuclear), medical devices, railroad control, defense, and aerospace systems. Companies that develop SCS must present evidence of their safety to obtain certification and authorization. This paper presents a Systematic Literature Review (SLR) to investigate processes, tools, and techniques for collecting and managing safety evidence in SCS. The authors conducted this SLR according to the guidelines proposed by Kitchenham and Charters. The SLR comprises seven (7) research questions that investigate essential aspects of collecting and managing safety evidence. The primary studies analyzed in this SLR were selected based on a search string applied into four data sources: ACM, IEEE Xplore, SpringerLink, and ScienceDirect. Data extraction considered (fifty-one) 51 primary studies. The authors identified eleven (11) different approaches covering processes, tools, and techniques for collecting and managing safety evidence. Despite other SLR works conducted about safety evidence, none of them focused on the details related to safety evidence collection. We found that very few approaches focused specifically on the process of collecting safety evidence.

Keywords: safety evidence collection — safety evidence model — safety-critical systems certification — systematic literature review

Resumo: Os Sistemas Críticos de Segurança (SCS) estão cada vez mais presentes no cotidiano das sociedades modernas, aumentando a dependência das pessoas em relação a eles. Os SCS atuais são firmemente baseados em tecnologia computacional; possíveis falhas na operação desses sistemas podem levar a acidentes e colocar em risco a vida humana, além de causar danos ao meio ambiente e ao patrimônio. Os SCS estão presentes em diversas áreas, como aviônica, sistemas automotivos, plantas industriais (química, óleo e gás e nuclear), dispositivos médicos, controle ferroviário, defesa e sistemas aeroespaciais. As empresas que desenvolvem SCS devem apresentar comprovação quanto a segurança para obter a certificação e autorização de comercialização. Este artigo apresenta uma Revisão Sistemática da Literatura (SLR) para investigar processos, ferramentas e técnicas para coletar e gerenciar evidências de segurança de SCS. Os autores conduziram esta SLR de acordo com as diretrizes propostas por Kitchenham e Charters. A SLR compreende sete (7) questões de pesquisa que investigam aspectos essenciais da coleta e gerenciamento de evidências de segurança. Os estudos primários analisados nesta SLR foram selecionados com base em uma string de pesquisa aplicada em quatro fontes de dados: ACM, IEEE Xplore, SpringerLink e ScienceDirect. A extração de dados considerou (cinquenta e um) 51 estudos primários. Os autores identificaram onze (11) abordagens diferentes abrangendo processos, ferramentas e técnicas para coletar e gerenciar evidências de segurança. Apesar de outros trabalhos de SLR realizados sobre evidências de segurança, nenhum deles focou nos detalhes relacionados à coleta de evidências de segurança. Verificamos que poucas abordagens se concentravam especificamente no processo de coleta de evidências de segurança.

Palavras-Chave: coleta de evidências de segurança — modelo de evidências de segurança — certificação de sistemas críticos de segurança — revisão sistemática da litetatura

[1] *Federal University of São Paulo (UNIFESP), São Paulo - São Paulo, Brazil*
[2] *Technological Institute of Aeronautics (ITA), São José dos Campos - São Paulo, Brazil*
***Corresponding author**: lilandra.oliveira@unifesp.br

DOI: http://dx.doi.org/10.22456/2175-2745.126544 • **Received:** 14/08/2022 • **Accepted:** 16/07/2023

1. Introduction

Technological advancement in contemporary society has benefited people's daily lives through systems and software that offer greater ease in carrying out activities. It may affect people to become increasingly dependent on technologies that have to work with precision to avoid and prevent damage and losses that can impact society in general [1]. These systems and software are defined as Safety-Critical Systems (SCS). SCS are heavily based on computational technology and are present in many domains, such as aviation systems, automotive systems, industrial plant control (chemical, oil, and nuclear), medical devices, railroad control, defense and aerospace systems, and others [2].

SCS must attend to the essential attributes of reliability and safety to prevent unexpected behaviors as much as possible so that any defects or failures can cause severe material, human or financial losses and damage to the environment or property [1]. Therefore, during the SCS development, the system supplier conducts several analyses and tests to provide safety evidence that may be used as part of a safety case. At the end of the SCS development lifecycle, the systems and products delivered must be submitted to a certification process. Regulatory entities guide such processes to ensure that the SCS developed by companies meet a certain minimum quality level and are authorized to be commercialized.

The certification process requires a set of safety evidence from the system suppliers, followed by convincing arguments that the SCS is safe enough to operate in a given environment [3, 4]. The safety standards recommend a set of procedures that the system suppliers should follow to satisfy safety objectives and mitigate the potential safety risks that a system can pose during operation [5]. Therefore, the need for safety evidence collection along the SCS development process is necessary to fulfill safety standards' compliance requirements.

The evidence collection process is essential to build convincing safety cases. A safety case contains three main parts: objectives, arguments, and evidence, in which the arguments relate the evidence to the objectives. Showing compliance with the safety case objectives involves obtaining evidence during the SCS development process [6, 7]. However, while aspects of safety case argumentation have been extensively studied, few studies focus on the precise characterization of evidence that should support the safety arguments in SCS [3, 4, 8]. In other words, there is still a lack of appropriate guidance on what evidence should be collected during the SCS development process and how they could be collected, stored, and managed along with the SCS certification and modification phases.

Companies increasingly need to develop proprietary tools besides popular software solutions, also known as commercial-off-the-shelf tools, to accommodate their needs for missing functionalities. It may lead to the internal development of models and tools to support the safety evidence collection along the SCS lifecycle [9]. A safety evidence management and collection process integrated with the SCS development is necessary due to the expensive and laborious activity of reconstructing the missing evidence artifacts after the fact [10].

This systematic literature review (SLR) investigates how the safety evidence provision and collection are addressed in the literature. Moreover, this SLR aims to identify how safety evidence is stored in an information system and managed along with the SCS development and certification phases. It also intends to identify the main evidence types addressed in the literature to understand involved authors producing the safety evidence. This SLR also aims to collect positive results of the used techniques to support safety evidence management addressed in the selected studies and identify gaps and discussion opportunities.

This work is organized into five sections in which the first one presents the contextualization of the research; the second one contains the background and related work; the third section describes the research methodology; section 4 highlights the SLR results and analysis, and section 5 presents results discussion and further works.

2. Background and related work

SCS are present in different domains such as avionics, automotive systems, industrial plants (chemical, oil & gas, and nuclear), medical devices, railroad control, defense, and aerospace systems. Possible failures in the operation of SCS can lead to accidents and endanger human life, as well as damage the environment and property. For that reason, critical systems' producers must present all the required evidence to demonstrate that the system operates safely even under unexpected scenarios. Collecting safety evidence during the system development cycle is a fundamental activity to show compliance with certification requirements and demonstrate that the SCS satisfy required safety levels to operate in established environments.

Considering the relevance of safety evidence collection process in SCS, this SLR was conducted to investigate how processes, tools, and techniques for SCS safety evidence provision and collection are general addressed in the literature. SLR conduction aimed to understand how safety evidence is managed during SCS development and certification phases, to identify existing models or tools related, and possible definition lacks regarding the process. In this section are presented the definitions that support this research and the related work to set the scope and clarify the adopted terms.

2.1 Background

The main concepts and definitions about this paper scope are following presented in order to explain the adopted terms.

2.1.1 Safety-Critical Systems

According to Nair et al. [11], SCS can be defined as one in which failure events "may cause death or injury to people, harm to the environment, or substantial economic loss". Systems known as SCS are increasingly present within people's

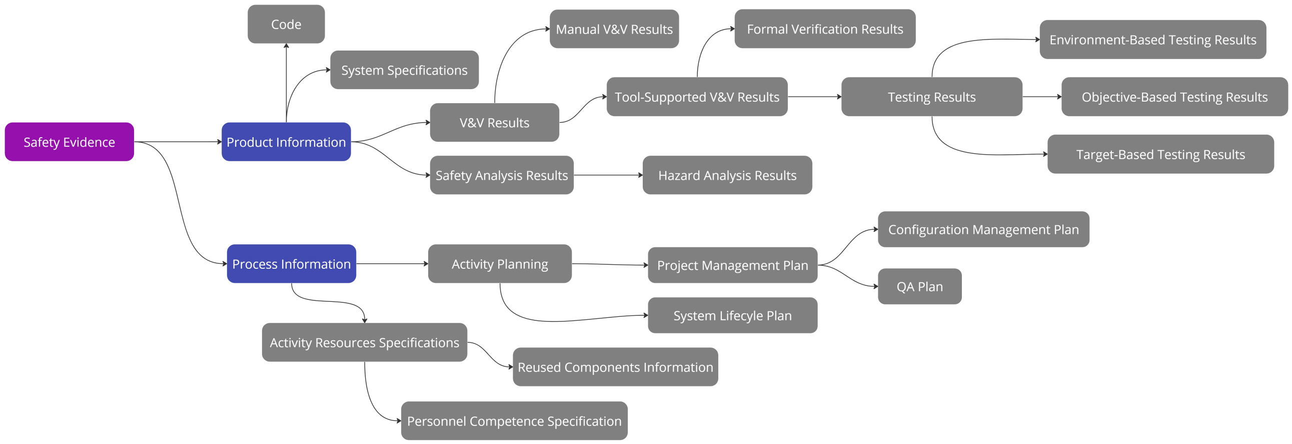

Safety Evidence
Product Information
Code
System Specifications
V&V Results
Manual V&V Results
Tool-Supported V&V Results
Formal Verification Results
Testing Results
Environment-Based Testing Results
Objective-Based Testing Results
Target-Based Testing Results
Safety Analysis Results
Hazard Analysis Results
Process Information
Activity Planning
Project Management Plan
Configuration Management Plan
QA Plan
System Lifecyle Plan
Activity Resources Specifications
Reused Components Information
Personnel Competence Specification

Figure 1. Evidence categories adapted from [11].

lives, in avionic systems, railway systems, and automotive systems [2].

According to Martins & Gorschek [4], safety is an essential attribute to qualify an SCS. Moreover, Lin et al. [18] cite reliability, security, and robustness as SCS quality-related attributes. Additionally, Bate & Burns [12] discuss distinguishing characteristics of SCS such as failure caused by system timing requirements e.g. latency or event sequence that may lead to catastrophic consequences when not given or properly considered.

Leveson [13] defines an accident as “an unplanned and undesired loss event”, which may include severe material, human or financial losses and damage to the environment or property [1]. System hazards may exist considering the environment of the system operation and its relationship with it and can be defined as a state of the system in which a set of “worst-case environmental conditions” that occurs because of the violation of system safety constraints [13]. Therefore, SCS must operate under prescribed specifications [14].

The increased development of complex systems caused by fast technological advancement in society reinforces the importance of safety engineering techniques and processes for preventing systematic failures in development of SCS [13, 15, 14, 16]. Leveson [13] explains that systems shall be designed considering each subsystem component integration and behavior given a social and technical context because of the complex and non-linear interactions among components preventing the system from jeopardize.

Due to the required level of safety to prevent damage to the society from more complex technological systems, the SCS development process must be submitted to safety certification in order to ensure that it meets required safety levels to operate by providing safety evidence to show compliance with the requirements. SCS safety evidence collection and certification process are detailed in sections 2.1.2 and 2.1.3.

2.1.2 Safety Evidence

Safety is a property achieved when the system is able to behave properly under chained conditions by meeting defined safety constraints [13]. Safety evidence can be defined as information or artifacts that contribute to developing confidence in a system’s safe operation and showing the fulfillment of the requirements of one or more safety standards [11]. It can also be considered as an artifact produced during a system’s lifecycle to a specific claim regarding system safety [2].

Nair et al. [11] classified safety evidence information in 49 basic types and offered a glossary to support common interpretation about each one. Figure 1 is adapted from their work and represents 20 safety evidence categories separated into product and process information. According to Nair et al. [11], product information is related to the performed activities while process information has a business perspective relating to activity planning and specifications.

Safety evidence collection is an SCS activity managed during the system development cycle where the produced evidence should be collected to create and support artifacts that can be used as based arguments to meet compliance with the required specifications and standards [17]. According to Martins & Gorschek [4], the safety evidence collection process is essential to build convincing safety cases.

A safety case relates the evidence to its objectives through arguments in order to demonstrate fulfillment with the defined standards and shall be developed in the early stages so that they can be incrementally developed within the product development phase [6, 18]. According to De la Vara et al. [18], safety cases are “arguably among the main evidence types for a safety-critical system” structured to provide convincing arguments that a system is “acceptably safe for a given application in a given operating environment”.

The safety evidence collection process should not be viewed as an after-the-fact activity in order to avoid rework needs that may be identified only during the product certification phase leading to costly and time-consuming activities in order to reconstruct missing evidence artifacts [6, 19, 10, 20, 21]. Safety evidence shall be collected on the appropriate phase and be properly stored on a repository to support evidence incremental development and future maintenance needs to compose certification compliance arguments [22, 23].

2.1.3 Certification Process

According to Silva & Vieira [24], the SCS certification process can be defined as ”a systematic presentation of evidence and justifications to prove the safety and the correct functionality of the system” which follows safety standards defined by organizations like ISO, IEC, IEEE, ABNT, and others [4], aiming to provide a formal assurance that it is “deemed safe by a licensing or regulatory body” [10].

Certification bodies regulate that developed systems are safe for operation, based on presented safety evidence that comply with safety requirements [5, 25]. A safety requirement is a type of requirement that specify what the system should do to prevent and mitigate potentially hazardous behavior of the software [26].

According to Panesar-Walawege et al. [5], the SCS certification is the major prerequisite for its operation and market. It aims to reveal potential gaps in SCS assurance [18]. According to Lin et al. [27], certifiers are usually not the system engineers that developed the system in order to avoid bias results. They also reinforce the importance of safety cases, a.k.a. assurance case, to set confidence about a “claim regarding software assurance from engineers to certifiers” [27].

Panesar-Walawege et al. [5] reinforce the importance of common interpretations of the requirements among certifiers and developers, so that they are able to agree on which evidence should be collected and maintained in for certification, and the use of specializing standards according to the system domain.

The size and complexity of evidence management due to growing systems complexity brought more difficulty for system suppliers that may jeopardize the safety certification [6, 28, 29]. Furthermore, there is an increasing demand for ensuring the SCS high-quality demands that may be contrary to the industry domains’ aim on reducing development costs and time-to-market [28].

According to Nair et al. [11], the certification of SCS is “regarded as being the most challenging”. Nair [10] also reinforces that assurance processes in which avionics, railways, and automotive systems are submitted when attending a certification process are means to not “pose undue risks to people, property, or the environment”.

2.2 Related work

It has been identified some related works addressed in the literature regarding the conducted SLR topic, e.g. on safety evidence types [11], on investigating state of the art about safety evidence [30, 1], on safety assurance case development [31], and about safety evidence management as part of an European initiative denominated OPENCOSS [11, 10, 32, 33, 16, 29, 34]. However, none of these works have focused on addressing the object of this study which is how provision and collection of safety evidence are made and how it is stored during SCS phases.

Nair et al. [11] have investigated safety evidence provision’s state-of-the-art by identifying evidence types addressed in the literature. Their work classified 49 safety evidence types divided into 20 categories related to process and product information. They also covered all evidence types related in OPENCOSS project.

Nair et al. [30] provided a concrete basis for learning about the various types of evidence that practitioners need to provide to support safety. Moreover, they suggest the need for more industry-oriented empirical studies in the area of evidence classification development. The authors recommend using this information as a basis for future research on safety evidence management tools to support the construction, storage, and manipulation of all the evidence types.

According to [29], OPENCOSS (Open Platform for EvolutioNary Certification of Safety-critical Systems) is a large-scale European research project, which goal is to create a common certification framework for the automotive, avionics, and railway domains in close collaboration with industry. When we first accessed the official project page (http://www.opencoss-project.eu/) in April 2020, it was possible to check out the project’s current status (completed), duration (42 months), and all public deliverables during the work phases. It included research papers, use case specifications based on business cases, mapped requirements, solution architecture design, and parts of the developed evidence management software prototype in OPENCOSS tools. Despite the documentation available about this initiative, some information is restricted only to partners, e.g., source code repository.

OPENCOSS related papers [11, 10, 32, 33, 16, 29, 34] commonly address that the certification process of SCS is one of the most expensive and laborious parts of the SCS development phase. Thus, the initiative aims to provide a framework capable of reusing safety evidence information, consolidate different terminology and requirements of industrial domains, and be able to communicate with existing development and assurance tools. It shall support reducing costs by time-consuming activities, eventual re-certification processes, and increasing safety assurance information reliability to different SCS industrial domains.

In summary, previous researches have reviewed the evidence taxonomy (e.g., definition, types, classification), and some of them have presented a developed tool that supports evidence management collection. Despite this, it was not possible to identify studies in the literature that have clearly explained how safety evidence management is supported during a Safety-Critical Systems development lifecycle.

This led us to the need for an SLR to gain new insights into collect, structure, assess, and manage safety evidence.

3. Research methodology

This section presents how the SLR was planned and conducted based on the guidelines proposed by Kitchenham and Charters [35]. Figure 2 illustrates the research process executed to collect the amount of the most relevant papers aiming to answer the SLR research questions. Which were first defined during the planning phase, as well as most relevant digital libraries definition regarding this research domain, and the search string protocol definition based on the performed tests.

Seven research questions were defined in the planning phase to support the SLR conduction, as listed in Table 1, each one aiming to identify pre-determined information. With these definitions, researchers were able to focus better on what information to extract from the studies by answering the questions in the SLR execution phase. Thereafter, the studies were classified based on the defined quality assessment criteria, so that analysis and documentation were executed considering relevant extracted data.

ID	Research Question	Aim
RQ1	What are the techniques, processes, and tools used to collect safety evidence in safety-critical systems?	To identify the techniques, methods, and processes used to collect safety evidence in SCS development.
RQ2	What are the sources/producers of safety evidence?	To identify the primary sources and producers of safety evidence along with SCS development.
RQ3	How are the safety evidence used during the systems development?	To understand how the safety evidence are related to the system artifacts produces along with the SCS development.
RQ4	How are the safety evidence used during the certification process?	To understand how the safety evidence collection im-pacts the certification process.
RQ5	Are the benefits of the used techniques cited? If so, which ones?	To collect positive results of previous studies.
RQ6	Are the difficulties of the used techniques cited? If so, which ones?	To collect gaps and opportunities for discussion.
RQ7	Are the safety evidence stored? If so, how are they recovered?	To understand how the project has developed a safety evidence collection model.

Table 1. Research questions for the systematic review

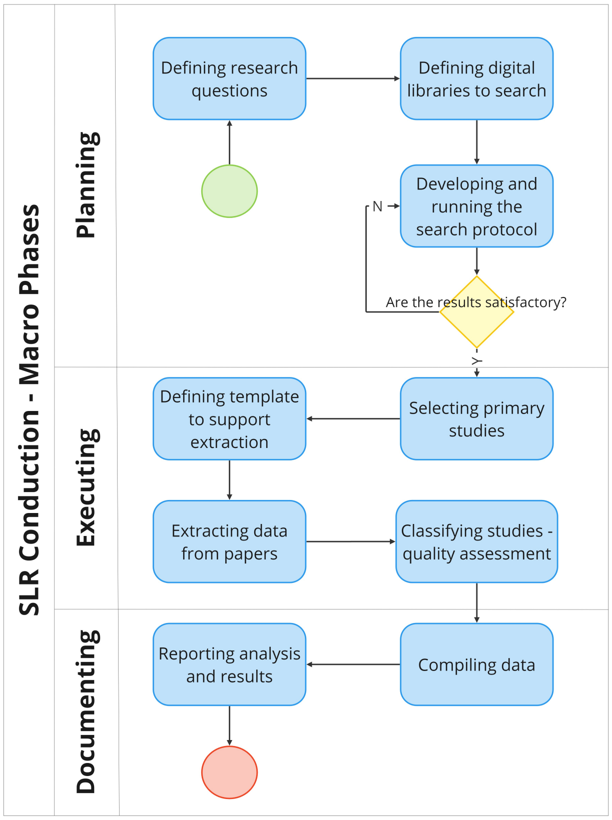

SLR Conduction - Macro Phases
Planning
Executing
Documenting
Defining research questions
Defining digital libraries to search
Developing and running the search protocol
N
Are the results satisfactory?
Y
Selecting primary studies
Defining template to support extraction
Extracting data from papers
Classifying studies - quality assessment
Compiling data
Reporting analysis and results

Figure 2. SLR conduction macro phases adapted from [35].

3.1 Search strategy

The adopted search strategy consists of searching papers through search engines of digital libraries by using the defined search string. Four digital libraries were used to select primary studies for the SLR, as presented in Table 2.

The following search string was informed on the search field of each digital library in order to select studies that match the filter information. The keywords were defined based on the results of some tests, in which most of the papers returned were related to SCS safety evidence. It is important to clarify that this search process was performed in September 2021.

”safety evidence” AND (collect* OR manag* OR analysis) AND (”safety-critical system” OR ”safety-critical software” OR ”critical system”) AND (”safety certification” OR ”embedded systems”)

Table 2. Searched bases

Base	Link
ACM Digital library	dl.acm.org
IEEE Xplore	ieeexplore.ieee.org
Science Direct	sciencedirect.com
Springer Link	link.springer.com

3.2 Review protocol

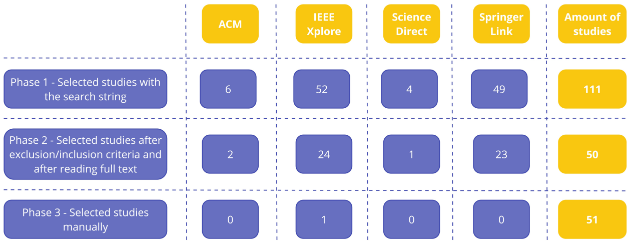
The review protocol was developed as follows: the selected resources chosen were ACM digital libraries, IEEE Xplore, Science Direct, and Springer Link; the search method used the digital libraries' search engines; the population was composed of publications reporting approaches to provide and collect pieces of safety evidence.

As studies selection criteria, we determined that papers should be scientific articles from journals, magazines, conferences, symposia, and workshops. Only articles wrote in English and free downloadable would be considered. We also defined the inclusion criteria as follows:

- The study suggests or relates to the certification process of critical systems;
- The study relates to relevant evidence used on argumentation during the certification process;
- The study relates to verification and validation during the software development life cycle;
- The study relates about management tools to control safety evidence to certification process;
- The study suggests or relates to the management and maintenance of safety cases;
- The study relates to certifications standards for control systems; and
- The study relates to safety evidence collection and argumentation (such as new methodologies).

We also defined the exclusion as:

- The study is not written in English;
- The study does not contain an abstract;
- The study is published just as an abstract;
- The study is a version older than another study already considered;
- Unable to access the study;
- The study abstract's content does not relate to any mentioned criteria in the eligible list; and
- The study is not a primary study (e.g., SLR, survey, course).



	ACM	IEEE Xplore	Science Direct	Springer Link	Amount of studies
Phase 1 - Selected studies with the search string	6	52	4	49	111
Phase 2 - Selected studies after exclusion/inclusion criteria and after reading full text	2	24	1	23	50
Phase 3 - Selected studies manually	0	1	0	0	51

Figure 3. Papers selection.

3.3 Procedure for studies selection

We manually selected articles from the protocol results based on reading each one's abstracts and applying the inclusion and exclusion criteria. Figure 3 details the final number of papers after three phases, in which the initial amount was 111 selected studies. In the second phase, we read each selected paper's abstract to filter manually, which could address the researched topic by strictly following the inclusion and exclusion criteria. Considering the 51 selected studies, the most relevant digital library sources were IEEE and Springer-Link. One primary study was included manually based on an expert's recommendation on this SLR subject in the third phase. Moreover, other relevant studies were added in order to complement the ones identified in the SLR.

3.4 Data extraction

Considering the inclusion and exclusion criteria presented, we selected fifty-one (51) studies for data extraction. The research questions presented in Table 1 were the main drivers to build the data extraction spreadsheet. We divided the spreadsheet into three following sections: article metadata (yellow color), research questions to SLR (blue color), quality assurance questions (green color).

In addition to the bibliographic information (title, authors, year, and source), for each study was extracted information considered as evidence in the literature, techniques for evidence structuring, techniques for assessing the evidence collected, tool support for evidence management, and challenges addressed related to evidence development, structuring, and assessment. The complete SLR data extraction spreadsheet is available at this link.

3.5 Study quality assessment

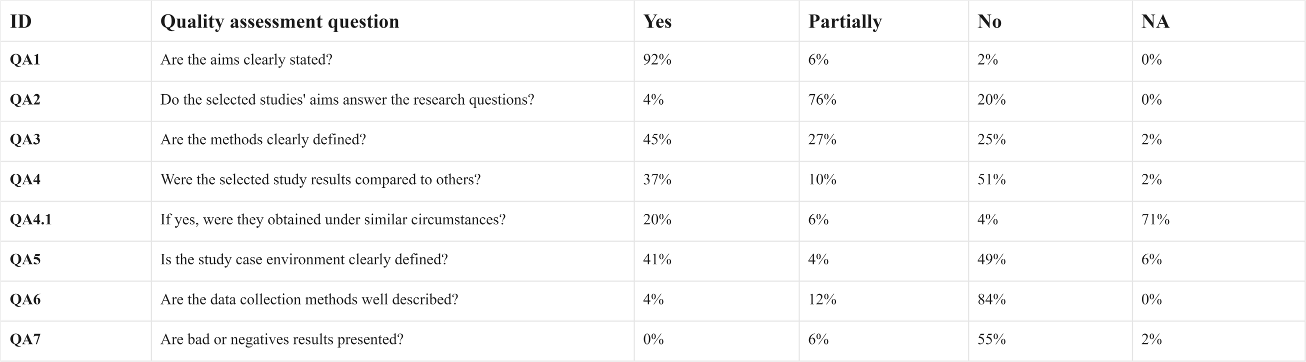
The study quality assessment evaluated how adherent the analyzed studies were concerning what the SLR is looking for. We defined eight quality assessment questions to assess the selected papers. Each question could be answered positively or negatively, or as partial or not applicable (NA). When the study did not address the intended topic, the question was answered as NA.

As shown in Figure 5, few papers (4%) could answer the complete set of research questions in this SLR. Hence, most of the questions could not be totally answered by the selected studies. The 45% of the selected papers defined the methods, 27% were partially defined, and 25% had no definition. It may indicate that most studies did not address how to collect and manage the pieces of evidence, whether by a model or a support tool.

Although 84% of the selected studies have not described the safety evidence collection methods in detail, it can still be considered that the information obtained brought relevant insights to this SLR.

3.6 Threats to validity

Following the guidelines proposed by Kitchenham and Charters [35], we elaborated a review protocol that mitigates the risks of biased results. Nevertheless, there is a possibility of interference in the validity of the results presented given the following considerations:



ID	Quality assessment question	Yes	Partially	No	NA
QA1	Are the aims clearly stated?	92%	6%	2%	0%
QA2	Do the selected studies' aims answer the research questions?	4%	76%	20%	0%
QA3	Are the methods clearly defined?	45%	27%	25%	2%
QA4	Were the selected study results compared to others?	37%	10%	51%	2%
QA4.1	If yes, were they obtained under similar circumstances?	20%	6%	4%	71%
QA5	Is the study case environment clearly defined?	41%	4%	49%	6%
QA6	Are the data collection methods well described?	4%	12%	84%	0%
QA7	Are bad or negatives results presented?	0%	6%	55%	2%

Figure 4. Quality assessment results.

- The final search string definition can be considered as a threat because even though successful pilot searches on the selected digital libraries, it is possible that some papers were not included based on the keywords' choice;
- The researchers started this SLR with limited knowledge about the related subjects. We decided to start with an automatic search based on the search strings. However, a manual filter based on the reading abstracts was cared out during phase 2, this procedure may bring the possibility of leaving some relevant papers behind;
- The possibility of not selecting all relevant papers for this SLR using the defined search protocol based on the inclusion and exclusion criteria (e.g., some papers were discarded during the selection phase because their content was not publicly available).

4. Results and analysis

In this section are presented the discussion and analysis of the 51 primary studies selected. Figure 5 shows the distribution of the selected studies by year of publication, which indicates that the number of studies related to this research theme has been increasing over the last years. As shown in Figure 6, about 86% of the studies used in this SLR are from the last ten years. It is important to emphasize that the publication year was not considered an exclusion criterion during the paper's selection phase.

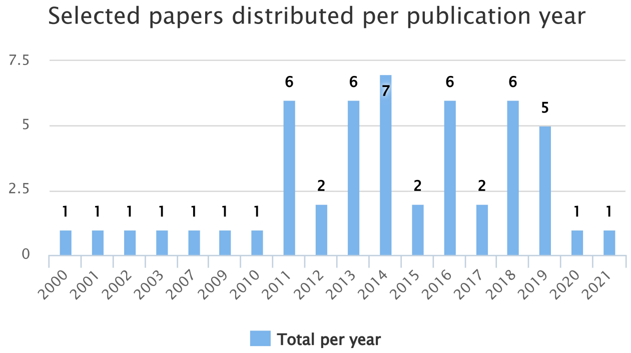


Figure 5. Selected papers distributed per publication year.

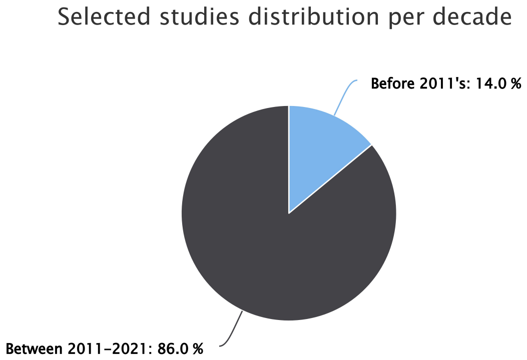


Figure 6. Selected papers distributed in the last decade.

4.1 Techniques, processes, and tools used to collect safety evidence (RQ1)

The purpose of this research question was to identify which approaches have been used currently to collect safety evidence. Table 3 shows the distribution of techniques, processes, and tools mentioned in the selected studies. Those that appear more frequently are: GSN (Goal Structuring Notation) [36, 22, 37, 38, 20, 39, 40, 12, 41, 42, 9], FTA (Fault Tree Analysis) [24, 43, 44, 15, 14, 45], UML (Unified Modeling Language) [5, 7, 22, 40, 16], SysML (Systems Modeling Language) [13, 26, 31, 38], and OCL (Object Constraint Language) [27, 5, 46, 40]. The usage of GSN can explicitly document an argument's elements and structure and the relationship between argument and safety evidence [47, 18, 27]. In GSN, the definition of argument's claims as goals and items of evidence are documented in solutions. Despite the GSN and the techniques mentioned above are not used to collect safety evidence, they were presented in selected studies to support this activity.

Nair et al. [10, 32] have developed the SafeTIM: A Traceability Information Model for safety evidence due to their previous works and as part of the OPENCOSS project. SafeTIM provides a basement of the evidence types and the relationship between the safety evidence to provide evidence traceability in real industrial settings. The authors consider that using this model with support tools can significantly facilitate evidence traceability in safety-critical systems development.

Table 3. Distribution of the techniques, paradigms, and tools used to collect safety evidence reported by the selected studies.

Description	Occurences	Definition
Goal Structuring Notation (GSN)	11	Technique
FTA (Failure Modes and Effects Analysis)	6	Technique
UML (Unified Modeling Language)	5	Technique
V&V (Verification and Validation)	4	Paradigm
Object Constraint Language (OCL)	4	Technique
SafeTIM (Traceability Information Model for safety evidence)	2	Tool
SAEM (Software Assurance Evidence Metamodel)	2	Tool
SACM (Structured Assurance Case Metamodel)	1	Technique
SysML language	1	Technique
CRESCO (Construction of Evidence REpositories for Managing Standards Compliance)	1	Tool
Model-based development (MBD) - CHESS	1	Paradigm
Model-Driven Safety Certification (MDSafeCer) method	1	Technique

De la Vara et al. [48] have presented SACM (Structured Assurance Case Metamodel), a case development and information exchange standard. SACM supports safety evidence management assurance (e.g., safety analysis results, system specifications, testing results). It identifies possible issues that could hinder this activity by using the hazard log and software verification results, common artifact types managed in the most safety-critical domains and used as safety evidence. According to Nair et al. [32], SACM also addresses safety evidence traceability scope. It provides an Evidence Metamodel that specifies relationships between evidence items and other assurance assets.

According to [49], the Safety Case Conceptual Model (SCCM) is used as a guide to help suppliers during the evidence collection phase. SCCM avoids redundancy and inconsistency artifacts and systematically detects missing evidence items to support suppliers when collecting evidence from the artifacts produced in the development process. The authors have compared SCCM with SafeTIM [32], and their consideration, i.e., SCCM focuses on the evidence collection process. SCCM is a smaller and more flexible approach to present the traceability between artifact and safety objectives.

According to Nair [10], the Software Assurance Evidence Metamodel (SAEM) is a standard metamodel that establishes the necessary models of evidence elements required for detailed compliance and risk analysis. According to Panesar-Walawege et al. [5], SAEM directs towards linking the certification evidence to safety claims and evaluating these claims subject to the evidence.

CRESCO is an acronym for Construction of Evidence REpositories for Managing Standards COmpliance. It is a web tool developed in Java EE that communicates to a relational database through Apache Derby and runs on Apache Tomcat. Server interfaces were built using JavaScript and JavaServer Pages. According to Panesar-Walawege et al. [23], the tool can store evidence in a centralized repository to be manipulated, generating evidence to demonstrate compliance reports in the certification phase.

Based on the presented papers, it is possible to observe that these approaches do not show the models and tools' insertion within the collection evidence process in SCS development. Based on the literature, Panesar-Walawege et al. [23] have mentioned an important gap in the certification safety process when considering the need for evidence repositories to demonstrate compliance to the standards. It means safety evidence must be collected and stored when produced, preventing the loss of information and allowing the usage during the certification phase.

4.2 Sources and producers of safety evidence (RQ2)

This research question aimed to identify the primary sources and producers of safety evidence along with the SCS development. Figure 7 presents the nomenclatures addressed in the selected studies to refer to sources and safety evidence producers. The three most common terminologies are system suppliers [10, 5, 7, 50, 51, 32, 29, 52, 45], software engineers [53, 47, 27, 43, 46, 15, 20, 16, 54, 55], and safety engineers [19, 44, 37, 16, 29, 41]. Additionally, it is possible to observe the non-mention of some of the important actors in the system lifecycle, e.g., software testers. Although they are part of the software developers, the term ”tester” was not particularly addressed. Software quality engineers and software testers are important actors who produce safety evidence throughout the SCS development process to meet the certification requirements. This finding might indicate that this specific role is already part of the responsibilities of the sources and producers and hence there is no mention of a specific role for tester. For instance, the roles of software engineer or software developer may include the role of software tester.

The system supplier has an essential role in generating safety evidence, e.g., when a company subcontracts software development, the subcontract is responsible for providing enough evidence to demonstrate that the software/system is safe. The safety engineer analyzes a series of supplier deliverables and verifies their compliance with standards and regulations to show the required compliance. They also contribute to disseminating a safety culture among the software developers and involved members and defining standards, processes, and methodologies according to the safety standards.

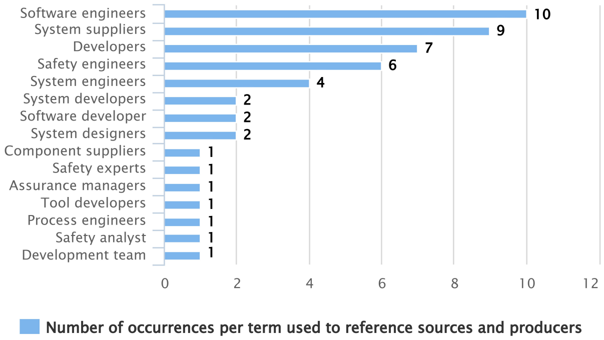


Figure 7. Distribution of terms used to reference sources and producers.

4.3 Use of safety evidence during systems development (RQ3)

The purpose of this research question was to understand how the safety evidence is related to the system artifacts produced along with SCS development. Lin et al. [27] have compared that engineers can generate system artifacts for a software application during the development process. During the software, engineers can modify those system artifacts due to various considerations. For instance, such modification may lead to the invalidation of a safety case and its argument structure because of out-of-date supporting evidence. The authors have mentioned an effective strategy on a support tool that could help certifiers assess the software assurance considering the system modifications would highlight the affected nodes in the safety case to concentrate analysis on the affected nodes.

According to Wu et al. [20], the production of safety evidence occurs after the design completion, and they recognized the need for incremental construction of safety evidence and corresponding safety cases. Jaradat et al. [21] complement it by addressing the system developer's difficulties in identifying modifications' direct and indirect impact due to the high level of dependency among safety case elements.

Based on the presented papers, it is possible to observe no clear definition that could help us answer how to use safety evidence during systems development. It is an open question for further investigation.

4.4 Use of safety evidence during certification process (RQ4)

The purpose of this research question was to understand how safety evidence collection impacts the certification process.

According to Sabetzadeh et al. [19], there is a need for more work on adapting additional demands that certification imposes in the expression of systems' design and that safety evidence collection is often (incorrectly) seen as an after-the-fact activity, rather than during the system development. They mentioned that this situation could give rise to several problems during the certification process because of the number of collected safety evidence necessary for certification during the development phase. Not making the design ”certification-aware”, the system would be inevitably not auditable for certification purposes, considering the potential omissions.

Gannous et al. [53] have mentioned that the verification phase of SCS is an essential part of the certification process and proposed a methodology to provide testing and verification activities to produce different safety evidence to support a successful and efficient safety certification process. In the same direction, De la Vara et al. [18] addressed that a safety case must be built for many safety-critical systems as part of the certification process. The safety case provides evidence to justify that the system's design and implementation avoid hazardous software behavior in its intended environment. It links system in-formation, called evidence, with the safety requirements, called safety claims, via arguments that show the relationship between the requirements and the system information.

According to Panesar-Walawege et al. [5], the safety standards recommend several procedures that the system suppliers should follow to create the necessary evidence during the development phase to meet the compliance requirements. Safety standards offer guidance to accumulate and share best practices in addition to building the certification process. However, the authors have mentioned that standards bring some challenges to system suppliers and certifiers. They emphasize the need for a systematic procedure for creating the necessary evidence by having common interpretations of all parties involved to know which evidence should be collected and maintained for certification. The certification body would be allowed to assess the evidence more effectively based on the application domain and the relevant standards.

4.5 Benefits of the used techniques (RQ5)

The purpose of this research question was to identify benefits addressed in the selected studies related to the use of techniques, processes, and tools to collect safety evidence. The most relevant studies discussed using models or tools during the safety evidence processes to fulfil safety standards' compliance requirements. Although few studies have presented in detail methods or tools, they mention future work opportunities and their possible benefits.

Sabetzadeh et al. [19] have specified a set of guidelines applied to functional requirements for the use of SysML for modeling SCS interfaces (hardware and software) aiming the improvement of the certification process by simplifying impact analysis on modifications made at a later stage and decreasing recurrent design specification issues identified on inspections and audits carried out by certification authorities. They addressed relevant topics for further investigation, e.g., development of guidelines to improve the quality of the specified design requirements, improve the developed guidelines for the use of SysML applied to non-functional requirements, and extend the mentioned technique's use to other domains besides maritime and energy systems.

Lin et al. [27] have proposed a framework to support the construction and maintenance of assurance cases and conducted a case study to simulate an intensive system's evolution. They concluded that the use of safety patterns adjusted to the necessary flexibility for each domain particularity allows the engineers and certifiers to develop and certify a software system more effectively and efficiently. They assimilated that automation integrates the generation and maintenance of safety cases during the complete software development process, not just at the end of the development phase. Romanski et al. [56] also recognize the use of automation tools as an essential factor to ensure accuracy and efficiency during the development lifecycle. Lin et al. [27] intend to extend the proposed technique by analyzing the explicit use of standards in assurance cases for intensive systems for further works.

Denney and Pai [36] have implemented an assurance case automation toolset named AdvoCATE. They illustrated its results by running a real example of an aircraft autopilot software's safety case through a hard-coded integration of AutoCERT, a static source-code analysis tool. The results suggest that an evidence' argument provides a more straightforward form of evidence management than a formal method or tool by integrating the evidence into the language of assurance arguments, not as a separated artifact. It also enables high-level requirements and its claims to low-level evidence traceability. The authors consider the unified and convenient interface of the developed tool to use verification tools. They intend to address its qualification and extent of assurance compared to formal methods or tools.

4.6 Difficulties of the used techniques (RQ6)

The purpose of this research question was to collect gaps and opportunities identified in the selected studies. Although few studies mentioned difficulties of the used techniques, models, or tools, it was possible to identify some challenges associated with the safety evidence management process. Mainly concerning the safety evidence lifecycle along the SCS lifecycle, i.e., none of the selected studies presented assumptions or guidelines about how to properly collect evidence according to the SCS phase considering its impacts commonly seen in the certification phase.

Panesar-Walawege et al. [5] have addressed some challenges/difficulties in the safety evidence management subject by relating the current certification standards. Some of these challenges/difficulties are: (i) the need to create common interpretations of all involved parties to avoid different interpretations of the standard used by the supplier and the certifier; (ii) the need to specialize standards to industrial contexts according to the domain; (iii) the need for aligning standards to the organizational practices reinforces the need for a systematic procedure to create the relevant evidence and share it with the certification body in a form to allow the assessment in terms of both the application domain and the applicable standard; (iv) the need for planning for certification refers to the inherent common interpretation needed between the supplier and certifier so that both have an upfront agreement concerning the evidence artifacts created during the SCS development phase; (v) the need of manage safety evidence electronically due to the difficult by using paper-based documents that form the basis of the certification evidence.

According to De la Vara et al. [29], system suppliers may have difficulties understanding safety standards, determining the evidence, or gaining confidence in evidence adequacy. The system supplier also manages a large amount of evidence and structure it to comply with a safety standard. The high volume and complexity of evidence managed by the practitioners could lead to not properly structured evidence and may jeopardize the safety certification. Huber et al. [9] presented as a result of an exploratory survey that all the participant companies use a variety of proprietary tools developed by themselves besides popular software solutions, also known as commercial-off-the-shelf tools, such as IBM Rational DOORS and PTC-Integrity (e.g., requirements engineering purposes), Enterprise Architect and Visio (e.g., system development) and Microsoft Word/Excel (e.g., risk management and assurance purposes). The companies have highly customized processes and try to accommodate their need for missing functionalities by developing intercommunication between different tools. It may lead to a complex toolchain and may jeopardize the safety certification due to the rigid and time-intensive change management.

The need for an integrated process to collect and manage safety evidence within the SCS development phase (generation of evidence), certification, and modification phases (potential maintenance of evidence) appropriate and flexible enough to each domain is observable. It is mainly due to the expensive (costly and time consuming) and laborious system supplier's activity to reconstruct the missing evidence artifacts after-the-fact [16]. Considering the benefits provided within the use of support tools [19, 27, 56, 36], it is possible to observe that based on a well-defined process, the development of a support tool may support the provision and collection of SCS evidences. A tool may integrate the certification requirements to improve the evidence control and management throughout the SCS lifecycle and mitigate the project impacts identified in the certification phase.

4.7 Safety evidence storage and recovery (RQ7)

The purpose of this research question was to understand how a support tool was developed in case that the selected study has presented one. As previously presented in Section 3.6, few studies have mentioned a developed tool to collect safety evidence. Most of the proposed studies do not show any technical details, such as the designed solution architecture or the programming languages used to build the system or specify how the evidence is stored or recovered. None of them were available in public source code repositories. The studies focused on addressing the support tool need and its high-level functionalities to present the results.

The most relevant initiative identified was the OPENCOSS [29], already mentioned in the Related Work Section, which presents the designed solution in detail through diagrams, use cases, architectural views, etc.

As Huber et al. [9] presented, companies use to develop various proprietary tools to satisfy their customized processes. Considering factors such as competitiveness, this knowledge about the process and tool shall be kept private by the company. It may lead us to assume that some companies could have developed their own safety evidence support tool based on its expertise, usually concentrated in senior professionals.

From the perspective of adopting a standard such as OPENCOSS for managing safety evidence, there might be some advantages such as leveraging a more mature tool/process from the starting point, considering that it was planned and built on top of a set of lessons learned from similar industries. Although such large initiatives can produce high impact, they also tend to move slowly and so companies might choose to move forward with their internal solutions as they can be fully customizable and meet their own roadmap priorities.

5. Conclusion

This work has presented the SLR results about the provision and collection of safety evidence, which have addressed techniques, processes, and tools used to collect safety evidence in SCS. We investigated the literature definitions about how safety evidence is produced, the professionals involved, and its use during the systems development and certification phases. We also identified some benefits and difficulties due to the techniques addressed in the selected studies. The most relevant findings and highlights to further research are as follows.

Safety evidence types definition. It was challenging to mine into the selected studies and identify evidence types that were not clearly named as one, except for the study conducted by Nair et al. [11], which performed an intensive analysis of evidence types and suggested the need for more detailed evidence collection models to address different industry domains.

Safety evidence supplier's and producer's definition and responsibilities. It was also challenging to identify the professional roles involved in the safety evidence management process due to the diversity of terms found. It was possible to observe that some of the important actors in the system development lifecycle were not mentioned, e.g., software testers. Although they are software developers, the term ”tester” was not particularly addressed. We consider that software testers are important actors who produce useful safety evidence throughout the SCS development process to meet the certification requirements.

Lack of definition of safety evidence in SCS development and certification phases. Based on the selected studies, it was possible to observe no clear definition that could help us answer how to use safety evidence during systems development. We saw that safety evidence as an after-the-fact activity, leading to significant omissions and making the system non-auditable for certification purposes [19, 10, 20, 21]. It is an open question for further investigation. Need for more adaptable and customizable safety patterns. There is a need for various safety patterns to address the specificity of the different domains. The most relevant benefits of using safety patterns were related to the necessary flexibility to each domain particularity, which allows the engineers and certifiers to develop and certify a software system more effectively and efficiently.

Need for common understanding about safety standards. Based on the safety standard analyzed along this SRL, we believe that system suppliers may have difficulties understanding the safety standards. The suppliers have a problem determining which evidence should be collected and maintained for the certification phase or gaining confidence in evidence adequacy due to the need for common interpretations of all parties involved [5, 29].

Safety evidence repositories. We observed that none of the presented approaches show integrating the models and tools with the collection evidence process in SCS development. Neither addressed the management of safety evidence repositories. It reinforces a gap in the certification process when considering the need for evidence repositories to demonstrate compliance with the safety standards [23]. We believe that an integrated safety evidence repository is fundamental for any safety evidence collection process.

Need for centralized and integrated safety evidence management. The most relevant difficulties were related to the certification process's challenges due to the use of a complex proprietary tools chain and the need to avoid safety evidence losses throughout the development phase to be stored when generated. [23, 9].

6. Authors contributions

Matheus Raffael Simon: Contributed to the development of this study; acted in the construction of the concept and methods for the conception of the research; participated in data acquisition and analysis; wrote, read and approved the final manuscript.

Adair Santa Catarina: Contributed to the development of this study; assisted in the construction of the concept and methods for designing the research; participated in data acquisition and analysis; wrote, read and approved the final manuscript.

Adriane Yaeko Togashi: Contributed to the development of this study; suggested the concept and methods for designing the research; participated in acquiring and analyzing the data; read and approved the final manuscript.

References

[1] CARDOSO, M. J. S. M. *Modelo de processo de testes para sistemas de software críticos*. Dissertação (Mestrado) — Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte. p. 171. 2010.

[2] NAIR, S. et al. Evidence management for compliance of critical systems with safety standards: A survey on the state of practice. *Information and Software Technology*, London, v. 60, p. 1–15, apr. 2015.

[3] MARTINS, L. E. G. ao. Desenvolvimento de um modelo de processo baseado em stamp para coleta e gerenciamento de evidências de seguran ça de sistemas críticos. (em preparação.). 2019.

[4] MARTINS, L. E. G. ao; GORSCHEK, T. A process model based on stamp for collecting and management of safety evidence. In: *SAFECOMP2020 - Position Papers International Conference on Computer Safety, Reliability and Security*. Lisbon, Portugal: HAL, 2020.

[5] PANESAR-WALAWEGE, R. K.; SABETZADEH, M.; BRIAND, L. Using model-driven engineering for managing safety evidence: Challenges, vision and experience. In: *2011 First International Workshop on Software Certification*. New York: IEEE, 2011. p. 7–12.

[6] KELLY, T. P. *Arguing safety - A systematic approach to managing safety cases*. 341 p. Tese (Doutorado) — University of York, York, 1998. Disponível em: ⟨https://www-users.cs.york.ac.uk/~tpk/tpkthesis.pdf⟩.

[7] PANESAR-WALAWEGE, R. K.; SABETZADEH, M.; BRIAND, L. A model-driven engineering approach to support the verification of compliance to safety standards. In: *22nd International Symposium on Software Reliability Engineering*. New York: IEEE, 2011. p. 30–39.

[8] WALKINSHAW, N. Software inspections, code reviews, and safety arguments. In: *Software Quality Assurance: Consistency in the Face of Complexity and Change*. Cham, Switzerland: Springer International Publishing, 2017. p. 127–140.

[9] HUBER, M. et al. Roadblocks on the highway to secure cars: An exploratory survey on the current safety and security practice of the automotive industry. Springer International Publishing, Cham, Switzerland, p. 157–171, 2018.

[10] NAIR, S. Evidence management for evolutionary safety assurance and certification. In: *21st IEEE International Requirements Engineering Conference (RE)*. New York: IEEE, 2013. p. 385–388.

[11] NAIR, S. et al. An extended systematic literature review on provision of evidence for safety certification. *Information and Software Technology*, London, v. 56, n. 7, p. 689–717, jul. 2014.

[12] BATE, I.; BURNS, A. An integrated approach to scheduling in safety-critical embedded control systems. *Real-Time Systems*, Netherlands, v. 25, n. 1, p. 5–37, jul. 2003.

[13] LEVESON, N. G. *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, USA: Massachusetts Institute of Technology, 2011.

[14] SHBOUL, B. A.; PETRIU, D. C. Pattern-based transformation of sysml models into fault tree models. In: *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*. Riverton, USA: IBM Corp., 2019. p. 214–223.

[15] WOLSCHKE, C. et al. Industrial perspective on reuse of safety artifacts in software product lines. In: *Proceedings of the 23rd International Systems and Software Product Line Conference - Volume A*. New York, NY, USA: Association for Computing Machinery, 2019. p. 143–154.

[16] LUO, Y.; SABERI, A. K.; BRAND, M. v. den. Safety-driven development and iso 26262. In: DAJSUREN, Y.; BRAND, M. van den (Ed.). *Automotive Systems and Software Engineering: State of the Art and Future Trends*. Cham, Switzerland: Springer International Publishing, 2019. p. 225–254.

[17] MACGREGOR, J.; BURTON, S. Challenges in assuring highly complex, high volume safety-critical software. In: *SAFECOMP 2018: Computer Safety, Reliability, and Security*. Cham, Switzerland: Springer International Publishing, 2018. p. 252–264.

[18] VARA, J. L. de la et al. An industrial survey of safety evidence change impact analysis practice. *IEEE Transactions on Software Engineering*, Piscataway, USA, v. 42, n. 12, p. 1095–1117, apr. 2016.

[19] SABETZADEH, M. et al. Using sysml for modeling of safety-critical software-hardware interfaces: Guidelines and industry experience. In: *2011 IEEE 13th International Symposium on High-Assurance Systems Engineering*. New York: IEEE, 2011. p. 193–201.

[20] WU, W.; KELLY, T. Towards evidence-based architectural design for safety-critical software applications. In: LEMOS, R. de; GACEK, C.; ROMANOVSKY, A. (Ed.). *Architecting Dependable Systems IV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 383–408.

[21] JARADAT, O.; BATE, I. Systematic maintenance of safety cases to reduce risk. In: *35th International Conference on Computer Safety, Reliability, and Security*. Cham, Switzerland: Springer International Publishing, 2016. p. 17–29.

[22] BRESSAN, L. et al. A systematic process for applying the chess methodology in the creation of certifiable evidence. In: *2018 14th European Dependable Computing Conference (EDCC)*. New York: IEEE, 2018. p. 49–56.

[23] PANESAR-WALAWEGE, R. K. et al. Cresco: Construction of evidence repositories for managing standards compliance. In: *ER 2011: Advances in Conceptual Modeling. Recent Developments and New Directions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 338–342.

[24] SILVA, N.; VIEIRA, M. Certification of embedded systems: Quantitative analysis and irrefutable evidences. In: *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. New York: IEEE, 2013. p. 15–16.

[25] REMPEL, P.; MäDER, P. Continuous assessment of software traceability. In: *Proceedings of the 38th International Conference on Software Engineering Companion*. New York, USA: Association for Computing Machinery, 2016. p. 747–748.

[26] HAWKINS, R. et al. Assurance cases and prescriptive software safety certification: A comparative study. *Safety Science*, Amsterdam, v. 59, p. 55–71, nov. 2013.

[27] LIN, C.-L.; SHEN, W.; DRAGER, S. A framework to support generation and maintenance of an assurance case. In: *2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. New York: IEEE, 2016. p. 21–24.

[28] ÖFIG, K. H.; ZELLER, M.; HEILMANN, R. Alfred: A methodology to enable component fault trees for layered architectures. In: *41st Euromicro Conference on Software Engineering and Advanced Applications*. New York: IEEE, 2015. p. 167–176.

[29] VARA, J. L. de la; PANESAR-WALAWEGE, R. K. Safetymet: A metamodel for safety standards. In: *MODELS 2013: Model-Driven Engineering Languages and Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 69–86.

[30] NAIR, S. et al. Classification, structuring, and assessment of evidence for safety – a systematic literature review. In: *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*. New York: IEEE, 2013. p. 94–103.

[31] ALMENDRA, C.; SILVA, C.; VILELA, J. Incremental development of safety cases: A mapping study. In: *Proceedings of the 34th Brazilian Symposium on Software Engineering*. New York, USA: Association for Computing Machinery, 2020. (SBES '20), p. 538–547.

[32] NAIR, S. et al. Safety evidence traceability: Problem analysis and model. In: SALINESI, C.; WEERD, I. van de (Ed.). *Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ*. Cham, Switzerland: Springer International Publishing, 2014. p. 309–324.

[33] VARA, J. L. de la et al. Towards a model-based evolutionary chain of evidence for compliance with safety standards. In: *Computer Safety, Reliability, and Security: 31st International Conference, Safecomp 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 64–78.

[34] LARRUCEA, X. et al. Analyzing a ros based architecture for its cross reuse in iso26262 settings. In: *MEDI 2018: New Trends in Model and Data Engineering*. Cham, Switzerland: Springer International Publishing, 2018. p. 167–180.

[35] KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Durham, UK, 2007. 65 p.

[36] DENNEY, E.; PAI, G. Evidence arguments for using formal methods in software certification. In: *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. New York: IEEE, 2013. p. 375–380.

[37] GROZA, A.; MARC, N. Consistency checking of safety arguments in the goal structuring notation standard. In: *10th International Conference on Intelligent Computer Communication and Processing (ICCP)*. New York: IEEE, 2014. p. 59–66.

[38] MURAM, F. U.; GALLINA, B.; RODRÍGUEZ, L. G. Preventing omission of key evidence fallacy in process-based argumentations. In: *11th International Conference on the Quality of Information and Communications Technology (QUATIC)*. New York: IEEE, 2018. p. 65–73.

[39] ŠLJIVO, I. et al. A method to generate reusable safety case argument-fragments from compositional safety analysis. *Journal of Systems and Software*, New York, v. 131, p. 570–590, sep. 2017.

[40] PANESAR-WALAWEGE, R. K.; SABETZADEH, M.; BRIAND, L. Using uml profiles for sector-specific tailoring of safety evidence information. In: *Conceptual Modeling – ER 2011*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 362–378.

[41] CÂRLAN, C. et al. Explicitcase: Integrated model-based development of system and safety cases. In: *The 36th International Conference on Computer Safety, Reliability and Security*. Cham, Switzerland: Springer International Publishing, 2017. p. 52–63.

[42] SUN, L.; SILVA, N.; KELLY, T. Rethinking of strategy for safety argument development. In: *The 33rd International Conference on Computer Safety, Reliability and Security*. Cham, Switzerland: Springer International Publishing, 2014. p. 384–395.

[43] MARTINS, L. E. G. a.; OLIVEIRA, T. de. A case study using a protocol to derive safety functional requirements from fault tree analysis. In: *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. New York: IEEE, 2014. p. 412–419.

[44] ARANDA, A.; DIESTE, O.; JURISTO, N. Evidence of the presence of bias in subjective metrics: Analysis within a family of experiments. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2014. p. 1–4.

[45] SOBRINHO, A. et al. Formal modeling of biomedical signal acquisition systems: source of evidence for certification. *Software & Systems Modeling*, Heidelberg, v. 18, n. 2, p. 1467–1485, apr. 2019.

[46] DECHEV, D.; STROUSTRUP, B. Model-based product-oriented certification. In: *16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*. New York: IEEE, 2009. p. 295–304.

[47] ROMANSKI, G. Combined safety and security certification. In: *7th IET International Conference on System Safety, incorporating the Cyber Security Conference*. New York: IEEE, 2012. p. 1–5.

[48] VARA, J. L. de la et al. An analysis of safety evidence management with the structured assurance case metamodel. *Computer Standards & Interfaces*, Amsterdam, v. 50, p. 179–198, feb. 2017.

[49] LIN, H. et al. A systematic approach for safety evidence collection in the safety-critical domain. In: *2015 Annual IEEE Systems Conference (SysCon) Proceedings*. New York, USA: IEEE, 2015. p. 194–199.

[50] FALESSI, D. et al. Planning for safety standards compliance: A model-based tool-supported approach. *IEEE Software*, Los Alamitos, USA, v. 29, n. 3, p. 64–70, may 2012.

[51] PANESAR-WALAWEGE, R. K. et al. Characterizing the chain of evidence for software safety cases: A conceptual model based on the iec 61508 standard. In: *Third International Conference on Software Testing, Verification and Validation*. New York: IEEE, 2010. p. 335–344.

[52] LUO, Y. et al. From conceptual models to safety assurance. In: YU, E. et al. (Ed.). *ER 2014: Conceptual Modeling*. Cham, Switzerland: Springer International Publishing, 2014. p. 195–208.

[53] GANNOUS, A.; ANDREWS, A.; GALLINA, B. Toward a systematic and safety evidence productive verification approach for safety-critical systems. In: *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. New York: IEEE, 2018. p. 329–336.

[54] LOVRIĆ, T. Requirements for the certification of safety critical railway systems. In: WIECZOREK, M.; MEYERHOFF, D. (Ed.). *Software Quality: State of the Art in Management, Testing, and Tools*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 225–240.

[55] BERTIERI, D. et al. Development and validation of a safe communication protocol compliant to railway standards. *Journal of the Brazilian Computer Society*, Heidelberg, Germany, v. 27, n. 1, p. 1–26, mar. 2021.

[56] ROMANSKI, G. Certification of an operating system as a reusable component. In: *Proceedings. The 21st Digital Avionics Systems Conference*. New York: IEEE, 2002. p. 5D3–5D3.