Unveiling the Safety Aspects of DevSecOps: Evolution, Gaps and Trends

Xhesika Ramaj[a], Mary Sánchez-Gordón[*a], Sabarathinam Chockalingam[b], and Ricardo Colomo-Palacios[a]

[a] *Department of Computer Science and Communication, Faculty of Computer Sciences, Engineering and Economics, Østfold University College, Halden, Norway;* [b] *Department of Risk, Safety and Security, Institute for Energy Technology, Halden, Norway*

Abstract:

Background: The popularity of DevSecOps is on the rise because it promises to integrate a greater degree of security into software delivery pipelines. However, there is also an unacceptable risk related to safety that cannot be overlooked, given the importance of this aspect in many industries.

Objective: The objective of this study is to provide an overview of the safety aspects reported in the literature on DevSecOps. This study also characterizes such aspects and identifies the gaps that may lead to future research work.

Method: A systematic literature review was conducted using five well-known academic databases. The search was executed in September 2021 and March 2022 to identify relevant studies.

Results: The search returned 114 academic studies. After the screening process, five primary studies published between 2019 and 2021 were selected. These studies were analyzed thoroughly to identify the safety aspects. Then, we categorized them into three main groups: (i) risk-related safety aspects, (ii) human-related aspects, and (iii) management aspects.

Conclusion: Safety is an important characteristic that is becoming more critical as the number of critical systems grows. This review reveals that only a scarce number of studies are focusing on safety in DevSecOps. However, those studies gave us some insights into this topic. Therefore, our main observation is that this topic has not yet been completely explored in the academic literature. This review can encourage reflection and discussion between the safety and security communities.

Keywords: DevSecOps, Safety, Security, Risk, Human factors, Systematic literature review

1. INTRODUCTION

Safety is of paramount importance in critical sectors of society such as defense, energy, healthcare, and transportation. Those are safety-critical domains in which software is increasingly being used to provide functionality [1], e.g., medical devices for diagnostic or treatment purposes [2]. Although these new devices offer high benefits, there are also safety risks that cannot be overlooked [3]. In these domains, software failures can result in serious injuries or human fatalities, e.g., Mars Polar Lander, the Patriot missile, and the Therac-25 radiation deaths [4].

The severity of potential failures related to safety has led to highly regulated environments in which approval procedures and certification are mandated by law, i.e., they comply with an appropriate standard. Such a certificate allows the organization to sell products on the market [5].

For instance, the US Food and Drug Administration (FDA) has to approve medical devices in the United States [5] and the development of railway applications in Europe has to fulfill the EN 50128 standard [6]. There are also several ISO standards, process models, and process maturity models that ensure that failures related to the safety of the software are avoided [1]. However, how safety-critical systems must be developed remains a complex problem.

Safety-critical software development has followed software industry trends and community [2]. Traditional software development processes such as waterfall have been adopted in response to the challenges of developing safety-critical software. Over time, changing requirements and an increasing need for short development cycles and quick time to the market have led to the use of agile processes in safety-critical software development [3], [7]. In recent years, big corporations such as Google, Apple, and Amazon have disrupted the automotive market, leading to a growing need to develop competencies in continuous software engineering (CSE) [7]. In particular, DevOps attempts at overcoming the lack of collaboration and communication between development and operations when implementing continuous integration (CI) and continuous deployment (CD) [8], [9]. DevOps has become a prominent trend that integrates development, delivery, and operations [8]. Therefore, DevOps has also been tailored for regulated development [2], [10], [11].

Given that security concerns are becoming a hot topic in several domains [12], including safety-critical domains [13], safety in the context of DevSecOps is relevant. A connected safety-critical system can only be considered safe when it is secure at the same time [14]. According to the state of DevOps report 2021 [15], DevSecOps is an explicit call to

action to “shift the left” of security —security is an integral part of the software development lifecycle from the beginning. However, there are other labels used in the industry like SecDevOps, DevOpsSec, Secure DevOps, or Rugged DevOps.

In light of that, although some secondary studies about DevOps exist (see Section 2.3), to the best of our knowledge, an overview of the state-of-the-art on safety of DevSecOps is not available in the literature. Therefore, we aim to bridge this gap by reviewing research on this field and provide a mapping of the safety aspects reported in the literature. In this study, we follow the guidelines for systematic literature review in software engineering proposed by Kitchenham et al. [16].

The structure of the remaining paper is as follows: Section 2 is focused on providing an essential foundation on safety and DevSecOps in addition to related work on secondary research studies in DevSecOps. Section 3 briefly describes our approach to conduct this review. Section 4 reports the findings of this review while Section 5 discusses them in light of the previous literature on the topic. Finally, Section 6 presents conclusions and potential future research directions.

2. BACKGROUND AND RELATED WORK

In this section, we provide an overview of the two main concepts that frame this review: safety and DevSecOps. Finally, we also outline secondary research studies related to DevSecOps.

2.1. Safety

The terms security and safety should not be confused even though in some languages like Spanish or German, they share the same word [17]. The International Organization for Standardization (ISO) defines safety as the freedom from unacceptable risk [18]. Moreover, risk is defined as the effect of uncertainty on objectives. On the contrary, security is defined as resistance to an intentional, unauthorized act(s) designed to cause harm or damage to a system [18]. In the USA, the National Institute of Standards and Technology (NIST) defines security as a state in which an organization can perform its mission and critical functions despite threats to its system utilization [19].

It means that both terms deal with risk, however, the origin of risk allows a clear distinction between safety and security [20]. Security risk is intentional, while safety risk is unintentional. Safety considers hazards, e.g., system failures or other accidental conditions, while security considers threats and potential attacks [20]. The nature of risk consequences differs as well [21]. Safety risk has a potential impact on the system environment, while security risk on the system itself [22]. Finally, the ways in which safety and security risk is assessed differ. In the case of security risk assessment, the sources of the threats to be examined are typically unknown to the analyst and cover a wide range of probable scenarios. On the other hand, safety risk assessment considers a limited number of scenarios, and accessible hazards [17].

Despite their differences, safety and security have similarities. Two decades ago, Eames and Moffett [23] pointed out that (i) both safety and security deal with risks, (ii) they result in constraints and require protective measures, and (iii) both create requirements. In 2015, Kriaa et al. [17] discussed four types of potential interactions between safety and security: (i) no interaction, (ii) conditional dependency, (iii) mutual reinforcement, and (iv) antagonism. More recently, in 2020, a systematic literature review (SLR) on safety and security co-analysis [14] highlights that bringing security and safety together is still challenging.

The traditional software development approaches for critical systems have been established on a principle of separation between development and operation that restricts and controls changes to ensure safety [24]. However, new market demands call for short development cycles with unclear or changing requirements that challenge the stability and repetition of safety activities [7], [24].

2.2. DevSecOps

DevSecOps stands for DEVelopment, SECurity, and OPerationS. The term was first introduced by Neil MacDonald in 2012, to emphasize the need to incorporate security into DevOps [9]. Since then, DevSecOps has not only aroused the concerns of DevOps practitioners but also DevSecOps has been increasingly recognized as a necessity [25].

According to [15], there is no agreement about the relationship between DevOps and DevSecOps, but two opposite views exist already. The first one claims that DevSecOps should not exist as a separate label since security is part of DevOps. On the other hand, DevSecOps drives the change to integrate security with DevOps practices, given that many practitioners have taken the label DevOps literally.

In the literature reviewed by Myrbakken and Colomo-Palacios [9], DevSecOps is seen as a necessary addition to DevOps that aims at the integration of security controls and processes into the DevOps software development lifecycle by promoting collaboration among development teams, security teams, and operation teams. Therefore, DevSecOps also integrates security as a part of the culture. Security culture in DevSecOps contributes to adopting a different way of working that highlights cross-team collaboration with a clear focus on security [13].

Myrbakken and Colomo-Palacios [9] identified five practices in DevSecOps: threat modeling and risk assessment, continuous testing, monitoring and logging, security code, red-team, and security drills. Moreover, they identified three benefits: security, shifting security to the left, and automating security. Particularly, the last one helps to reduce risk, save time and facilitate understanding risk and create policies and procedures. In this way, DevSecOps helps to ensure that security is implemented at the right level and at the right time [26].

An essential task in DevSecOps is tool integration, however, practitioners find it challenging because it is a manual and time-consuming task [25]. Given that tools enable automation, they are a critical element in continuous practices and DevSecOps, just like in DevOps. In this scenario, the automation and efficiency of security practices may be the cornerstones of DevSecOps [27].

2.3. Related work

An initial search was conducted to identify other secondary research studies related to DevSecOps. We found six studies that review the literature on different aspects of DevSecOps.

Mohan and Ben Othman [28] conducted a systematic mapping on DevSecOps to identify its definition and other aspects related to security best practices, process automation, tools, compliance, team collaboration, availability of activity data, and information secrecy. Only 8 out of 66 artifacts were identified as relevant and 6 of those were academic research papers. The authors conclude that the variety of these aspects suggests that SecDevOps is not a buzzword.

Myrbakken and Colomo-Palacios [9] conducted a multivocal literature review (MLR) to provide DevSecOps definition, challenges, benefits, and evolution. Their review included 8 out of 52 artifacts and only 2 of those were academic research papers. This was the first MLR on this topic and the results contributed to identifying challenges and benefits as well as to highlight the evolution of the concept.

Prates et al. [29] conducted a MLR to identify metrics for measuring the effectiveness of applying the DevSecOps methodology. After screening 296 artifacts, 11 were included in their review. They reported nine metrics: defect density, defect burn rate, critical risk profiling, top vulnerability types, adversary return rate, point of risk per device, number of adversaries per application, number of continuous delivery cycles per month, and number of issues during red teaming drills.

Sanchez-Gordon and Colomo-Palacios [13] carried out a SLR on the cultural aspect of DevSecOps. From the 148 search results, they included 11 studies in their review. The findings were categorized into 13 attributes. The three most cited were collaboration, sharing knowledge, and feedback.

Rafi et al. [30] conducted a SLR to explore security challenges. Their review included 44 out of 110 studies retrieved in the initial search. As a result, 18 security challenges were identified and evaluated by experts to develop a taxonomy of security challenges using PROMETHEE.

Mao et al. [25] conducted a grey literature review to report the state of practice on DevSecOps. Out of 141 artifacts, three major software security risks were identified along with security opportunities.

Rajapakse et al. [27] carried out a SLR on the challenges and solutions when practitioners adopt DevSecOps. By screening 460 studies, 48 of them were included as primary studies. After snowballing, six further studies were also included. Then, 21 challenges and 31 proposed solutions were identified. Findings were classified into four themes: people, practices, tools, and infrastructure.

These reviews present important insights into some aspects of DevSecOps, but they are not focused on safety. In fact, none of them mentioned safety. However, four reviews [25], [27], [29], [30] gave us a glimpse of the risk aspects that will be discussed in Section 5. The goal of the present study is to provide an overview of the safety aspects reported in the literature on DevSecOps.

3. MATERIALS AND METHODS

This study was conducted following the guidelines for SLRs in software engineering proposed by Kitchenham and Charters [16]. In this section, we present our research goals, questions, and search strategy. Authors also outline how we filtered and selected the relevant studies followed by the corresponding data extraction process. Figure 1 shows an overview of the whole research process.

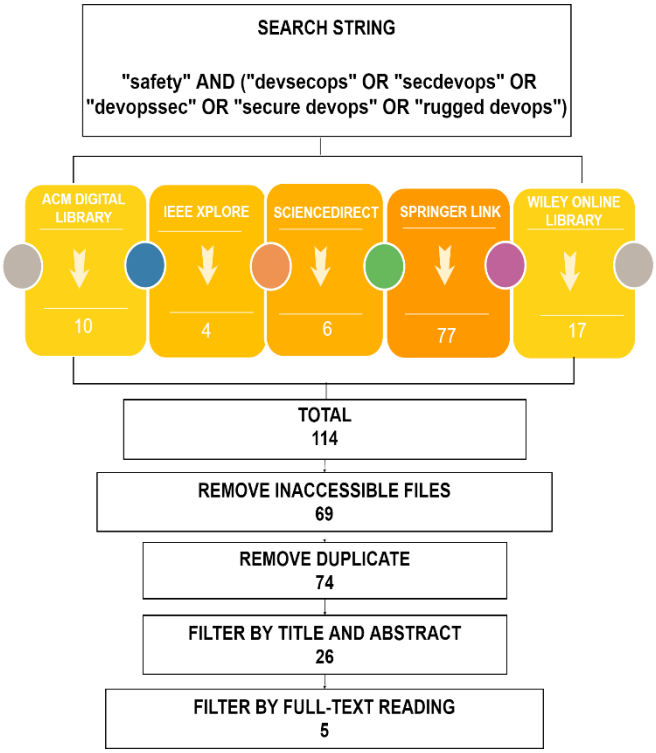

Figure **1.** The research process overview.

3.1 Research goals and questions

This SLR on safety in the context of DevSecOps is conducted keeping the following three specific objectives in mind: (i) identify the safety aspects reported in the scientific literature, (ii) reveal how the safety aspects are integrated into DevSecOps, and (iii) identify the evolution of this research field. Based on the above-mentioned objectives, we formulated the following research questions (RQs):

RQ1: What are the safety aspects reported in the scientific literature about DevSecOps? DevSecOps ensure security, however, safety is an important characteristic that is related to security. However, this aspect has not been addressed by previous secondary research studies about DevSecOps.

RQ2: How are the safety aspects integrated into DevSecOps? The integration of safety and security into DevSecOps is a need, particularly in safety-critical domains. This review would provide important insights to adequately address the unreasonable risk of harm caused by both malfunctioning and malicious intent.

RQ3: What is the evolution of the scientific literature on safety in the context of DevSecOps? DevSecOps is a growing research field that has appealing potential benefits. Therefore, it is important to explore the evolution of this topic.

3.2 Search strategy

The search string is based on two keywords: *safety* and *DevSecOps*. Both these keywords are directly related to the subject of this review and allow us to identify relevant studies that address safety in the context of DevSecOps. However, there are also other terms for DevSecOps that can

be used as synonyms, namely: *DevOpsSec, SecDevOps, Secure DevOps, or Rugged DevOps* [31]. To build the search string, synonyms were joined with OR and the keywords were joined with AND. Finally, we specify the following search string:

"safety" AND ("devsecops" OR "secdevops" OR "devopssec" OR "secure devops" OR "rugged devops")

In line with best practices [16], the following five academic databases have been used to conduct searches: ACM Digital Library, IEEE Xplore, Wiley Online Library, SpringerLink, and ScienceDirect. The final search string was executed on the aforementioned databases during September – 2021. The search was not limited by the date of publication. Table 1 shows the summary of the results in each database.

Table **1**. Summary of the first search process.

Database	Initial search	Title, abstract, keywords	Full text
ACM Digital Library	10	4	1
IEEE Xplore	3	3	1
ScienceDirect	5	2	1
Springer Link	39	11	1
Wiley	15	1	-
TOTAL	**72**	**21**	**4**

All identified studies were stored in a reference manager, namely, Zotero. After retrieving the basic information from each source, we conducted a duplication-identifying process. Only two duplicates were identified.

3.3 Selection criteria

The selection criteria of this study aim to identify those studies that answer the research questions posed in this review [16]. The studies were selected based on a set of inclusion and exclusion criteria, as shown in Table 2.

Table **2.** Inclusion and Exclusion Criteria.

Criteria	Inclusion	Exclusion
Exposure	- Studies explicitly focused on DevSecOps - Studies that explicitly identify/address at least a safety aspect in the context of DevSecOps	- Studies not addressing safety in DevSecOps
Period	- The search was not limited by the date of publication.	
Language	- Studies that are written in English	
Accessibility	- All accessible studies, without including duplicated studies	- Inaccessible studies - Duplicated studies

Application of the inclusion and exclusion resulted in some *drawbacks*, which we have classified according to selection criteria as follows:

Exposure: Only a few studies explicitly mention the term "safety".
Period: The period was not limited although DevSecOps was first mentioned in 2012. We found a scarce number of studies published between 2012 and early 2022 (first trimester).
Language: There may be studies in other languages that are relevant, e.g., the German language.
Accessibility: There may be studies that are relevant but cannot be accessed.

The initial search returned 72 studies. Then, we excluded (3) inaccessible studies and (2) duplicated studies. One author selected (21) relevant studies by screening title and abstract screening. A second author confirmed the selection of eligible studies. Table 1 shows the summary of results after each phase of the selection process. For each study, notes were kept regarding the decision. In case of any doubt during the first-level screening, the corresponding study was included for full-text screening. Then, the full text of each selected study was screened, and 15 studies were excluded. Therefore, four studies were included as the primary studies (see Table 4). Most of the studies were excluded because safety is simply mentioned by passing, e.g., [32], [33].

Due to the scarce number of studies selected, two authors conducted a second search process after six months (October 2021 to March 2022). The search string was executed in the same databases and retrieved a total of 42 studies. Most of the studies came from Springer Link, with only four studies published in the other databases (see Table 3).

Table **3.** Summary of the second search process.

Database	2nd search	Title, abstract, keywords	Full text
ACM Digital Library	-	-	-
IEEE Xplore	1	1	-
ScienceDirect	1	1	-
Springer Link	38	5	1
Wiley	2	-	-
TOTAL	**42**	**7**	**1**

Seven books were removed since there were their book chapters, i.e., the book they belong to was excluded. As a result, a total of 35 studies were selected for further review. Then, seven studies were chosen for full text reading and one study [38] was identified as relevant after reading the title, abstract, and keywords (see Table 4).

Table **4.** List of primary studies.

Ref.	Year	Title / Authors
[34]	2019	A systems-of-systems security framework for requirements definition in cloud environment Carturan, Sara B. O. Gennari; Goya, Denise Hideko
[35]	2019	Airline Application Security in the Digital Economy: Tackling Security Challenges for Distributed Applications in Lufthansa Systems Somoskői, Balázs; Spahr, Stefan; Rios, Erkuden; Ripolles, Oscar; Dominiak, Jacek; Cserveny, Tamás; Bálint, Péter; Matthews, Peter; Iturbe, Eider; Muntés-Mulero, Victor
[36]	2020	Usability Testing within a Devsecops Environment Burkard, Emerson Czerwinski
[37]	2020	Assurance for CyberPhysical Systems: Addressing Supply Chain Challenges to

		Trustworthy Software-Enabled Things Martin, Robert Alan
[38]	2021	Security Issues in Android Application Development and Plug-in for Android Studio to Support Secure Programming Tran, Anh-Duy; Nguyen, Minh-Quan; Phan, Gia-Hao; Tran, Minh-Triet

3.4 Data extraction strategy and process

The selection and data extraction process were done using a data extraction form. Such a form was created to capture details from the data sources including bibliographic information and relevant information to answer the above-mentioned research questions.

The bibliographic information was automatically extracted using Zotero. One author conducted this data extraction. The information is descriptive information about each study, such as title, authors' names, type of publication (conference/journal), year of publication, number of pages, keywords, and abstract. As some inconsistencies were found during the automatic extraction, the extracted information was updated manually. A second author confirmed this data. Moreover, two authors independently extracted data from selected studies to answer the research questions. Due to the limited number and focus of this review, a narrative synthesis of the selected studies was conducted. Extracted data were presented according to the research questions.

4. RESULTS

The first observation from our SLR is that only 5 out of 114 studies passed our selection criteria which represent studies on safety in the context of DevSecOps. This section presents our findings grouped by each research question.

4.1. RQ1: What are the safety aspects addressed or investigated in the literature about DevSecOps?

Although the main focus of the selected studies is not safety, they mention safety as an important characteristic. Table 5 presents the main safety aspects identified in this review.

Table **5:** Mapping of safety aspects and selected studies.

Category	Ref.	Safety aspect
Risk	[34]	Risk identification
	[34]	Risk monitoring
	[34], [37]	Risk mitigation
	[36]	Risk reduction
	[38]	Risk prevention
	[35]	Risk assessment
	[35]	Risk analysis tools
	[37], [38]	Assurance cases
Human	[36]	Frequent Feedback
	[36], [38]	Minimize use errors
	[38]	Users' safety
	[34], [35]	Cultural and behavioral changes
Management	[35]	Change work and responsibilities
	[34]	Review of security environments
	[38]	Secure software development
	[34]	Review of standards

Carturan and Goya [34] state that organizations must establish a strategy for risk management by *identifying* the information assets and understanding their vulnerabilities. Moreover, they must identify the current (and future) threats that may put them at risk by exploiting such vulnerabilities. Therefore, constant *monitoring* is important, too. Carturan and Goya also point out the need to have a solution that goes beyond *risk mitigation* to improve service to clients. In cloud environments, to balance, empower users, or minimize risk (safety) apart from technical skills, *cultural and behavioral changes* (human factors) are needed [34]. Although security in the operational reference model and IT process model is discussed, safety is not explicitly mentioned. A safe operation of software requires a check on the environment settings, *review of security environments*, and *review of standards* [34]. Furthermore, it needs to comply with a defined governance process.

Somoskői et al. [35] state that a safer environment requires changes not only in the processes conducted or the technology used but also in architecture and organizational *culture*. Implementing a DevSecOps approach is a cultural change in organizing teams, work, and responsibilities. Moreover, the importance of taking security decisions based on sound *risk analysis* is highlighted. Somoskői et al. [35] also point out a *risk assessment* and *risk mitigation*, but it is not clear to what extent safety is considered. However, it is worth noting that two of the ten authors are safety experts, according to the information provided.

Burkard [36] states that a smaller *feedback* loop with frequent deployment *reduces risk* by increasing opportunities for realignment. This idea of "safe change" is consistent with the general purpose of *risk reduction.* Subjective feedback from users led to prioritizing and reviewing critical *errors.* Tran et al. [38] suggest also actions that developers can perform to minimize the *errors* and *prevent risks.* For example, setting permissions for each activity. In this way, you ask for permission each time you want to call such an activity.

Martin [37] states that safety involves risks associated with connectivity. These risks entail a loss of safety from an operational risk viewpoint. He also points out that a methodology for trustworthiness across a marketplace requires building *assurance cases*. Moreover, he lists some *mitigation* strategies for attack patterns.

Tran et al. [38] identify the main causes of unsafe Android software related to a lack of security in software development process or a delay of traditional security *assurance* methods. They propose that Android developers need a *secure software development* process to ensure the safety of users when using their apps. They must adhere to a secure development process to counter Android application *risk*. Furthermore, authors state that security issues and secure programming are unavoidable aspects in ensuring the safety of Android applications while maintaining development speed.

4.2. RQ2: How are the safety aspects integrated into DevSecOps?

Carturan and Goya [34] conducted a project that provides empirical evidence on the viability of safe cloud environment usage. As a result, they propose a System-of-

Systems (SoS) security framework for requirements definition in a cloud environment. To do so, they identified security aspects that should be included by using a checklist and some questions. Bearing in mind the perspectives of the existing IT Governance Model, IT Operational Model, and IT Processes, security drivers to integrate cloud computing in a SoS context were also identified. In this proposal, "deployment and safe operation" is one of the safe components of the SoS Security Governance Model.

Somoskői et al. [35] use the MUSA framework to: (i) identify the cloud providers that best fulfilled application security requirements based on a new mechanism for assessing risk using agile approaches, (ii) create Service Level Agreements (SLAs) based on the detected security requirements, (iii) automatically deploy the prototype application components using the identified providers, and (d) monitoring the application aspects related to security and to control compliance with agreed SLAs.

Burkard [36] uses usability testing to ensure safety and effectiveness. Identifying security issues at the earliest stage means enhancing the quality of software and reducing the risk. Bringing end-users into the feedback loop is seeming also as the logical step for fixing the mismatch between end-user needs and the teams' interpretation of the product. This could be achieved by increasing the frequency check in every stage of software creation and protection. This process involves inviting end-users to utilize the main features, ensuring not only effectivity but safety as well. The frequent style checking and testing of DevSecOps is aligned with the philosophy of usability testing, allowing in this way a harmonized integration with each other.

Martin [37] provides insights into assurance for cyber-physical systems. This study highlights that software is enabled and connected, therefore it should be trustworthy (not just secure). In the light of trustworthiness, safety, privacy, resilience, reliability, and security behaviors of systems are all interacting, although such interaction is not always the same in proportion. Then, a composition of assurance cases is presented. Martin also mentions that gathering and sharing evidence-based standards is as important as selecting appropriate testing and assessment methods.

Tran et al. [38] summarize the common security issues in Android applications and develop a plug-in for Android Studio to support secure programming, called 9Fix. Authors state that integration of security throughout the application development process will secure the development life cycle and provide safe Android software. 9Fix plug-in can inspect vulnerable code and prevent the risk. This is achieved by instantly suggesting an alternative secure code for developers during their programming time. Such a solution will not only help to improve the security but also instruct the developers on how to write a secure code.

4.3. RQ3: What is the evolution of the scientific literature on safety in the context of DevSecOps?

As mentioned before, safety is not the main focus of the selected studies. It means that the academic literature is not addressing safety in the context of DevSecOps or at least is not explicitly distinguished from security. It seems that potential interactions between safety and security are in place. DevSecOps is a rising research field that has appealing potential benefits, but the area related to safety has not been explored in the academic literature yet. The consequence of this fact is the scarce number of papers on the topic and, with that, the impossibility to draw an evolution of the topic taking into account literature. Thus, RQ3 cannot be answered.

5. DISCUSSION AND LIMITATIONS

Despite that the search was not limited by the date of publication, we found only a scarce number of studies that are published between 2012 and early 2022. Although it seems to be reasonable since DevSecOps was first mentioned in 2012, it is worth noting that safety is not the main focus of these publications. Only a few studies explicitly mention the term "safety".

Safety is an important characteristic that organizations have to consider. They have to make sure to protect themselves, their assets, and customers regardless of the domain on which they operate, e.g., financial [34], aviation [35], [36], software development [38], or cyber-physical systems [37]. While some organizations tend to focus on the loss of data and services they offer, another aspect to be considered is the loss of safety itself [39].

As more organizations in safety-critical domains adopt DevSecOps, the necessity to investigate safety aspects in DevSecOps increases. An issue of high interest is to understand how organizations can integrate these safety aspects while adopting DevSecOps in their working environments.

According to the definition of ISO 23643 [34], the main safety aspects to consider are those that influence the level of risk, such as risk identification, monitoring, and mitigation. In the context of DevSecOps, automation can reduce downtime on a given product, which can reduce the deployment risk [36]. In turn, frequent deployment is one of the ways to reduce risk since it increases the opportunities for realignment. Frequent deployment together with continuous feedback can contribute to assure that every change is a "safe" one. Therefore, *risk reduction and frequent feedback* are other relevant aspects.

Automation is taking place in many organizations by implementing DevOps and DevSecOps, however, there are a lot of processes that rely on humans and manual work. In fact, human factors have been reported as one of the biggest vulnerabilities that require high consideration at all levels [40], e.g., executive, managerial, or operational [34].

Software practitioners can use the categorization of safety aspects to create a priority list or indicators to evaluate the impact of these aspects in the software development process. In some cases, safety engineers must have active participation in the software development lifecycle.

Previous secondary studies also provide some insights into the risk aspects.

Rajapakse et al. [27] identify the limitations of dynamic analysis (DAST) tools as one of the reasons that restrict their usage in DevSecOps. For instance, the use of dynamic analysis tools allows us to identify a vast number of vulnerabilities, but it requires the execution of the software,

i.e., building, installing, and configuring it [41]. In a DevSecOps environment, the code is frequently released and those tasks are not trivial [27].

Rafi et al. [30] identify "lack of automated testing tools"' and "lack of secure coding standards" as the most critical challenges related to risk. To monitor and control the security risks, there should be adequate automation testing tools, and adequate security implementations and countermeasures.

Prates et al. [29] identify five metrics explicitly related to risk. *Critical risk profiling* is the relation between issue criticality and the value of that vulnerability to attackers. *Top vulnerability types* list the top vulnerability types and the most recurring ones. *The point of risk per device* tracks the number of vulnerabilities per server. The last two metrics are *the number of adversaries per application* and *the adversary return rate*.

Mao et al. [25] analyze the impacts of risks to security by identifying three major challenges (i) sacrifice of security for speed/agility, (ii) afterthought in the process, and (iii) environment risk. However, these authors also point out that many DevOps practices provide fertile ground for integrating security and audit capabilities as a built-in component of DevOps processes.

This review, as any other secondary study, faced limitations and threats to validity. In what follows, authors overview the main limitations of the study conducted.

As mentioned before, the major limitation of this study is the scarce number of primary studies that discuss safety in the context of DevSecOps. Such a low number of studies was a result of both the first and the second search process. Studies that did not use any of the search terms defined in the protocol may not have been found, e.g., studies that implement risk mitigation but did not explicitly use the word "safe". However, the limited findings reveal that safety in the context of DevSecOps is a research area to be explored.

Another common limitation in an SLR is related to bias in study selection and data extraction. A protocol was defined to reduce this bias. Moreover, a data extraction sheet based on a data extraction form was created. However, it is assumed that nonwritten knowledge could be substantial and should be captured by other methods.

6. CONCLUSION

This SLR provides an overview of safety in the context of DevSecOps. It was conducted by performing the guidelines for systematic literature reviews in software engineering, proposed by Kitchenham and Charters [16]. Although we found a scarce number of studies reported in the literature, we categorized our findings into the following three main categories: risk, human, and management.

Risk-related: Incorporates all processes related to risk, for instance, risk identification, monitoring, mitigation, reduction, assessment, and automation of risk analysis (tools).

Human-related: This group covers issues related to people from all levels considered as security drivers, e.g., executives, managers, key users, and technicians related to security; frequent feedback; minimal use error.

Management-related: Includes those managing processes of safety, e.g., operational safety, review of security environments, and review of standards

Despite the limitations, our findings bring some insights that we hope can encourage reflection and discussion between the safety and security communities.

Current & Future Developments

Current developments are reported in industries such as aviation, financial, and cyber-physical systems. We hope this review fosters further research on safety aspects in the context of DevSecOps and other critical domains such as health.

It could be interesting to explore the documentation of the integration process of safety aspects. The documentation process helps in analyzing the different situations that an organization may face. It facilitates risk analysis and may assist in creating best practices in this area.

Further research is also needed on how the safety aspects can be integrated automatically into DevSecOps pipelines. Therefore, the development of continuous safety assessment tools could be another research line along with the development of metrics to evaluate the results generated during the integration of safety aspects into DevSecOps.

This review can encourage reflection and discussion between the safety and security communities. For instance, to what extent the safety aspects could be automated into a pipeline or reduce the agility of the processes is not clear.

STANDARDS OF REPORTING

PRISMA guideline has been followed.

FUNDING

This paper is partially funded by the Research Council of Norway (RCN) in the INTPART program under the project "Reinforcing Competence in Cybersecurity of Critical Infrastructures: A Norway-US Partnership (RECYCIN)" with the project number #309911 and by the project "User-centred Security Framework for Social Robots in Public Space", project number #321324. Moreover, the first author is supported by a scholarship from Østfold University College, Halden, Norway.

ACKNOWLEDGEMENTS

This paper is partially funded by the Research Council of Norway (RCN) in the INTPART program under the project "Reinforcing Competence in Cybersecurity of Critical Infrastructures: A Norway-US Partnership (RECYCIN)" with the project number #309911 and by the project "User-centred Security Framework for Social Robots in Public Space", project number #321324. Moreover, the first author is supported by a scholarship from Østfold University College, Halden, Norway.

REFERENCES

[1] A. B. Bujok, S. T. MacMahon, P. Grant, D. Whelan, W. J. Rickard, and F. McCaffery, "Approach to the development of a Unified Framework for Safety Critical Software Development", *Comput. Stand. Interfaces*, Vol. 54, pp. 152–161, Nov. 2017. DOI: 10.1016/j.csi.2016.11.013.

[2] M. F. Lie, M. Sánchez-Gordón, and R. Colomo-Palacios, “DevOps in an ISO 13485 Regulated Environment: A Multivocal Literature Review”, in *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, New York, NY, USA, Oct. 2020, pp. 1–11. DOI: 10.1145/3382494.3410679.

[3] L. T. Heeager and P. A. Nielsen, “A conceptual model of agile software development in a safety-critical context: A systematic literature review”, *Inf. Softw. Technol.*, Vol. 103, pp. 22–39, Nov. 2018. DOI: 10.1016/j.infsof.2018.06.004.

[4] P. A. McQuaid, “Software disasters—understanding the past, to improve the future”, *J. Softw. Evol. Process*, Vol. 24, No. 5, pp. 459–470, 2012. DOI: 10.1002/smr.500.

[5] U.S. FDA, “FDA Agents - FDA Registration and U.S. Agent Representation’. https://www.fdaagents.com/ (accessed Sep. 24, 2021).

[6] ‘EN 50128 Railway applications - Communication, signalling and processing systems’. European Committee for Electro-technical Standardization, 2012.

[7] R. Kasauli, E. Knauss, B. Kanagwa, A. Nilsson, and G. Calikli, “Safety-Critical Systems and Agile Development: A Mapping Study”, in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Prague, Aug. 2018, pp. 470–477. DOI: 10.1109/SEAA.2018.00082.

[8] M. Sánchez-Gordón and R. Colomo-Palacios, “Characterizing DevOps Culture: A Systematic Literature Review”, in *Software Process Improvement and Capability Determination*, Cham, 2018, pp. 3–15. DOI: 10.1007/978-3-030-00623-5_1.

[9] H. Myrbakken and R. Colomo-Palacios, “DevSecOps: A Multivocal Literature Review”, in *International Conference on Software Process Improvement and Capability Determination*, 2017, pp. 17–29.

[10] T. Laukkarinen, K. Kuusinen, and T. Mikkonen, “Regulated software meets DevOps”, *Inf. Softw. Technol.*, Vol. 97, pp. 176–178, May 2018. DOI: 10.1016/j.infsof.2018.01.011.

[11] M. Olszewska and M. Waldén, “DevOps meets formal modelling in high-criticality complex systems”, in *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, Bergamo Italy, Sep. 2015, pp. 7–12. DOI: 10.1145/2804371.2804373.

[12] X. Larrucea, A. Berreteaga, and I. Santamaria, “Dealing with Security in a Real DevOps Environment”, in *Systems, Software and Services Process Improvement*, Cham, 2019, pp. 453–464. DOI: 10.1007/978-3-030-28005-5_35.

[13] M. Sánchez-Gordón and R. Colomo-Palacios, “Security as Culture: A Systematic Literature Review of DevSecOps”, in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, Seoul Republic of Korea, Jun. 2020, pp. 266–269. DOI: 10.1145/3387940.3392233.

[14] E. Lisova, I. Šljivo, and A. Čaušević, “Safety and Security Co-Analyses: A Systematic Literature Review”, *IEEE Syst. J.*, Vol. 13, No. 3, pp. 2189–2200, Sep. 2019. DOI: 10.1109/JSYST.2018.2881017.

[15] ‘State of DevOps Report 2021”, 2021.

[16] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering’. 2007.

[17] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, “A survey of approaches combining safety and security for industrial control systems”, *Reliab. Eng. Syst. Saf.*, Vol. 139, pp. 156–178, Jul. 2015. DOI: 10.1016/j.ress.2015.02.008.

[18] ‘ISO/IEC 23643:2020(en), Software and systems engineering — Capabilities of software safety and security verification tools’. https://www.iso.org/obp/ui/fr/#iso:std:iso-iec:23643:ed-1:v1:en (accessed Sep. 28, 2021).

[19] C. Paulsen and R. Byers, “Glossary of Key Information Security Terms”, National Institute of Standards and Technology, NIST Internal or Interagency Report (NISTIR) 7298 Rev. 3, Jul. 2019. DOI: 10.6028/NIST.IR.7298r3.

[20] A. J. Kornecki and M. Liu, “Fault Tree Analysis for Safety/Security Verification in Aviation Software”, *Electronics*, Vol. 2, No. 1, Art. no. 1, Mar. 2013. DOI: 10.3390/electronics2010041.

[21] L. Piètre-Cambacédès and M. Bouissou, “Cross-fertilization between safety and security engineering”, *Reliab. Eng. Syst. Saf.*, Vol. 110, pp. 110–126, Feb. 2013. DOI: 10.1016/j.ress.2012.09.011.

[22] L. Piètre-Cambacédès and C. Chaudet, “The SEMA referential framework: Avoiding ambiguities in the terms “security” and “safety”“, *Int. J. Crit. Infrastruct. Prot.*, Vol. 3, No. 2, pp. 55–66, Jul. 2010. DOI: 10.1016/j.ijcip.2010.06.003.

[23] D. P. Eames and J. Moffett, “The Integration of Safety and Security Requirements”, in *Computer Safety, Reliability and Security*, Vol. 1698, M. Felici and K. Kanoun, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 468–480. DOI: 10.1007/3-540-48249-0_40.

[24] C. Fayollas, H. Bonnin, and O. Flebus, “SafeOps: A Concept of Continuous Safety”, in *2020 16th European Dependable Computing Conference (EDCC)*, Munich, Germany, Sep. 2020, pp. 65–68. DOI: 10.1109/EDCC51268.2020.00020.

[25] R. Mao *et al.*, “Preliminary Findings about DevSecOps from Grey Literature”, in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, Macau, China, Dec. 2020, pp. 450–457. DOI: 10.1109/QRS51102.2020.00064.

[26] K. Carter, “Francois Raynaud on DevSecOps”, *IEEE Softw.*, Vol. 34, No. 5, pp. 93–96, 2017. DOI: 10.1109/MS.2017.3571578.

[27] R. N. Rajapakse, M. Zahedi, M. A. Babar, and H. Shen, “Challenges and solutions when adopting DevSecOps: A systematic review”, *Inf. Softw. Technol.*, Vol. 141, p. 106700, Jan. 2022. DOI: 10.1016/j.infsof.2021.106700.

[28] V. Mohan and L. B. Othmane, “SecDevOps: Is It a Marketing Buzzword? - Mapping Research on

Security in DevOps", in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Aug. 2016, pp. 542–547. DOI: 10.1109/ARES.2016.92.

[29] L. Prates, J. Faustino, M. Silva, and R. Pereira, "DevSecOps Metrics", in *Information Systems: Research, Development, Applications, Education*, Cham, 2019, pp. 77–90. DOI: 10.1007/978-3-030-29608-7_7.

[30] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, and A. Gumaei, "Prioritization Based Taxonomy of DevOps Security Challenges Using PROMETHEE", *IEEE Access*, Vol. 8, pp. 105426–105446, 2020. DOI: 10.1109/ACCESS.2020.2998819.

[31] A. A. U. Rahman and L. Williams, "Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices", in *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED)*, May 2016, pp. 70–76. DOI: 10.1109/CSED.2016.021.

[32] R. Chatterjee, "Security in DevOps and Automation", in *Red Hat and IT Security: With Red Hat Ansible, Red Hat OpenShift, and Red Hat Security Auditing*, R. Chatterjee, Ed. Berkeley, CA: Apress, 2021, pp. 65–104. DOI: 10.1007/978-1-4842-6434-8_3.

[33] X. Larrucea, A. Berreteaga, and I. Santamaria, "Dealing with Security in a Real DevOps Environment", in *Systems, Software and Services Process Improvement*, Cham, 2019, pp. 453–464. DOI: 10.1007/978-3-030-28005-5_35.

[34] S. B. O. G. Carturan and D. H. Goya, "A systems-of-systems security framework for requirements definition in cloud environment", in *Proceedings of the 13th European Conference on Software Architecture - ECSA '19 - volume 2*, Paris, France, 2019, pp. 235–240. DOI: 10.1145/3344948.3344977.

[35] B. Somoskői *et al.*, "Airline Application Security in the Digital Economy: Tackling Security Challenges for Distributed Applications in Lufthansa Systems", in *Digitalization Cases: How Organizations Rethink Their Business for the Digital Age*, N. Urbach and M. Röglinger, Eds. Cham: Springer International Publishing, 2019, pp. 35–58. DOI: 10.1007/978-3-319-95273-4_3.

[36] E. C. Burkard, "Usability Testing within a Devsecops Environment", in *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)*, Sep. 2020, pp. 1C1-1-1C1-7. DOI: 10.1109/ICNS50378.2020.9222919.

[37] R. A. Martin, "Assurance for CyberPhysical Systems: Adressing Supply Chain Challenges to Trustworthy Software Enabled-Things", presented at the 2020 IEEE Systems Security Symposium (SSS), Jul. 2020. DOI: 10.1109/SSS47320.2020.9174201.

[38] A.-D. Tran, M.-Q. Nguyen, G.-H. Phan, and M.-T. Tran, "Security Issues in Android Application Development and Plug-in for Android Studio to Support Secure Programming", in *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications*, Singapore, 2021, pp. 105–122. DOI: 10.1007/978-981-16-8062-5_7.

[39] 'Assurance and Sustainability", in *Security Engineering*, John Wiley & Sons, Ltd, 2020, pp. 1015–1058. DOI: 10.1002/9781119644682.ch28.

[40] T. Limba, T. Plėta, K. Agafonov, and M. Damkus, "Cyber security management model for critical infrastructure", *Entrep. Sustain. Issues*, vol. 4, No. 4, pp. 559–573, Jun. 2017. DOI: 10.9770/jesi.2017.4.4(12).

[41] J. A. Kupsch, B. P. Miller, V. Basupalli, and J. Burger, "From continuous integration to continuous assurance", in *2017 IEEE 28th Annual Software Technology Conference (STC)*, Sep. 2017, pp. 1–8. DOI: 10.1109/STC.2017.8234450.