The Journal of Systems and Software 214 (2024) 112063

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

Identifying the primary dimensions of DevSecOps: A multi-vocal literature review☆

Xiaofan Zhao *, Tony Clear, Ramesh Lal

Auckland University of Technology, 55 Wellesley Street East, Auckland Central, Auckland, New Zealand

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.7959584

Keywords:
Multivocal literature review
DevSecOps
DevOps
Security
Global software engineering

ABSTRACT

Context: Security as a key non-functional requirement of software development is often ignored and devalued in DevOps programs, with security seen as an inhibitor to high velocity required in DevOps implementation. Hence, the DevSecOps approach as a security-orientated expansion to DevOps, has aimed to integrate security into DevOps implementation by promoting collaboration among development, operation and security teams. DevSecOps is a topical concept and rapidly emerging area of practice in both academic and industrial settings.
Objective: We reviewed both the white and grey literature to identify recent researches and practical trends of DevSecOps, aiming to: (a) review, document and analyze the current state of DevSecOps in the existing literature; (b) investigate the application of DevSecOps in Global Software Engineering (GSE) contexts.
Method: A Multi-vocal Literature Review on DevSecOps and its global application was conducted, by executing a dual-track strategy including white (104 studies) and grey (43 studies) literature from 2012 to 2021. A Thematic Analysis was performed to identify, synthesize and analyze the themes within data for reporting the MLR results.
Results: Through the Multi-vocal Literature Review and Thematic Analysis, this paper identifies five major aspects of DevSecOps (Definitions, **C**hallenges, **P**ractices, **T**ools/Technologies, and **M**etrics/Measurement); collects related themes of each aspect; and generates a **Challenge-Practice-Tool-Metric (CPTM) model** by integrating the themes of the latter four aspects within a lifecycle model. Moreover, an unexplored area relating to the global application of DevSecOps has been identified.
Conclusion: Based on MLR results, a CPTM (Challenge-Practice-Tool-Metric) model is built to reveal the current status of DevSecOps. The model provides a breakdown and a broad landscape of DevSecOps, from which researchers and practitioners may select an area of focus to improve their knowledge or practice. With DevSecOps spanning the many stages of the lifecycle, we believe the model will enable emphases and absences such as global aspects to be investigated.

Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.

1. Introduction

DevOps is a trending term and has gained popularity in the Software Engineering (SE) industry and academia. It aims to improve the performance of software development implementation by enhancing software development (Dev) practices with IT Operations (Ops) practices as part of the SE process (Hussain et al., 2017). However, security as a key non-functional requirement is often ignored and devalued with DevOps programs, due to security being seen as an inhibitor to the high velocity required in DevOps implementation (Myrbakken and Colomo-Palacios, 2017).

Software security can be divided into security of the software development environment, and security of the software in the production environment (Morales et al., 2020). The growing importance of security with SE for development and deployment of software products includes: needs to focus on end-user privacy for larger scale systems; the emergence of Software as a Service (SaaS) as an alternate deployment model; globally distributed systems; and the requirements of rapid delivery cycles. Besides, the utilization of technologies such as cloud, container and serverless computing requires upfront consideration of security requirements with feature implementation (Fernandez and Brito, 2019). According to SANS 2022 DevSecOps survey (Edmundson

☆ Editor: Christoph Treude.

* Corresponding author.
E-mail addresses: gavin.zhao@autuni.ac.nz (X. Zhao), tony.clear@aut.ac.nz (T. Clear), ramesh.lal@aut.ac.nz (R. Lal).

https://doi.org/10.1016/j.jss.2024.112063
Received 10 July 2023; Received in revised form 20 February 2024; Accepted 12 April 2024
Available online 23 April 2024

and Hartman, 2022): in 2022, nearly 92% of responding organizations are using cloud and 25% are using multiple cloud providers; 65% of responding organizations run over 25% of their applications in the cloud; 8% run 100% of their applications in the cloud. Moreover, virtual machines, containers and serverless are the top three cloud-hosted technologies. The security implication to these figures is that cloud resources should be properly secured, as the use of multi-clouds and cloud-based technologies would not only benefit organizations but also cause security complications. However, the survey (Edmundson and Hartman, 2022) reveals the reality of many companies is that they under-utilize security methods such as Cloud Security Posture Management and Cloud Workload Protection Platform.

To build security in DevOps, the term 'DevSecOps' has been created as a security-oriented variant of DevOps. It aims to integrate security into DevOps without impacting the development speed and quality, addressing risks and security issues, through enhanced collaboration amongst security, development, and operations teams (Zaydi and Nassereddine, 2020). A key benefit of DevSecOps is that it shifts security and testing upfront with development (shift-left) and enables continuous security implementation throughout the Software Development Lifecycle (SDLC), to reduce security threats earlier and address security issues faster (Carter, 2017). Another benefit is that manual security tests and support activities are reduced by the automation, so that teams can focus more on policies (Ahmed and Francis, 2019). SANS survey (Edmundson and Hartman, 2022) shows that in 2022, 58% of responding organizations adopt DevSecOps to varying extent; 21% do not; and 18% deem their DevSecOps adoptions to be "spurious", SANS therefore advises organizations to sustain promoting DevSecOps practices. In addition to SANS, Gartner Betts (2022) forecasts that 85% of organizations will adopt DevSecOps practices by 2027, migrating from DevOps to DevSecOps. Thus, we believe that the current state of DevSecOps is worth studying systematically, from the perspectives of SE academia and industry.

Meanwhile, software academia and industry's interest in another trend - Global Software Engineering (GSE), has also been increasing during this decade. GSE is a business strategy to arrange project teams distributed and geographically separated (Grande et al., 2024). The benefits include: specialized and diverse skilled human resources from all over the globe, reduction of costs due to the possible salary savings, and reduction of duration by leveraging time-zone effectiveness and round-the-clock productivity (Conchuir et al., 2009; Vizcaíno et al., 2016). GSE and DevOps/DevSecOps essentially belong to Collaborative Software Engineering (CoSE), which is *"about creating the organizational structures, reward structures, and work breakdown structures that afford effective work towards goal"* (Whitehead et al., 2010). Thus, a growing number of researchers and organizations pay attention to the adoption of DevOps in the GSE context, to achieve further success in SE (Cico et al., 2021). As an expansion of DevOps, we also believe that adopting DevSecOps in global settings deserves careful academic study, in order to enhance the security aspects for global DevOps applications.

This paper aims to review, document and analyze the current state of DevSecOps in the existing literature, and to investigate its adoption in GSE contexts. DevSecOps is topical in academic and industrial settings, so that the investigations from academia and industry are equally essential to learn from. Thus, a Multi-vocal Literature Review (MLR) was conducted by executing a dual-track strategy covering the published and unpublished literature, to identify recent researches and practical trends and to find out opportunities for further research. MLR is a special form of Systematic Literature Review (SLR) which uses not only formally and commercially published literature (called White Literature, e.g. journal and conference papers) but also includes unpublished work (called Grey Literature, e.g. technical reports, websites, blogs, etc.) (Garousi et al., 2019). Although the existing MLRs (Myrbakken and Colomo-Palacios, 2017; Prates et al., 2019; Akbar et al., 2022), SLRs (Sanchez-Gordon and Colomo-Palacios, 2020; Rajapakse et al., 2022), Grey Literature Review (GLR) (Mao et al., 2020) and mapping study (Mohan and Othmane, 2016) have made contributions to the topic of DevSecOps, this MLR reports some new findings for this fast moving new field, e.g., the analysis of differences between academia and industry, the new taxonomy, a novel Challenge-Practice-Tool-Metric (CPTM) model for DevSecOps, and the absence of global applications. We believe that the MLR could consolidate, confirm, update and add value to the extant literature. We give the review and comparison of all existing review papers (Table 1) in Section 2.2.1.

The rest of the paper is organized as follows: Section 2 introduces the key concepts and reviews the related work; Section 3 describes the research methodology, including multivocal literature review and thematic analysis; Section 4 reports the results, presents a Challenge-Practice-Tool-Metric (CPTM) model for DevSecOps, and discusses the findings along with study implications; Section 5 provides the threats to validity; Section 6 concludes the paper and provides the future work.

2. Key concepts and related work

This section introduces three key concepts, namely, DevOps, DevSecOps, and GSE; and reviews the existing related work by comparison with this paper.

2.1. Key concepts

2.1.1. DevOps

There are different ways to define DevOps. Simply, DevOps is a compound of development (Dev) and operations (Ops) (Sebastian et al., 2020). DevOps can be defined as a culture (Soni, 2015), aiming to bridge the gaps between developers and operations (Huttermann, 2012), emphasizing the collaboration within and between teams involved in the Software Development Lifecycle (SDLC) (Dyck et al., 2015; Humble and Molesky, 2011). Bass et al. (2015) defines DevOps as a process which is *"a set of practices aimed to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality"*. Loukides (2012) defines DevOps as a technology, emphasizing the automation for software delivery and infrastructure changes. Humble and Molesky (2011) defines Culture, Automation, Measurement, and Sharing (CAMS) as four pillars of DevOps. Smeds et al. (2015) defines Capabilities, Cultural Enablers, and Technological Enablers as critical related elements of DevOps while identifying Capabilities as the most important of all three. Jabbari et al. (2016) based on previous studies, defines DevOps as *"a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices"*. Different perspectives and ways to define DevOps inspire various taxonomies used to define DevSecOps, we explain this in Section 4.1.2.

2.1.2. DevSecOps

Software security can be divided into security of the software development environment (security threats in the factory) and security of the software being developed (software security testing) (Morales et al., 2020). The importance of security consideration with SE has triggered the emergence of DevSecOps, which is a security-orientated expansion to DevOps. The most common definition of DevSecOps is: "*the concept of incorporating security practices in the DevOps processes by promoting collaboration between development, operations and security teams*" (Mohan and Othmane, 2016). Essentially, the DevOps/DevSecOps approach belongs to Collaborative Software Engineering (CoSE), which is *"about creating the organizational structures, reward structures, and work breakdown structures that afford effective work towards goal"* (Whitehead et al., 2010). The terms 'SecDevOps', 'DevOpsSec', and 'Security in DevOps' are the aliases to DevSecOps (Rahman and Williams, 2016).

Table 1
Comparison among review papers.

Reference	Year	Research methods	Search sources	Included studies	Aspects involved
Mohan and Othmane (2016)	2016	Mapping study	Google Scholar, IEEE, OWASP and RSA conferences	5 WL + 3 Presentations	Definition, Practices, Compliance, Automation, Tools, Configuration management, Team collaboration, Availability of activity data, Information secrecy
Myrbakken and Colomo-Palacios (2017)	2017	MLR	Google Scholar, Google	2 WL + 50 GL	Definitions, Characteristics, Benefits, Challenges, Evolution
Prates et al. (2019)	2019	MLR	ACM, IEEE, Scopus, Google Scholar, Google	2 WL + 11 GL	Metrics
Sanchez-Gordon and Colomo-Palacios (2020)	2020	SLR	Google Scholar	11 WL	Cultural aspects
Mao et al. (2020)	2020	GLR	Google	141 GL	Security risks, Practices
Akbar et al. (2022)	2022	MLR + Survey	ACM, IEEE, Wiley, Springer Link, Science Direct, Google Scholar, Google	46 WL + 41 GL	Challenges
Rajapakse et al. (2022)	2022	SLR + Thematic analysis	ACM, IEEE	54 WL	Challenges, Solutions
Ours	2022	MLR + Thematic analysis	ACM, IEEE, Scopus, Google	104 WL + 43 GL	Definitions, Challenges, Practices, Tools, Metrics, Global applications

2.1.3. Global software engineering

Global Software Engineering (GSE) is a business strategy to have the project teams distributed and geographically separated (Grande et al., 2024), aimed to find specialized and diverse skilled human resources from "a global pool", to promote competitiveness by accessing a global market (Vizcaíno et al., 2016), to reduce software development costs due to the possible salary savings, and to shorten development duration by leveraging time-zone effectiveness and round-the-clock productivity (Conchuir et al., 2009). GSE depends on the distributed teams comprising of stakeholders from different geographic locations, different time zones, and even different organizational and national cultures (Jalali et al., 2010; Tamburri et al., 2012). Thus, GSE also faces challenges from geographical, temporal, linguistic and cultural distances so that it is particularly associated with the 3C Collaboration model (Communication, Coordination and Cooperation) (Conchuir et al., 2009; Tamburri et al., 2012). Same as DevOps/DevSecOps, GSE also belongs to CoSE (Whitehead et al., 2010). Collaboration is vital to the success of DevOps, DevSecOps, and GSE. As GSE becomes a prevalent approach, it warrants investigating the adoption of DevOps and DevSecOps in GSE settings.

2.2. Related work

2.2.1. Existing reviews on DevSecOps

Some review studies have been conducted on DevSecOps to identify its definitions, benefits, challenges, practices, tools, metrics, applications, etc. We chronologically summarized the existing review work and demonstrated the comparison in Table 1. Historically, the findings of early studies were basic but premature and unshaped, due to the extremely limited data sources of this emerging area. Along with the development of DevSecOps, the research aspects of DevSecOps have been becoming clear and taking shape, the majority of studies focus on the challenges when adopting DevSecOps, and the practices applied in DevSecOps. Recently, the research trend is going to explore the interrelation of challenges, practices and tools, further to build theoretical frameworks for DevSecOps.

In 2016, Mohan and Othmane (2016) conducted a mapping study to identify DevSecOps definition, practices, compliance requirements, automation, tools, configuration management, team collaboration, availability of activity data and information secrecy. To our knowledge, this paper might be the earliest review study on DevSecOps, it reviewed a limited amount of studies, including five academic papers and three conference presentations. This paper serves a pioneer role but the findings are relatively premature and unshaped.

Myrbakken and Colomo-Palacios (2017) presented an MLR to identify definition, characteristics, benefits, challenges and evolution. The MLR was conducted in 2017 when DevSecOps was still a new approach, so limited white literature (WL) (2) and grey literature (GL) (50) were available for reviewing. WL results were fewer than GL results including sources that had not been peer-reviewed. In 2019, Prates et al. (2019) presented an MLR to identify 9 DevSecOps metrics from 2 WL papers and 11 GL articles. These two papers selected MLR as review method due to the lacking academic studies at that time (mentioned in their limitations). In contrast, using MLR is an active choice to this paper, not a compromise, so that it covers both of the researcher-oriented and practitioner-oriented sources to analyze DevSecOps from dual perspectives.

In 2020, Sanchez-Gordon and Colomo-Palacios (2020) presented an SLR to identify DevSecOps from a cultural perspective. Mao et al. (2020) conducted a GLR to identify security risks on DevOps and to collect a set of DevSecOps practices. By contrast, we did not define a specific aspect, when we raised research questions and formulated the search string. Our attempt was to cover the entire software development life cycle, from development to operation, instead of a single point of view or a specific step such as security testing. In this way, the MLR would provide a general and broad coverage of this topic.

In 2022, an MLR by Akbar et al. (2022) revealed 18 DevSecOps challenges and grouped them in 10 categories; subsequently surveyed with practitioners to assess findings and the result showed that identified challenges and categories were relevant to the industry. Another recent SLR by Rajapakse et al. (2022) identified 21 challenges and 31 solutions by applying thematic analysis, further classified them in four categories: People, Practices, Tools, and Infrastructure. To a certain extent, these two review papers are similar to ours, since the research trend is going to explore the interrelation of challenges, practices and tools, further to create theoretical frameworks for DevSecOps. Thus, our paper serves a confirmatory role for the existing literature in addition to its more specific contributions. However, in addition to the partial replication of existing work, we use different classifications; and we provide a general and broad coverage of this topic, rather than a very focused aspect or step, in order to cover the entire SDLC. This work identifies multiple research aspects of DevSecOps and models

their links. There are four levels of interpretation in Thematic Analysis (Cruzes and Dyba, 2011a): Text, Code, Themes, and Model. The mentioned review papers (Akbar et al., 2022; Rajapakse et al., 2022) both stopped at the third level - Themes, while our MLR has reached the final level - Model. Moreover, our MLR involves the exploration of adopting DevSecOps in GSE, and identifies the absence of global dimension of DevSecOps in the existing literature.

Overall, the existing review studies on DevSecOps are relatively early or focus on a single perspective or aspect, therefore a substantial body of academic research on the topic has not been completely built. This paper provides a more comprehensive review, throughout the decade (2012–2022) of DevSecOps development, so that it aims to update and add value to the extant literature.

2.2.2. Existing studies on global DevOps

Some papers reflect the global context of DevOps. Gupta et al. (2019) presented their experience in a global DevOps project across India, the USA and Germany, that had successfully established continuous delivery and short release cycles with agile. Hussain et al. (2017) investigated online job advertisements and combined with interviews; identified required knowledge, skills and capabilities for DevOps roles in New Zealand; further revealed that the global dimension of DevOps roles were apparent in most job ads sometimes by explicit mention (16% job postings explicitly mentioned global aspects) but more often by implication, reflected that global DevOps has been involved but not importantly. Diel et al. (2016) identified sets of communication challenges (geographical, socio-cultural, temporal distance) and strategies (frequency, direction, modality and content) in distributed DevOps, through exploratory observations and interviews. A recently published paper by Grande et al. (2024) presented the results of a systematic mapping study in the adoption of DevOps in Global Software Development (GSD). This paper proposed the definition of DevOps in global settings; captured the goals of adopting DevOps in GSD along with 5 motivating issues; identified 11 benefits, 9 challenges and 15 practices of DevOps in GSD; mapped the identified challenges with a list of well-known GSD risks; also mapped the links between the motivating issues, benefits, challenges and practices.

However, these studies do not mention any security aspect in the 'global DevOps' setups. Security is one of the motivations behind this MLR, to investigate the academic and industrial work, further to identify evidence-based practices for bridging the gap between DevSecOps and GSE.

3. Research method

A Multi-vocal Literature Review (MLR) was conducted for the research. MLR is a special form of Systematic Literature Review (SLR) which use not only formally and commercially published literature (called White Literature, e.g., journal and conference papers) but also includes unpublished work (called Grey Literature, e.g., technical reports, websites, blogs, etc.) (Garousi et al., 2019). The most common definition of GL called 'Luxembourg definition' was presented at the 3rd International Conference on Grey Literature in Luxembourg in 1997 (Farace and Schopfel, 2010), it defined: *"Grey literature is that which is produced on all levels of government, academics, business and industry in print and electronic formats, but which is not controlled by commercial publishers"*. In 2004, a postscript was added: *"…not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body"*. In 2010, Schopfel who is member of GreyNet defined (Schopfel, 2010): *"Grey literature stands for manifold document types produced on all levels of government, academia, business and industry in print and electronic formats that are protected by intellectual property rights, of sufficient quality to be collected and preserved by libraries and institutional repositories, but not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body"*. There is no need to define WL as it is a relative concept to GL. According to the definitions, the boundary between white and grey literature is whether it is controlled by commercial publishers.

3.1. Motivation and process

The reason for using MLR is that the emerging topic is very significant in industry since many software development companies apply DevSecOps nowadays. Practitioners are the first to adopt and review new approaches and emerging technologies. To get the best outcome, both white and grey literature should be included. DevSecOps is a topical area in industrial settings, practitioners constantly produce technical reports, feedbacks and reviews. Hence, it would be useful to learn about DevSecOps. Fig. 1 depicts the MLR process adapted from Garousi's guidelines (Garousi et al., 2019), and which also shows Thematic Analysis (TA) (Cruzes and Dyba, 2011a) was adopted for data synthesis. A review protocol was developed and updated over the research timeline. The latest version of the review protocol is available in an open repository at zenodo.org (https://doi.org/10.5281/zenodo.7959584), attached with data extraction form and the search records of databases.

3.2. Philosophical stance

From the philosophical stance, the research design involves both positivist and interpretive paradigms, and *"although combined approaches are methodologically challenging"* (Cruzes and Dyba, 2010), the study adopts a pragmatic strategy as recommended by both Hoda (2021), and Cruzes and Dyba (2010). The earlier steps of the research, from the beginning of the MLR to the data extraction step in Fig. 1, belong more to the positivist perspective underlying the evidence based SE movement (Kitchenham et al., 2004), which aims to derive an objective reality from pure data by using quantitative analysis, without undue influence of researchers' interpretations (Alharahsheh and Pius, 2020). Although bias in study selection, quality assessment, and data extraction were inevitable, they could be mitigated by the formulation and implementation of the review protocol. Nonetheless, a literature review considers synthesis and interpretation as a mandatory property (Rowe, 2014), hence the later steps from data synthesis in Fig. 1 can be addressed through an interpretivist stance, which advocates a relativist ontology and subjective epistemology, using qualitative methods (Alharahsheh and Pius, 2020). In this case, we collected white and grey literature in strict accordance with the search strategy and selection criteria, and conducted quantitative analysis on the selected papers. Afterwards, we employed (Reflexive) Thematic Analysis (Braun and Clarke, 2021) as a synthesis method to conduct qualitative analysis on the collected data, through coding, theming and modeling, to gain further depth through seeking the existing perception and experience of DevSecOps.

3.3. Objectives and research questions

The objectives of this MLR were to: (a) review, document and analyze the current state of DevSecOps in the existing literature; (b) investigate the adoption of DevSecOps in GSE contexts. The following were the research questions and associated sub-questions which were investigated.

- *RQ1: What is the current state of DevSecOps (namely, aspects involved, related themes in each aspect and their links) in the existing (white and grey) literature?*
  Sub-question 1.1: What aspects of DevSecOps can be found in the existing (white and grey) literature?
  Sub-question 1.2: What themes do these aspects contain?
  Sub-question 1.3: How do the identified aspects and themes link to each other?

- *RQ2: How is DevSecOps adopted in the Global Software Engineering (GSE) contexts?*

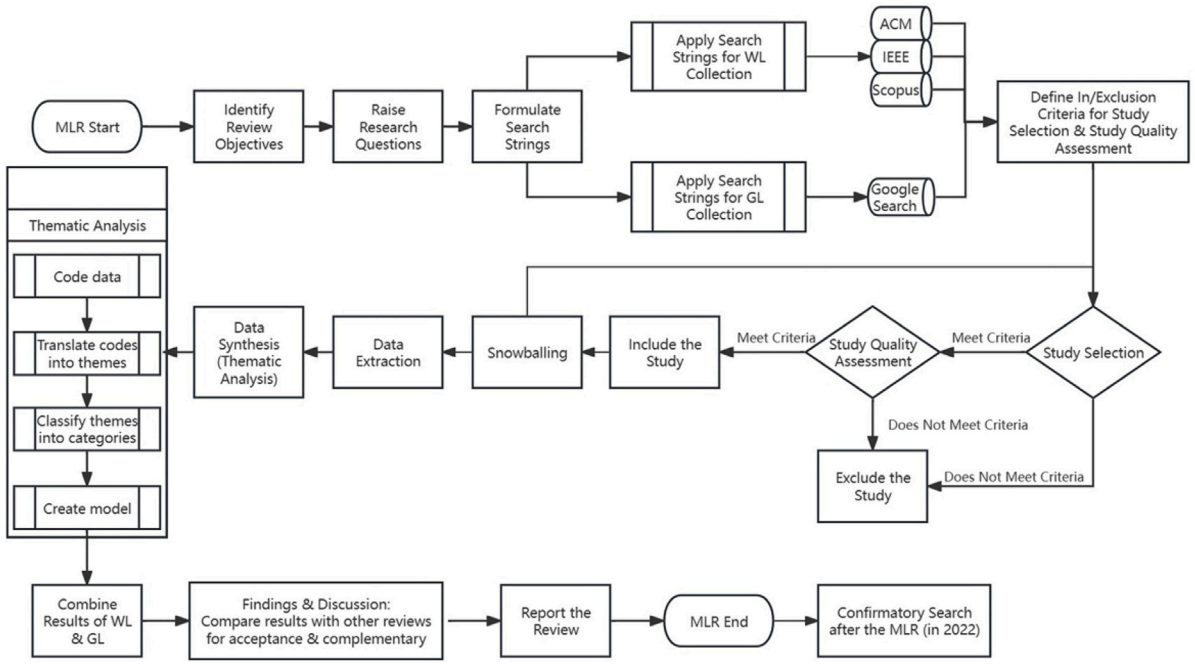

Fig. 1. An overview of the MLR process.

3.4. Search strategy

Based on the research questions, the search strategy was been defined and discussed, including search sources, search strings, and snowballing.

3.4.1. Search sources

Three digital databases with advanced or constructed search features were used for White Literature (WL) collection: ACM Digital Library, IEEE Xplore, and Scopus, as these databases in combination were considered to comprehensively cover the computing and software engineering discipline publications. ScienceDirect and Scopus are both owned by Elsevier and somewhat overlap. ScienceDirect was not selected because it is more suitable for searching journal papers on specific topics. Our attempt to search with ScienceDirect did not produce sufficient results, so we used Scopus, which is the biggest abstract and citation database, providing a wide range of literature from multiple publishers. Similarly, Springer was not used as it did not provide us a sufficient number of empirical studies relating to this topic. However, some high-quality secondary papers were found via ScienceDirect and Springer, hence, these were used during snowballing search and the confirmatory search after MLR. Google was used to search the Grey Literature (GL).

3.4.2. Search strings

To address RQ1, Search String 1 = *(devops AND (security OR secure OR safe)) OR secdevops OR devsecops*. The search string does not enclose the terms 'secure' and 'safe' in double-quotes, so variations such as 'securely', 'safely' and 'safety' are included too. Some dictionaries define security in terms of safety, and vice versa (Burns et al., 1992). Germanic languages, Romanic languages and Chinese do not even distinguish them (Line and Rostad, 2006). Although the definitions of security/secure and safety/safe have a lot in common, they are not identical but complementary (Line and Rostad, 2006). In context of information technology, Line (Line and Rostad, 2006) indicates that safety focuses on the undesirable effects that are unintentional; security focuses on the undesirable effects that are caused by malicious parties out of the system. Burns et al. (1992) distinguishes safety and security by causalities and failure consequences: a system is safety-critical if failure could immediately and directly cause absolute harm; a system is security-critical if failure could only cause relative harm, or could raise the possibility of harm. Both papers (Burns et al., 1992; Line and Rostad, 2006) underline that security and safety connotations are not separate: security flaws compromise safety while safety breaches make security impossible. Considering the above, safe was included in the string.

After applying Search String 1 in all search sources, the results did not include any studies involving the adoption of DevSecOps in GSE settings. To address RQ2, we used an additional Search String 2 = *(devops AND (security OR secure OR safe) OR secdevops OR devsecops) AND ("global software engineering" OR "global software development" OR gse OR gsd OR "globally distributed" OR "distributed software development" OR "distributed software engineering" OR "multi-site" OR "multi-nation" OR "transnational" OR "remote work")*. The intention of Search String 2 was to narrow the search to a global-orientation by specifying additional keywords related to GSE, because Search String 1 retrieved hundreds of papers from each database, potentially led to accidentally missing a few GSE-related papers.

Search strings might be adapted due to the differences between databases and the acceptability of Boolean operators. Search strings were also applied to Google search for GL collection. After eliminating advertising, the first 18 results pages (180 GL articles) were browsed, as the relevance became extremely weak after page 19. Searches were limited as follows: strings were searched within Metadata (title, abstract and keywords); books, posters and abstracts were excluded; language was set to English; and publication year was set between 2012 and 2021. This was the decade within which the DevOps concept became common while DevSecOps was first mentioned in 2012 (Sanchez-Gordon and Colomo-Palacios, 2020), actually the earliest related paper was published in 2013. Search results were sorted by relevance enabling us to know when the relevance was extremely weak and to stop the search. Search strings were also applied on Google for GL collection. Pre-selection criteria were applied for this, by reading the titles, abstracts (for WL), and summaries (for GL), to identify appropriate literature.

3.4.3. Snowballing search

In addition to the database searches, snowballing was applied to locate relevant studies. In a normal way, we conducted database searches

firstly, and later complemented with snowballing searches. Though Wohlin et al. (2022) recommend that using snowballing as the first search strategy is also a good alternative. In this case, the main purpose of snowballing was to validate the reliability of relevant studies, and its minor purpose was to search for more papers. Because database searches had already identified a large collection of papers, and snowballing could was unlikely to find more papers beyond the result of database searches, hence, we mainly applied backward snowballing, rather than forward. For example, backward snowballing was always applied to trace and validate the literature review sections of the included papers. It was also applied on some secondary studies to search for their selected primary studies. The process mainly relied on Google Scholar, and a few additional databases were included, e.g., Springer, ScienceDirect, and the university library's databases. Like snowballing in the WL search, 'back-links' within Google were navigated either forward or backward to detect additional GL articles and validate the reliability.

3.5. Study selection and quality assessment

Study inclusion and exclusion criteria were defined to ensure that selected studies provided data to answer the research questions. In practice, GL selection criteria usually overlap and are integrated with a quality assessment guide (Garousi et al., 2019).

Inclusion criteria: *(a) The study mentions one or more primary aspects related to the topic of DevSecOps, e.g., definition, challenges, practices/activities/solutions, tools/technologies, metrics/measurement, and global applications; (b) The study is written in English; (c) The study is published from 2012; (d) The study has a clearly stated methodology/research design; (e) The study has credible source.*

Exclusion criteria: *(a) The study does not have a full-text; (b) The study is external to the subject area of computer science and software engineering; (c) The study does not have a rigorous research method to prove the correctness of findings; (d) Duplicate studies; (e) Secondary studies.*

Quality assessment was applied to ensure further inclusion/exclusion criteria, to determine the validity of sources, to assess the importance of studies, and to minimize bias (Kitchenham, 2007). Fig. 2 shows a screenshot of the quality assessment criteria defined in our MLR protocol, adapted from Garousi's MLR guidelines (Garousi et al., 2019) and Kitchenham's SLR guidelines (Kitchenham, 2007). Garousi presented a QA checklist for GL only, it was adapted and extended to also cover WL. The first 14 questions were grouped into 6 categories and would be answered YES/NO, so the criteria would be marked 0/1. ''Literature Type'' would be marked on a scale from 0 to 4. Hence, the full mark is 18 (14 + 4), and we set 11 (60% of 18) as the passing score. QA scores of the included papers are available at zenodo.org (https://doi.org/10.5281/zenodo.7959584).

3.6. Replication and snowballing

The exclusion criterion (e) states that secondary studies should be excluded to ensure the credibility of this MLR. However, six important secondary studies (Myrbakken and Colomo-Palacios, 2017; Mohan and Othmane, 2016; Prates et al., 2019; Sanchez-Gordon and Colomo-Palacios, 2020; Akbar et al., 2022; Rajapakse et al., 2022) were identified and used to validate our findings. According to Wohlin et al. (2022), the replication in SLRs can be used as a method for the acceptance of new knowledge. Researchers sometimes rely on intentional replication to validate their own results, by comparing other SLRs/MLRs on the same topic, but raising different questions, performing different search strategies, selecting different primary studies, and using different analytical methods (Wohlin et al., 2022).

Snowballing was applied to these review papers to identify the overlaps of included primary studies compared with our MLR. Table 2 reports the number of overlapping and non overlapping papers. Only the WL papers are compared, as some GL articles are not available or not accessible anymore. For non-overlapping papers, there are two situations: the chief one is that the paper had been selected during database searches but was excluded by our selection criteria or quality assessment; another one is that the paper had not been selected during database searches but fulfilled our criteria, therefore would be complementary.

3.7. Search execution

Table 3 shows a summary of the search execution. The numbers of collected studies were counted along with implementing the search procedure. We conducted Search 1 by applying Search String 1, but its result did not include any studies involving the global aspect of DevSecOps. To address RQ2, we performed Search 2 by using Search String 2 which includes additional global-related keywords. The full included white and grey papers/articles are listed in Appendices A.1–A.2.

The work of paper collection and selection had been finished in July 2021. To avoid staleness and to continue validation, we conducted a continuous confirmatory process termed 'Confirmatory Search' after the MLR to find the latest literature. The confirmatory search used the same search strings with the MLR but included additional databases such as ScienceDirect and Springer. The selection criteria were broadened (including secondary studies) to enable us to find more recent studies, to be used for validation purposes rather than the continuation of the MLR. By 2022, 13 new WL papers and 7 GL articles have been added (Appendix A.3). However, these new papers and articles which were collected from the confirmatory search were not taken into the thematic analysis, and were not integrated in the final CPTM model, in order to avoid affecting the original MLR results. Fig. 3 depicts the number of included articles based on the source types and the published years. There are 147 articles collected from this MLR, and 20 new articles collected from the confirmatory search after MLR. Conference papers and grey literature articles play significant roles in DevSecOps research. The peak of published years is between 2019 and 2020. Besides, the number of GL articles was lacking before 2017, but it has been increasing since 2018.

3.8. Data extraction and data synthesis (Thematic analysis)

Data extraction was performed to gather relevant information from selected studies. This was done by using an adapted data extraction form (Kitchenham, 2007). Data synthesis was performed to collate and summarize the result of data extraction (Kitchenham, 2007).

3.8.1. Thematic analysis

Data synthesis was conducted by using Thematic Analysis (TA), which is a method for identifying, analyzing and reporting themes with data, combining qualitative (text segments, codes, themes) and quantitative (frequency statistics) evidence (Braun and Clarke, 2006). Flexibility is a key advantage of TA method, enabling researchers to provide a wide range of analytic options (Braun and Clarke, 2006). Compared to other methods, TA is said to be relatively easy to learn and perform, thereby being accessible to inexperienced researchers (Braun and Clarke, 2006). Thus, TA is one of the most frequently used methods for data synthesis in SE, 2/3 of the systematic reviews in SE employed TA to synthesize the data from primary studies (Cruzes and Dyba, 2011b). The key distinction between TA and another classic synthesis method - Grounded Theory (GT), is that GT uses an ongoing process to code data throughout data collection (Cruzes and Dyba, 2011b), while TA is applied after data collection. Another distinction is that GT aims to create a new theory, but TA is used to capture themes and summarize key features based on existing frameworks. Therefore, the latter is a more appropriate choice for this study. Considering the above, TA was selected as the synthesis method.

Braun and Clarke (2021) specify three types of TA: Coding reliability, Codebook, and Reflexive. We used reflexive TA, which fully

Criteria	Questions
Authority of the producer (Measure = 0 or 1)	• Is the author or the publishing organization reputable? • Has the author published other work in the field? • Does the author have expertise in the area?
Methodology (Measure = 0 or 1)	• Does the work have a clearly stated aim? • Does the work have a stated methodology? • Does work have authoritative and contemporary references? • Are any limits clearly stated?
Objectivity (Measure = 0 or 1)	• Does the work provide objective statements or credible findings? • Is there vested interest? E.g., a tool comparison by authors that are working for particular tool vendor. • Is the conclusion supported by the data?
Publication Date (Measure = 0 or 1)	• Does the work have a clearly stated date?
Novelty (Measure = 0 or 1)	• Does the work have a novel idea or something unique? • Does the work strengthen or refute a current position?
Impact (Measure = 0 or 1)	• For WL, is the author's work cited often? / For GL, is the source viewed/shared/discussed often?
Literature Type (Measure = 0 to 4)	• WL: peer-reviewed academic papers (Measure = 4). • WL: PhD/Master thesis (Measure = 3). • GL with high credibility, such as books, magazines, specialized databases, white papers, method creators and consultants' websites and case studies (Measure = 2). • GL with medium credibility, such as technical reports, news, Q/A sites, blogs, presentations and videos (Measure = 1). • GL with low credibility, like ideas/opinions/thoughts/commentaries without evidences (Measure = 0).

Fig. 2. Study quality assessment form.

Table 2
Overlaps of included primary studies between review papers.

Existing review papers	Year	Included primary studies (WL)	Overlapped with this MLR	Not overlapped	Overlapping percentage
Mapping study by Mohan and Othmane (2016)	2016	5	2	3	40%
MLR by Myrbakken and Colomo-Palacios (2017)	2017	2	2	0	100%
MLR by Prates et al. (2019)	2019	2	1	1	50%
SLR by Sanchez-Gordon and Colomo-Palacios (2020)	2020	11	8	3	73%
SLR by Rajapakse et al. (2022)	2022	54	26	28	48%
MLR by Akbar et al. (2022)	2022	46	26	20	57%

Table 3
Summary of MLR search execution.

Search steps	Search 1 results WL/GL	Search 2 results WL/GL
Applying search string	692 (acm-416, ieee-100, scopus-176)/400 m studies	216 (acm-97, ieee-27, scopus-92)/150k studies
Study pre-selection	327 (acm-113, ieee-90, scopus-124)/180 studies	66 (acm-35, ieee-21, scopus-10)/100 studies
Study selection	238 (acm-101, ieee-88, scopus-49)/56 studies	8 (acm-7, ieee-0, scopus-1)/ 3 studies
Study quality assessment	96 (acm-26, ieee-39, scopus-31)/43 studies	2 (acm-2, ieee-0, scopus-0)/0 study
Snowballing	102 (acm-26, ieee-45, scopus-31)/43 studies	2 (acm-2, ieee-0, scopus-0)/0 study

embraces qualitative research values and researchers' subjective skills, thereby fitting an experiential (e.g., critical realist, contextualist) and critical (e.g., relativist, constructionist) framing of language, data and meaning (Braun and Clarke, 2021). In reflexive TA, analysis is a situated interpretative reflexive process and can be conducted inductively or deductively; coding is open and organic without a coding framework; themes are the final outcome of data coding and iterative theme development. In comparison with 'reflexive TA' (informed by interpretivism, which advocates a relativist ontology and subjective epistemology (Alharahsheh and Pius, 2020)), 'coding reliability TA' is concerned with objective and unbiased coding (informed by positivism, which aims to achieve an objective reality from pure data without human interpretation (Alharahsheh and Pius, 2020)); 'codebook TA' uses their developed hybrid variant of a structured codebook or coding framework with coding reliability approaches (informed by pragmatism, which is driven by pragmatic demands around pre-determined information needs (Braun and Clarke, 2021)). Except for 'coding reliability TA', agreement between researchers and inter-rater reliability are not required as measures of quality for 'codebook TA' and 'reflexive TA'. Braun and Clarke (2021) rather critically stress that it is *"illogical, incoherent and ultimately meaningless"* to require coding reliability and bias suppression in reflexive TA, *"because meaning and knowledge are understood as situated and contextual, and researchers' subjectivity is conceptualized as a resource for knowledge production, which inevitably sculpts the knowledge produced, rather than a must-be-contained threat to credibility"*.

3.8.2. Model creation

Cruzes and Dyba (2011a) present four levels of interpretation and abstraction in TA: Text, Code, Themes, and Model. According to the recommended steps (Cruzes and Dyba, 2011a), the first author initially read the text from many pages of included papers, identified specific

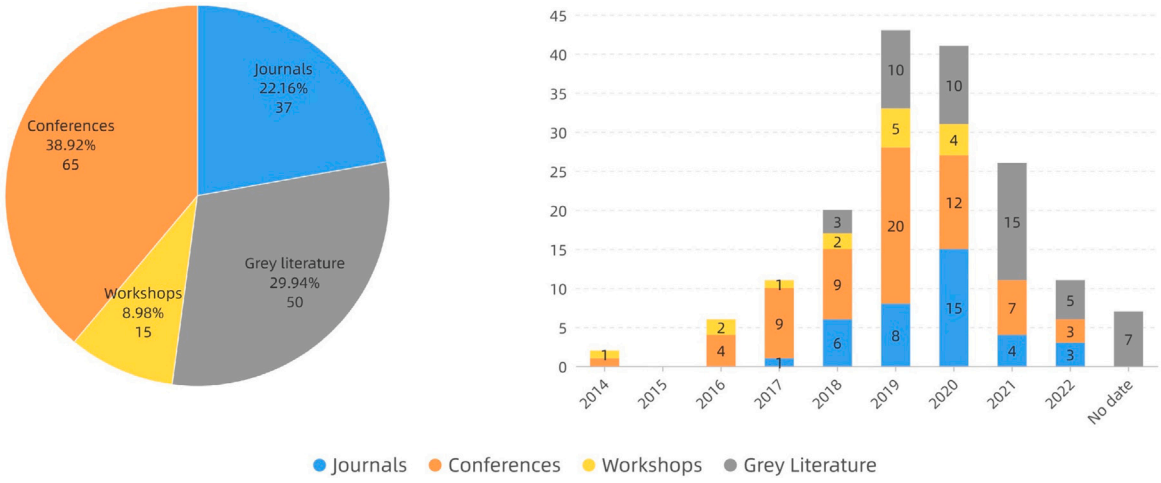

Fig. 3. Number of included papers based on source types and published years.

segments of text, and labeled segments into codes. Subsequently, code overlaps were reduced and the codes were translated into themes. Themes were further classified into categories (high-order themes), and we finally created a conceptual model. TA can be performed either by manual methods or using a software programme (Braun and Clarke, 2006). We analyzed manually because initial text segments and codes capturing concepts (e.g., the definitions) were so long that they were difficult to fit to software, which favors short and descriptive codes. The TA process was performed manually, by searching for concepts and highlighting segments of text on included papers, extracting data to word documents, writing notes for potential themes, and making tables for numeric counts. TA associated materials are available at zenodo.org (https://doi.org/10.5281/zenodo.7959584)

The TA process initially followed an inductive approach (coding and theming were directed by the content of data (Braun and Clarke, 2020)). After generating sets of codes/themes, it was supplemented by a deductive approach (coding and theming were directed by existing concepts (Braun and Clarke, 2020)). Specifically, MLR data was collected from WL and GL. WL data was initially coded and themed in an inductive approach; GL data was subsequently analyzed mainly in a deductive approach, based on the codes/themes generated during the TA of WL. In developing the model we were cognizant of the view that *"thematic analysis has limited interpretative power beyond mere description if it is not used within an existing theoretical framework"* (Cruzes and Dyba, 2010). So in addition to the four inductively derived categories we complemented the thematic dimensions of the developing model with a DevSecOps lifecycle framework (MacDonald and Head, 2016), based on the established concept of the SDLC (Pothukuchi et al., 2023; Mohammed et al., 2017). This framework was then applied deductively, to map the themes and categories to each applicable lifecycle stage to derive the final CPTM model (Fig. 5).

Coding tasks were completed by the first author, because reflexive TA does not demand to have multiple coders who work independently, and to measure the agreement between coders (inter-rater reliability) (Braun and Clarke, 2021). However, to strengthen the process, the output was reviewed and evaluated in consultation with the second and third authors by weekly or bi-weekly meetings to achieve consensus. Sets of developing codes and themes were shared progressively. There was a good level of agreement on the early TA steps such as texting, coding and early theming, until we were classifying themes and naming categories (higher-order themes). A main discrepancy among us was on how to accurately name the categories and classify related themes. Another discrepancy was about the design of the model. We designed more than three versions, where incorporating the SDLC framework and mapping the themes to lifecycle stages developed the themes usefully, but this integration process took a considerable amount of time to design, complete and refine, from 2021 to 2023. The reason is that there are a large number of items, elements and categories to be covered in such a broad landscape. How to present the model in a perfect format is still a somewhat open question to us. Indeed the challenge posed for practitioners in categorizing the broad based phenomenon of DevOps has been recognized by the authors of the State of the DevOps report (Puppet, 2023), who have settled on a new focus on 'Platform Engineering'. Nonetheless, after iterative negotiations in joint meetings, three authors reached a consensus on coding, theming, classifying, mapping to stages and modeling.

3.8.3. Trustworthiness assessment

To assess the trustworthiness of the synthesis, in terms of the credibility, confirmability, dependability and transferability (Cruzes and Dyba, 2011a), TA tasks were reviewed by leveraging Braun's checklist (Braun and Clarke, 2021). Credibility is importantly concerned with the quality of selected primary studies, a quality assessment was therefore conducted on the included papers. Another concern for achieving credibility is the suitability of text segments, specifically long segments, in this case, e.g., definitions of DevSecOps. Confirmability is mainly focused on the agreement among researchers, which has been mentioned above. Besides, the second and third authors are experienced scholars, their recognition could also be an aspect of confirmability. Dependability refers to the stability of findings (Cruzes and Dyba, 2011a), which was validated by comparing with the findings of other SLRs/MLRs on the same topic, but with different questions, different search strategies, different primary studies and different analytical methods. Transferability refers to the extent to which the findings can be transferred to other settings (Cruzes and Dyba, 2011a), which would be assessed by the further work which is a Delphi study currently underway to validate and refine the findings.

3.9. Combination of WL and GL

Finally, all the processed data, codes and themes from the white and grey literature were combined, analyzed, reported and discussed, concluding the MLR to answer the research questions.

4. Results and discussion

Results are reported and discussed in this section to answer the research questions and associated sub-questions. Furthermore, we compared the results with four previous review papers (Myrbakken and Colomo-Palacios, 2017; Mohan and Othmane, 2016; Prates et al., 2019; Sanchez-Gordon and Colomo-Palacios, 2020) and two newly published review papers (Akbar et al., 2022; Rajapakse et al., 2022), which have

been summarized on Tables 1 and 2 in Section 2 (the GLR by Mao et al. (2020) was abandoned as it totally used grey literature), in order to validate and complement our findings, meanwhile, using our new findings to update the extant literature. In addition, several study implications for researchers and practitioners are provided.

4.1. RQ1 - Current state of DevSecOps in literature

To answer RQ1 regarding the current state of DevSecOps in the existing literature, Search String 1 was applied to ACM, IEEE and Scopus databases, so that 327 academic papers were initially collected. After pre-selecting and eliminating duplicates, 238 papers remained. After performing the study selection, quality assessment and snowballing, 102 WL papers were finally included. On the other hand, Search String 1 was applied on Google to search GL work. The first 18 pages (180 search results, because results had relevance till page 18) were browsed, so that 56 GL articles were collected. After the quality assessment, 43 GL articles were finally included. Considering the rigor, timeliness, and replicability of the study, it is necessary to state that GL sites were accessed and collected by June 30, 2021.

4.1.1. Five aspects of DevSecOps

To answer Sub-question 1.1 *"What aspects of DevSecOps can be found in the existing literature?"*, we read through all included WL and GL primary studies, additionally appealed to three MLRs (Myrbakken and Colomo-Palacios, 2017; Prates et al., 2019; Rajapakse et al., 2022), two SLRs (Sanchez-Gordon and Colomo-Palacios, 2020; Akbar et al., 2022), and a mapping study (Mohan and Othmane, 2016), to identify common terms. As a result, five major aspects of DevSecOps in the existing white and grey literature are identified:

- Definitions: definitions for the term DevSecOps and equivalent terms;
- Challenges: the problems, concerns and uphill tasks that are faced when adopting DevSecOps;
- Practices: DevOps and security activities suited for DevSecOps;
- Tools/Technologies: specific tools and technical approaches that are used for DevSecOps;
- Metrics/Measurement: means for measuring the effect and maturity of DevSecOps practices.

If the wording of studies were different and confusing, synonyms would be considered. For example, "Meanings", "Perceptions" and "Concepts" were categorized as fitting the "Definitions" aspect. "Activities", "Approaches", "Solutions" and "Strategies" were categorized as fitting the "Practices" aspect. Several minor aspects were omitted and regarded as a part of the major aspect. For instance, characteristics and benefits of DevSecOps were often mentioned in definitions, we therefore considered them as codes/themes under the major aspect of "Definitions".

Fig. 4 depicts the distribution of the five aspects of DevSecOps, i.e., the total frequency of the initially identified text segments of each aspect, including similarities and repetitions, that would be further coded and themed. We believed that the initial text segments, e.g., phrases, clauses, or long sentences, could reflect the result more realistically than codes and themes, which had been artificially processed. "Practices" was the most widely focused aspect in the literature, while "Metrics/Measurement" had the least coverage. WL work gave more results on definitions, challenges and practices, while GL work focused mostly on tools and metrics. This reflects that DevSecOps investigation from academia and industry are equally essential and complementary for learning and practice. From Fig. 4, it can be seen that scholars tend to do phenomenological research on this topic, by defining concepts and identifying challenges and practices when adopting DevSecOps. On the other hand, industrial practitioners take a pragmatic look at their DevSecOps approach, focusing on tools and metrics to offer solutions to customers.

Table 4 shows the included WL and GL work relating to each aspect. For examples: paper S1-IEEE-08 (Rafi et al., 2020) identifies DevSecOps challenges and classifies them into a Culture-Automation-Measure-Sharing (CAMS) model; S1-IEEE-06 (Tomas et al., 2019) presents results from interviews on DevSecOps challenges and practices and also classifies results using the CAMS model; S1-IEEE-12 (Rahman and Williams, 2016) summarizes experiences in utilizing DevSecOps practices based on a survey; S1-IEEE-06 (Tomas et al., 2019), S1-IEEE-57 (Wagner and Ford, 2020) and S1-GL-43 (Chickowski, 2018) identify DevSecOps metrics; S1-SC-01 (Sen, 2021) and S1-GL-42 (Blogumas, 2020) list DevSecOps tools with their functions. In addition, the newly included papers from confirmatory search after MLR and the covered aspects of DevSecOps are also reported. For examples: CS-ACM-01 (Rajapakse et al., 2021) identifies 14 challenges and 23 practitioners' recommendations (practices) in integrating security tools; CS-ACM-02 (Gonzalez et al., 2021) reveals 7 pain points (challenges) of writing automated security tests; CS-ACM-03 (Brasoveanu et al., 2022) proposes a Security Maturity self-Assessment Framework adapted from three well-known models, i.e., OWASP DevSecOps Maturity Model (DSOMM), OWASP Software Assurance Maturity Model (SAMM), Building Security In Maturity Model (BSIMM), and ISO/IEC 27001 standard, in order to measure how the security practices work; CS-IEEE-02 (Ahamed et al., 2022) presents a DevOps framework to systematize security analyses in multi-cloud application development; CS-IEEE-03 (Sojan et al., 2021) provides a solution based on micro-service architectural style to monitor the cloud-native infrastructure involving automation for easy deployment and event-triggered alerting; CS-IEEE-04 (Angermeir et al., 2021) identifies five types of automated security activities along with relevant tools; CS-IEEE-05 (Ibrahim et al., 2022) proposes an automated DevSecOps module for infrastructure as code; CS-SC-02 (Nisha, 2022) proposes a DevSecOps migration model to comprehensively cover the migration challenges, migration strategies, migration procedure, support functions, tools, and evaluation factors.

4.1.2. Themes and classification

Table 5 summarizes the TA results to help answer Sub-question 1.2 *"What themes do these aspects contain?"*. We discovered two common taxonomies used to identify DevSecOps. The first taxonomy by Smeds et al. (2015) identifies the following elements: Capabilities, Cultural Enablers, and Technological Enablers. A second taxonomy by Humble and Molesky (2011) identifies Culture, Automation, Measurement, and Sharing (CAMS) as key elements of DevSecOps. While the latter is more widely used. These two taxonomies are derived from DevOps principles and thus do not fully capture the DevSecOps approach (e.g., DevSecOps technology includes but is not limited to automation). Hence, based on our own MLR results, all the emerging themes are classified into the four categories (high-order themes):

- Organization, People and Culture (OPC): includes themes relating to organizational structure, people management, and cultural strategies, e.g., breaking silos, collaboration, communication, sharing, training, recruiting, etc.
- Process Capabilities (PC): includes themes relating to the capabilities of DevSecOps process, e.g., integration of security, security-left, continuous activities, risk management, faster lifecycle, etc. Acuna and Juristo (2004) defined the term Capabilities as *"the skill or attribute of the personal behavior of a person that can be considered as a behavioral characteristic and according to which activity-oriented behavior can be logically and reliably classed"*. Mao et al. (2020) defined it as *"the processes that an organization should be able to carry out, while the enablers allow a fluent, flexible and efficient way of working"*. Wang and Ahmed (2007) defined Capabilities as *"the firm's capacity to deploy resources, usually in combination, and encapsulate both explicit processes and those tacit elements embedded in the processes"*; and argued that *"capabilities are not simply processes, but embedded in processes"*. The definitions

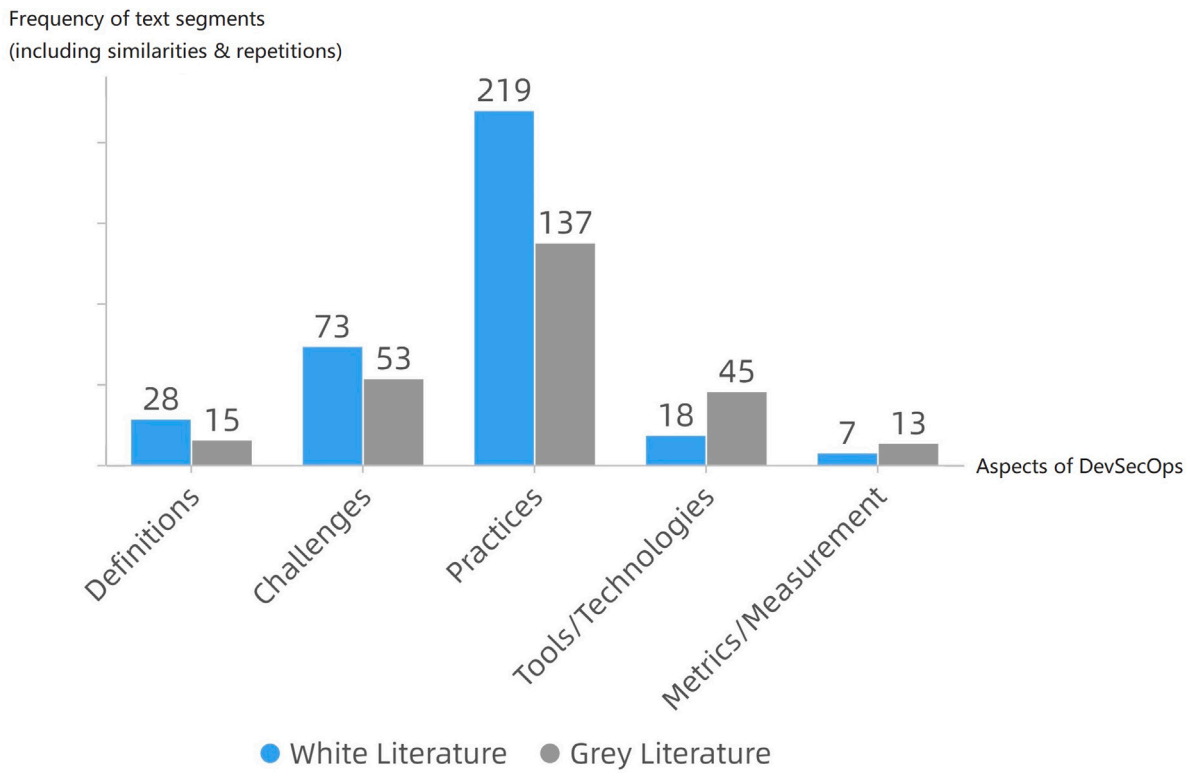

Fig. 4. Aspects of DevSecOps.

Table 4
Work relating to each aspect.

Aspects	Related WL work (Paper ID)	Related GL work (Paper ID)
Definitions	S1-ACM-04, 07, 45, 50, 68, S1-IEEE-03, 05, 06, 08, 10, 12, 21, 22, 24, 26, 44, S1-SC-01, 02, 03, 04, 09, 10, 11, 14, 21, 22, 31, CS-ACM-01, 02, 03, 04, CS-IEEE-01, 02, 03, 04, 05, 06, CS-SC-01, 02, 03	S1-GL-01, 02, 04, 05, 10, 11, 12, 13, 15, 16, 19, 23, 26, 27, 33, CS-GL-01, 02, 03, 04, 05, 06, 07
Challenges	S1-ACM-01, 05, 06, 19, 52, 59, 64, 66, 95, S1-IEEE-01, 04, 06, 07, 08, 11, 12, 16, 25, 28, 33, 39, 42, S1-SC-08, 26, CS-ACM-01, 02, 04, CS-IEEE-01, 06, CS-SC-01, 02, 03	S1-GL-13, 15, 17, 18, 19, 20, 24, 28, 29, 30, 37, 38, 39, 40, CS-GL-07
Practices	S1-ACM-01, 02, 03, 08, 09, 15, 45, 49, 50, 52, 69, 71, 72, 81, 95, S1-IEEE-02, 04, 06, 07, 09, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 24, 26, 29, 30, 31, 33, 34, 36, 38, 39, 40, 41, 43, 52, 54, 55, 57, 61, 71, 84, 86, S1-SC-07, 08, 09, 11, 15, 17, 18, 20, 22, 26, 27, 32, 34, 36, 38, 40, 41, 42, CS-ACM-01, 03, 04, CS-IEEE-01, 02, 03, 04, 05, 06, CS-SC-02, 03	S1-GL-02, 04, 06, 08, 09, 10, 11, 13, 14, 15, 17, 18, 19, 22, 23, 24, 25, 28, 30, 31, 32, 35, 36, 41, CS-GL-01, 02, 03, 04, 06, 07
Tools/Technologies	S1-ACM-52, 76, 89, 95, 99, S1-IEEE-06, 07, 18, 31, 33, 39, 55, S1-SC-01, 09, 12, 18, 20, 26, 29, 34, 42, 45, 48, CS-ACM-01, CS-IEEE-01, 02, 03, 04, 05, 06, CS-SC-02	S1-GL-01, 03, 04, 10, 21, 23, 42, CS-GL-01, 02, 03, 05
Metrics/Measurement	S1-IEEE-06, 57, CS-ACM-03, CS-SC-02	S1-GL-01, 18, 43

Table 5
Thematic analysis and synthesis results.

Aspects	Extracted data (text segments) WL/GL	Coded data	Translated codes into themes	Classified themes into categories
DevSecOps definitions	28/15 definitions	74 codes	21 themes	4 categories: OPC, PC, Technology, Business
DevSecOps challenges	73/53 challenges	85 codes	23 themes	4 categories: OPC, PC, Technology, Business
DevSecOps practices	219/137 practices	142 codes	56 themes	4 categories: OPC, PC, Technology, Business
DevSecOps metrics	7/13 metrics	20 codes	16 themes	3 categories: OPC, PC, Technology
DevSecOps tools	18/45 tools	56 codes	16 themes	Single category: Technology

emphasize that person/organization/firm are part of capabilities, that partly overlap "Organization, People and Culture" category. Thus, the composite term "Process Capabilities" is defined to avoid confusion.
- Technology: includes themes relating to technological approaches and software and hardware tools, e.g., automation, cloud, containerization, testing techniques and tools.
- Business: includes themes relating to business benefits, customers, quality of product and service, e.g., increasing value, higher quality, less impacts to users, etc. The reason for adding the Business category was that the MLR specially the GL results showed a business perspective on DevSecOps.

A. DevSecOps definitions. 28 and 15 DevSecOps definitions were extracted from WL and GL respectively, including similarities and repetitions. We labeled 74 codes from the extracted data; and translated codes into 21 themes. These themes were further classified into four categories: OPC, Process Capabilities, Technology, and Business. The four categories were elicited from our synthesis of the definitions, and formed the basis for later grouping. Table 6 lists the themes and codes with categories, frequencies and sources.

The data showed some DevSecOps definitions were repeatedly quoted or paraphrased in the papers we reviewed. Snowballing was applied with WL studies to trace the sources of these definitions, while GL has no references to enable snowballing. However, the traced sources are all secondary studies, hence, they were not included into the MLR. Table 7 identifies the authors who presented the common DevSecOps definitions that were quoted or paraphrased by selected papers. These codes were grouped into a special theme "Authors of common definitions". The frequencies of these codes (names) were counted to measure the commonality and reliability of definitions. Results show that the definition by Mohan and Othmane (2016) is the most frequently cited (9 counts): *"DevSecOps is seen as a necessary expansion to DevOps, refers to incorporating security practices in the DevOps processes by promoting collaboration between the development, operations and security teams"*.

B. DevSecOps challenges. 73 and 53 DevSecOps challenges were extracted from WL and GL respectively, including similarities and repetitions. We labeled 85 codes and classified them into 23 themes which were further classified into four categories: OPC, Process Capabilities, Technology, and Business. We validated our findings of challenges with three review papers: Myrbakken and Colomo-Palacios' MLR (Myrbakken and Colomo-Palacios, 2017), Akbar's MLR (Akbar et al., 2022), and Rajapakse's SLR (Rajapakse et al., 2022). Eventually, 28 DevSecOps challenges have been identified - 23 were from our findings and another 5 were included from the findings of Myrbakken and Colomo-Palacios' MLR (Myrbakken and Colomo-Palacios, 2017). According to frequency statistics, OPC category contains 9 challenges and ranks first, followed by PC (8), Technology (7) and Business (4). This ranking reflects the degree of attention to types of challenges in literature. However, future work such as our current Delphi study needs to be conducted to verify whether it is consistent in the real world. Moreover, it is worth noting that all the challenges identified by this study can match or at least partly match the findings of two newly published review papers (Akbar et al., 2022; Rajapakse et al., 2022), though using various categories, therefore the commonality of findings on DevSecOps challenges can be verified. Hence, the SE community needs to be aware of these identified challenges and continue doing research to improve the DevSecOps approach.

Challenges in "Organization, People and Culture" category. Table 8 lists themes and codes related to Organization, People and Culture. Items marked with an asterisk wholly or partly matched the findings of two MLR studies (Myrbakken and Colomo-Palacios, 2017; Akbar et al., 2022) and an SLR study (Rajapakse et al., 2022). C09 is an additional challenge identified by Myrbakken and Colomo-Palacios' MLR study (Myrbakken and Colomo-Palacios, 2017). Statistics report the most frequently mentioned OPC-related challenge is "C02-Challenges of collaboration, communication and coordination". This challenge reveals that the isolation between developers, operation and InfoSec would cause the lack of collaboration, communication and coordination among teams, further lead to friction, conflicts and mistrust (Rafi et al., 2020). Another most important OPC-related challenge is "C05-Lack of security knowledge and skills, need for training", that stresses the necessity of training security knowledge and skills, especially for developers (Tomas et al., 2019). In addition, "C01-Cultural resistance and organizational opposition" also deserves attention, it can be seen as the primary cause for all OPC-related challenges. Without addressing this issue at the beginning, projects may not start off well (Rafi et al., 2020; Tomas et al., 2019).

Challenges in "Process Capabilities" category. Table 9 lists themes and codes related to Process Capabilities. The asterisked items wholly or partly matched the findings of Myrbakken and Colomo-Palacios (2017), Akbar et al. (2022) and Rajapakse et al. (2022). The most frequently mentioned one is "C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance". The traditional way of including security at the end of the SDLC is inherently slow and becomes the reason to decelerate delivery, so that the main challenge of migration from DevOps to DevSecOps is to maintain the agility and speed of DevOps after implanting security (Nisha, 2022). Also, the adoption of incompatible or immature practices would hinder the DevOps environment running (Kumar and Goyal, 2020).

Challenges in "Technology" category. Table 10 lists the themes and codes related to Technology. The asterisked items partly or wholly matched the findings of Myrbakken and Colomo-Palacios (2017), Akbar et al. (2022) and Rajapakse et al. (2022). C19 and C23 are two additional challenges which were identified by Myrbakken and Colomo-Palacios' MLR study (Myrbakken and Colomo-Palacios, 2017). The most frequently mentioned technological challenge is "C21-Use of cloud and serverless computing brings security complications". It is indisputable that the increasing use of cloud and cloud-based technologies is an accelerator of DevSecOps development. However, the misconfigured cloud environments become the cause of security breaches for attacks (Fernandez and Brito, 2019). Another crucial technological challenge is "C18-Lack of mature tools for automation and security". Various prioritization-based studies (Akbar et al., 2022; Rafi et al., 2020) also identify this challenge and rank it in the top tier, thus reflecting the real needs of DevSecOps organizations.

Challenges in "Business" category. Table 11 lists the themes and codes related to Business. One notable finding is that there is no business-related challenge identified from GL. This reflects the nature of grey literature which, from the authors' experience, mostly paints a positive picture of the phenomenon being analyzed or promoted. We included two additional business challenges (C27-28) from Myrbakken and Colomo-Palacios' MLR study (Myrbakken and Colomo-Palacios, 2017).

C. DevSecOps practices. 219 and 137 DevSecOps practices were extracted from WL and GL respectively, including similarities and repetitions. We labeled 142 codes and classified them into 56 themes, which were further classified into four categories: OPC, Process Capabilities, Technology, and Business. We compared our findings with five previous review papers (Myrbakken and Colomo-Palacios, 2017; Mohan and Othmane, 2016; Prates et al., 2019; Sanchez-Gordon and Colomo-Palacios, 2020; Rajapakse et al., 2022) to validate our identified DevSecOps practices. 60 practices have been identified - 56 were identified from our included primary studies; 2 were complemented from Sánchez-Gordón and Colomo-Palacios' SLR (Sanchez-Gordon and Colomo-Palacios, 2020); and 2 were complemented from Rajapakse's SLR (Rajapakse et al., 2022). Statistics show that Technology category ranks first with 23 practices, followed by PC (17), OPC (15) and

Table 6
Thematic analysis on DevSecOps definitions.

Categories	Themes (Frequency)	Codes [Papers contributed to the code]
OPC	Dev, Sec and Ops (10)	Development, operations and security teams [S1-IEEE-05, 08, 12, S1-SC-10, 21, S1-ACM-68, S1-GL-15, 19, 27]
		dev/sec/ops [S1-IEEE-26]
	Expansion to DevOps (4)	Expansion to DevOps [S1-IEEE-08, S1-SC-21]
		Extension to DevOps [S1-SC-01]
		Extension of the DevOps [S1-GL-33]
	Culture (8)	Culture [S1-ACM-45, S1-GL-10, 13, 26]
		Cultural approach [S1-IEEE-26]
		Cultural shift [S1-ACM-50, S1-GL-11]
		Shift the mindset [S1-IEEE-10]
	Collaboration (9)	Collaboration/collaborate [S1-IEEE-08, 12, 26, S1-SC-10, 21, S1-ACM-45, 68, S1-GL-26]
		Team work [S1-GL-02]
	Breaking silos of security (4)	Breaking silos of security [S1-IEEE-08, 24, 26]
		Break down the barrier [S1-IEEE-22]
	Sharing knowledge (3)	Sharing that knowledge [S1-IEEE-08]
		Giving that knowledge to the different teams [S1-IEEE-24, 26]
	Shared responsibility (6)	Shared responsibility [S1-GL-10, 33]
		Everyone's responsibility [S1-GL-10]
		Security is a part of everyone's job [S1-GL-12]
		Make everyone accountable for security [S1-GL-27]
		At the top of every developer's mind [S1-GL-12]
	Philosophy (3)	Philosophy [S1-GL-02, 19, 26]
	Communication (1)	Communication [S1-GL-19]
	Combination of DevOps and SecOps (1)	Combination of DevOps and SecOps [S1-GL-13]
PC	Integration of security into DevOps (21)	Incorporating security practices in the DevOps processes [S1-IEEE-08, S1-SC-21]
		Incorporation of security practices in a DevOps environment [S1-SC-10, 11]
		IT processes with security approach [S1-ACM-04, S1-IEEE-21]
		Integration of security with development and operation [S1-SC-09]
		Integrating security principles [S1-IEEE-12]
		Integration of security processes and practices [S1-IEEE-10, S1-GL-10]
		Introduction of more security-oriented processes [S1-SC-22]
		Integrates continuous security into the original DevOps process [S1-IEEE-03]
		Injection of security principles and controls into the DevOps [S1-ACM-50]
		Integrating secure development best practices and methodologies into development and deployment processes [S1-IEEE-44]
		Integrating the software development and operation processes considering security and compliance requirements [S1-SC-11]
		Integrating security methods into a DevOps process [S1-GL-02]
		Integrating security practices within the DevOps process [S1-GL-26]
		Adding security components to each step of the DevOps [S1-GL-23]
		Bake security into the rapid-release cycles [S1-GL-11]
		Integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline[S1-GL-16]
		Built-in security [S1-GL-04]
	Agile (4)	Agile [S1-ACM-45, S1-IEEE-03, S1-GL-05]
		Smart and lightweight approach [S1-SC-31]
	Security is the main concern throughout the SDLC (7)	Security is the main emphasis [S1-SC-14]
		Security is given high priority throughout the SDLC [S1-ACM-07]
		Key concern throughout all phases of the development lifecycle and even post deployment [S1-SC-31]
		Security practices are implemented at each stage of the cycle [S1-ACM-07]
		Security is implemented at the right level and at right time [S1-IEEE-24]
		Emphasizes the importance of sound information security practices [S1-GL-01]
		Security through the entire SDLC [S1-GL-19]
	Shifting security to the start (8)	Puts security at the forefront of requirements [S1-IEEE-24]
		Shifting security to the early stages [S1-IEEE-06]
		Security from the start/beginning [S1-GL-04, 15, 33]
		Integrate security objectives as early as possible [S1-GL-10]
		Placing security practices early during the SDLC [S1-GL-05]
		Avoids any risk of security being an afterthought [S1-GL-01]
	Time reduction (4)	Time reduction [S1-ACM-04, S1-IEEE-21]
		Increase deployment rates [S1-IEEE-22]
		Shorten the SDLC [S1-GL-23]
	Security assurance (3)	Maintaining a secure operational atmosphere [S1-IEEE-22]
		Identifying security vulnerabilities [S1-SC-31]
		Responsible for application security [S1-IEEE-05]
Technology	Tooling (2)	Reliance on operational tools [S1-ACM-45]
		Tooling [S1-GL-10]
	Automation (2)	Automation/automating [S1-GL-04, 19]
	Security-as-Code (1)	Security as code [S1-GL-26]
Business	High quality (4)	Without lost quality [S1-ACM-04, S1-IEEE-21]
		Quality affirmation [S1-SC-14]
		High software quality [S1-GL-23]

Table 7
Authors of common definitions.

Themes	Codes [Papers contributed to the code] (Counts)
Authors of common definitions	Mohan and Othmane [S1-IEEE-08, 26, S1-SC-09, 10, 11, 21, 22, S1-ACM-45, 68] (9) Rahman and Williams [S1-IEEE-08, 12, 44, S1-SC-22] (4) Carter [S1-IEEE-24, 26] (2) Carturan and Goya [S1-IEEE-21, S1-ACM-04] (2) Myrbakken and Colomo-Palacios [S1-IEEE-10] (1) Mohan, Othmane, and Kres [S1-SC-11] (1)

Table 8
Thematic analysis on DevSecOps challenges related to OPC.

Themes/Challenges (Freq)	Codes [Papers contributed to the code]
C01-Cultural resistance and organizational opposition (7)*	Developer resistance to integrate security protocol [S1-IEEE-08, S1-ACM-05] Developers lose autonomy [S1-IEEE-06] Resistance to change [S1-GL-15] Challenge of the shifting role of security [S1-GL-37] Organizational opposition [S1-GL-24] Cultural resistance [S1-GL-20]
C02-Challenges of collaboration, communication and coordination (20)*	Teams working towards conflicting objectives [S1-SC-08] Insufficient monitoring of collaboration [S1-ACM-01] Challenge of unrestricted collaboration [S1-IEEE-08, S1-ACM-05] Coordination of security team and DevOps team [S1-IEEE-08, S1-ACM-05] Untrusted inputs causing isolation [S1-IEEE-08, S1-ACM-05] Conflict between security and development [S1-IEEE-06] Collaboration challenges [S1-GL-28, 29] Conflicting aims [S1-GL-38, 40] Failing to collaborate with the InfoSec team [S1-GL-18] Lack of coordination between InfoSec team and developers [S1-GL-19] Gaps between DevOps and Security teams [S1-GL-20] Disconnect between security and development [S1-GL-39] Friction between development and security [S1-GL-13] Communication requirements [S1-GL-15] Lack of common process and platform for communication, collaboration, and sharing information and feedback [S1-SC-08]
C03-Neglecting security (3)*	Not prioritize security [S1-IEEE-06] Focused on velocity, not security [S1-GL-17] Neglect security [S1-GL-30]
C04-Lack of security awareness and responsibility (3)*	Security awareness [S1-IEEE-06] Nobody is responsible for security [S1-IEEE-06] Security push-pull [S1-IEEE-06]
C05-Lack of security knowledge and skills, need for training (9)*	Lacking security education [S1-IEEE-06] Lacking knowledge and training [S1-IEEE-06] Lack of security knowledge [S1-IEEE-08, S1-GL-38, 40] Developers are not security specialists [S1-GL-15] Unfamiliar with common security risks [S1-GL-18] The skills gap [S1-GL-37] Not enough security savvy [S1-GL-39]
C06-Recruiting challenges (3)	Recruiting challenges [S1-GL-24] Understaffing InfoSec teams and engaging too late with the InfoSec team [S1-GL-18] Boundary between specialist and generalist [S1-IEEE-06]
C07-Inconsistent security polices design (2)*	Inconsistent security polices design [S1-ACM-05, S1-IEEE-08]
C08-Challenges of governance and leadership (1)*	Insufficient level of governance on DevSecOps adoption [S1-SC-08] Lack of clarity and transparency in strategy [Myrbakken and Colomo-Palacios' MLR] Lack of commitment of leadership and senior management [Myrbakken and Colomo-Palacios' MLR]
C09-Lacking confidence*	Low or no confidence in DevSecOps [Myrbakken and Colomo-Palacios' MLR]

Business (5). Unsurprisingly, the literature mostly covers technology-related practices, that suggests the research and practitioner focus, not necessarily the real importance and adoption of DevSecOps practices based on facts.

Practices in "Organization, People and Culture" category. Table 12 lists themes and codes related to OPC. Sánchez-Gordón and Colomo-Palacios's SLR (Sanchez-Gordon and Colomo-Palacios, 2020) characterized DevSecOps from people and cultural perspective. Rajapakse's SLR (Rajapakse et al., 2022) mentioned DevSecOps solutions related to people. Hence, these two SLR studies were used to validate our findings in terms of OPC-related practices. The asterisked items wholly or partly matched the findings of the SLR studies (Sanchez-Gordon and Colomo-Palacios, 2020; Rajapakse et al., 2022), and two additional practices (P14-15) from Sanchez-Gordon and Colomo-Palacios (2020) were included to make our findings more complete. Statistics show that the most frequently mentioned OPC-related practice is "P02-Improving collaboration, communication and cooperation", that exactly corresponds to the most cited OPC-related challenge identified before "C02-Challenges of collaboration, communication and coordination". Rahman and Williams (2016) observed and analyzed the collaboration and communication between Dev and Ops, between Dev and Sec, and between Ops and Sec, and findings show that Sec teams actively collaborate with Dev and Ops teams in established DevOps organizations, and a supervised collaboration among teams might help to improve the automated deployment for system's security. However,

Table 9
Thematic analysis on DevSecOps challenges related to Process Capabilities.

Themes/Challenges (Freq)	Codes [Papers contributed to the code]
C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance (11)*	Integrate security practices into a fast moving DevOps pipeline without slowing down [S1-SC-08] Implementing security in CI/CD [S1-GL-28] Rapid pace of change [S1-GL-29] Faster development process [S1-GL-28] Keep up with the pace of DevOps [S1-GL-30] DevOps velocity [S1-GL-37] Slow security testing [S1-GL-38, 40] Running current product and services in parallel to its transformation to DevSecOps [S1-SC-08] Tradeoff between security measures and CI system performance [S1-ACM-95] Interconnectedness of the DevOps process [S1-GL-28]
C11-Using unsuitable metrics (3)*	Using unsuitable metrics [S1-ACM-01, 05, S1-IEEE-08]
C12-Compliance requirements (5)*	Compliance requirements [S1-IEEE-07, 08, 11, S1-ACM-05, S1-GL-39]
C13-Neglecting change control in security (1)*	Neglecting change control in security [S1-IEEE-08]
C14-Lack of standards (2)*	Lack of security standards [S1-IEEE-08] Lack of tool standards [S1-IEEE-06]
C15-Ignoring processes and security essentials leading to technical and security debt (1)	Ignoring processes and security essentials leading to technical debt and security debt [S1-SC-08]
C16-Poor visibility of security track record (1)	Poor visibility of security track record [S1-GL-19]
C17-Inadequate privileged credentials and access controls causing cyber attacks (2)	Inadequate controls provide an opening for attack [S1-GL-30] Privileged credentials used in DevOps are targeted by cyber attackers [S1-GL-17]

Table 10
Thematic analysis on DevSecOps challenges related to Technology.

Themes/Challenges (Freq)	Codes [Papers contributed to the code]
C18-Lack of mature tools for automation and security (19)*	Lack of automated testing tools [S1-IEEE-06, 08, S1-ACM-05] Lack of integrated testing tools [S1-IEEE-08, S1-ACM-05] Wrong automated deployment tools [S1-IEEE-12, S1-ACM-01] Immature automated tools [S1-IEEE-08, 12, S1-ACM-01, 05] Need for automated testing [S1-IEEE-08, S1-ACM-05] Mismatched tools [S1-GL-15] Tool-centric approaches to secrets management create security gaps [S1-GL-17] Inefficient SAST tools [S1-GL-19] Manual pen-testing becomes a bottleneck [S1-GL-19] Threat modeling scalability issue [S1-IEEE-08, S1-ACM-05]
C19-Complexity in managing different tools*	Complexity in managing different tools [Myrbakken and Colomo-Palacios' MLR]
C20-Challenges of legacy system refactoring (4)*	Challenging to automate legacy system [S1-IEEE-06] Lack of cloud support [S1-GL-19] Systems are not scalable [S1-GL-19] Legacy infrastructure [S1-GL-24]
C21-Use of cloud and serverless computing brings security complications (21)*	Cloud security complications [S1-SC-25, 44, S1-IEEE-06, 16, 25, 39, S1-ACM-19, 52, 59, 66, S1-GL-24, 29, 38, 40] Attacks due to miss-configured cloud environments [S1-IEEE-33, 42] Security smells in Infrastructure as Code [S1-ACM-06, S1-IEEE-28, S1-SC-26] Security smells in serverless computing [S1-GL-28] Cloud and open source environments lead to compromise of critical information, configuration errors, compliance issues and security breaches [S1-GL-20]
C22-Containers and other tools come with their own risks (3)*	Container and other tools can often be the reason for security concerns [S1-GL-20] Workload containerization [S1-GL-29] Tools come with their own risks [S1-GL-30]
C23-Availability and reliability of infrastructure, tools, automation, and network bandwidth*	Availability and reliability of infrastructure resources, tools, automation, and network bandwidth for shorter and frequent deployment cycle [Myrbakken and Colomo-Palacios' MLR]
C24-Continuous deployment chaos (1)	Continuous deployment chaos [S1-GL-19]

Table 11
Thematic analysis on DevSecOps challenges related to Business.

Themes/Challenges (Freq)	Codes [Papers contributed to the code]
C25-Challenges of cost control (2)	High cost such as salaries for security experts, costs on new tools [S1-IEEE-04] Risk and cost battle [S1-IEEE-06]
C26-Conflicts between security and business (2)	Security and business objectives are implemented using conflicting approaches [S1-ACM-64] Dilemma in selection of business processes in product and service delivery for transformation to DevSecOps [S1-SC-08]
C27-Customer readiness for frequent releases*	Customer readiness for applying frequent releases [Myrbakken and Colomo-Palacios' MLR]
C28-Training users for using advanced tools*	Users need to be properly trained when using advanced tools [Myrbakken and Colomo-Palacios' MLR]

Table 12
Thematic analysis on DevSecOps practices related to OPC.

Themes/Practices (Freq)	Codes [Papers contributed to the code]
P01-Cultural shift to security (3)*	Cultural shift [S1-GL-41] Change the security mindset [S1-GL-32] Make security a priority [S1-GL-32]
P02-Improving collaboration, communication and cooperation (34)*	Work collaboratively [S1-ACM-02] Enhanced collaboration [S1-ACM-02] Cross-departmental collaboration [S1-IEEE-04] Collaborating development, operation and security [S1-IEEE-04, 12] Close collaboration [S1-IEEE-12] Collaboration within and between different teams [S1-IEEE-12] Collaboration amongst different departments [S1-IEEE-12] Collaboration between Dev and Ops [S1-IEEE-12] Collaboration between Dev and Sec [S1-IEEE-12] Collaboration between Sec and Ops [S1-IEEE-12] Team collaboration [S1-IEEE-15] Strong collaboration [S1-IEEE-15] Strong communication [S1-IEEE-12] Close communication [S1-ACM-02, S1-IEEE-09] Communication of security requirements [S1-ACM-02] Virtual communication [S1-ACM-02] Face-to-face communication [S1-ACM-02] Physical communication [S1-ACM-02] Trust [S1-ACM-02, S1-IEEE-29] Trustworthy [S1-ACM-02] Trusted relationships [S1-ACM-02] Mutual trust [S1-ACM-02] Implicit trust [S1-ACM-02] Trust within the teams [S1-IEEE-29] Cross-functional collaboration [S1-GL-30] Foster collaboration [S1-GL-25] Open contribution and collaboration [S1-GL-24] Collaboration and integration [S1-GL-02] Communicate and collaborate [S1-GL-32] Improving empathy and cooperation [S1-GL-10] Reducing friction [S1-GL-10]
P03-Shared and collective responsibility for security (3)*	Shared responsibility for security [S1-ACM-02] Collective responsibility [S1-GL-02] Assign security responsibility to one person from DevOps team [S1-GL-28]
P04-Shared knowledge (3)*	Knowledge sharing [S1-ACM-02] Learn from each other [S1-GL-32] Shared threat intelligence [S1-GL-24]
P05-Training, learning and education for security (6)*	Training [S1-GL-06, 10, 32] Cross-training [S1-GL-35] Educate developers [S1-GL-25] Security learning [S1-GL-14]
P06-Security champions (2)*	Security champions [S1-ACM-02, S1-GL-10]
P07-Recruiting success (1)*	Recruiting success [S1-GL-10]
P08-Continuous feedback loop (6)*	Feedback loop [S1-ACM-15] Continuous feedback loops [S1-GL-09, 13, 15, 22, 35]
P09-Be reactive and responsive (1)	Be reactive and responsive [S1-GL-32]
P10-Shameless retrospectives (1)*	Shameless retrospectives [S1-IEEE-09]
P11-Impose security policies (2)*	Impose policy and governance [S1-GL-41] Implement security policies [S1-GL-30]
P12-Commitment and agreement (1)*	Commitment and agreement [S1-IEEE-29]
P13-Enhance transparency (2)*	Transparency [S1-IEEE-29, S1-SC-09]
P14-Continuous improvement mindset*	Continuous improvement mindset [Sánchez-Gordón and Colomo-Palacios' SLR]
P15-Leadership support*	Leadership support [Sánchez-Gordón and Colomo-Palacios' SLR]

unrestricted collaboration might cause inappropriate access to system resources, and further might hurt system's security. The second most cited OPC-related practice is "P05-Training, learning and education for security", which also corresponds to the No. 2 challenge "C05-Lack of security knowledge and skills, need for training".

Practices in "Process Capabilities" category. Table 13 lists themes and codes related to PC. The asterisked items wholly or partly matched the findings of Myrbakken and Colomo-Palacios' MLR (Myrbakken and Colomo-Palacios, 2017) and Rajapakse's SLR (Rajapakse et al., 2022). One additional practice (P31) was complemented from Rajapakse's SLR study (Rajapakse et al., 2022). In this category, the most frequently mentioned practice is "P16-Shifting security to the left (early)", which could address the challenge "C10-Difficulties in integrating security into DevOps without losing speed and affecting current process and performance". In a variety of practices and applications, shift-security-left is always regarded as the core idea of DevSecOps.

Practices in "Technology" category. Table 14 lists the themes and codes related to Technology. The asterisked items matched or partly matched the finding of Mohan and Othmane's mapping research (Mohan and Othmane, 2016), Myrbakken and Colomo-Palacios' MLR (Myrbakken

Table 13
Thematic analysis on DevSecOps practices related to Process Capabilities.

Themes/Practices (Freq)	Codes [Papers contributed to the code]
P16-Shifting security to the left (early) (18)*	Shifting security to the left [S1-IEEE-04, 24, 26, S1-SC-08, 11, S1-ACM-50, 81] Moving security to the left [S1-GL-08, 09, 13, 15, 18, 31, 35, 36] Integrate security during the planning phase [S1-GL-35] Take a proactive approach to security [S1-GL-17] Include security early [S1-GL-28]
P17-Security-by-Design (12)*	Security by design [S1-SC-07, 08, 18, 20, 22, S1-IEEE-16, 29, 30, 36, S1-ACM-45, 69, S1-GL-31]
P18-Increase the visibility (2)	Increase the visibility [S1-SC-09] Enhance visibility [S1-GL-41]
P19-Good documentation, logging and reporting (3)*	Good documentation and logging [S1-IEEE-15] Better reporting [S1-GL-02, 19]
P20-Compliance control (6)	Compliance control [S1-IEEE-11, S1-SC-27, S1-GL-10, 24] Identify compliance requirements beforehand [S1-GL-28] Bridging the divide between compliance and development [S1-GL-02]
P21-Risk management (9)*	Risk management (including risk assessment, risk treatment and risk control) [S1-SC-11, 18, 20, 22, 26, 40, 41, S1-ACM-03, S1-IEEE-34]
P22-Vulnerability and incident management (5)*	Vulnerability and incident management [S1-GL-14] Incident management [S1-GL-08, 10] Vulnerability management [S1-GL-23, 30]
P23-Privilege management (3)*	Least privilege controls [S1-IEEE-33] Privileged access management [S1-GL-30] Secure access via secrets management [S1-GL-41]
P24-Configuration management (1)	Configuration management [S1-GL-10]
P25-Patch management (1)	CI/CD for patching management [S1-GL-10]
P26-Define metrics (3)*	Define metrics [S1-GL-06, 19] Measurement [S1-GL-02]
P27-Software process maturity (2)*	Software process maturity [S1-SC-32] Building Security In Maturity Model (BSIMM) model [S1-ACM-01]
P28-Define security requirements (2)*	Define security requirements [S1-GL-06] Security requirements and design [S1-GL-14]
P29-Security review and evaluation (8)*	Security reviews [S1-GL-18] Security evaluation [S1-GL-14] Proactive security assessments [S1-GL-10] Detect existing security flaws [S1-SC-09] Make sure the basics of host and network security are in place [S1-SC-09] Host hardening [S1-GL-10] Application-level assessment [S1-GL-10] Operational controls validation and improvement [S1-GL-14]
P30-Keep credentials safe (1)	Keep credentials safe [S1-GL-06]
P31-Common weaknesses enumeration (1)	Common weaknesses enumeration [S1-GL-08]
P32-Hybrid life cycles with data-security focus*	Combining data security and software development life cycles [Rajapakse's SLR]

and Colomo-Palacios, 2017), and Rajapakse's SLR (Rajapakse et al., 2022). One additional practice (P55) was complemented from Rajapakse's SLR study (Rajapakse et al., 2022). As expected, the most frequently mentioned practice is "P33-Automate tools and security processes", not only in this category, but also in all DevSecOps practices. Automation is one of the pillars of DevSecOps and DevOps (Humble and Molesky, 2011), typical applications include: automated testing, automated code reviews, automated scans and automated monitoring. From the grey literature, we find that the leading DevSecOps organizations are committed to automate their security process as much as possible, hence, selecting and using appropriate automated tools is a key factor to DevSecOps' success.

Practices in "Business" category. Table 15 lists the themes and codes related to Business. The business-related practices were discovered from GL. WL studies appear to have ignored this category. While the practitioners from industry provide insights into DevSecOps practices from the real business perspective, this category at the moment lacks the academic view.

D. DevSecOps metrics. The result showed that only two WL papers and three GL articles mentioned the measurement or metrics of DevSecOps. 7 and 13 metrics were extracted from WL and GL respectively, and no duplicates and similarities between WL and GL. This reflects that the existing literature (particularly WL) is lacking on DevSecOps metrics, and there has been no adequate exchange and consensus on this aspect between academia and industry. We identified 20 codes and 16 themes, further classified into three categories: OPC, Process Capabilities, and Technology. Tables 16 and 17 depict the TA results on DevSecOps metrics, with measuring and goal. Two previous MLR studies (Prates et al., 2019; Myrbakken and Colomo-Palacios, 2017) were used to validate our "Metrics" findings. Asterisked items represent the matched metrics (M02, 06, 09-12). Three additional metrics (M07, 08, 19) were complemented from Prates et al. (2019) and were grouped into "PC" and "Technology" categories. It is worth noting that Prates' MLR (Prates et al., 2019) is the only review work which involved DevSecOps metrics so far, and the results were also identified mainly from GL work. "M20-Business metrics" were complemented from Myrbakken and Colomo-Palacios (2017) and grouped into "Business" category. There are no business-related metrics identified from our included studies. Eventually, a total of 20 DevSecOps metrics are identified. OPC-related and business-related metrics are relatively scarce, compared with the other two categories. Moreover, a newly published paper by Amaro et al. (2023) elicited 24 DevOps metrics through MLR and interview. When we mapped our 20 identified metrics to their findings, 13 DevSecOps metrics can match 10 DevOps metrics. (Table 18, M stands for DevSecOps metrics; Me stands for DevOps metrics). The comparison shows that approximately half of the DevOps metrics are

Table 14
Thematic analysis on DevSecOps Practices related to Technology.

Themes/Practices (Freq)	Codes [Papers contributed to the code]
P33-Automate tools and security processes (93)*	Automation [S1-ACM-01, 09, 49, 71, 72, 81, 95, S1-IEEE-06, 07, 09, 10, 12, 13, 15, 20, 21, 26, 38, 41, 54, 57, S1-SC-08, 09, 11, 17, 18, 20, 22, 26, 27, 32, 40, S1-GL-02, 04, 06] Automated/automating test/testing [S1-ACM-01, 09, 49, 81, 95, S1-IEEE-06, 07, 09, 10, 12, 15, 21, 26, 38, 41, 54, 57, S1-SC-08, 09, 11, 17, 18, 22, 26, 27, S1-GL-08,11,13,15, 35] Automated monitoring [S1-ACM-01, 71, 72, 81, S1-IEEE-07, 12, 13, 15, 21, 26, 38, S1-SC-08, 09, 18, 20, 26, 40] Automated/automating scans [S1-IEEE-07, S1-SC-32] Automated/automating code review [S1-IEEE-07, 12, S1-GL-23] Automate as much as possible [S1-GL-25, 28] Automate protection of business logic flaws [S1-GL-09] Automate tools and security processes [S1-GL-17, 30] Use automated security tools [S1-GL-41]
P34-Security-as-Code (5)*	Security as code [S1-SC-08, 09, 18, S1-IEEE-06, S1-GL-32]
P35-Threat modeling (15)*	Threat modeling/analysis [S1-IEEE-02, 04, 07, 11, 30, 36, 39, 61, 71, S1-SC-26, S1-GL-06, 10, 14, 25, 28]
P36-Continuous monitoring (22)*	Continuous monitoring [S1-IEEE-07, 12, 13, 15, 21, 26, 38, S1-SC-08, 09, 18, 20, 26, 40, S1-ACM-01, 15, 71, 72, 81, S1-GL-02, 06, 25, 31] 24 x 7 proactive monitoring [S1-GL-24]
P37-Secure coding (6)	Source code repository and scanning [S1-GL-10] Secure coding [S1-GL-10, 14, 28] Build preapproved code [S1-GL-18] Conduct code dependency checks regularly [S1-GL-25]
P38-Advanced malware detection (1)	Advanced malware detection employs machine learning and deep learning [S1-SC-32]
P39-Cloud security (4)	Verify cloud infrastructure [S1-GL-28] MUSA Security DevOps framework [S1-ACM-52] MUSA DevOps framework for security in multi-cloud applications [S1-IEEE-16, 40]
P40-Container security (14)*	Container/Containerization security [S1-ACM-52, S1-IEEE-55, S1-GL-28, 41] Run container as non-root users [S1-IEEE-55, S1-SC-09, 34] Use the latest version of image [S1-SC-42] Conduct deep scanning of container image [S1-IEEE-04] Enhance security of Docker [S1-IEEE-31, S1-GL-10] Security practices in Kubernetes [S1-IEEE-18, S1-GL-10] Version control, metadata and orchestration [S1-GL-10]
P41-Sensitive information scan (1)	Sensitive information scan [S1-GL-23]
P42-Software Composition Analysis (2)	Software composition analysis [S1-GL-06, 23]
P43-Red team security drills (2)*	Red team security drills [S1-IEEE-04] Red and blue team exploit testing [S1-GL-24]
P44-Fault injection (chaos engineering) (1)	Fault injection (chaos engineering) [S1-IEEE-13]
P45-RASP (4)	Runtime Application Self-Protection (RASP) [S1-SC-32, S1-GL-02, 08, 25]
P46-SAST (4)*	Static Application Security Testing (SAST) [S1-GL-02, 08, 23, 25]
P47-DAST (5)	Dynamic Application Security Testing (DAST) [S1-IEEE-10, S1-GL-02, 08, 23, 25]
P48-IAST (5)*	Interactive Application Security Testing (IAST) [S1-IEEE-15, S1-GL-02, 08, 19, 25]
P49-Immutable-as-Code (1)	Immutable-as-code ensures the immutability of infrastructure and avoids accidental configuration drifts [S1-IEEE-33]
P50-Policy-as-Code (2)	Policy-as-Code is an attempt to code the policy itself [S1-IEEE-33, S1-GL-17]
P51-Design-as-Code (1)	Design-as-code: CAIRIS (Computer Aided Integration of Requirements and Information Security) model [S1-IEEE-36]
P52-Compliance-as-Code (1)	Compliance as code [S1-GL-23]
P53-Adopting DevSecOps in microservices-based applications (8)	Adopting DevSecOps in microservices-based applications [S1-IEEE-17, 43, 52, 57, 84, 86, S1-SC-15, 36]
P54-Integrate security issues within your general bug tracker (1)	Integrate security issues within your general bug tracker [S1-GL-19]
P55-Big data and behavioral analytic techniques*	Obtain fast feedback from end users and predictive analytic for trends in user behaviors [Rajapakse's SLR]

Table 15
Thematic analysis on DevSecOps practices related to Business.

Themes/Practices (Freq)	Codes [Papers contributed to the code]
P56-Consumable security services with APIs (1)	Consumable security services with APIs [S1-GL-24]
P57-Separation of duties (2)	Separation of duties [S1-GL-14, 17]
P58-Business-driven security (1)	Business-driven security [S1-GL-24]
P59-Linear scalability and affordable cost (1)	Linear scalability and affordable cost [S1-GL-19]
P60-Availability and business continuity management (1)	Availability and business continuity management [S1-GL-14]

Table 16
Thematic analysis on DevSecOps metrics I.

Categories	Themes/Metrics (Frequency)	Codes [Papers contributed to the code]
OPC	M01-Security-trained rate (1)	The ratio of developers that have gone through security-training in the team [S1-IEEE-06] *Measuring: The number of developers that have gone through security-training divided by the total number of developer in the team. Higher rate means better training.* *Goal: Know the number and the level of developers with good security mindset, knowledge and skills.*
PC	M02-Top vulnerability (3)*	Number of mistakes in different security categories [S1-IEEE-06] OWASP top 10 [S1-IEEE-06] Top vulnerability types and recurring bugs [S1-GL-43] *Measuring: Count the number of different types of mistakes and keep track of most recurring vulnerabilities.* *Goal: Help planning training provided to developers accordingly and capacitate them with knowledge to handle and mitigate returning vulnerabilities*
	M03-Time spent correcting mistakes in each category (1)	Time spent correcting mistakes in each category [S1-IEEE-06] *Measuring: Count the time spent correcting mistakes different vulnerability types. The shorter, the easier.* *Goal: Assess the difficulties of addressing different vulnerability types.*
	M04-Security review performance (3)	Whether features undergo a security review [S1-GL-18] *Measuring: The percentage of features that undergo security review early in the design process. This percentage should go up over time.* *Goal: Know the current state and progress of security reviews.* Whether security review slows down the development cycle [S1-GL-18] *Measuring: How much time the reviews add to the development process. The time that security reviews take should go down until it reaches an agreed-to minimum.* *Goal: Assess the efficiency of security reviews.* How well security is integrated into the delivery lifecycle [S1-GL-18] *Measuring: Measure the number of security reviews captured at each of the stages of the software development lifecycle (design, develop, test, and release). This number should go up until it reaches a value that suggests that InfoSec is fully integrated into the lifecycle.* *Goal: Know the degree of InfoSec team's involvement in each step of the software delivery lifecycle.*
	M05-SLA performance (1)	SLA performance [S1-GL-43] *Measuring: set up service level agreements (SLAs) based on criticality and tracking the SLA performance religiously* *Goal: Assess the SLA performance*
	M06-Critical risk profiling (1)*	Critical risk profiling–the relation between issue criticality and the value of that vulnerability to possible attackers [S1-GL-43] *Measuring: Vulnerability should be associated with a score for a criticality and another that defines the value of that vulnerability to attackers. Vulnerabilities that have high scores in both criticality and value should be addressed first. The scores are expected being as small as possible.* *Goal: Prioritize the order of addressing issues.*
	M07-Point of risk per device*	Point of risk per device [Prates' MLR] *Measuring: Identify and keep track of un-patched vulnerabilities per server. The number of vulnerabilities should tend to zero.* *Goal: Prioritize vulnerabilities according to their criticality giving special attention to the ones that are most exposed to attack from the internet.*
	M08-Number of continuous delivery cycles per month*	Number of continuous delivery cycles per month [Prates' MLR] *Measuring: Count the number of attempts to deploy versus the number of successful attempts. A positive value is to have the highest number of successful attempts.* *Goal: Measure how quickly code changes can be deployed to production.*

related to security, which further implies that a fair portion of DevOps teams have always been attaching great importance to the security aspects, even though they do not claim to be adopting DevSecOps.

E. DevSecOps tools. 18 and 45 tools were extracted from WL and GL respectively. We used tools' names as codes, and grouped 56 tools/codes into 16 themes based on their functions. The theme was based on the core function if a tool had multiple functions. All themes of tools were classified into the "Technology" category. Table 19 reports the identified tools. We compared our findings with Mohan and Othmane's mapping study (Mohan and Othmane, 2016) and complemented sets of monitoring and alerting tools, cyber security tools, and logging tools. Hence, we finally identified 18 tool groups. The result shows that container tools such as Docker and Kubernetes are the most prominent in the existing literature, especially white literature. To enhance the security of containers, container security tools such as Twistlock, Notary and Aqua Security could be selected. The second most frequently mentioned type of tool is an automation platform, e.g., Chef, Jenkins and Puppet. This reflects that automation plays a key role in DevSecOps and DevOps projects. From another point of view, namely, the grey literature or practitioners, the theme mentioned the most is vulnerability management tools, covering a variety of brands and products, e.g., Snyk, ArcherySec, Defect Dojo, HackerOne, etc.

4.1.3. Links between aspects and themes - CPTM model for DevSecOps

There are no mature DevSecOps models created by the white literature. For examples, Mohammed et al. (2017) defines the main steps of the SDLC: Requirements, Design, Coding, Testing, Deployment, and Maintenance; Pothukuchi et al. (2023) defines SDLC steps: Discovery, Design, Development, Testing and QA, Release, and Maintenance. GL work such as Jireh (2016) presents a DevOps model to depict the SDLC with eight steps: Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor. Similar to the DevOps model but more security-oriented, Gartner (MacDonald and Head, 2016) presents a DevSecOps model, which decomposes DevSecOps lifecycle into ten steps: Plan, Create, Verify, Preproduce, Release, Prevent, Detect, Respond, Predict, and Adapt. These steps form a loop which starts with planning, where each iteration is completed and the next one is improved. The DevOps model by Jireh (2016) and the DevSecOps model by Gartner (MacDonald and Head, 2016) have been universally accepted by both of the academia and industry, and are the most frequently cited DevOps/DevSecOps models in the existing white and grey literature (Myrbakken and Colomo-Palacios, 2017). Table 20 defines the steps of Gartner (MacDonald and Head, 2016) DevSecOps model, and Table 21 maps our identified themes to these steps.

To answer Sub-question 1.3 *"How do the identified aspects and themes link to each other?"*, a Challenge-Practice-Tool-Metric (CPTM) model for

Table 17
Thematic analysis on DevSecOps metrics II.

Categories	Themes/Metrics (Frequency)	Codes [Papers contributed to the code]
Technology	M09-Number of adversaries per application (1)*	Number of adversaries per application–is associated with the practice of Threat Modeling and Risk Analysis [S1-GL-43] *Measuring: Team exercise where the objective is to think how many adversaries they think an application as and register those findings.* *Goal: Identify the applications inside an organization that are more exposed to possible attacks and prepare accordingly.*
	M10-Adversary return rate (1)*	Adversary return rate–Measures how often an adversary will use the same strategy and procedures [S1-GL-43] *Measuring: Measure is done by counting the number of times adversaries use the same attacking strategy and compiling into a ranking that visible for every team member. Ideal is to have a plan to handle each attacking strategy.* *Goal: Define appropriate training and preparing to better handle these known attacks.*
	M11-Defect density (1)*	Defect density–the number of confirmed defects detected in software/component during a defined period of development/operation divided by the size of the software/component [S1-GL-43] *Measuring: Defect density is measured by dividing the total number of confirmed defects by the total line of codes of all the modules in the new release. Ideal is to have the lowest density value possible.* *Goal: Helps Sec team and developers negotiate reasonable goals to reduce defect density over time.*
	M12-Defect burn rate (1)*	Defect burn rate–indicates how quickly the team is addressing defects. [S1-GL-43] *Measuring: Take the total number of defects found in development and divided it by the sum of defects found in development and production and multiplied by 100. The rate is higher, the team is more effective.* *Goal: Measure Dev team productivity solving defects.*
	M13-Penetration test pass rate (1)	Systems that are affected by internal and external penetration testing [S1-IEEE-06] *Measuring: The degree of system that passed authorized and simulated cyberattacks.* *Goal: evaluate the security of the system in a simulated scenario.*
	M14-Security test pass rate (1)	Security test pass rate [S1-IEEE-57] *Measuring: The ratio of failed-versus-pass static security source code scans in a given time period.* *Goal: Identify security vulnerabilities in the build stage.*
	M15-Code scanning detection rate (1)	Code scanning detection rate [S1-IEEE-57] *Measuring: Count the number of security scans that come back with a problem in a given timeframe or given process phase, as well as the number of problems. This rate should decrease with time or with movement from one stage to the next.* *Goal: Improvements in this metric over time can increase confidence in the safety and security of the product.*
	M16-Whether automated testing covers security requirements (1)	Whether automated testing covers security requirements [S1-GL-18] *Measuring: As InfoSec gains greater input into the testing process, the number or percentage of security requirements that are included in the automated testing process. This percentage should go up over time.* *Goal: Know the degree of InfoSec team's involvement in writing automated tests.*
	M17-Use of preapproved libraries, packages, tool chains, and processes (1)	Use of preapproved libraries, packages, tool chains, and processes [S1-GL-18] *Measuring: Initially, measure whether InfoSec is engaged in tools development. As work progresses, the number of InfoSec-approved libraries, packages, and tool chains that are available, or the number of these resources that are used by the development and operations teams. Engagement should increase over time until the organization agrees that InfoSec oversight of tools is at the correct level. Similarly, the percentage or number of preapproved tools in use should increase until the team uses all the tools that InfoSec has created or approved.* *Goal: Know the degree of InfoSec team's engagement in tools development and the usage of preapproved libraries, packages, tool chains.*
	M18-Use of SAFe DevOps Health Radar (1)	Use of SAFe DevOps Health Radar [S1-GL-01] *Measuring: Use SAFe DevOps Health Radar to measure DevOps performance, by assessing the maturity of four aspects and 16 activities of the CI/CD pipeline.* *Goal: know the maturity of DevOps.*
	M19-Number of issues during red teaming drills*	Number of issues during red teaming drills [Prates' MLR] *Measuring: Count the number of defects found and fixed by the Red Team.* *Goal: Measure the effectiveness of Red Team.*
Business	M20-Business metrics*	Business metrics [Myrbakken and Colomo-Palacios' MLR] Revenue [Myrbakken and Colomo-Palacios' MLR] Key performance indicators (KPI) [Myrbakken and Colomo-Palacios' MLR] *Measuring: Define suitable DevOps KPIs for the organization and assess the revenue accurately.* *Goal: Know the current state in business views, and find out how to improve it.*

DevSecOps (Fig. 5) has been created to cover the identified challenges, practices, tools and metrics; and to show the links between these four elements associated with the four categories, which are: "Organization, People and Culture" is shaded in yellow; "Process Capabilities" in blue; "Technology" in green; and "Business" in red. The ten steps of the Gartner DevSecOps model (MacDonald and Head, 2016) have been integrated into this CPTM model, as discussed in Section 3.8.2 on model creation, and all identified themes of the four elements have been allocated to these lifecycle steps. The CPTM model is the result of conducting the TA process and the main contribution of this MLR. Due to layout constraints, its full version is shared at zenodo.org (https://doi.org/10.5281/zenodo.7959584). In addition to the interlink, the CPTM model is decomposed into separate Figs. 6–9 for readability.

In the CPTM model, within each step of the DevSecOps lifecycle, there are four columns to indicate the four elements, namely, Challenges, Practices, Tools and Metrics. The connecting lines demonstrate the relationships between the four elements. For example, it shows what practices can be adopted to overcome the challenges; what tools can be used in the practices; and what metrics can be applied to measure the performance of DevSecOps practices. However, in the model, one challenge may correspond to multiple practices; and not each practice has its corresponding tools and metrics. A few items in

Table 18
DevSecOps metrics mapped to DevOps metrics.

DevOps metrics by Amaro et al. (2023)	DevSecOps metrics
Me01-Mean time to recover/restore	M03-Time spent correcting mistakes in each category
Me03-Deployment frequency	M08-Number of continuous delivery cycles per month
Me07-Mean time to detection	M15-Code scanning detection rate
Me09-Defect escape rate	M11-Defect density; M12-Defect burn rate
Me11-SLAs and SLOs	M05-SLA performance
Me13-Production error and Incident rate	M04-Security review performance; M06-Critical risk profiling; M07-Point of risk per device
Me14-Customer tickets volume and feedback	M20-Business metrics
M17-Pipeline automated tests pass rate	M14-Security test pass rate
Me18-Westrum culture measures	M01-Security-trained rate
Me19-Automated test code coverage	M16-Whether automated testing covers security requirements

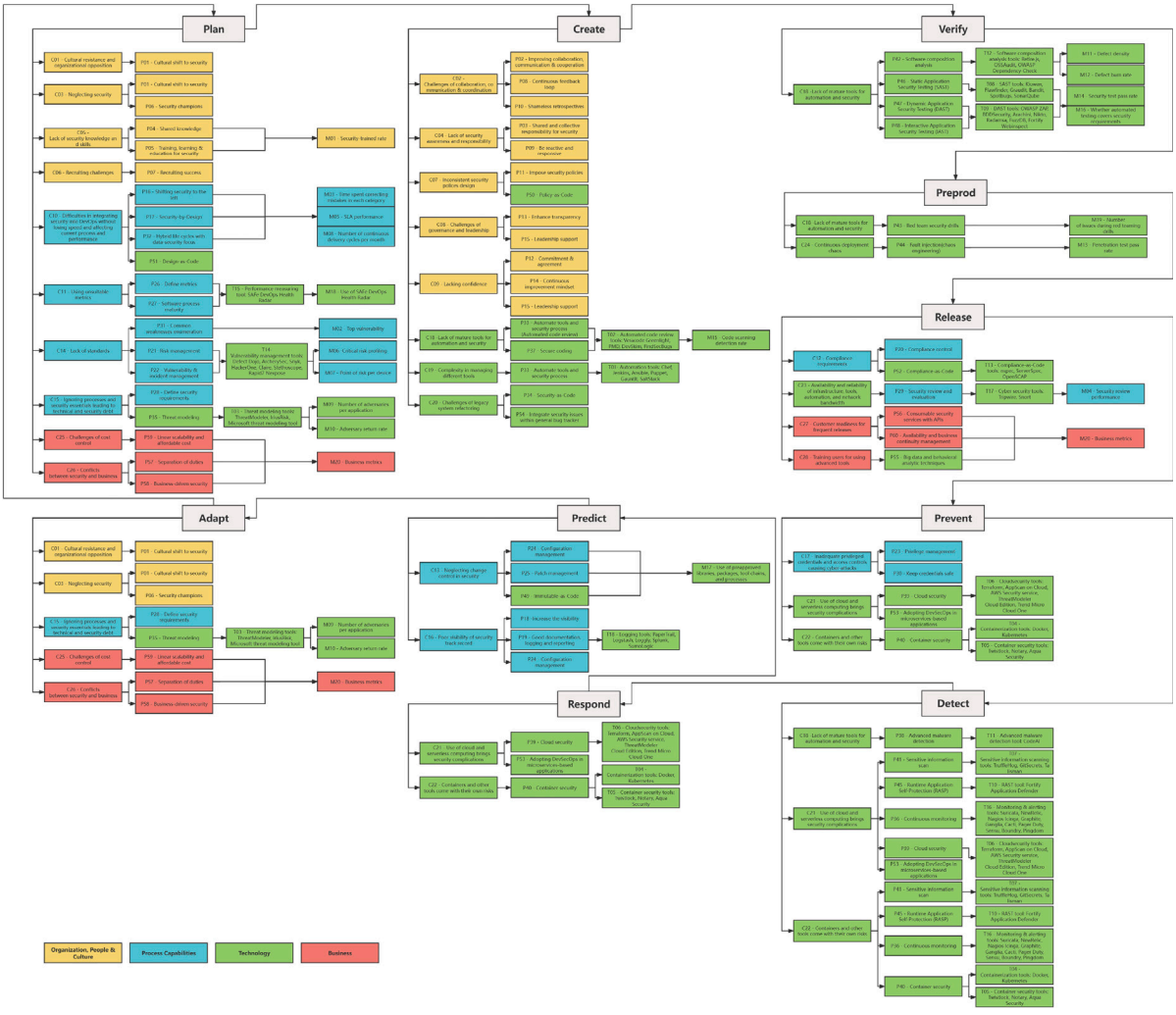

Fig. 5. Challenge-Practice-Tooling-Measurement (CPTM) model for DevSecOps.

the model appear to be cross-cutting themes across categories, and their categories potentially differ from those in our thematic analysis. For instance, all tools are grouped into Technology category, but some of them may appear on other categories in the model to match their corresponding practices. All items in the model were identified from our MLR findings, without artificial reconstructions. Thus, the model will be evaluated, upgraded and further validated in subsequent work.

The distribution of the four categories (OPC, PC, Technology, and Business) associated with ten DevSecOps steps can be analyzed, by viewing the four colors in the model. Fig. 6 shows that most challenges and practices in OPC (yellow) and PC (blue) categories are in the Plan and Create steps, and might be adapted (Fig. 9) and re-planned. This reveals that a number of challenges would occur in the beginning of the DevSecOps process, from Plan to Create steps. In which case, the corresponding practices, tools and metrics should be planned, created and adapted as early as possible. This exactly reveals the spirit of DevSecOps - shift security to the left. A definite plan and a thorough execution is the key to DevSecOps' success. Moreover, it can be seen from Fig. 7, several business-related challenges and practices (red) appear in the Release step, this reflects the fact that the business perspective is also important for releasing the product, not only planning and adapting, therefore the organizations who adopt DevSecOps should be concerned with their business performance at the beginning, the middle and the end of the lifecycle.

In comparison with the previous three categories, Figs. 7 and 8 depict technology-related challenges and practices (green) are mainly

Table 19
Thematic analysis on DevSecOps tools.

Themes/Functions (Frequency)	Codes/Tools [Papers contributed to the code]
T01-Automation tools (11)	Chef [S1-IEEE-07, S1-SC-12, 20, 26], Jenkins [S1-SC-12], Ansible [S1-SC-20, S1-GL-04], Puppet [S1-SC-20], Gauntlt [S1-IEEE-06], SaltStack [S1-SC-01, 20]
T02-Automated code review tools (4)	Veracode Greenlight [S1-SC-01], PMD [S1-GL-23], DevSkim [S1-GL-23], FindSecBugs [S1-GL-23]
T03-Threat modeling tools (2)	IriusRisk [S1-SC-01], Microsoft threat modeling tool [S1-IEEE-39]
T04-Containerization tools (22)	Docker [S1-SC-09, 18, 20, 29, 34, 42, 45, 48, S1-ACM-95, 99, S1-IEEE-31, 55, S1-GL-03, 10], Kubernetes [S1-ACM-52, 76, 89, S1-SC-20, 29, S1-IEEE-18, S1-GL-03, 10]
T05-Container security tools (3)	Twistlock [S1-GL-42], Notary [S1-GL-42], Aqua Security [S1-GL-42]
T06-Cloud security tools (7)	Terraform [S1-SC-12, 20, S1-IEEE-33], AppScan on Cloud [S1-GL-42], AWS Security service [S1-GL-42], ThreatModeler Cloud Edition [S1-GL-42], Trend Micro Cloud One [S1-GL-42]
T07-Sensitive information scanning tools (3)	TruffleHog [S1-GL-23], GitSecrets [S1-GL-23], Talisman [S1-GL-23]
T08-SAST tools (7)	Kiuwan [S1-SC-01], Flawfinder [S1-GL-23], Graudit [S1-GL-23], Bandit [S1-GL-23], Spotbugs [S1-GL-23], SonarQube [S1-GL-23, 42]
T09-DAST tools (7)	OWASP ZAP [S1-GL-23], BDD Security [S1-GL-23], Arachini [S1-GL-23], Nikto [S1-GL-23], Radamsa [S1-GL-23], FuzzDB [S1-GL-23], Fortify Webinspect [S1-GL-42]
T10-RAST tool (1)	Fortify Application Defender [S1-GL-42]
T11-Advanced malware detection tool (1)	CodeAI [S1-SC-01]
T12-Software composition analysis tools (3)	Retire.js [S1-GL-23], OSSAudit [S1-GL-23], OWASP Dependency-Check [S1-GL-23]
T13-Compliance-as-Code tools (3)	nspec [S1-GL-23], ServerSpec [S1-GL-23], OpenSCAP [S1-GL-23]
T14-Vulnerability management tools (8)	Defect Dojo [S1-GL-23], ArcherySec [S1-GL-23], Snyk [S1-GL-10, 21], HackerOne [S1-GL-21], Claire [S1-GL-21], Stethoscope [S1-GL-21], Rapid7 Nexpose [S1-GL-21]
T15-DevOps performance measuring tool (1)	SAFe DevOps Health Radar [S1-GL-01]
T16-Monitoring and alerting tools (2)*	Suricata [S1-GL-21], NewRelic [S1-GL-42] Nagios Icinga, Graphite, Ganglia, Cacti, Pager Duty, Sensu, Boundry, Pingdom [Mohan and Othmane's mapping study]
T17-Cyber security tools*	Tripwire, Snort [Mohan and Othmane's mapping study]
T18-Logging tools*	PaperTrail, Logstash, Loggly, Splunk, SumoLogic [Mohan and Othmane's mapping study]

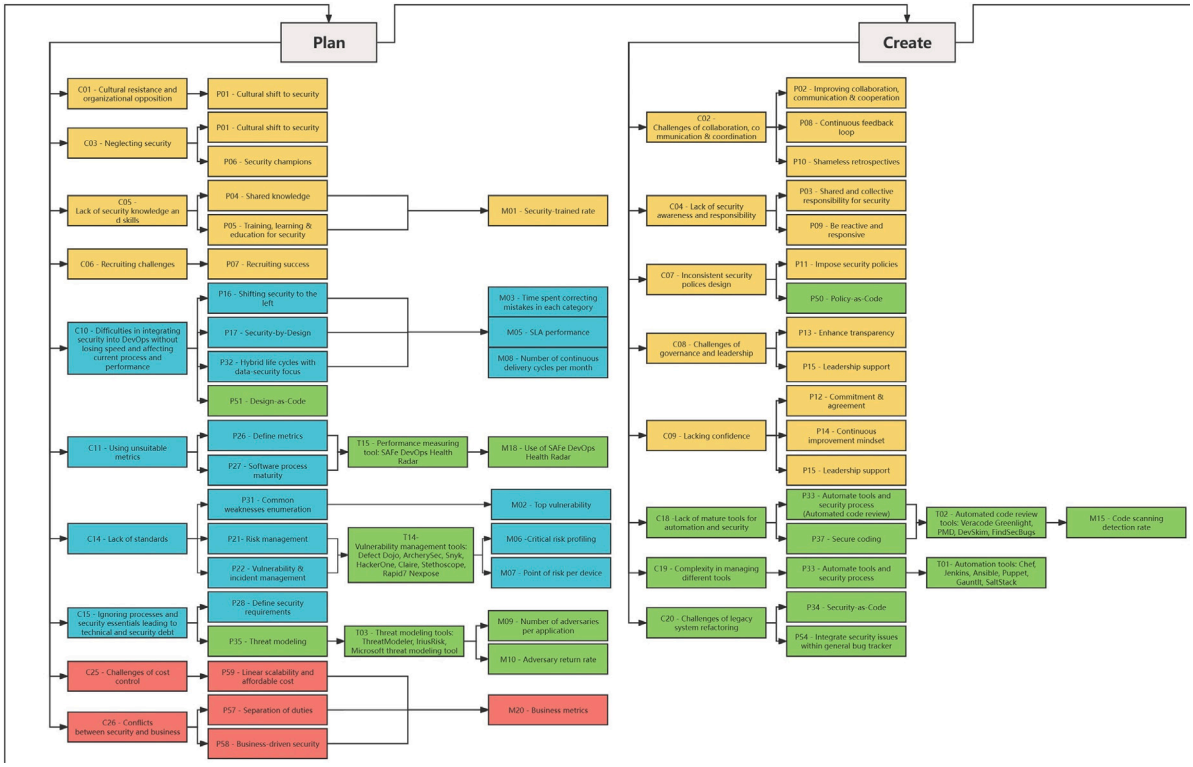

Fig. 6. CPTM model - Plan and create.

Fig. 7. CPTM model - Verify, preprod and release.

distributed in the steps of Verify, Preproduction, Prevent, Detect, Respond and Predict. This reveals the implementation of DevSecOps relies principally on technological enablers and tools. For instance, the most important technology-related challenge in Verify step (security testing) is "the lack of mature tools for automation and security", so that sets of practices and tools are identified, e.g., SCA (Software Composition Analysis), SAST (Static Application Security Testing), DAST (Dynamic Application Security Testing) and IAST (Interactive Application Security Testing). Another example is in the operation steps, i.e., Prevent, Detect, Respond and Predict, "the use of cloud and containers brings certain security complication", so that sets of practices and tools for cloud security and containers security can be selected. In addition, some other technological practices e.g., RASP (Runtime Application Self-Protection), Continuous Monitoring, Sensitive Information Scan, etc, are also adopted by the operation.

4.1.4. *Summary of the answer to RQ1*

In summary, based on the included white and grey literature, five aspects of the DevSecOps topic have been identified: Definitions, Challenges, Practices, Measurements/Metrics, and Technologies/Tools. A Thematic Analysis process was conducted to collect, analyze and report the related themes of each aspect, further four categories have been identified: OPC (Organization, People and Culture), PC (Process Capabilities), Technology, and Business. These categories and themes have in turn been mapped to the ten stages of a lifecycle model. On this

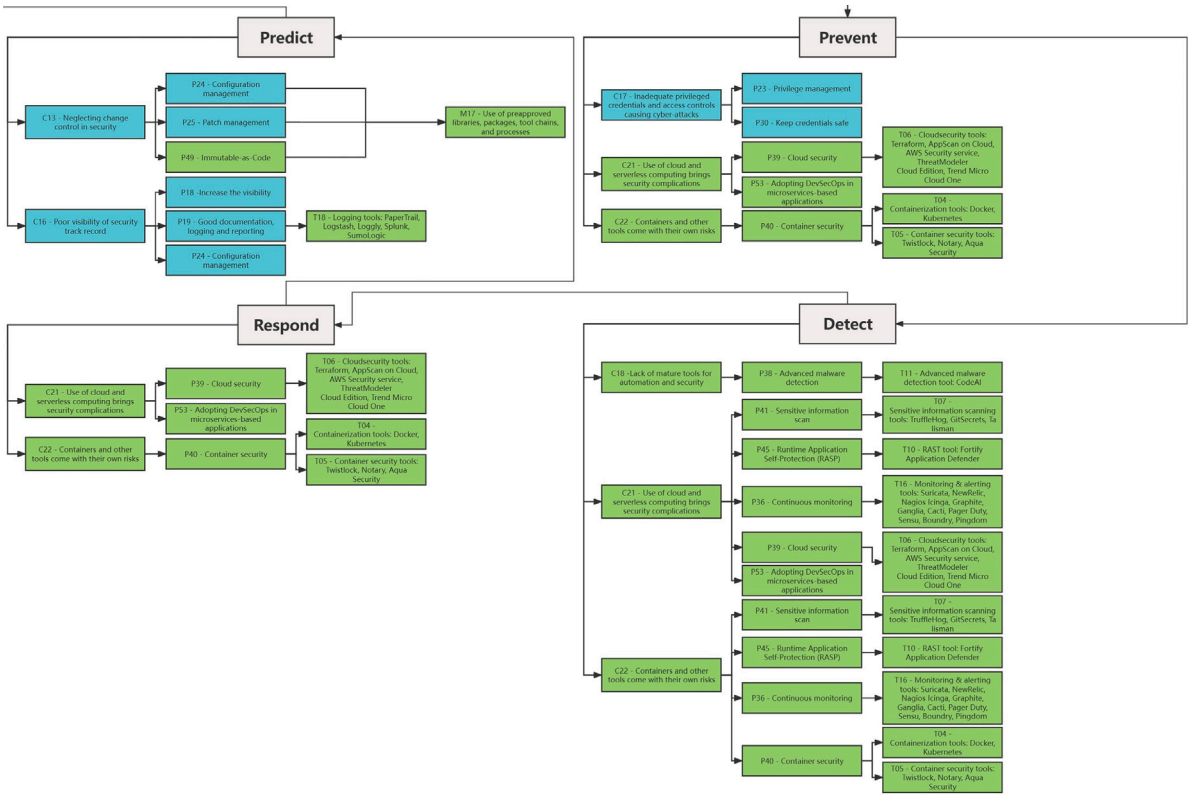

Fig. 8. CPTM model - Prevent, detect, respond and predict.

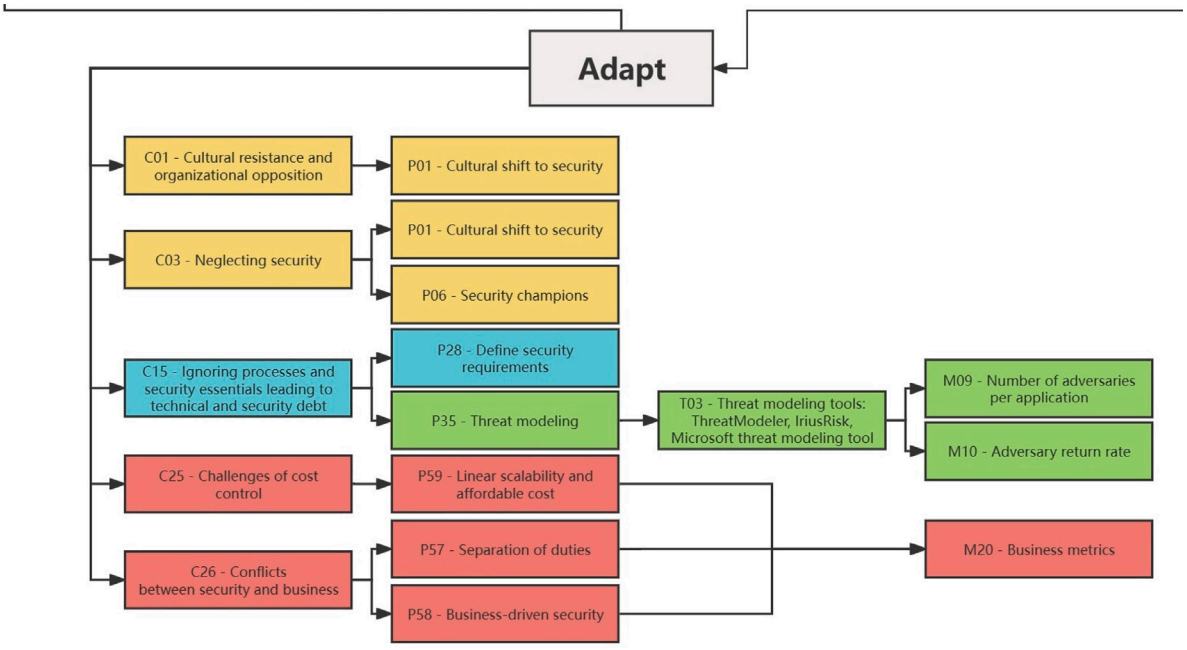

Fig. 9. CPTM model - Adapt.

basis, a Challenge-Practice-Tool-Metric (CPTM) model for DevSecOps is deduced, covering 28 challenges, 60 practices, 20 metrics, and sets of tools based on their functions. The CPTM model reveals the current state of DevSecOps in academia and industry, and captures the existing experience in this area. In comparing academia and industry (WL and GL), scholars have tended to contribute to research studies into the phenomenon, by defining concepts and identifying challenges and practices. By contrast, the practitioners from industry have contributed more to the business perspective and pragmatic implications of the DevSecOps approach, focusing on practical tools and metrics to provide solutions.

4.2. RQ2 - Adopting DevSecOps in GSE

After applying Search String 1 in all search sources, the results did not include any work involving the adoption of DevSecOps in GSE

Table 20
Steps of DevSecOps model by Gartner (MacDonald and Head, 2016).

Steps	Definitions
Plan	The step to set project objectives, identify security requirements, plan security measures, define metrics and policies, prepare organizations/teams, select technologies/tools, and develop budgets
Create	The step to start executing the plan, prepare security practices, and set up security tools
Verify	The step to conduct security practices by using appropriate (automated) tools and technologies, such as security tests (SAST, DAST, IAST) and software composition analysis (SCA)
Preproduction	The step to include further security tests, such as chaos engineering and red team drilling
Release	The step to sign the software and get it ready to be released and build it into the production environment, by reviewing configuration, infrastructure, network bandwidth, compliance, etc
Prevent	The step to protect the runtime environment architecture, such as cloud, containers, serverless, user access control, etc
Detect	The step to continuously monitor and scan the runtime environment architecture, such as runtime application self-protection, sensitive information scan, malware detection, etc
Respond	The step to address the vulnerabilities detected in the previous step
Predict	The step to analyze the vulnerabilities to identify the causes
Adapt	The step to improve security processes and re-plan the DevSecOps lifecycle, based on the lessons learned from the previous steps

contexts. To address RQ2, the additional Search String 2 was applied and resulted in 126 WL papers. After eliminating duplicates, 66 papers remained. However, most of them talk about global DevOps, without involving the security aspect. After study selection and QA, only 2 papers (both from ACM) involving GSE, DevOps and security were finally included. The search results held even when Search String 2 had been adjusted numerous times such as by trying other additional keywords, e.g., 'multi-site', 'multi-nation', 'transnational', etc.

4.2.1. Lack of global dimension in WL

Paper S2-ACM-04 (Gupta et al., 2019) presents an empirical study on a global software project of the development team distributed geographically across India, the USA and Germany, that successfully adopted DevOps. Authors identified challenges and practices when adopting global DevOps. Comparing their challenges and practices to our CPTM model, we identified several matched challenges (e.g. DevOps team setting-up; people's mindset; continuous requirements; and new architecture, tools and technologies) and practices (e.g. "shift right-move left" expectation; setting environment where experimentation and failure are safe, recommend early and fast failure; employing automation and cloud to achieve everything-as-code).

Paper S2-ACM-05 (Viggiato et al., 2019) presents an exploratory and inductive research study to identify similarities and differences of GSE practices from different domains. Some of the findings are related to DevOps and security: (a) Continuous integration is not always a homogeneous GSE practice in some domains, e.g. banking and e-commerce domains usually interrupt continuous integration during critical commerce periods (like Black Friday), aiming at avoiding inserting bugs in systems; (b) the E-commerce domain particularly values UX and non-functional requirements (performance, usability, security), so they are accustomed to conduct extensive tests instead of frequent continuous delivery; (c) Healthcare domains give high priority to reliability, privacy and security, so more security practices are needed; (d) Social network domain typically has no dedicated test team, tests are conducted by developers, relying on modern architectures, such as micro-services. These findings provide a practical guide on selecting appropriate domains for further research on Global DevSecOps.

4.2.2. Absence of global dimension in GL

Search String 2 and its variants were also applied on Google to search GL. After browsing the first 10 pages (100 results), no GL work involving the three terms (GSE, DevOps, and security) had been found. This reveals that the existing GL does not provide any practical experience on Global DevSecOps.

4.2.3. Summary of the answer to RQ2

In summary, the results report a notable absence of the global dimension in the white and grey literature. To our knowledge, we find that most of the existing literature only covers two of the three terms (DevOps, security, and GSE) simultaneously: some papers focus on the adoption of DevOps in GSE, excluding security; others cover DevOps and security (DevSecOps), excluding GSE. There are four possibilities for the results in our analysis. The first one is that no significant correlation exists between GSE and DevSecOps, namely, there are no distinguishing characteristics of DevSecOps whether it is adopted in a local or in a global setting. Or it may be that security is typically a centralized and control-oriented function in organizations, so global aspects are not prominent. The third possibility is that there is a research

Table 21
Identified themes mapped to steps by Gartner (MacDonald and Head, 2016).

Steps	Challenges	Practices	Tools	Metrics	Steps	Challenges	Practices	Tools	Metrics
Plan	C01	P01	NA	NA	Release	C12	P20, P52	T13	NA
	C03	P01, P06	NA	NA		C23	P29	T17	M04
	C05	P04, P05	NA	M01		C27	P56, P60	NA	M20
	C06	P07	NA	NA		C28	P55	NA	M20
	C10	P16, P17, P32, P51	NA	M03, M05, M08					
	C11	P26, P27	T15	M18	Prevent	C17	P23, P30	NA	NA
	C14	P21, P22, P31	T14	M02, M06, M07		C21	P39, P53	T06	NA
	C15	P28, P35	T03	M09, M10		C22	P40	T04, T05	NA
	C25	P59	NA	M20	Detect	C18	P38	T11	NA
	C26	P57, P58	NA	M20		C21	P36, P39, P41, P45, P53	T06, T07, T10, T16	NA
Create	C02	P02, P08, P10	NA	NA		C22	P36, P40, P41, P45	T04, T05, T07, T10, T16	NA
	C04	P03, P09	NA	NA	Respond	C21	P39, P53	T06	NA
	C07	P11, P50	NA	NA		C22	P40	T04, T05	NA
	C08	P13, P15	NA	NA	Predict	C13	P24, P25, P49	NA	M17
	C09	P12, P14, P15	NA	NA		C16	P18, P19, P24	T18	NA
	C18	P33, P37	T02	M15	Adapt	C01	P01	NA	NA
	C19	P33	T01	NA		C03	P01, P06	NA	NA
	C20	P34, P54	NA	NA		C15	P28, P35	T03	M09, M10
Verify	C18	P42, P46, P47, P48	T08, T09, T12	M11, M12, M14, M16		C25	P59	NA	M20
Preprod	C18	P43	NA	M19		C26	P57, P58	NA	M20
	C24	P44	NA	M13					

gap in this area. The fourth possibility, also maybe a limitation, is that some terminologies were missed in determining our search string, though we have revised our search strings to verify this negative result. To prove the above possibilities, further work is needed to seek more concrete proof from academic and industrial sources.

4.3. Confirmatory search after MLR

To avoid staleness and to continue validation for the MLR findings, we conducted a continuous confirmatory process of what we have termed 'Confirmatory Search' after the MLR to find the latest literature. By 2022, 13 academic papers and 7 grey articles have been newly included (Appendix A.3). The new papers and articles which were collected from the confirmatory search were not taken into the thematic analysis, and were not integrated in the final CPTM model, because the confirmatory search was conducted after the MLR and TA processes, in order to find the latest literature while not affecting the original MLR results.

From newly included academic papers, we find that the main research aspects of DevSecOps are still its definition, challenges, practices, and tools, involving a few new contributions to metrics/measurement, e.g., Brasoveanu et al. (2022) and Nisha (2022). We have reported in Table 4. For the Global aspect of DevSecOps, paper CS-ACM-04 (Liu et al., 2021) designed a DevOps architecture scheme for the cross-network and multiple environment CoSE. It revealed traditional large-scale enterprises in China faced two challenges when adopting DevOps in high security environment: (1) physical isolation of multiple environments (development, test and production environment); (2) cross regional collaboration of teams which have complex compositions, large number of employees, but very limited resources. These are also the challenges of GSE.

As mentioned in Section 2, two recently published review papers are similar to ours. CS-SC-01 (Akbar et al., 2022) conducted an MLR, which revealed 18 DevSecOps challenges (all challenges can match or partly match our findings) and grouped them into 10 categories. CS-SC-03 (Rajapakse et al., 2022) conducted an SLR and also applied TA, identified 21 challenges (all can match our findings) and 31 solutions (29 can match our identified practices) of DevSecOps, and classified their findings into four categories: People, Practices, Tools, and Infrastructure.

From the new GL work, we find that the new material is only novel by date, not data. Most GL articles slip into a routine and repeat similar stories. They introduce repetitive contents in a conventional form, which consists of a common definition, and sets of challenges, practices and tools. This reflects that practitioners' perspectives on DevSecOps have been converging, since the DevSecOps pattern has shaped up. In contrast, although there are no apparent new findings/themes identified from the new WL work, some recent publications make new contributions to framework design for DevSecOps, by using the known findings. For example, 7 of the 13 new WL papers proposed their new models or frameworks. This reveals that DevSecOps research has been towards the next stage. Scholars summarize the first decade of DevSecOps development (2012–2022), so that the research trend is moving to framework design on this topic.

4.4. Study implications

We believe that this study could provide some valuable contributions for both researchers and practitioners working in the area of DevSecOps.

For researchers, the study provides a systematic state-of-the-art overview of DevSecOps in the past decade, by executing a dual-track strategy including white and grey literature. Firstly, the paper identifies five main aspects of DevSecOps studies in the existing literature, namely, Definition, Challenges, Practices, Tools/Technologies, and Metrics/Measurement. Of these, challenges and practices seem to be of most concern for researchers. So we believe that this finding could help researchers to see a body of knowledge and choose research directions in this area. Secondly, the study provides a framework covering identified challenges, practices, tools and metrics within the DevSecOps lifecycle, so that researchers could learn about the detailed implementation and the existing experience of DevSecOps process, Thirdly, the model will enable researchers to select areas of focus, themes and stages in the lifecycle which have not been well covered for further investigation. Fourthly, the study reveals that there is extremely limited literature related to adopting DevSecOps in GSE contexts, thereby helping researchers to avoid unnecessary work in this direction, or in contrast, offering a potential research gap.

In addition to researchers, the study could provide knowledge and experience for the practitioners who adopt the DevSecOps paradigm. For example, practitioners could refer to the CPTM model as a road map during the execution of DevSecOps projects, as it depicts what practices can be adopted to address corresponding challenges; what tools can be selected to use; and what metrics can be applied to measure the performance. The model also covers various categories (i.e., OPC, PC, Technology and Business), so that it could guide DevSecOps teams to consider work items for different roles and from different perspectives and identify areas of weakness that could benefit from increased attention.

Furthermore, from this study, we find that researchers and practitioners have different emphases and strengths in this area, and they are complementary. According to recent literature, researchers have summarized the first decade of DevSecOps development and have been striving to develop frameworks for DevSecOps, highlighting major challenges, crucial practices, relevant tools and their links. Meanwhile, industrial organizations are committed to the development and application of more pertinent tools and metrics which can be adopted by DevSecOps teams. Thus, we are looking forward to strengthening the cooperation between academia and industry, to achieve the unity of DevSecOps work from a variety of perspectives.

However, there remain some open areas for the CPTM model to incorporate and developing trends for further consideration. While the model addresses the full software development lifecycle from plan to operate and refine, that is inherently based on the concept of a project, and does not directly address the layered dimensions of the enterprise and distributed organization (GSE), with portfolio and program dimensions augmenting that of the project (Antil, 2023; Beecham et al., 2021; Lal and Clear, 2021). A further area warranting attention for cybersecurity professionals is the rapidly developing set of developments in Artificial Intelligence (Chakrabarty et al., 2023) and their implications for security.

5. Threats to validity

This MLR faces several potential threats to validity, including study selection bias, quality assessment subjectivity, data extraction bias, trustworthiness of synthesis, and construction of search string.

5.1. Bias of study selection, quality assessment, and data extraction

Study inclusion/exclusion bias, quality assessment subjectivity and data extraction bias are the common threats to validity in SE secondary studies (Ampatzoglou et al., 2019). Particularly in this study, the first author drove the tasks of paper collection, without an additional execution by the other researchers. To mitigate the threats, we clearly defined inclusion/exclusion criteria, study quality assessment form and data extraction form when we developed the review protocol, discussed among authors, and updated over the research timeline. Nonetheless, while we believe that the results are representative, it cannot be certain that all literature has been included and all useful data has been covered. For this reason, we decided to apply snowballing on previous MLR/SLR papers, to further compare and validate our findings by referring to these papers.

5.2. Trustworthiness of synthesis

As mentioned in the research methods section, the tasks of coding and theming were mainly completed by the first author, and the output was reviewed and evaluated in consultation with the second and third authors by weekly or bi-weekly meetings. Although using reflexive TA does not demand to measure agreement (Braun and Clarke, 2021), the trustworthiness of synthesis is still a threat needed to be assessed. Inevitable biases which emerged from researchers' subconscious preferences could affect the trustworthiness of synthesis. In this case, researchers had certain preconceived notions in this topic, e.g., the identified elements of DevOps/DevSecOps (Capabilities, Cultural Enablers, and Technological Enablers) (Smeds et al., 2015) and the CAMS (Culture, Automation, Measurement, and Sharing) model (Humble and Molesky, 2011), that might have influenced the ways of coding and theming. Also, the four elements of the CPTM model (Challenge, Practices, Tool, and Metrics) might be identified by the influence of existing review papers or our preconceived notions of DevSecOps. To ensure the trustworthiness of synthesis, TA tasks were reviewed by leveraging Braun's checklist (Braun and Clarke, 2021); and the four components of trustworthiness, i.e., credibility, confirmability, dependability and transferability (Cruzes and Dyba, 2011a) have also been carefully assessed (Section 3.8.3). It is important to distinguish between bias and subjectivity, especially in this research, when performing a Reflexive Thematic Analysis, which is based on relativist ontology and subjective epistemology. Subjectivity which was caused by researchers' knowledge, experiences, roles and backgrounds, could affect data collection, analysis and interpretation. However, it should be considered as a strength for knowledge production, rather than a threat to credibility (Braun and Clarke, 2021).

5.3. Construction of search string

Inappropriate construction of search string might return redundant or lacking search results (Ampatzoglou et al., 2019). This MLR found extremely limited primary studies related to the adoption of DevSecOps in global settings. Some terminologies specific to GSE and DevSecOps were possibly missed in determining our search string. Although we had modified the search string multiple times with additional keywords and employed the snowballing technique, the results did not change. Thus, we could safely draw a negative conclusion that there is an absence of the global dimension of DevSecOps in the existing white and grey literature.

6. Conclusion and future work

This paper reviews the existing white and grey literature in the area of DevSecOps and its adoption in the GSE contexts. The study identifies five major aspects of DevSecOps (Definitions, Challenges, Practices, Tools/Technologies, Metrics/ Measurement); collects the related themes of each aspect by performing a Thematic Analysis (TA) process; and builds a **Challenge-Practice-Tool-Metric (CPTM)** model by integrating the included themes of the latter four aspects, within a staged lifecycle model. Moreover, the paper explores the adoption of DevSecOps in global settings, from the existing white and grey literature, so that it identifies the missing global dimension of DevSecOps and analyzes relevant reasons.

In the future work, we intend to conduct an empirical investigation to verify the utility of the CPTM model, and thus to improve and refine it. Hence, we are conducting a Delphi study on the identified Challenges, Practices, Tools and Metrics of DevSecOps, gathering opinions and comments from both academic and industrial experts, to determine the degree of emphasis and priority allocated to specific aspects by using the Analytic Hierarchy Process (AHP) (Brunelli, 2014). Another potential further research direction may be conducting a field study with a global software vendor with the cloud deployment of its software products, to validate the efficacy of the CPTM model in a global setting.

Furthermore, we notice that a State of DevOps Report 2023 presented by Puppet (2023) highlights a recent industry trend - many organizations do not use the term DevOps anymore, since they have internalized all its lessons. The report also suggests a clear pattern emerged that mature DevOps organizations tend to instead use Platform Engineering, which is "*the discipline of designing and building self-service capabilities to minimize cognitive load for developers and to enable fast flow software delivery*". Platform teams provide shared infrastructure platforms to internal users, i.e., software developers and engineers; and they continuously develop, build, maintain and support underlying infrastructures, aiming to provide self-service solutions, which enable development teams to deliver fast, and ensure consistency for the rest of the organization (Puppet, 2023). Thus, a potential future direction may be deduced to extend the topic of DevSecOps by including security in Platform Engineering, if the research trend also begins to evolve from DevOps to Platform Engineering. A further area warranting attention for cybersecurity professionals is the rapidly developing set of developments in Artificial Intelligence (Chakrabarty et al., 2023) and their implications for security.

In conclusion, the **Challenge-Practice-Tool-Metric (CPTM)** model we have presented here provides a breakdown and a broad landscape of DevSecOps, from which researchers and practitioners may select an area of focus to improve their knowledge or practice. With DevSecOps spanning the many stages of the lifecycle, we believe the model will enable emphases and absences such as global aspects to be investigated.

CRediT authorship contribution statement

Xiaofan Zhao: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tony Clear:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Ramesh Lal:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Associated materials are available in an open repository at zenodo.org https://doi.org/10.5281/zenodo.7959584, including:

- MLR protocol
- List of included papers along with quality assessment score
- Raw data/text and codes (definitions, challenges and practices)
- Thematic synthesis for white and grey literature
- Thematic analysis tables (first edition)
- Thematic analysis tables (completed edition)
- CPTM model (full version).

Appendix. List of included papers

A.1. White literature papers

S1-ACM-01: M.G. Jaatun, Software security activities that support incident management in secure DevOps, Proceedings of the 13th International Conference on Availability, Reliability and Security. (2018). doi:10.1145/3230833.3233275.

S1-ACM-02: D. Ashenden, G. Ollis, Putting the SEC in DevSecOps: Using social practice theory to improve secure software development, New Security Paradigms Workshop 2020. (2020). doi:10.1145/3442167.3442178.

S1-ACM-03: M.G. Jaatun, D.S. Cruzes, J. Luna, DevOps for better software security in the cloud invited paper, Proceedings of the 12th International Conference on Availability, Reliability and Security. (2017). doi:10.1145/3098954.3103172.

S1-ACM-04: S.B. Carturan, D.H. Goya, A systems-of-systems security framework for requirements definition in cloud environment, Proceedings of the 13th European Conference on Software Architecture. (2019). doi:10.1145/3344948.3344977.

S1-ACM-05: S. Rafi, W. Yu, M.A. Akbar, Towards a hypothetical framework to secure devops adoption, Proceedings of the Evaluation and Assessment in Software Engineering. (2020). doi:10.1145/3383219.3383285.

S1-ACM-06: A. Rahman, M.R. Rahman, C. Parnin, L. Williams, Security smells in Ansible and Chef Scripts, ACM Transactions on Software Engineering and Methodology. 30 (2021). doi:10.1145/3408897.

S1-ACM-07: J.A. Morales, H. Yasar, A. Volkman, Implementing devops practices in highly regulated environments, Proceedings of the 19th International Conference on Agile Software Development: Companion. (2018). doi:10.1145/3234152.3234188.

S1-ACM-08: M. Anisetti, C.A. Ardagna, F. Gaudenzi, E. Damiani, A continuous certification methodology for DevOps, Proceedings of the 11th International Conference on Management of Digital EcoSystems. (2019). doi:10.1145/3297662.3365827.

S1-ACM-09: J. A. Morales, T. P. Scanlon, A. Volkmann, J. Yankel, H. Yasar, Security impacts of sub-optimal devsecops implementations in a highly regulated environment, Proceedings of the 15th International Conference on Availability, Reliability and Security. (2020). doi:10.1145/3407023.3409186

S1-ACM-15: E. Di Nitto, P. Jamshidi, M. Guerriero, I. Spais, D.A. Tamburri, A software architecture framework for quality-aware DevOps, Proceedings of the 2nd International Workshop on Quality-Aware DevOps. (2016). doi:10.1145/2945408.2945411.

S1-ACM-45: T. Lopez, H. Sharp, T. Tun, A. Bandara, M. Levine, B. Nuseibeh, "hopefully we are mostly secure": Views on Secure Code in professional practice, 2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering. (2019). doi:10.1109/chase.2019.00023.

S1-ACM-49: S. Vost, S. Wagner, Keeping continuous deliveries safe, 2017 IEEE/ACM 39th International Conference on Software Engineering Companion. (2017). doi:10.1109/icse-c.2017.135.

S1-ACM-50: J. Nguyen, M. Dupuis, Closing the feedback loop between UX design, software development, security engineering, and Operations, Proceedings of the 20th Annual SIG Conference on Information Technology Education. (2019).doi:10.1145/3349266.3351420.

S1-ACM-52: G.P. Fernandez, A. Brito, Secure container orchestration in the cloud, Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. (2019). doi:10.1145/3297280.3297296.

S1-ACM-59: E. Rios, E. Iturbe, M.C. Palacios, Self-healing multicloud application modelling, Proceedings of the 12th International Conference on Availability, Reliability and Security. (2017). doi:10.1145/3098954.3104059.

S1-ACM-64: K. Rindell, S. Hyrynsalmi, V. Leppänen, Aligning security objectives with Agile Software Development, Proceedings of the 19th International Conference on Agile Software Development: Companion. (2018). doi:10.1145/3234152.3234187.

S1-ACM-66: K.A. Torkura, M.I.H. Sukmana, C. Meinel, Integrating Continuous Security Assessments in microservices and cloud native applications, Proceedings of the 10th International Conference on Utility and Cloud Computing.(2017).doi:10.1145/3147213.3147229.

S1-ACM-68: Y. Rouf, J. Mukherjee, M. Fokaefs, M. Shtren, J. Le, M. Litoiu. Rule-based security management system for data-intensive applications, Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, IBM Corp, USA, 254–263. (2019).

S1-ACM-69: K. Tuma, D. Hosseini, K. Malamas, R. Scandariato, Inspection guidelines to identify security design flaws, Proceedings of the 13th European Conference on Software Architecture. (2019). doi:10.1145/3344948.3344995.

S1-ACM-71: M. Miglierina, D.A. Tamburri, Towards Omnia, Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion. (2017). doi:10.1145/3053600.3053629.

S1-ACM-72: J. Winter, M. Aniche, J. Cito, A.van Deursen, Monitoring-aware IDES, Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. (2019).
doi:10.1145/3338906.3338926.

S1-ACM-76: L. F. Rivera, N. M. Villegas, G. Tamura, M. Jiménez, H. A. Müller. UML-Driven Automated Software Deployment, Proceedings of 28th Annual International Conference on Computer Science and Software Engineering, (2018). doi: 10.475/123-4

S1-ACM-81: A. Wiedemann, N. Forsgren, M. Wiesche, H. Gewald, H. Krcmar, Research for practice, Communications of the ACM. 62 (2019). doi:10.1145/3331138.

S1-ACM-89: E. Yuan, Architecture interoperability and repeatability with microservices: An industry perspective, 2019 IEEE/ACM 2nd International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering. (2019). doi:10.1109/ecase.2019.00013.

S1-ACM-95: M. Hilton, N. Nelson, T. Tunnell, D. Marinov, D. Dig, Trade-offs in Continuous Integration: Assurance, security, and flexibility, Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. (2017). doi:10.1145/3106237.3106270.

S1-ACM-99: Z. Sampedro, A. Holt, T. Hauser, Continuous integration and delivery for HPC, Proceedings of the Practice and Experience on Advanced Research Computing. (2018). doi:10.1145/3219104.3219147.

S1-IEEE-02: A. Valani, Rethinking secure devops threat modeling: The need for a dual velocity approach, 2018 IEEE Cybersecurity Development (SecDev). (2018). doi:10.1109/secdev.2018.00032.

S1-IEEE-03: K. Zunnurhain, S.R. Duclervil, A new project management tool based on devsecops, 2019 International Conference on Computational Science and Computational Intelligence. (2019). doi:10.1109/csci49370.2019.00049.

S1-IEEE-04: C. Fayollas, H. Bonnin and O. Flebus, SafeOps: A Concept of Continuous Safety, 2020 16th European Dependable Computing Conference. (2020). doi: 10.1109/EDCC51268.2020.00020.

S1-IEEE-05: Z. Ahmed, S. C. Francis, Integrating security with devsecops: Techniques and challenges, Proceedings of the 2019 International Conference on Digitization. (2019). doi:10.1109/icd47981.2019.9105789.

S1-IEEE-06: N. Tomas, J. Li, H. Huang, An empirical study on culture, automation, measurement, and sharing of devsecops, Proceedings of 2019 International Conference on Cyber Security and Protection of Digital Services. (2019).
doi:10.1109/cybersecpods.2019.8884935.

S1-IEEE-07: M. Z. Abrahams, J. J. Langerman, Compliance at Velocity within a DevOps Environment, 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany. (2018) doi:10.1109/ICDIM.2018.8847007.

S1-IEEE-08: S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, A. Gumaei, Prioritization based taxonomy of devops security challenges using promethee, IEEE Access 8 (2020). doi:10.1109/ACCESS.2020.2998819.

S1-IEEE-09: L. Williams, Continuously integrating security, Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment. (2018). doi:10.1145/3194707.3194717.

S1-IEEE-10: T. Rangnau, R.v. Buijtenen, F. Fransen, F. Turkmen, Continuous Security Testing: A case study on Integrating Dynamic

Security Testing Tools in CI/CD pipelines, 2020 IEEE 24th International Enterprise Distributed Object Computing Conference. (2020). doi:10.1109/edoc49727.2020.00026.

S1-IEEE-11: J.R. Michener, A.T. Clager, Mitigating an oxymoron: Compliance in a DevOps Environments, 2016 IEEE 40th Annual Computer Software and Applications Conference. (2016). doi:10.1109/compsac.2016.155.

S1-IEEE-12: A. A. U. Rahman, L. Williams, Software security in devops: Synthesizing practitioners' perceptions and practices, Proceedings of the International Workshop on Continuous Software Evolution and Delivery, ACM, New York, NY, USA, 2016, pp. 70–76. doi:10.1145/2896941.2896946.

S1-IEEE-13: T.F. Düllmann, C. Paule, A. van Hoorn, Exploiting devops practices for dependable and secure continuous delivery pipelines, Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering.(2018).doi:10.1145/3194760.3194763.

S1-IEEE-15: V. Mohan, L. ben Othmane, A. Kres, BP: Security Concerns and best practices for automation of software deployment processes: An industrial case study, 2018 IEEE Cybersecurity Development (SecDev). (2018). doi:10.1109/secdev.2018.00011.

S1-IEEE-16: E. Rios, E. Iturbe, W. Mallouli, M. Rak, Dynamic Security Assurance in multi-cloud DevOps, 2017 IEEE Conference on Communications and Network Security (CNS). (2017). doi:10.1109/cns.2017.8228701.

S1-IEEE-17: A. Avritzer, Challenges and approaches for the assessment of Micro-Service Architecture Deployment Alternatives in devops : A tutorial presented at ICSA 2020, 2020 IEEE International Conference on Software Architecture Companion. (2020). doi:10.1109/icsa-c50368.2020.00007.

S1-IEEE-18: M.S. Islam Shamim, F. Ahamed Bhuiyan, A. Rahman, Xi commandments of Kubernetes Security: A systematization of knowledge related to Kubernetes Security Practices, 2020 IEEE Secure Development (2020). doi:10.1109/secdev45635.2020.00025.

S1-IEEE-20: A. Rahman, Characteristics of defective infrastructure as code scripts in DevOps, Proceedings of the 40th International Conference on Software Engineering. (2018). doi:10.1145/3183440.3183452.

S1-IEEE-21: S. Carturan, D. Goya, Major challenges of systems-of-systems with cloud and devops – a financial experience report, 2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES). (2019). doi:10.1109/sesos/wdes.2019.00010.

S1-IEEE-22: E.C. Burkard, Usability testing within a Devsecops environment, 2020 Integrated Communications Navigation and Surveillance Conference. (2020). doi:10.1109/icns50378.2020.9222919.

S1-IEEE-24: Francois raynaud on devsecops, IEEE Software 34 (5) (2017) 93–96. doi:10.1109/ms.2017.3571578.

S1-IEEE-25: M.H. Syed, E.B. Fernandez, Cloud ecosystems support for internet of things and devops using patterns, 2016 IEEE First International Conference on Internet-of-Things Design and Implementation. (2016). doi:10.1109/iotdi.2015.31.

S1-IEEE-26: J. Diaz, J.E. Perez, M.A. Lopez-Pena, G.A. Mena, A. Yague, Self-service cybersecurity monitoring as enabler for devsecops, IEEE Access. 7 (2019) 100283–100295. doi:10.1109/access.2019.2930000.

S1-IEEE-28: A. Rahman, C. Parnin, L. Williams, The Seven sins: Security smells in infrastructure as code scripts, 2019 IEEE/ACM 41st International Conference on Software Engineering. (2019). doi:10.1109/icse.2019.00033.

S1-IEEE-29: P. Frijns, R. Bierwolf, T. Zijderhand, Reframing security in Contemporary Software Development Life cycle, 2018 IEEE International Conference on Technology Management, Operations and Decisions. (2018). doi:10.1109/itmc.2018.8691277.

S1-IEEE-30: L. Sion, K. Tuma, R. Scandariato, K. Yskout, W. Joosen, Towards Automated Security Design Flaw Detection, 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop. (2019). doi:10.1109/asew.2019.00028.

S1-IEEE-31: Amith Raj MP, A. Kumar, S.J. Pai, A. Gopal, Enhancing security of Docker using linux hardening techniques, 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology(2016).doi:10.1109/icatcct.2016.7911971.

S1-IEEE-33: R. Rompicharla, B.R. P. V, Continuous compliance model for hybrid multi-cloud through self-service orchestrator, 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics. (2020).doi:10.1109/icstcee49637.2020.9276897.

S1-IEEE-34: N. Ferry, P.H. Nguyen, Towards model-based continuous deployment of secure IOT Systems, 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion. (2019). doi:10.1109/models-c.2019.00093.

S1-IEEE-36: S. Faily, C. Iacob, Design as code: Facilitating collaboration between usability and Security Engineers using Cairis, 2017 IEEE 25th International Requirements Engineering Conference Workshops. (2017). doi:10.1109/rew.2017.23.

S1-IEEE-38: B.S. Farroha, D.L. Farroha, A framework for managing mission needs, compliance, and trust in the devops environment, 2014 IEEE Military Communications Conference. (2014). doi:10.1109/milcom.2014.54.

S1-IEEE-39: M.G. Jaatun, Architectural risk analysis in agile development of cloud software, 2019 IEEE International Conference on Cloud Computing Technology and Science. (2019). doi:10.1109/cloudcom.2019.00050.

S1-IEEE-40: V. Casola, A. De Benedictis, M. Rak, U. Villano, E. Rios, A. Rego, et al. Musa deployer: Deployment of Multi-cloud applications, 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises. (2017). doi:10.1109/wetice.2017.46.

S1-IEEE-41: Y. Wang, M. Pyhajarvi, M.V. Mantyla, Test Automation Process Improvement in a DevOps Team: Experience Report, 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops. (2020).
doi:10.1109/icstw50294.2020.00057.

S1-IEEE-42: Tran Quang Thanh, S. Covaci, T. Magedanz, P. Gouvas, A. Zafeiropoulos, Embedding security and privacy into the development and operation of cloud applications and services, 2016 17th International Telecommunications Network Strategy and Planning Symposium. (2016). doi:10.1109/netwks.2016.7751149.

S1-IEEE-43: J. McZara, S. Kafle, D. Shin, Modeling and analysis of dependencies between microservices in devsecops, 2020 IEEE International Conference on Smart Cloud. (2020). doi:10.1109/smartcloud49737.2020.00034.

S1-IEEE-44: C. Izurieta, M. Prouty, Leveraging secdevops to tackle the technical debt associated with cybersecurity attack tactics, 2019 IEEE/ACM International Conference on Technical Debt. (2019). doi:10.1109/techdebt.2019.00012.

S1-IEEE-52: T. Soenen, S. Van Rossem, W. Tavernier, F. Vicens, D. Valocchi, P. Trakadas, et al. Insights from Sonata: Implementing and integrating a microservice-based NFV service platform with a DevOps methodology, 2018 IEEE/IFIP Network Operations and Management Symposium. (2018). doi:10.1109/noms.2018.8406139.

S1-IEEE-54: M. Johnson, D. Cummings, B. Leinwand, C. Elsberry, Continuous testing and deployment for Urban Air Mobility, 2020 AIAA/IEEE 39th Digital Avionics Systems Conference. (2020). doi:10.1109/dasc50938.2020.9256435.

S1-IEEE-55: A.J. Younge, K. Pedretti, R.E. Grant, R. Brightwell, A tale of two systems: Using containers to deploy HPC applications on supercomputers and clouds, 2017 IEEE International Conference on Cloud Computing Technology and Science. (2017). doi:10.1109/cloudcom.2017.40.

S1-IEEE-57: T.J. Wagner, T.C. Ford, Metrics to meet Security and Privacy Requirements with Agile Software Development Methods in a

regulated environment, 2020 International Conference on Computing, Networking and Communications. (2020).
doi:10.1109/icnc47757.2020.9049681.

S1-IEEE-61: L. Sion, D.V. Landuyt, W. Joosen, The never-ending story: On the need for Continuous Privacy Impact Assessment, 2020 IEEE European Symposium on Security and Privacy Workshops. (2020). doi:10.1109/eurospw51379.2020.00049.

S1-IEEE-67: D. Preuveneers, W. Joosen, Towards multi-party policy-based access control in federations of cloud and edge microservices, 2019 IEEE European Symposium on Security and Privacy Workshops. (2019). doi:10.1109/eurospw.2019.00010.

S1-IEEE-71: C. Paule, T.F. Dullmann, A. Van Hoorn, Vulnerabilities in continuous delivery pipelines? A case study, 2019 IEEE International Conference on Software Architecture Companion. (2019). doi:10.1109/icsa-c.2019.00026.

S1-IEEE-84: J. Bogner, J. Fritzsch, S. Wagner, A. Zimmermann, Microservices in industry: Insights Into Technologies, characteristics, and software quality, 2019 IEEE International Conference on Software Architecture Companion. (2019). doi:10.1109/icsa-c.2019.00041.

S1-IEEE-86: A. Luntovskyy, B. Shubyn, Highly-distributed systems based on micro-services and their construction paradigms, 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering. (2020). doi:10.1109/tcset49122.2020.235378.

S1-SC-01: A. Sen, Devops, devsecops, aiops- paradigms to it operations, Lecture Notes in Electrical Engineering. (2021). doi:10.1007/978-981-15-7804-5-16.

S1-SC-06: G. Siewruk, W. Mazurczyk, A. Karpiński, Security assurance in DevOps methodologies and related environments, INTL Journal of Electronics and Telecommunications, 65 (2019) 211-216. doi: 10.24425/ijet.2019.126303.

S1-SC-07: V. Casola, A. De Benedictis, M. Rak, G. Salzillo, A cloud secdevops methodology: From design to testing, Communications in Computer and Information Science. (2020) 317–331. doi:10.1007/978-3-030-58793-2-26.

S1-SC-08: R. Kumar, R. Goyal, Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over Cloud (ADOC), Computers and Security. 97 (2020) 101967. doi:10.1016/j.cose.2020.101967.

S1-SC-09: K.V.D.Kiran, P.J.R.Shalem Raju, Performance Analysis of Automation Monitoring System shifting from devops to devsecops, International Journal of Emerging Trends in Engineering Research. 8 (2020) 5128–5134. doi:10.30534/ijeter/2020/40892020.

S1-SC-10: F. Moyón, R. Soares, M. Pinto-Albuquerque, D. Mendez, K. Beckers, Integration of security standards in DevOps Pipelines: An industry case study, Product-Focused Software Process Improvement. (2020) 434–452. doi:10.1007/978-3-030-64148-1-27.

S1-SC-11: X. Larrucea, A. Berreteaga, I. Santamaria, Dealing with security in a real devops environment, Communications in Computer and Information Science. (2019) 453–464. doi:10.1007/978-3-030-28005-5-35.

S1-SC-12: S. Vignesh, B.R. Kanna, AWS Infrastructure Automation and Security Prevention using DevOps, Advances in Intelligent Systems and Computing. (2020) 537–549. doi:10.1007/978-981-15-0199-9-46.

S1-SC-14: R. Ravinder, V. Sucharita, A Secure Cloud Service Deployment Framework for DevOps, Indonesian Journal of Electrical Engineering and Computer Science. 21 (2021) 874. doi:10.11591/ijeecs.v21.i2.pp874-885.

S1-SC-15: U. Zdun, E. Wittern, P. Leitner, Emerging trends, challenges, and experiences in DevOps and microservice apis, IEEE Software. 37 (2020) 87–91. doi:10.1109/ms.2019.2947982.

S1-SC-17: L. Bass, The software architect and DevOps, IEEE Software. 35 (2018) 8–10. doi:10.1109/ms.2017.4541051.

S1-SC-18: E. Zheng, P. Gates-Idem, M. Lavin, Building a virtually air-gapped secure environment in AWS, Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security. (2018). doi:10.1145/3190619.3190642.

S1-SC-19: S. Schork, F. Zahid, D. Pradhan, S. Kicin, A. Schwichtenberg, Building an open-source Cross-Cloud devops stack for a CRM enterprise application: A case study, IFIP Advances in Information and Communication Technology. (2019) 3–11. doi:10.1007/978-3-030-20883-7-1.

S1-SC-20: S.D. Duque Anton, D. Fraunholz, D. Krohmer, D. Reti, H.D. Schotten, F. Selgert, et al. Creating it from scratch: A practical approach for enhancing the security of IOT-Systems in a devops-enabled software development environment, Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops. (2020) 266–281. doi:10.1007/978-3-030-55583-2-20.

S1-SC-21: M.A. Akbar, S. Mahmood, M. Shafiq, A. Alsanad, A.A.-A. Alsanad, A. Gumaei, Identification and prioritization of devops success factors using Fuzzy-AHP approach, Soft Computing. (2020). doi:10.1007/s00500-020-05150-w.

S1-SC-22: V. Casola, A. De Benedictis, M. Rak, U. Villano, A novel security-by-design methodology: Modeling and assessing security by SLAS with a quantitative approach, Journal of Systems and Software. 163 (2020) 110537. doi:10.1016/j.jss.2020.110537.

S1-SC-25: Y. Verginadis, I. Patiniotakis, M. Prusinski, M. Rozanska, S. Schork, G. Mentzas, A security and privacy-preserving path for enhancing information systems that manage Cross-Cloud Applications, Advances in Intelligent Systems and Computing. (2020) 1119–1132. doi:10.1007/978-3-030-44038-1-103.

S1-SC-26: S. Almuairfi, M. Alenezi, Security controls in infrastructure as code, Computer Fraud and Security. (2020) 13–19. doi:10.1016/s1361-3723(20)30109-3.

S1-SC-27: C. Dyess, Maintaining a balance between agility and security in the cloud, Network Security. (2020) 14–17.
doi:10.1016/s1353-4858(20)30031-3.

S1-SC-29: N.C. Mendonca, P. Jamshidi, D. Garlan, C. Pahl, Developing self-adaptive microservice systems: Challenges and directions, IEEE Software. 38 (2021) 70–79. doi:10.1109/ms.2019.2955937.

S1-SC-31: B. Fitzgerald, K.-J. Stol, Continuous Software Engineering and beyond: Trends and challenges, Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014. (2014). doi:10.1145/2593812.2593813.

S1-SC-32: E. Amoroso, Recent progress in software security, IEEE Software. 35 (2018) 11–13. doi:10.1109/ms.2018.1661316.

S1-SC-34: A. Martin, S. Raponi, T. Combe, R. Di Pietro, Docker ecosystem – vulnerability analysis, Computer Communications. 122 (2018) 30–43. doi:10.1016/j.comcom.2018.03.011.

S1-SC-36: F. Boyer, X. Etchevers, N. de Palma, X. Tao, Architecture-based automated updates of distributed microservices, Service-Oriented Computing. (2018) 21–36. doi:10.1007/978-3-030-03596-9-2.

S1-SC-38: D. Klein, Micro-segmentation: Securing Complex Cloud Environments, Network Security. (2019) 6–10. doi:10.1016/s1353-4858(19)30034-0.

S1-SC-40: N. Ferry, J. Dominiak, A. Gallon, E. Gonzalez, E.ider Iturbe, S. Lavirotte, S. Martinez, A. Metzger, V. Muntes-Mulero, P. H. Nguyen, A. Palm, A. Rego, E. Rios, D. Riviera, A. Solberg, H. Song, J. Tigli, T. Winter, Development and operation of trustworthy smart IoT systems: the ENACT framework, DEVOPS 2019, (2020) 121–138, doi: 10.1007/978-3-030-39306-9-9

S1-SC-41: T. Pawlik, P.H. Meland, T. Stålhane, G.K. Hanssen, The agile RAMSS lifecycle for the future, Proceedings of the 29th European Safety and Reliability Conference. (2019). doi:10.3850/978-981-11-2724-3-0170-cd.

S1-SC-42: S. Kitajima, A. Sekiguchi, Latest image recommendation method for automatic base image update in dockerfile, Service-Oriented Computing. (2020) 547–562. doi:10.1007/978-3-030-65310-1-40.

S1-SC-44: J. Sandobalin, E. Insfran, S. Abrahao, Towards model-driven infrastructure provisioning for multiple clouds, Lecture Notes in

Information Systems and Organisation. (2019) 207–225. doi:10.1007/978-3-030-22993-1-12.

S1-SC-45: S. Sugandi, I. Riadi, A. Sugandi, Forensic analysis of docker swarm cluster using GRR Rapid Response Framework, International Journal of Advanced Computer Science and Applications. 10 (2019). doi:10.14569/ijacsa.2019.0100260.

S1-SC-48: S. Abraham, A.K. Paul, R.I. Khan, A.R. Butt, On the use of containers in high performance computing environments, 2020 IEEE 13th International Conference on Cloud Computing. (2020). doi:10.1109/cloud49709.2020.00048.

S2-ACM-04: R. K. Gupta, M. Venkatachalapathy, F. K. Jeberla, Challenges in adopting continuous delivery and devops in a globally distributed product team: A case study of a healthcare organization, Proceedings of 2019 ACM/IEEE 14th International Conference on Global Software Engineering. (2019). doi:10.1109/ICGSE.2019.00020.

S2-ACM-05: M. Viggiato, J. Oliveira, E. Figueiredo, P. Jamshidi, C. Kastner, Understanding similarities and differences in software development practices across domains, Proceedings of 2019 ACM/IEEE 14th International Conference on Global Software Engineering. (2019). doi:10.1109/icgse.2019.00013.

A.2. Grey literature articles (accessed June 30, 2021)

S1-GL-01: DevOps, Scaled Agile Framework. (2021). https://www.scaledagileframework.com/devops/.

S1-GL-02: What is the difference between DevOps and DevSecOps?, PVS. (2020). https://pvs-studio.com/en/blog/posts/0710/.

S1-GL-03: M. Foster, DevOps vs. devsecops - here's how they fit together, Red Hat OpenShift Makes Container Orchestration Easier. (2021). https://www.openshift.com/blog/devops-vs.-devsecops-heres-how-they-fit-togethe.

S1-GL-04: What is DevSecOps?, Red Hat - We Make Open Source Technologies for the Enterprise. (2018).
https://www.redhat.com/en/topics/devops/what-is-devsecops.

S1-GL-05: A. Singh, DevOps vs devsecops – what is the difference? Security Boulevard. (2020).
https://securityboulevard.com/2020/08/devops-vs-devsecops-what-is-the-difference/.

S1-GL-06: Microsoft Security devops, Microsoft Security DevOps. (n.d.).
https://www.microsoft.com/en-us/securityengineering/devsecops.

S1-GL-07: Security – Disciplined Agile (DA) - PMI, (n.d.). https://www.pmi.org/disciplined-agile/process/security.

S1-GL-08: C. Maerz, What's the difference between DevOps and DevSecOps? AppDynamics. (2021).
https://www.appdynamics.com/blog/product/devops-vs-devsecops/.

S1-GL-09: E. Miller, Difference between DevOps and devsecops. Invensis Learning Blog. (2019). https://www.pmi.org/disciplined-agile/process/security.

S1-GL-10: What is DevSecOps?: Devsecops model, Snyk. (2021). https://snyk.io/devsecops/.

S1-GL-11: L. Constantin, What is devsecops? Why it's hard to do well? CSO Online. (2020).
https://www.csoonline.com/article/3245748/what-is-devsecops-developing-more-secure-applications.html.

S1-GL-12: K. Magowan, What is devsecops? Combining development, Security and Operations, BMC Blogs. (2020).
https://www.bmc.com/blogs/devops-devsecops/.

S1-GL-13: M, Preston, DevOps VS DevSecOps: The differences. (2020). https://www.clouddefense.ai/blog/devops-vs-devsecops-the-differences.

S1-GL-14: M. Spisak and J. Darwin, Secure DevOps architecture. IBM. (n.d.). https://www.ibm.com/cloud/architecture/architectures/secure-devops-arch/.

S1-GL-15: What is DevSecOps: Devops security tools: Imperva, Learning Center. (2021). https://www.imperva.com/learn/application-security/devsecops-devops-security/.

S1-GL-16: K. Zettler, DevSecOps Tools, Atlassian. (2021). https://www.atlassian.com/devops/devops-tools/devsecops-tools.

S1-GL-17: DevOps security, CyberArk. (2021). https://www.cyberark.com/what-is/devops-security/.

S1-GL-18: DevOps Tech: Shifting left on security, Google. (2021). https://cloud.google.com/architecture/devops/devops-tech-shifting-left-on-security.

S1-GL-19: R. Velasco, DevSecOps: The 7 key factors to secure your DevOps practice, Hdiv Security. (2020).
https://hdivsecurity.com/bornsecure/devsecops-the-7-key-factors-to-secure-your-devops-practice/.

S1-GL-20: VeritisAdmin, DevOps security: An overview of Challenges and Best Practices, Go to Veritis Group Inc. (n.d.). https://www.veritis.com/blog/devops-security-an-overview-of-challenges-and-best-practices/.

S1-GL-21: I. Eldridge, SecDevOps: Injecting Security into DevOps Processes, New Relic. (2018). https://newrelic.com/blog/best-practices/what-is-secdevops.

S1-GL-22: S. Bocetta, How to seamlessly evolve DevOps into devsecops, InfoQ. (2019). https://www.infoq.com/articles/evolve-devops-devsecops/.

S1-GL-23: P. Academy, DevSecOps: Integrating security with DevOps, Medium. (2021).
https://blog.pentesteracademy.com/devsecops-learning-path-integrating-security-with-devops-1cc03670552f.

S1-GL-24: B. Dobran, Why you should be using devops security to deliver secure software, PhoenixNAP Blog. (2019).
https://phoenixnap.com/blog/devops-security-best-practice.

S1-GL-25: Top 10 devsecops best practices for building secure software: Synopsys, Application Security Blog. (n.d.).
https://codedx.com/blog/how-to-join-devops-and-security-best-practices-in-devsecops/.

S1-GL-26: What is DevSecOps?, Sumo Logic. (2019). https://www.sumologic.com/insight/devsecops-rugged-devops/.

S1-GL-27: What is DevSecOps?, Forcepoint. (2021). https://www.forcepoint.com/cyber-edu/devsecops.

S1-GL-28: G. Maayan, DevOps security challenges and how to overcome them, CCSI. (2019).
https://www.ccsinet.com/blog/devops-security-challenges/.

S1-GL-29: A. Uss, DevOps security challenges and best practices, Snyk. (2021). https://snyk.io/learn/devops-security.

S1-GL-30: DevOps security challenges and how to deal with them: Scalyr, SentinelOne. (2019).
https://www.sentinelone.com/blog/devopssec-challenges/.

S1-GL-31: Why security testing should be a part of the DevOps process, 6point6. (2021). https://6point6.co.uk/insights/why-security-testing-should-be-a-part-of-the-devops-process/.

S1-GL-32: P. Cheslock, How to integrate security into a DevOps World, Threat Stack. (2021).
https://www.threatstack.com/blog/how-to-integrate-security-into-a-devops-world.

S1-GL-33: L. Terquem, How to apply devops principles to increase security? (2020). https://www.padok.fr/en/blog/devsecops-security.

S1-GL-34: C. Brimhall, Closer than you think: Bridging the devops-security gap, Anitian. (2019). https://www.anitian.com/closer-than-you-think-bridging-the-devops-security-gap/.

S1-GL-35: M. Rimkus, From DevOps to devsecops: Securing the CI/CD pipeline, Cherry Servers. (2020).
https://blog.cherryservers.com/from-devops-to-devsecops-securing-the-cicd-pipeline.

S1-GL-36: F. Reimer, Cybersecurity for Business Leaders, SecurityRoundTable.org. (n.d.). https://www.securityroundtable.org/.

S1-GL-37: A. Arampatzis, Why is it such a challenge to integrate security into devops?, DATAVERSITY. (2021). https://www.dataversity.net/why-is-it-such-a-challenge-to-integrate-security-into-devops/.

S1-GL-38: H. Bavati, From DevOps to DevSecOps: The Security Challenges of DevOps. Datafloq. (2019). https://datafloq.com/read/from-devops-devsecops-security-challenges.

S1-GL-39: R. Annadi, Overcoming the Top 3 DevOps Security Challenges. Devopsdigest. (2020). https://www.devopsdigest.com/overcoming-the-top-3-devops-security-challenges.

S1-GL-40: S. Ben-Hador, From devops to devsecops: The security challenges of devops, Exabeam. (2019). https://www.exabeam.com/information-security/devsecops-and-the-security-challenges-of-devops/.

S1-GL-41: M. Vernon, Devsecops: The intersection of devops and security, Victorops Blog. (2019). https://victorops.com/blog/devsecops-the-intersection-of-devops-and-security.

S1-GL-42: T. Blogumas, Top 15 devsecops tools for an enterprise CI/CD pipeline, Medium. (2020). https://levelup.gitconnected.com/top-15-devsecops-tools-for-an-enterprise-ci-cd-pipeline-bd865b47ed5f.

S1-GL-43: E. Chickowski, Seven winning DevSecOps metrics security should track. (2018). https://businessinsights.bitdefender.com/seven-winning-devsecops-metrics-security-should-track.

A.3. New literature from confirmatory search (2021–2022)

CS-ACM-01: R.N. Rajapakse, M. Zahedi, M.A. Babar, An empirical analysis of practitioners' Perspectives on Security Tool Integration into DevOps, Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. (2021). doi:10.1145/3475716.3475776.

CS-ACM-02: D. Gonzalez, P.P. Perez, M. Mirakhorli, Barriers to shift-left security, Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. (2021). doi:10.1145/3475716.3475786.

CS-ACM-03: R. Brasoveanu, Y. Karabulut, I. Pashchenko, Security maturity self-assessment framework for software development lifecycle, Proceedings of the 17th International Conference on Availability, Reliability and Security.(2022).doi:10.1145/3538969.3543806.

CS-ACM-04: L. Liu, D. Xie, Y.C. Cheng, G. Li, Architecture scheme of devops for Cross Network and multiple environment collaboration, The 5th International Conference on Computer Science and Application Engineering. (2021). doi:10.1145/3487075.3487116.

CS-IEEE-01: S. Throner, H. Hutter, N. Sanger, M. Schneider, S. Hanselmann, P. Petrovic, et al. An advanced devops environment for Microservice-based applications, 2021 IEEE International Conference on Service-Oriented System Engineering. (2021). doi:10.1109/sose52839.2021.00020.

CS-IEEE-02: S.F. Ahamed, M. Dhar M S, S.K. Kishore, M.P. Borawake, T.D. R, M. Thenmozhi, DevOps security and privacy in the development of Multicloud Applications, 2022 International Conference on Electronics and Renewable Systems. (2022). doi:10.1109/icears53579.2022.9752387.

CS-IEEE-03: A. Sojan, R. Rajan, P. Kuvaja, Monitoring solution for cloud-native devsecops, 2021 IEEE 6th International Conference on Smart Cloud. (2021). doi:10.1109/smartcloud52277.2021.00029.

CS-IEEE-04: F. Angermeir, M. Voggenreiter, F. Moyon, D. Mendez, Enterprise-driven open source software: A case study on security automation, 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice. (2021). doi:10.1109/icse-seip52600.2021.00037.

CS-IEEE-05: A. Ibrahim, A.H. Yousef, W. Medhat, DevSecOps: A security model for infrastructure as code over the cloud, 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference. (2022). doi:10.1109/miucc55081.2022.9781709.

CS-IEEE-06: Y. Yang, W. Shen, B. Ruan, W. Liu, K. Ren, Security challenges in the container cloud, 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications. (2021). doi:10.1109/tpsisa52974.2021.00016.

CS-SC-01: M.A. Akbar, K. Smolander, S. Mahmood, A. Alsanad, Toward successful DevSecOps in software development organizations: A decision-making framework, Information and Software Technology. 147 (2022) 106894. doi:10.1016/j.infsof.2022.106894.

CS-SC-02: Nisha T. N., A. Khandebharad, Migration from devops to devsecops, International Journal of Cloud Applications and Computing. 12 (2022) 1–15. doi:10.4018/ijcac.2022010102.

CS-SC-03: R.N. Rajapakse, M. Zahedi, M.A. Babar, H. Shen, Challenges and solutions when adopting DevSecOps: A systematic review, Information and Software Technology. 141 (2022) 106700. doi:10.1016/j.infsof.2021.106700.

CS-GL-01: S. Ingalls, Best DevSecOps Tools for 2022: eSecurity Planet, ESecurityPlanet. (2022). https://www.esecurityplanet.com/products/devsecops-tools/.

CS-GL-02: What is DevSecOps? JFrog. (2022). https://jfrog.com/devops-tools/what-is-devsecops/.

CS-GL-03: A. Neto, What is devsecops: Top 5 automation tools for CI pipelines, RSS. (n.d.). https://bluelight.co/blog/what-is-devsecops.

CS-GL-04: DevSecOps Best practices, Tigera. (2022). https://www.tigera.io/learn/guides/devsecops/devsecops-best-practices/.

CS-GL-05: S. Manjaly, The top 10 best devsecops tools for 2022, IT Management Software. (2022). https://blog.invgate.com/devsecops-tools.

CS-GL-06: J. Hirschauer, Top 10 best practices for devsecops, Harness.io. (2022). https://harness.io/blog/best-practices-devsecops.

CS-GL-07: M. Hales, Devsecops challenges, DevSecOps Challenges. (2021). https://www.adaptavist.com/blog/8-common-devsecops-challenges-and-how-to-overcome-them.

References

Acuna, S.T., Juristo, N., 2004. Assigning people to roles in software projects. Qual. Res. Psychol. 34 (7), 675–696. http://dx.doi.org/10.1002/spe.586.

Ahamed, S.F., Murali Dhar, M.S., Kishore, S.K., p. Borawake, M., Thirupurasundari, D.R., Thenmozhi, M., 2022. DevOps security and privacy in the development of multicloud applications. In: Proceedings of the International Conference on Electronics and Renewable Systems. IEEE, pp. 1631–1635. http://dx.doi.org/10.1109/icears53579.2022.9752387.

Ahmed, Z., Francis, S.C., 2019. Integrating security with devsecops: Techniques and challenges. In: 2019 International Conference on Digitization. IEEE, pp. 178–182. http://dx.doi.org/10.1109/icd47981.2019.9105789.

Akbar, M.A., Smolander, K., Mahmood, S., Alsanad, A., 2022. Toward successful DevSecOps in software development organizations: A decision-making framework. Inf. Softw. Technol. 147 (1), 1–21. http://dx.doi.org/10.1016/j.infsof.2022.106894.

Alharahsheh, H.H., Pius, A., 2020. A review of key paradigms: positivism VS interpretivism. Glob. Acad. J. Humanit. Soc. Sci. 2 (3), 39–43. http://dx.doi.org/10.36348/gajhss.2020.v02i03.001.

Amaro, R., Pereira, R., da Silva, M.M., 2023. Capabilities and metrics in DevOps: A design science studys. Inf. Manag. 60 (5), http://dx.doi.org/10.1016/j.im.2023.103809.

Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., Chatzigeorgiou, A., 2019. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. Inf. Softw. Technol. 106 (1), 201–230. http://dx.doi.org/10.1016/j.infsof.2018.10.006.

Angermeir, F., Voggenreiter, M., Moyon, F., Mendez, D., 2021. Enterprise-driven open source software: A case study on security automation. In: Proceedings of the 43rd International Conference on Software Engineering: Software Engineering in Practice. ICSE-SEIP, IEEE, pp. 278–287. http://dx.doi.org/10.1109/icse-seip52600.2021.00037.

Antil, P., 2023. Requirements Prioritization in Scaled Agile Distributed Software Development (Ph.D. thesis). Auckland University of Technology, URL http://hdl.handle.net/10292/16580.

Bass, L., Weber, I.M., Zhu, L., 2015. DevOps: A Software Architect's Perspective. Addison-Wesley.

Beecham, S., Clea, T., Lal, R., Noll, J., 2021. Do scaling agile frameworks address risk in global software development? An empirical study. J. Syst. Softw. 171 (110823), http://dx.doi.org/10.1016/j.jss.2020.110823.

Betts, D., 2022. 3 Essential Steps to Enable Security in DevOps. URL https://www.gartner.com/en/documents/4261699.

Blogumas, T., 2020. Top 15 devsecops tools for an enterprise CI/CD pipeline. URL https://levelup.gitconnected.com/top-15-devsecops-tools-for-an-enterprise-ci-cd-pipeline-bd865b47ed5f.

Brasoveanu, R., Karabulut, Y., Pashchenko, I., 2022. Security maturity self-assessment framework for software development lifecycle. In: Proceedings of the 17th International Conference on Availability, Reliability and Security. ACM, Vienna, Austria, pp. 1–8. http://dx.doi.org/10.1145/3538969.3543806.

Braun, V., Clarke, V., 2006. Using thematic analysis in psychology. Qual. Res. Psychol. 3 (2), 77–101. http://dx.doi.org/10.1191/1478088706qp063oa.

Braun, V., Clarke, V., 2020. Thematic analysis: a reflexive approach. URL https://www.psych.auckland.ac.nz/en/about/thematic-analysis.html.

Braun, V., Clarke, V., 2021. One size fits all? What counts as quality practice in (reflexive) thematic analysis? Qual. Res. Psychol. 18 (3), 328–352. http://dx.doi.org/10.1080/14780887.2020.1769238.

Brunelli, M., 2014. Introduction to the Analytic Hierarchy Process. Spring.

Burns, A., McDermid, J., Dobson, J., 1992. On the meaning of safety and security. Comput. J. 35 (1), 3–15. http://dx.doi.org/10.1093/comjnl/35.1.3.

Carter, K., 2017. Francois Raynaud on devsecops. IEEE Softw. 34 (5), 93–96. http://dx.doi.org/10.1109/ms.2017.3571578.

Chakrabarty, A., Hanley, M., Daugherty, R., O'Shea, B., 2023. Redefining the next decade of cybersecurity: AI-powered security built to empower developers [plenary presentation SEC2732m]. In: GitHubUniverse. GitHub, URL https://reg.githubuniverse.com/flow/github/universe23/sessioncatalog/page/sessioncatalog/session/1689094392389001bUiL.

Chickowski, E., 2018. Seven winning DevSecOps metrics security should track. URL https://businessinsights.bitdefender.com/seven-winning-devsecops-metrics-security-should-track.

Cico, O., Jaccheri, L., Nguyen-Duc, A., Zhang, H., 2021. Exploring the intersection between software industry and software engineering education - A systematic mapping of software engineering trends. J. Syst. Softw. 172, http://dx.doi.org/10.1016/j.jss.2020.110736.

Conchuir, E.O., Ågerfalk, P.J., Olsson, H.H., Fitzgerald, B., 2009. Global software development: Where are the benefits? Commun. ACM 52 (8), 127–131. http://dx.doi.org/10.1145/1536616.1536648.

Cruzes, D.S., Dyba, T., 2010. Synthesizing evidence in software engineering research. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, Bolzano-Bozen, Italy, pp. 1–10. http://dx.doi.org/10.1145/1852786.1852788.

Cruzes, D.S., Dyba, T., 2011a. Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement. IEEE, pp. 275–284. http://dx.doi.org/10.1109/ESEM.2011.36.

Cruzes, D.S., Dyba, T., 2011b. Research synthesis in software engineering: A tertiary study. Inf. Softw. Technol. 53 (5), 440–455. http://dx.doi.org/10.1016/j.infsof.2011.01.004.

Diel, E., Marczak, S., Cruzes, D.S., 2016. Communication challenges and strategies in distributed DevOps. In: 2016 IEEE 11th International Conference on Global Software Engineering. IEEE, pp. 24–28. http://dx.doi.org/10.1109/ICGSE.2016.28.

Dyck, A., Penners, R., Lichter, H., 2015. Towards definitions for release engineering and devops. In: 2015 IEEE/ACM 3rd International Workshop on Release Engineering. IEEE, http://dx.doi.org/10.1109/RELENG.2015.10, 3–3.

Edmundson, C., Hartman, K., 2022. SANS 2022 DevSecOps Survey: Creating a Culture to Significantly Improve Your Organization's Security Posture. URL https://www.sans.org/white-papers/sans-2022-devsecops-survey-creating-culture-improve-organization-security/.

Farace, D.J., Schopfel, J., 2010. Grey Literature in Library and Information Studies. De Gruyter Saury.

Fernandez, G.P., Brito, A., 2019. Secure container orchestration in the cloud: Policies and implementation. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. ACM, pp. 138–145. http://dx.doi.org/10.1145/3297280.3297296.

Garousi, V., Felderer, M., Mäntylä, M., 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Inf. Softw. Technol. 106, 101–121. http://dx.doi.org/10.1016/j.infsof.2018.09.006.

Gonzalez, D., Perez, P.P., Mirakhorli, M., 2021. Barriers to shift-left security: The unique pain points of writing automated tests involving security controls. In: Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM, ACM, Bari, Italy, pp. 1–12. http://dx.doi.org/10.1145/3475716.3475786.

Grande, R., Vizcaino, A., Garcia, F.O., 2024. Is it worth adopting DevOps practices in global software engineering? possible challenges and benefits. Comput. Stand. Interfaces 87 (1), http://dx.doi.org/10.1016/j.csi.2023.103767.

Gupta, R.K., Venkatachalapathy, M., Jeberla, F.K., 2019. Challenges in adopting continuous delivery and DevOps in a globally distributed product team: A case study of a healthcare organization. In: 2019 ACM/IEEE 14th International Conference on Global Software Engineering. IEEE, pp. 30–34. http://dx.doi.org/10.1109/ICGSE.2019.00020.

Hoda, R., 2021. Socio-technical grounded theory for software engineering. IEEE Trans. Softw. Eng. 48 (10), 3808–3832. http://dx.doi.org/10.1109/TSE.2021.3106280.

Humble, J., Molesky, J., 2011. DevOps: a software revolution in the making? CutterIT J. 24 (8), 6–24.

Hussain, W., Clear, T., MacDonell, S., 2017. Emerging trends for global DevOps: A New Zealand perspective. In: 2017 IEEE 12th International Conference on Global Software Engineering. IEEE, pp. 21–30. http://dx.doi.org/10.1109/icgse.2017.16.

Huttermann, M., 2012. DevOps for Developers, first ed. A Press, Berkeley, CA, http://dx.doi.org/10.1007/978-1-4302-4570-4.

Ibrahim, A., Yousef, A.H., Medhat, W., 2022. DevSecOps: A security model for infrastructure as code over the cloud. In: Proceedings of the 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference. MIUCC, IEEE, pp. 284–288. http://dx.doi.org/10.1109/miucc55081.2022.9781709.

Jabbari, R., bin Ali, N., Petersen, K., Tanveer, B., 2016. What is DevOps?: A systematic mapping study on definitions and practices. In: Proceedings of the Scientific Workshop Proceedings of XP2016. In: XP '16 Workshops, ACM, New York, NY, USA, http://dx.doi.org/10.1145/2962695.2962707.

Jalali, S., Gencel, C., Šmite, D., 2010. Trust dynamics in global software engineering. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM '10, ACM, New York, NY, USA, http://dx.doi.org/10.1145/1852786.1852817.

Jireh, 2016. What is DevOps. URL http://www.jirehtechconsulting.com/what-is-devops/.

Kitchenham, B., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report EBSE-2007-01, Keele University and Durham University, ST5 5BG, UK and Durham, UK.

Kitchenham, B.A., Dyba, T., Jorgensen, M., 2004. Evidence-based software engineering. In: Proceedings of the 26th International Conference on Software Engineering. IEEE, pp. 273–281. http://dx.doi.org/10.1109/ICSE.2004.1317449.

Kumar, R., Goyal, R., 2020. Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC). Comput. Secur. 97, http://dx.doi.org/10.1016/j.cose.2020.101967.

Lal, R., Clear, T., 2021. Three levels of agile planning in a software vendor environment. In: Proceedings of the 2021 Australasian Conference on Information Systems. pp. 1–12, URL https://aisel.aisnet.org/acis2021/48/.

Line, M.B., Rostad, L., 2006. Safety vs. Security? In: Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management. ASME, pp. 1202–1210. http://dx.doi.org/10.1115/1.802442.paper151.

Liu, L., Xie, D., Cheng, Y., Li, G., 2021. Architecture scheme of DevOps for cross network and multiple environment collaboration. In: Proceedings of the 5th International Conference on Computer Science and Application Engineering. ACM, Sanya, China, pp. 1–5. http://dx.doi.org/10.1145/3487075.3487116.

Loukides, M., 2012. What is devops?

MacDonald, N., Head, I., 2016. Devsecops: How to seamlessly integrate security into devops. URL https://https://www.gartner.com/en/documents/3463417.

Mao, R., Zhang, H., Dai, Q., Huang, H., Rong, G., Shen, H., Chen, L., Lu, K., 2020. Preliminary findings about DevSecOps from grey literature. In: 2020 IEEE 20th International Conference on Software Quality, Reliability and Security. QRS, IEEE, http://dx.doi.org/10.1109/QRS51102.2020.00064.

Mohammed, N.M., Niazi, M., Alshayeb, M., Mahmood, S., 2017. Exploring software security approaches in software development lifecycle: A systematic mapping study. Comput. Stand. Interfaces 50, 107–115.

Mohan, V., Othmane, L.B., 2016. Secdevops: Is it a marketing buzzword? - mapping research on security in devops. In: 2016 11th International Conference on Availability, Reliability and Security. IEEE, pp. 205–210. http://dx.doi.org/10.1109/ares.2016.92.

Morales, J.A., Scanlon, T.P., Volkmann, A., Yankel, J., Yasar, H., 2020. Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. ACM, New York, NY, USA, http://dx.doi.org/10.1145/3407023.3409186.

Myrbakken, H., Colomo-Palacios, R., 2017. DevSecOps: A multivocal literature review. In: Software Process Improvement and Capability Determination. Vol. 770, Springer, Cham, pp. 17–29. http://dx.doi.org/10.1007/978-3-319-67383-7_2.

Nisha, T.N., 2022. Migration from DevOps to DevSecOps: A complete migration framework, challenges, and evaluation. Int. J. Cloud Appl. Comput. 12 (1), 1–15. http://dx.doi.org/10.4018/ijcac.2022010102.

Pothukuchi, A.S., Kota, L.V., Mallikarjunaradhya, V., 2023. Impact of generative AI on the software development lifecycle (SDLC). Int. J. Creat. Res. Thoughts 11 (8).

Prates, L., Faustino, J., Silva, M., Pereira, R., 2019. DevSecOps metrics. In: Information Systems: Research, Development, Applications, Education. Vol. 359, Springer, Cham, pp. 77–90. http://dx.doi.org/10.1007/978-3-030-29608-7_7.

Puppet, 2023. State of DevOps Report 2023. URL https://www.puppet.com/success/resources/state-of-platform-engineering.

Rafi, S., Yu, W., Akbar, M.A., Alsanad, A., Gumaei, A., 2020. Prioritization based taxonomy of DevOps security challenges using PROMETHEE. IEEE Access 8, 105426–105446. http://dx.doi.org/10.1109/ACCESS.2020.2998819.

Rahman, A.A.U., Williams, L., 2016. Software security in devops: Synthesizing practitioners' perceptions and practices. In: Proceedings of the International Workshop on Continuous Software Evolution and Delivery. ACM, New York, NY, USA, pp. 70–76. http://dx.doi.org/10.1145/2896941.2896946.

Rajapakse, R.N., Zahedi, M., Babar, M.A., 2021. An empirical analysis of practitioners' perspectives on security tool integration into DevOps. In: Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM, ACM, Bari, Italy, pp. 1–12. http://dx.doi.org/10.1145/3475716.3475776.

Rajapakse, R.N., Zahedi, M., Babar, M.A., Shen, H., 2022. Challenges and solutions when adopting DevSecOps: A systematic review. Inf. Softw. Technol. 141 (1), 1–22. http://dx.doi.org/10.1016/j.infsof.2021.106700.

Rowe, F., 2014. What literature review is not: Diversity, boundaries and recommendations. Eur. J. Inf. Syst. 23 (3), 241–255. http://dx.doi.org/10.1057/ejis.2014.7.

Sanchez-Gordon, M., Colomo-Palacios, R., 2020. Security as culture: A systematic literature review of DevSecOps. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. ACM, New York, NY, USA, pp. 266–269. http://dx.doi.org/10.1145/3387940.3392233.

Schopfel, J., 2010. Towards a Prague Definition of Grey Literature. URL https://greynet.org/images/GL12_S1S,_Sch_pfel.pdf.

Sebastian, I.M., Ross, J.W., Beath, C., Mocker, M., Moloney, K.G., Fonstad, N.O., 2020. How big old companies navigate digital transformation. In: Strategic Information Management, fifth ed. Routledge, pp. 133–150. http://dx.doi.org/10.4324/9780429286797-6, (inbook).

Sen, A., 2021. DevOps, DevSecOps, AIOPS- paradigms to IT operations. Lect. Notes Electr. Eng. 211–221. http://dx.doi.org/10.1007/978-981-15-7804-5_16.

Smeds, J., Nybom, K., Porres, I., 2015. Devops: A definition and perceived adoption impediments. In: Agile Processes in Software Engineering and Extreme Programming. Vol. 212, Springer, pp. 166–177. http://dx.doi.org/10.1007/978-3-319-18612-2_14.

Sojan, A., Rajan, R., Kuvaja, P., 2021. Monitoring solution for cloud-native DevSecOps. In: Proceedings of the 6th International Conference on Smart Cloud. SmartCloud, IEEE, pp. 125–131. http://dx.doi.org/10.1109/smartcloud52277.2021.00029.

Soni, M., 2015. End to end automation on cloud with build pipeline: The case for DevOps in insurance INDUSTRY, continuous integration, continuous testing, and continuous delivery. In: 2015 IEEE International Conference on Cloud Computing in Emerging Markets. IEEE, pp. 85–89. http://dx.doi.org/10.1109/CCEM.2015.29.

Tamburri, D.A., Razo-Zapata, I.S., Fernández, H., Tedeschi, C., 2012. Simulating awareness in global software engineering: A comparative analysis of scrum and agile service networks. In: 2012 4th International Workshop on Principles of Engineering Service-Oriented Systems. IEEE, pp. 1–7. http://dx.doi.org/10.1109/PESOS.2012.6225933.

Tomas, N., Li, J., Huang, H., 2019. An empirical study on culture, automation, measurement, and sharing of DevSecOps. In: 2019 International Conference on Cyber Security and Protection of Digital Services. Cyber Security, IEEE, http://dx.doi.org/10.1109/cybersecpods.2019.8884935.

Viggiato, M., Oliveira, J., Figueiredo, E., Jamshidi, P., Kastner, C., 2019. Understanding similarities and differences in software development practices across domains. In: 2019 ACM/IEEE 14th International Conference on Global Software Engineering. IEEE, pp. 84–94. http://dx.doi.org/10.1109/icgse.2019.00013.

Vizcaíno, A., García, F., Piattini, M., Beecham, S., 2016. A validated ontology for global software development. Comput. Stand. Interfaces 46, 66–78. http://dx.doi.org/10.1016/j.csi.2016.02.004.

Wagner, T.J., Ford, T.C., 2020. Metrics to meet security and privacy requirements with agile software development methods in a regulated environment. In: 2020 International Conference on Computing, Networking and Communications. ICNC, IEEE, pp. 17–23. http://dx.doi.org/10.1109/icnc47757.2020.9049681.

Wang, C.L., Ahmed, P.K., 2007. Dynamic capabilities: A review and research agenda. Int. J. Manag. Rev. 9 (1), 31–51. http://dx.doi.org/10.1111/j.1468-2370.2007.00201.x.

Whitehead, J., Mistrik, I., Grundy, J., van der Hoek, A., 2010. Collaborative software engineering: Concepts and techniques. In: Collaborative Software Engineering. Springer, pp. 1–30. http://dx.doi.org/10.1007/978-3-642-10294-3, (inbook).

Wohlin, C., Kalinowski, M., Felizardo, K.R., Mendes, E., 2022. Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. Inf. Softw. Technol. 147 (1), 1–12. http://dx.doi.org/10.1016/j.infsof.2022.106908.

Zaydi, M., Nassereddine, B., 2020. DevSecOps practices for an agile and secure it service management. J. Manag. Inf. Decis. Sci. 23 (2), 134–149, doi:1532-5806-23-2-186. URL https://www.abacademies.org/articles/DevSecOps-practices-for-an-agile-and-secure-it-service-management-1532-5806-23-2-186.pdf.

Xiaofan Zhao is a Ph.D. candidate and researcher in the Department of Computer Science and Software Engineering, Auckland University of Technology (AUT), New Zealand. He received his master and bachelor degrees at the same university. He is active in research within the security aspects of DevOps (DevSecOps) and global software engineering.

Tony Clear is an Associate Professor in the Department of Computer Science and Software Engineering, Auckland University of Technology (AUT), New Zealand, and an ACM Distinguished Member. He is also Co-Director of the Software Engineering Centre (SERC - https://serc.aut.ac.nz/) with Prof. Jacqueline Whalley. He holds positions as an Associate Editor for ACM Transactions on Computing Education (TOCE), for the journal Computer Science Education, and ACM Inroads for which he is also a regular columnist and Editorial Board member. He is active in research within the global software engineering and computer science education communities. With Professor Daniela Damian of University of Victoria, Canada he has been working on a Royal Society of NZ International Leaders Fellowship Grant titled - "Leading the Way in Software Ecosystems for NZ". Tony has served on the steering committee for ICGSE and has chaired or served on the programme committee for conferences such as ICGSE, EASE, ITiCSE, ICER, ACE, FIE, CITRENZ, APRES, ECIS, SIESC, and reviewed for journals such as TSE, IST, JSS, JSEP, IEEE S/W, IJEE, CLEIej. He supervises and has examined doctoral students in Global Software Engineering, CS Education and interdisciplinary topics, and has chaired or participated in several doctoral consortia including ICER 2023.

Ramesh Lal is a senior lecturer in the department of Computer Science and Software Engineering, Auckland University of Technology (AUT), New Zealand. He is active in research within agile software engineering processes and practices and agile project management. Ramesh has served on the programme committee for conference such as ACIS and reviewed for journals such as Journal of Software: Evolution and Process, and Australasian Journal of Information Systems. He supervises doctoral and master students in agile software engineering processes and agile project management including interdisciplinary topics such as data mining.