



ORIGINAL RESEARCH

Revisit security in the era of DevOps: An evidence-based inquiry into DevSecOps industry

Xin Zhou¹  | Runfeng Mao¹ | He Zhang¹ | Qiming Dai² | Huang Huang³  |
Haifeng Shen⁴ | Jingyue Li⁵ | Guoping Rong¹

¹State Key Laboratory for Novel Software Technology, Software Institute, Nanjing University, Nanjing, China

²Huatai Securities Co., Ltd., Nanjing, China

³State Grid Nanjing Power Supply Company, Nanjing, China

⁴Peter Faber Business School, Australian Catholic University, Sydney, New South Wales, Australia

⁵Norwegian University of Science and Technology, Trondheim, Norway

Correspondence

He Zhang.

Email: hezhang@nju.edu.cn

Funding information

Innovation Project and Overseas Open Project of State Key Laboratory for Novel Software Technology (Nanjing University), Grant/Award Numbers: KFKT2022A09, ZZKT2022A25; National Key Research and Development Program of China, Grant/Award Number: 2019YFE0105500; Research Council of Norway, Grant/Award Number: 309494; National Natural Science Foundation of China, Grant/Award Numbers: 62072227, 62202219; Key Research and Development Program of Jiangsu Province, Grant/Award Number: BE2021002-2

Abstract

By adopting agile and lean practices, DevOps aims to achieve rapid value delivery by speeding up development and deployment cycles, which however lead to more security concerns that cannot be fully addressed by an isolated security role only in the final stage of development. *DevSecOps* promotes security as a shared responsibility integrated into the DevOps process that seamlessly intertwines development, operations, and security from the start throughout to the end of cycles. While some companies have already begun to embrace this new strategy, both industry and academia are still seeking a common understanding of the DevSecOps movement. The goal of this study is to report the state-of-the-practice of DevSecOps, including the impact of DevOps on security, practitioners' understanding of DevSecOps, and the practices associated with DevSecOps as well as the challenges of implementing DevSecOps. The authors used a mixed-methods approach for this research. The authors carried out a grey literature review on DevSecOps, and surveyed the practitioners of DevSecOps in industry of China. The status quo of DevSecOps in industry is summarized. Three major software security risks are identified with DevOps, where the establishment of DevOps pipeline provides opportunities for security-related activities. The authors classify the interpretations of DevSecOps into three core aspects of DevSecOps capabilities, cultural enablers, and technological enablers. To materialise the interpretations into daily software production activities, the recommended DevSecOps practices from three perspectives—people, process, and technology. Although a preliminary consensus is that DevSecOps is regarded as an extension of DevOps, there is a debate on whether DevSecOps is a superfluous term. While DevSecOps is attracting an increasing attention by industry, it is still in its infancy and more effort needs to be invested to promote it in both research and industry communities.

KEYWORDS

software development management, software engineering

1 | INTRODUCTION

Given the diverse customer demands and rapidly changing marketplace, a common desire about Software Engineering (SE) in industry is for agility in order to timely realise and/or adapt business value [1]. As a result, various agile methodologies, such as Scrum [2], eXtreme Programming (XP) [3], and

Kanban [4] have been widely adopted in software development. By seamlessly spreading the agile culture across development and operations, and by emphasising software quality and collaboration between development and operation teams, DevOps [5] has emerged as a paradigmatic shift towards evolving software at a continuous pace and streamlining all parts of the software lifecycle. It is crucial that software teams

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Authors. *IET Software* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

have ownership and responsibility to deploy software changes in DevOps [6], which allows the software to be delivered quickly [7]. At such a frequent deployment and delivery (e.g. up to 500 times a day in Facebook [8]), the software might not undergo adequate security reviews [9]. In this context, MacDonald from Gartner pointed out that [10]:¹

Development, operations and security are fundamentally intertwined and DevOps must evolve to a new vision of DevOpsSec¹.

As shown in Figure 1, Gartner describes the new and updated services cycle through an iterative DevSecOps process in their report, where security becomes a shared responsibility integrated into the entire DevOps process [11].

While software industry has been increasingly adopting DevSecOps in their projects and the academia has begun to pay attention to this new paradigm [12–14], a shared understanding of the DevSecOps movement underpinned by solid evidence is generally missing. As there is very little academic literature on DevSecOps, like the emerging stage of many other trendy topics in SE such as microservices [15], evidence has to first come in the form of Grey Literature (GL), which is mostly produced by industry practitioners [16] and can serve as an important supplement to the shortage of academic literature [17].

Although DevSecOps has gained an increasing attention in academia [12–14], industry still leads this movement. According to our survey, the Chinese industry's attention to DevSecOps is increasing year by year, but good practices are not sufficient. We need to face the international industry to seek more practices and discoveries. Given the absence of academic literature on DevSecOps, to gain the state-of-the-practice of DevSecOps, we carried out a Grey Literature Review (GLR) on DevSecOps with reference to the guidelines [18]. From the 215 grey articles retrieved by Google search engine, we analysed the impacts of DevOps on software security and identified three major challenges that DevOps brings to software security including sacrifice of security for speed/agility, afterthought in the process, and environment risks. Nevertheless, the centralised and standardised DevOps pipeline also provides opportunities for performing security related activities. We then classified the interpretations of DevSecOps into three core aspects including DevSecOps capabilities, cultural enablers, and technological enablers. We further elaborate the recommended DevSecOps practices from the three perspectives of people, process and technology in order to materialise the three interpretations of DevSecOps into daily software production activities.

This article is a significant extension of a prior conference short paper version [19] that initially reports the limited findings from the early stage of the review. This article extensively

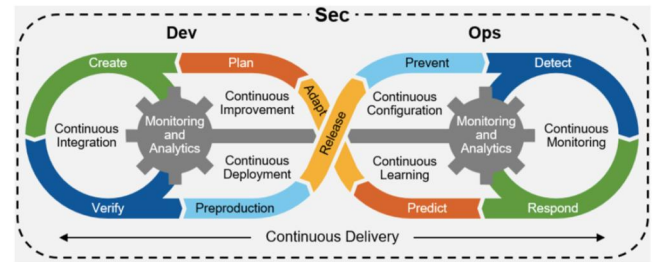


FIGURE 1 DevSecOps in Gartner Report [11].

describes the work we carried out after the preliminary findings, in particular achieves five major contributions as below:

1. The state-of-the-research on DevSecOps by academic literature is presented to justify the adoption of GLR.
2. The major impacts of DevOps on software security are identified with the synthesised understanding of DevSecOps from the perspectives of capabilities and enablers.
3. The recommended DevSecOps practices are classified in terms of people, process and technology.
4. The challenges of implementing DevSecOps and the contentious views on the relationship between DevOps and DevSecOps are discussed to arouse the interests of both research and industry communities.

The rest of this paper is organised as follows. Section 2 introduces the background of DevSecOps and Grey Literature. Section 3 describes the research questions and the methodology of our review. Sections 4 and 5 present the findings from the synthesis of academic and grey literature respectively, followed by the discussion related to this review in Section 6. Section 7 compares this study with the related secondary studies on DevSecOps and Section 8 shares lessons in conducting GLR and identifies potential threats to the validity of our study. Section 9 draws the conclusions with a summary of the findings, the implications, and the directions for future work.

2 | THE VOICE ON THE SE COMMUNITY

This section introduces grey literature and its use in SE as well as the background of DevSecOps from the perspectives of DevOps and software security respectively.

2.1 | Software security in DevOps

Software security has always been a key concern of Chinese enterprise segments, especially when cyber crime is accelerating nowadays. If software security is not adequately addressed, security breaches could result in massive losses in an enterprise. For example, British Airways was attacked due to 22 lines of unsafe code, causing personal information leak involving

¹There exist three synonyms describing the integration of security into DevOps, that is, *SecDevOps*, *DevSecOps*, and *DevOpsSec*. Although there might be some subtle differences among them, for the ease of discussion, we use *DevSecOps* consistently throughout the paper.

approximately 380,000 customers in 2018 [20]. Kraemer's study [21] showed that programmers tend to neglect security issues when they are affected by certain external factors such as time pressure or high workload. When software is deployed and delivered at a rapid pace by adopting DevOps, developers are more susceptible to these external factors. If the changed software is deployed in a production environment without undergoing sufficient security reviews, vulnerabilities are likely to occur and consequently the risk of being attacked is high.

Hence, many organisations and practitioners attempt to integrate security into the DevOps pipeline by applying some protection practices, such as providing security training for developers and adopting some traditional security activities [9]. The concept of DevSecOps [22] opens up new horizons for holistically bringing security principles into the DevOps process, which would help organisations in developing and delivering high quality and more secure software. Various dimensions (culture [23], challenge [24, 25] etc.) of DevSecOps have been discussed in academia to form the full view of this concept.

2.2 | Grey literature in Software Engineering

SE practitioners often share their knowledge and experience through channels that are not as rigorous as academic (peer-reviewed) publications, such as free online books and blogs [26–28]. Scientific information produced and published in this fashion is commonly referred to as grey literature. Because of the scarcity of academic literature on certain topics in SE, especially those that are trendy or industry driven, SE researchers have recently started to pay more attention to grey literature. Shpilko et al. [29] proposed a model for GL by following Kepes's study [30] that divides literature into four grey scales according to the difference in scope.

As SE is a practitioner-oriented engineering field, it is vital to establish connections between research and practice. Garousi et al. [31] stated that it is essential and useful to use GL to achieve

the research-practice balance in SE research. In ref. [32], we identified five reasons why SE researchers considered GL in their studies, including *seeking more related studies*, *avoiding publication bias*, *comparing different perspectives*, *understanding the views of the practitioner's community*, and *exploring uncharted research areas*. We also proposed a conceptual model to understand how GL works in SE research lifecycle.

With reference to Garousi's guidelines [31], Soldani et al. [15] conducted a systematic GLR on microservices with rapidly evolving state-of-the-practice in industry. They stated that the efforts on microservices are still at an early stage in academia. They tried to bridge the gap between industry and academia by analysing 51 grey (industrial) studies published from 2014 (when Lewis and Fowler enumerates the characteristics of microservice architecture) through 2017 (when their review was carried out). Garousi et al. [18] further proposed guidelines for conducting Multivocal Literature Reviews (MLR) in SE by including GL and consulting the existing SLR guidelines, MLR guidelines and experience papers from other disciplines.

3 | RESEARCH METHODOLOGY

This section elaborates the research design of this research with its processes depicted in Figure 2. The thin arrowed lines in this figure connect two different steps, while the artefacts produced by these steps are indicated by thick arrowed lines.

Our whole research process contains four main phases. In Phase 1, through an ongoing survey of the status quo of the use of DevOps in the Chinese industry, we have learnt that Chinese practitioners have maintained a high degree of attention to the specific practices of DevOps Security. Therefore, the results of the survey have motivated us to understand the dynamics of international practitioners. In Phase 2, a pre-search was first conducted in the middle of 2019 to get an

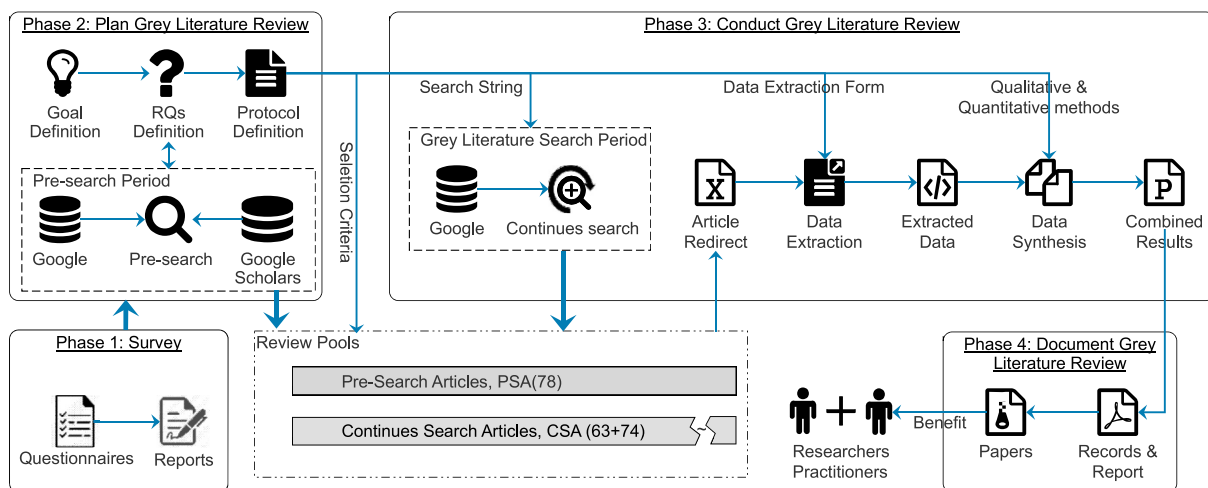


FIGURE 2 Research process of this review.

overview of this emerging new field and formulate our research questions (RQ2-RQ4). We started Phase 3 of this GLR in October 2019 and reported the preliminary findings [19] in SE community. The positive feedback from the community motivated us to expand our review pools in SE. After systematic synthesis of the extracted data, we documented our findings in Phase 4. The following subsections elaborate these phases of the review process. The research team consists of three research students (one PhD and two master students) and their supervisors. All the research students have gained extensive knowledge and experience in Empirical Software Engineering through research training and participation in previous projects.

3.1 | Research questions

This study aims to address the following research questions:

RQ1 What is the current situation of DevOps security in industry of China?

RQ2.1 What are the impacts of DevOps on software security in industry?

RQ2.2 From what aspects do practitioners understand DevSecOps?

RQ2.3 Which practices in industry are associated with DevSecOps?

RQ1 is designed to seek more detailed information specific to the understanding and the use of DevSecOps in Chinese enterprise segments. RQ2.1 is designed to examine how software security is affected by adopting DevOps in industry. RQ2.2 aims to explore the practitioners' understanding on the conception of DevSecOps. RQ2.3 steers our investigation towards the practices that support DevSecOps.

3.2 | Opinion survey

In **Phase 1**, we seek more detailed information specific to the understanding and the use of DevSecOps in industry of China by means of questionnaires to both the DevOps practitioners and SE experts on the Global Conference of Software Development (NJSD)². NJSD conference participants belonged to organisations from multinational companies and some local Chinese companies. Since 2016, we have carried out the annual survey of 'DevOps' almost every year and reported it on the NJSD.

3.2.1 | Target invitees and sampling frame

The survey instrument was designed from the respondent's perspective, which requires a clear identification of the

population and target invitees [33]. We hence chose the purposive sampling frame in this case.

In recent years, we have insisted on publishing questionnaires online, spreading through the DevOps community and InfoQ China³ and other technical communities in SE. In addition, we are still publicising the questionnaires at conferences such as NJSD to attract more developers to participate in the survey. In order to collect as much data as possible, in the process of distributing the questionnaire, we also sent the questionnaire to relevant practitioners who have participated in or downloaded the report we released, because they have a higher probability of participating investigation. Taking into account that the interviewed groups are all high-income groups with a strong interest in new technologies. In the way of motivating participation in the questionnaire, we use the respondent's ability to obtain the survey report in the first time as an incentive. This incentive method can also better exclude developers who have no interest and experience in DevOps, and the collected data can be guaranteed to be of higher quality.

Respondents in this survey cover a wide range of industries, positions, departments, scale of organisations, DevOps experience, and cloud native usage. According to the questionnaire distribution form, it will be conducted through technical communities and related technical conferences. Distributed, the quality of participants is high (excluding invalid questionnaires), and it is representative.

3.2.2 | Questionnaire design

We designed questionnaires to collect data (shown in Table 1). In the questionnaire, we designed 12 questions from basic information, organisational information, performance, organisational culture, security practices, and security tools.

3.2.3 | Questionnaire administration

For respondents who had participated in our survey and experts in the field of security, we distributed the invitations to participate in the survey via emails, which gave the reply period of 4 weeks for each sector of invitees. The invitees with no reply received were gently reminded of the survey 1 week after the first round invitations. In distributing the invitations to their survey, once any reply from them was received (either a decline to reply or an automatic reply for unavailability), we immediately stop sending the next round of email (reminder) to them.

3.2.4 | Data synthesis and analysis

In this survey, quantitative statistics is the most basic and commonly used method. In order to better measure the impact of DevSecOps on an organisation's production activities, this

²<http://www.njsd-china.org>.

³<https://www.infoq.com>.

TABLE 1 DevSecOps questionnaire.

No.	Question	Choice options
1	How much time do you have in DevOps?	<ul style="list-style-type: none"> A. Less than 6 months. B. 6–12 months. C. 1–2 years. D. 3–5 years. E. More than 5 years.
2	Which of the following titles matches you best?	<ul style="list-style-type: none"> A. Development Engineer. B. Testing Engineer. C. Architect Engineer. D. Automation or Tools Engineer. E. Devops Cheng Shi. F. SRE (Station Reliability Engineer). G. Build or Release Engineer. H. System Engineer. I. Project Manager. J. Technical Director. K. Network Engineer. L. Product Manager. M. Operations Engineer. N. Security Engineer. O. Senior Management. P. Others.
3	What kind of industry does your organisation belong to?	<ul style="list-style-type: none"> A. Technology. B. Internet. C. Banking or Finance. D. Education. E. Communication. F. Consultant. G. Entertainment or Media. H. Government. I. Healthcare. J. Retail. K. Others.
4	Which of the following security practices does your organisation implement?	<ul style="list-style-type: none"> A. Security and development teams collaborate on threat models. B. Security tools are integrated into the development integration pipeline, so engineers can be confident that they will not inadvertently introduce known security issues into their code base. C. Security requirements, both functional and non-functional, are prioritised as part of the product backlog. D. Security experts evaluate automated testing and ask them to check for changes in high-risk areas of code (e.g. authentication systems, cryptography etc.). E. Before deploying, check the security policies related to the infrastructure. F. Security personnel review and approve major code changes prior to deployment.
5	Which of the following accord with your organisational structure?	<ul style="list-style-type: none"> A. Centralised security capabilities, support delivery teams on demand. B. Centralised security capabilities, with designated security experts for each delivery team. C. Decentralised security capabilities, each delivery team has a security expert. D. Other organisational structures.
6	Which of the following phases of the software development cycle does your organisation involve in security practices?	<ul style="list-style-type: none"> A. Requirements phase. B. Design phase. C. Build phase. D. Test phase. E. Deployment phase.
7	Does your organisation have a professional security team?	<ul style="list-style-type: none"> A. No, there is no professional security team. B. No, there is no professional security team, but there are security management posts and personnel. C. Yes, there is a special security management team and security supervisor.

(Continues)

TABLE 1 (Continued)

No.	Question	Choice options
		D. Yes, there is a high-level security management organisation and a team of security experts with perfect skills. E. Yes, the team of security experts has made outstanding contributions to the industry. F. I don't know.
8	For DevSecOps, which of the following aspects do your organisation pay more attention to?	A. Whether the design comply with security standards and specifications. B. Code security. C. Security of the third party open source library. D. Automated configuration security. E. Efficiency of automated security tools. F. Security of cloud platform.
9	Which of the following phases does your organisation add automated security testing?	A. Application architecture design phase B. Code development phase C. QA/test phase. D. Deployment phase E. Pre-production phase F. Production phase. G. Continuous delivery of the whole process F. Others (please add) _____.
10	Which of the following is more in line with your organisation's security management during software delivery?	A. No security management. B. The security management of source code and dependent components is carried out, and the security management is included in the test. C. Perfect security scanning and testing toolchain, pipeline integration, automatic security testing. D. All participants are responsible for security. E. With intelligent security delivery (code scanning, testing etc.) platform. F. I don't know. G. Others (please add) _____.
11	Which of the following is more suitable for the security management of your organisation in the process of software operation?	A. No Security management. B. Security Monitoring covers some business scenarios, and periodic security reporting is carried out. C. Automatic security monitoring covers the whole business, the security process is perfect, and the security problems are continuously summarised and analysed. D. Automatic security monitoring covers the whole business and infrastructure, and some security events can be intelligent early warning and self-healing. E. With an intelligent comprehensive security monitoring system. F. I don't know. G. Others (please add) _____.
12	What security tools are being used in your organisation?	Your answer: _____.

survey selected questions about the organisation's "security content, security practices, automation and security tools".

3.3 | Literature review

3.3.1 | Search process

Developing a protocol is an important activity in a review process as it gives details of the plan for the review [34], including process to be followed and allocation of reviewers to particular activities. In **Phase 2**, we first held a meeting to develop an initial protocol, and then conducted a pilot study to refine the protocol until the final protocol was reached. We conducted pre-search to unified the knowledge of DevSecOps obtained from GL and

scoped our final research questions based on these knowledge. A pre-search was conducted via Google and Google Scholar with the following string to retrieve online articles relevant to DevSecOps as many as possible.

DevSecOps OR SecDevOps OR DevOpsSec OR
(DevOps AND Security) OR "Continuous
Security"

Each research student quickly scanned the retrieved articles and regular meetings were held to discuss the results and manage discrepancies. In the end of this phase, the state of research and industrial adoptions of DevSecOps emerged. According to our established understanding, we defined the research questions and developed the review protocols.

After the protocol was finalised, we executed the search strategy defined in the protocol to identify relevant literature both in industry and in academia with the identical search string.

For searching grey literature in **Phase 3**, we set a continued search period in order to collect as many relevant articles as possible using the identical search string. We noticed that some of the highly relevant results retrieved in Phase 2 were no longer available despite that their URLs still exist. To synthesise as much evidence as possible for the research questions, we build a final GL review pool by preserving all the relevant articles retrieved in both pre-search and continued search in PDF format. This pool consists of 78 articles from pre-search and 137 (63 + 74) articles from continued search after applying inclusion and exclusion criteria listed in Table 2.

3.3.2 | Study selection

All the retrieved articles were divided into three groups, and each of the three student researchers independently reviewed two different groups, which ensures that each article was reviewed by at least two different researchers. Any disagreement was discussed in routine group meetings involving the student researchers. The outstanding issues would be escalated to their supervisors for final decision.

For the 174 grey articles collected in the pre-search of Phase 2 (in the middle of 2019), 78 were selected after applying the inclusion and exclusion criteria for grey literature defined in Table 2. For the 115 more articles collected in Phase 3 (in October 2019), 63 were selected after applying the same selection criteria. Then for the 119 supplementary articles collected in January 2020, 74 were selected. In total, 215 articles⁴ were retained for data extraction and synthesis.

Compared to academic literature, GL may display some distinct characteristics. One is that a grey article may contain product marketing related material. Such an article typically makes an objective statement of facts or describes a particular problem of interest, followed by recommended products from either the vendor itself or external organisations. To obtain as much evidence as possible in this review, we thoroughly read the full text of such an article and selected it for inclusion if its statement is neutral (not biased towards the recommended products) and the product recommendation portion is less than half of the article. Otherwise, the article would be considered as a *product teaser* and excluded from the review. For example, the majority of S [13] discusses the necessity of embedding security in DevOps and how to achieve security at speed, followed by promoting their own product at the end. Accordingly, this article is selected as evidence into this review.

Another characteristic is that the link to a grey article may be randomly redirected to another URL whenever it is accessed. To avoid infinite snowballing, we only included the very first GL page in the review and ignore all subsequences.

3.3.3 | Data extraction

The data extracted from grey literature are basically similar, and here we only describe the process of extracting data from GL. Before the extraction, we redirected articles that clearly stated they were reproduced from other sources, which means that the extracted data is drawn from the original sources rather than from the reprinted ones. Table 3 lists the data extraction items that pertain to the research questions in addition to citation information. The three student researchers independently read the full text of the articles assigned and extracted the required data items. Any discrepancy in the extracted data was discussed in routine meetings involving the student researchers. The controversial questions were presented to their supervisors who made expert decisions. All extracted data was later cross-checked by all the researchers involved after the extraction was done.

3.3.4 | Data synthesis

Both *quantitative* and *qualitative* methods were used to synthesise the extracted data in order to answer the research questions. We applied *thematic analysis* in combination with *narrative summaries*. *Coding* was carried out for *thematic*

TABLE 2 Inclusion and exclusion criteria for grey literature.

Inclusion criteria	
IN1	Written in English
IN2	Article content related to DevSecOps
IN3	The full text of the article is accessible
Exclusion criteria	
EX1	Advertisement of vendor product or job recruitment
EX2	Books without full text access and videos
EX3	Product marketing articles
EX4	New pages randomly redirected to the original literature
EX5	Duplicated content

TABLE 3 Data extraction items.

Data item	RQ to be answered
Title	-
Year	-
The organisation where the website belong	-
Terms related to 'DevSecOps'	-
How DevOps affects software security?	RQ2
The definitions of DevSecOps	RQ3
The principles of DevSecOps	RQ3
The characteristics of DevSecOps	RQ3
Practices	RQ4

⁴The list of all the selected articles and their URLs are available at <https://figshare.com/s/c90cd0c94c6b14cc6a15>.

analysis in our review. Whilst *statistical analysis*, shown in the figures and tables, was used for illustrating the distributions of the selected articles, *descriptive statistics* was used to illustrate practitioners' views from different perspectives of DevSecOps.

Table 4 shows an example of coding for thematic analysis, where the three data items were initially coded with the labels of 'Challenges from the speed of DevOps', 'Challenges from the agility in DevOps' and 'Ignore security for the sake of speed/agility' respectively. After the thorough investigation and discussion within the research team, these three codes were finally consolidated into one label, that is, 'Sacrifice security for speed/agility'.

4 | RESULTS FROM SURVEY (RQ1)

DevSecOps integrates security into each stage of DevOps. The development, security, and operation departments work closely together, emphasising that under the premise of controllable security risks, it helps companies improve IT efficiency and better realise DevSecOps.

This survey also set relevant questions for the security in DevOps. From the reply, it is found that the security issues concerned by different industries are different (as shown in Figure 3). Among them, the entertainment industry has the largest difference, and its concern for the security of third-party open source libraries is only 6.70%. The Internet and technology industries pay more attention to code security, but the overall attention to all aspects of security is relatively average.

We have received 239 effective feedback questionnaires within 45 days after the questionnaire was released. **Among these replies, over 40% of Chinese enterprise segments have introduced DevSecOps.** The survey results show that 41.4% (99/239) of Chinese enterprise segments have introduced DevSecOps; at the same time, 58.6% (140/239) of Chinese enterprise segments have not introduced DevSecOps. **More than 40% of Chinese enterprise segments have professional security teams, and security investment has been valued by Chinese enterprise segments and developed rapidly.** As shown in Figure 4, the survey results show that 28.9% (69/239) of companies don't have a professional security team; 28.5% (68/239) of companies do not have a professional security team, but have security management posts and personnel. At the same time, 29.7% (71/239) of

Chinese enterprise segments have a special security management team and security supervisor; 9.6% (23/239) of Chinese enterprise segments have a high-level security management organisation and a well-skilled team of security experts; only 2.9% (7/239) of Chinese enterprise segments have a team of security experts, and have a good knowledge of the industry outstanding contribution.

Automated security testing gradually covers the entire process, which can help companies find and solve security problems as early as possible (as show in Figure 5). The survey results show that 27.2% (65/239) of companies can introduce automated security testing in the application architecture design phase; 40.6% (97/239) of companies have added

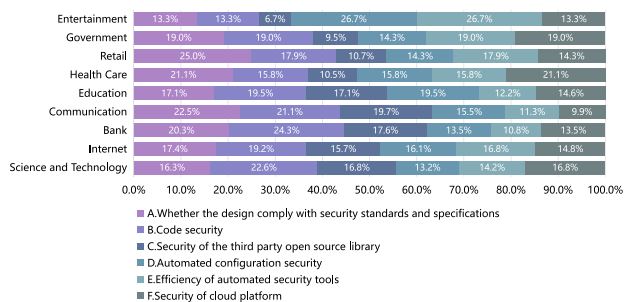


FIGURE 3 Distribution of security issues concerned by different industries.

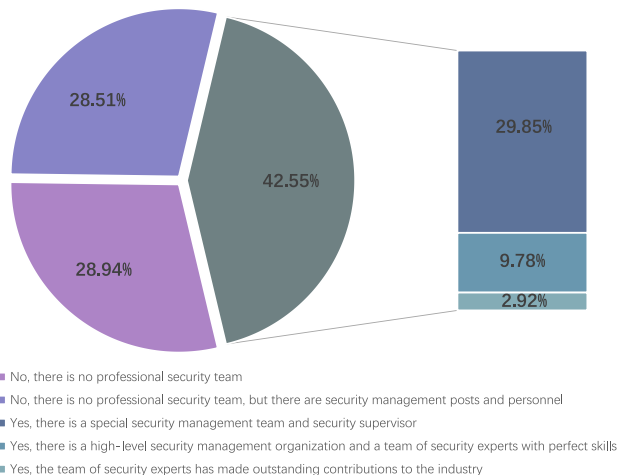


FIGURE 4 Current situation of professional security team.

TABLE 4 An example of coding for thematic analysis.

Data item	Initial code	New code
'DevOps pushes and modifies batches of code over very short time frames, which may far outpace the speed at which security teams can keep up with code review'	Challenges from the speed of DevOps	Sacrifice security for speed/agility
'How do you fit in security while staying agile in DevOps'	Challenges from the agility in DevOps	
'In their quest for speed, DevOps professionals potentially taking shortcuts that are leaving their systems open to exploitation'	Ignore security for the sake of speed	

automated security testing in the code development phase; 47.7% (114/239) of companies have introduced automated security testing in the QA/testing phase; other additions The stages of automated security testing are deployment (41.4%, 99/239), pre-production (34.3%, 82/239), production (38.5%, 92/239) and the whole process of continuous delivery (28.5%, 68/239).

The security management of the software delivery process covers source code and dependent components, and the assembly line generally integrates automated security testing. The survey results show that 19.2% (46/239) of companies have no security management in the software delivery process; 29.4% (70/239) of companies conduct security management on source code and dependent components, and include security management in testing; 23.4% (56/239) of companies have complete security scanning and testing tools Chain and assembly lines integrate automated security testing; 15.9% (38/239) of companies incorporate security management into the entire process of R&D and delivery, and

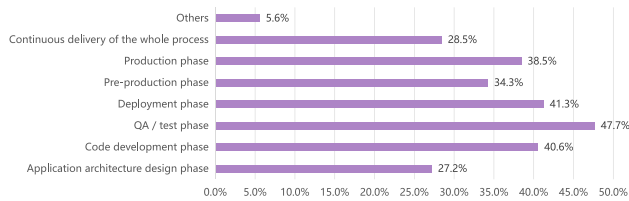


FIGURE 5 Automated security testing phase.

all participants are responsible for security; 12.1% (29/239) of companies have an intelligent security delivery (code scanning, testing etc.) platform.

Security operations are developing towards normalisation, and the coverage is extended to business scenarios and infrastructure. The survey results show that 18.8% (45/239) of companies have no security management in the software operation process; 34.3% (82/239) of companies' security monitoring covers some business scenarios and will periodically report on security; 21.0% (50/239) of companies' automated security monitoring covers the entire business, and the security processing process is complete, Security issues will continue to be summarised and analysed; 18.4% (44/239) of Chinese enterprise segments' automated security monitoring covers the entire business and infrastructure, and some security incidents can be intelligently warned and self-healed; 7.5% (18/239) of Chinese enterprise segments have an intelligent comprehensive security monitoring system.

The application of security tools is diversified, and the penetration rate of containers and network security-related tools needs to be improved. In Figure 6, the survey shows that the host security tool NSFOCUS, the web security tool AppScan and the code security tool Fortify are the three most widely used security tools for Chinese enterprise segments, accounting for 22.2% (53/239), 21.8% (52/239), and 19.7% (47/239) respectively. Other security tools selected by more than 15% are the code security tool Coverity (17.2%, 41/239) and the host security tool Nmap (16.3%, 39/239).

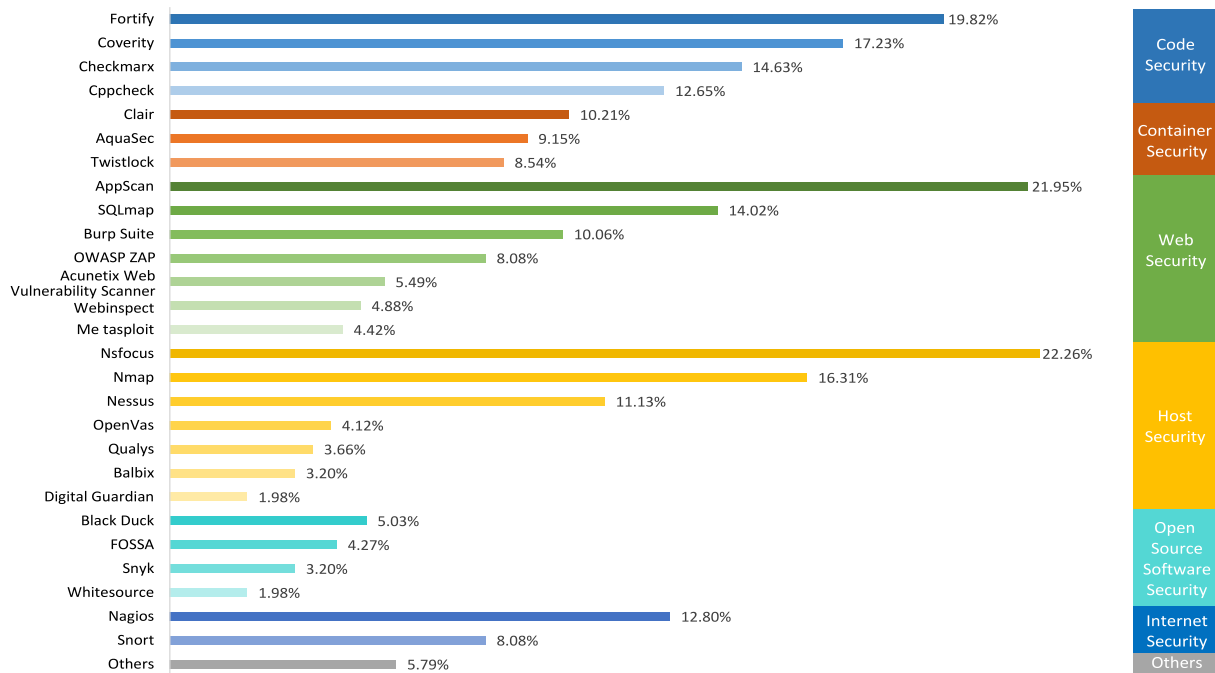


FIGURE 6 Current status of the use of security tools.

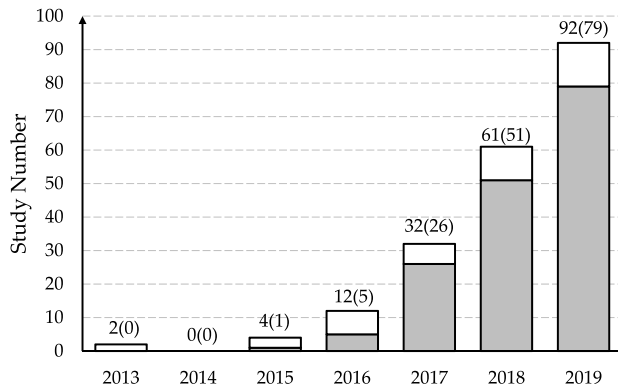


FIGURE 7 Distribution of grey literature over years.

Finding 1: What Chinese enterprise segments are considering is no longer ‘whether they need to embrace DevSecOps’, but ‘how to do a good job of DevSecOps implementation practice’.

Finding 2: Along with the increasing improvement of DevSecOps strategic framework, the construction of related industries of China has also been carried out rapidly, and the practice effect of head industries such as finance, operators, communication and Internet is also gradually improved.

Finding 3: Chinese enterprises segments have made progress in project management, tool chain usage and construction, security protection etc. ‘Continuous automation testing’ is an important focus of DevSecOps.

The results of the survey urge us to further understand the current status of DevSecOps in the industry through literature review. In addition, in the future, we can continue to iterate the questionnaire regularly and conduct surveys on practitioners/experts in the industry.

5 | RESULTS AND SYNTHESIS OF GREY LITERATURE

This section first presents the demographic results of GL on DevSecOps and then discusses the answers to the research questions by synthesising the evidence.

5.1 | Study statistics

Figure 7 shows the distribution of the 215 grey articles published between 2013 and 2019, where the numbers in brackets and the shaded portion indicate the articles mentioning the term of

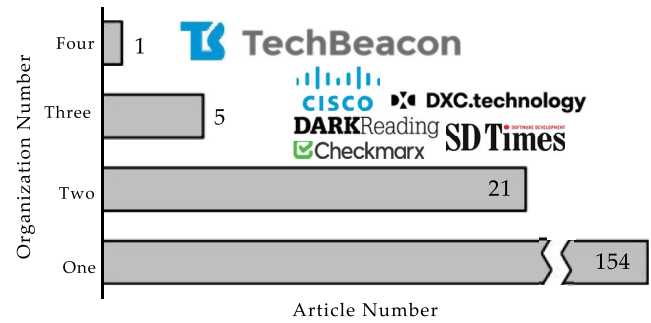


FIGURE 8 Partial distribution of organisations where the articles were published.

‘DevSecOps’. It is worth noting that among the 215 articles, 12 did not record the exact dates of publication, including 2 articles that never mention ‘DevSecOps’. Figure 7 shows a clear increasing trend of articles on DevSecOps with an exponential growth in recent years, indicating practitioners’ growing interest in adopting DevOps in practice recently. From 2016 to 2018, the annual growth of GL on DevSecOps holds at approximately 20–30 more articles, while the number peaked at 92 in 2019 most recently. In particular, the vast majority of the GL (72.5%) was published in the period between 2018 and 2019. It is worth noting that the number of articles explicitly mentioning DevSecOps follows a similar trend, which confirms that the concept of DevSecOps has been increasingly accepted by practitioners since its inception in 2012 [10]. For the articles that do not explicitly mention the term of ‘DevSecOps’, they are still relevant to this review and included in further analysis and discussion as they pertain to a similar concept, such as ‘DevOpsSec’, ‘Sec-DevOps’, or the like.

We further analysed the distribution of organisations where the articles were published in order to reveal their varying degrees of interest in DevSecOps. As shown in Figure 8, the selected articles are posted in 181 different organisations, where TechBeacon⁵ contributes the most with four articles. CISCO⁶, DXC technology⁷, Dark Reading⁸, SDTimes⁹ and Checkmarx¹⁰ were each associated with three selected articles. The other 21 organisations are each associated with two selected articles. Each of the remaining articles is only associated with one of 154 organisations.

5.2 | Impacts of DevOps on software security (RQ2.1)

We synthesised the impacts of DevOps on software security from the two perspectives of *security risks* and *security opportunities*. Security risks in DevOps can be further classified

⁵Homepage available at <https://www.techbeacon.com/>.

⁶Homepage available at <https://www.cisco.com/>.

⁷Homepage available at <https://www.dxc.technology/>.

⁸Homepage available at <https://www.darkreading.com/>.

⁹Homepage available at <https://sdtimes.com/>.

¹⁰Homepage available at <https://www.checkmarx.com/>.

into three categories through thematic synthesis, including *sacrifice security for speed/agility*, *afterthought in the process*, and *environment risks*. Among the selected 215 articles, 142 (64.0%) are relevant to the impacts of DevOps on software security and the vast majority are concerned with security risks (as shown in Figure 9). It should be noted that one article may describe multiple impacts, hence the total number of the articles in Figure 9 is more than 142.

5.2.1 | Security risks in DevOps

Sacrifice security for speed/agility

While many organisations have embraced the integrated approach to development and operations, they are often slow to include security within the DevOps framework. DevOps' focus on speed/agility through cloud deployment, rapid application development, frequently changing application features and configurations, and speed-prioritised and varying workloads, often leaves security teams flat-footed and reactive. More than one hundred of the selected articles point out that DevOps practitioners degrade the priority of security since they regard security as the biggest hurdle to rapid application development considering traditional security methods do not fit the DevOps pipeline and are an inhibitor to DevOps agility. For instance, (S10) illustrates DevSecOps community survey' result to present the phenomenon that 'security is an inhibitor to DevOps agility'. It is highlighted in 20 articles that some developers might simply use the same security credentials for multiple assets simply for the sake of convenience, introducing more risks to data protection. Particularly, (S138) points out that developers are unlikely to keep tabs on different credentials.

Afterthought in the process

The data sources show that security experts typically conduct tests at the end of the software development lifecycle, leading to a situation where the security team works out of the DevOps paradigm. Many companies find that increasing the rate at which new iterations are released may cause teams to bypass certain information security efforts. However, in a world where code changes frequently, attack surfaces and risk

profiles can also change quickly, making security a critical concern for DevOps initiatives. The development team rarely have enough time to address all the security issues before the product goes online, which means that there are potential security risks on the Internet. All issues associated with a team's structural division increase the development cycle time, delay delivery of valuable functionality or corrections, reduce collaboration, and increase frustration and lack of trust within the team. Therefore, a systematic approach is required to improve the whole organisation's culture and structure, fostering collaborative actions, in particular between security and other divisions.

Environment risks

The affinity for DevOps teams to take to the cloud, however, creates new complications for security teams because conventional security measures mostly pertain to on-premise infrastructure. In addition, with the deployment of containers and microservices in cloud, DevOps teams also need to take security considerations of these techniques into account.

5.2.2 | Security opportunities in DevOps

DevOps advocates that organisations build a centralised and standardised delivery pipeline, which can help the security team grasp what is being built so that they can take every opportunity to inject various kinds of security activities into the pipeline. As discussed in (S3, S65), DevOps' high speed is achieved through a controlled and structured environment, instead of cutting corners and skipping important steps. Many of the practices coming with DevOps, such as automation, emphasis on testing, short feedback loops, improved visibility, collaboration, consistent release practices and more, are a fertile ground for integrating security and audit capability as a built-in component of a DevOps process.

5.3 | Practitioners' understanding of DevSecOps (RQ.2)

Although there is a general consensus that DevSecOps is an extension of DevOps [35], no commonly accepted definition of DevSecOps has been formulated in both academia and industry. Practitioners may possess different understanding of DevSecOps that depends on their own professions, their DevOps practice maturity levels, and the purposes of their articles. Our discussion adopts the three core aspects of DevOps, namely *engineering capabilities*, *cultural enablers* and *technological enablers*, identified by Smeds et al. [36], to look into the understanding of DevSecOps. Capabilities represent the processes that an organisation is able to execute, while the enablers allow a fluent, flexible, and efficient way of working. Figure 10 categorise the 121 identified articles that provide detailed information about their understanding of DevSecOps [36].

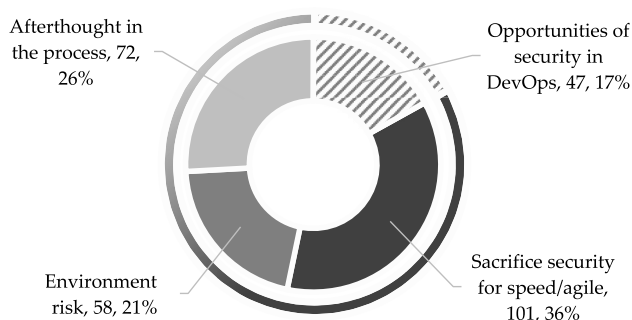


FIGURE 9 Impacts of DevOps on software security.

5.3.1 | DevSecOps capabilities

The capabilities of DevSecOps include *shift security to left* and *continuous security*. Among the 68 articles relevant to the capabilities, 47 of them focus on *shift security to left* and the remaining 21 focus on *continuous security*. Shift security to left is not only about introducing security activities into the early phase of development, but also about integrating security into the entire DevOps lifecycle. Furthermore, continuous security is more about continuous learning and continuous improvement of projects and security delivery.

5.3.2 | Cultural enablers

The cultural enablers list the traits that a DevSecOps team should exhibit. In the 121 articles, 60 (49.6%) highlight the importance of sharing responsibility, which means everyone in the value chain should be responsible for the security of the end product. This shift of mindset makes the development and operations teams to take some of the load off security and have a deeper understanding of how each discipline functions. The improvement of communication is also emphasised in 18 articles as a smooth communication through the project cycle facilitates cross-departmental collaboration, which supports the creation of a security-aware culture as a focus area of DevSecOps and raise people's concerns on security spontaneously.

5.3.3 | Technological enablers

The selected GL shows that DevSecOps stresses the need for automating security tasks since most of the practitioners regard it as a technological enabler of DevSecOps. It is widely acknowledged that implementing automated security checks in the DevOps pipeline will substantially reduce the time and

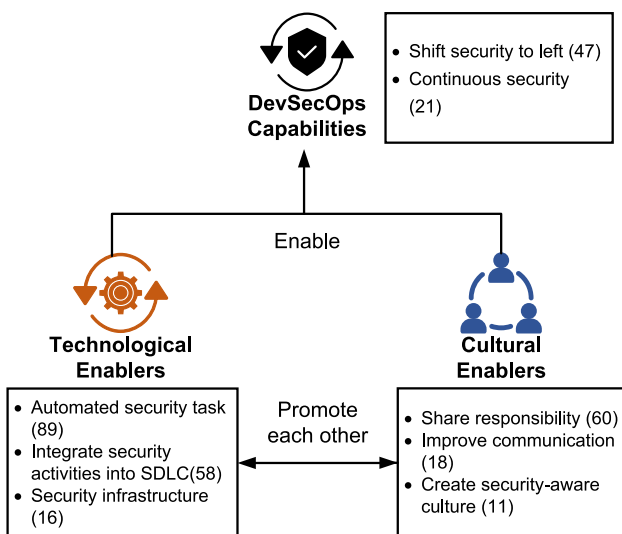


FIGURE 10 DevSecOps capabilities and enablers.

the eventual cost of discovering errors by manual processes. Moreover, automation helps organisation integrate security activities into SDLC (Software Development Life Cycle) without slowing it down and enables developers to improve code security without professional security knowledge. Security also needs to be integrated into the infrastructure since the greater scaled and more dynamic infrastructure enabled by containers have changed the software production environment.

5.4 | Practices associated with DevSecOps (RQ2.3)

While practitioners' understanding reveals the fundamental ideas and values characterising DevSecOps, the practices materialise them into daily software production activities [37]. We summarised the extracted practices from three main perspectives: People, Process and Technology (PPT), referred to by some researchers as the “Golden Triangle” of organisation operation fundamental principles [38]. Widely recognised as the three key elements for process improvement [39], PPT has been applied to the analysis of DevOps [40], cloud security [41], information security [38], and other areas related to DevSecOps.

5.4.1 | People

From the people's perspective, we identified the human factors in DevSecOps and summarised the main practices as shown in Figure 11. Practitioners often discuss the human factors in terms of culture, organisation, collaboration or communication. The main roles involved are DevOps teams and security teams, where the relationship between developers and security teams are most discussed. Specifically, *security training* and *security champion* are the most discussed practices in the people dimension of DevSecOps due to their positive effect on breaking down traditional silos and improving communication.

Almost a third of the selected articles (72 out of 215) mention security training, where the trainees include both

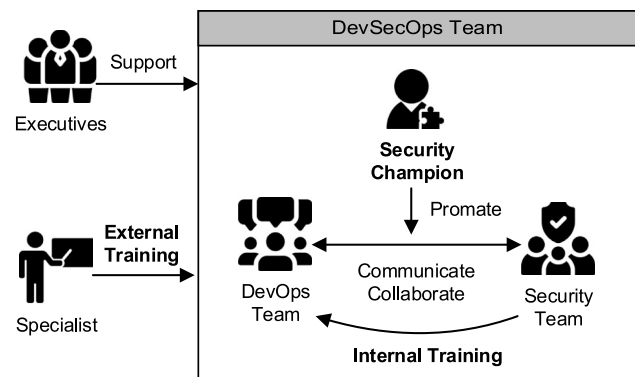


FIGURE 11 People-based DevSecOps practices.

DevOps teams and security teams. For software and IT engineers, security-related training can equip them with the guidelines for setting routines to improve security in the coding phase (S92), while the trainer can be a security specialist from an external DevOps training organisation (S121) or from the internal security teams (S32). The training may cover attack's perspectives, practical hacking exercises, vulnerable applications, and secure coding rules (S15, S146). For security teams, the objective of training is to be imbued with the DevSecOps ethos and to learn about coding and APIs (S44).

Whilst security training is effective for DevOps teams and security teams to share security responsibility and reduce communication barriers, they are still two separated groups with different specialities. To promote the communication and collaboration between them, practitioners recruit or train security champions in their teams. The security champions act as the voice of security for a given product or team, and they assist in the triage of security bugs for their team or area (S121). From the 16 selected articles that introduce the concept of 'security champion', 9 of them support nominating security champions from DevOps teams to become the security conscience of the teams. Moreover, the assignment of security champions is also the first step towards creating a cross-functional team focused on application security and security operations (S146).

Besides the DevSecOps teams, we observe that other stakeholders especially executives also play an important role in DevSecOps. Although only 9 selected articles mention the importance of getting buy-in from stakeholders, it is likely that these stakeholders are the key to a cultural change. The data on the latest security and data breaches and the consequences showing how the involved companies were negatively affected could make a strong case for convincing executives to get on board with a cultural shift from DevOps to DevSecOps (S131).

5.4.2 | Process

From the process' perspective, we gathered the security practices and organised them according to the software

development processes with the continuous delivery workflow [42], which divides the processes in terms of artefacts and environment. Compared to the plan-code-build-test-operation model used in the conference version [19], the model in this article (as shown in Figure 12) is relevant to CI/CD and delineates the boundaries of processes. The workflow is divided into five phases, where **Pre-Commit** involves the activities before the changed code is checked-in to the source repository, followed by **Commit** phase that is triggered to build the changed code, then transfer the binary files to the binary repository and subsequently to an **Acceptance** test environment. Finally the change is deployed to **Production** after passing all the previous phases. We separate the operations and production because the post-deployment is a long-term process which is distinct from the production deployment.

The number attached to every practice in Figure 12 denotes the number of the selected articles mentioning the practice. In general, practitioners are more focused on Pre-Commit and Commit phases, which coincides with the concept of shifting security to left. As the most discussed practice in Pre-Commit phase, threat modelling ensures that security is considered from the beginning of development. It identifies potential threats, estimates possible outcomes, and creates a proactive mitigation strategy resulting in a solid threat model (S24). However, threat modelling needs to be automated or simplified because of its perceived slowness in DevOps (S15). To address security issues during coding, 14 articles suggest including checks for defensive coding and security vulnerabilities during peer code review, which could be a touch point for security teams to collaborate with developers (S215). Another alternative to improve code security is security code scanning with automated static analysis software testing (SAST) tools. We classified SAST into three practices: IDE security checks, static code analysis, and deep SAST scan based on the processes it is associated with. The IDE security checks stress on the code consistency, maintainability, and clarity (S118), and the static code analysis looks for common coding bugs and bug patterns to catch subtle logic mistakes and errors that could lead to runtime failures or security vulnerabilities, while the deep SAST scan identifies security vulnerabilities

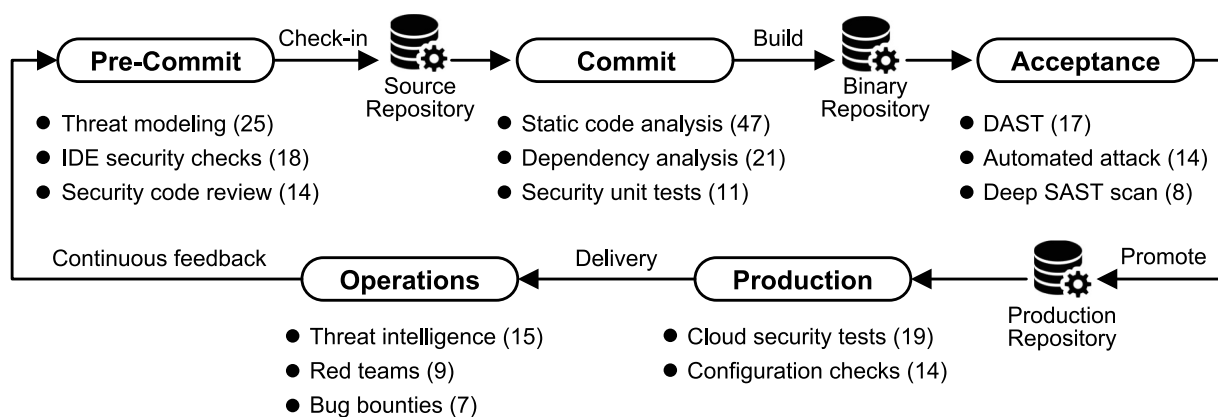


FIGURE 12 Process-based DevSecOps practices.

through taint analysis, control flow, and other techniques. The dependency analysis and security unit tests are also conducted in Commit phase to reduce risks in the early stage.

Following a successful build, the change is deployed to the test environment for acceptance tests. Although more than 10 articles introduce Dynamic Application Security Testing (DAST) and deep SAST scan as DevSecOps practices that can be partly automated, they are still time-consuming. Automated attacks (S53) can simulate attacks on a running application by executing a basic set of targeted automated pen tests against the system as part of the automated test cycle. The security checks and controls in production are mainly cloud security tests and configuration checks, which are automated to ensure the security of production environment. We find that the agility and speed of tests are less discussed after Commit phase, and the automation is emphasised to reduce the manual work.

In Operations phase, practitioners pay more attention to vulnerability discovery and feedback. Red teams and Bug bounties (S118) in this phase can demonstrate what is wrong and provide a solution, creating a positive feedback loop between security teams and the developers. Furthermore, organisations can address security issues using cyber threat intelligence solutions, which collect and process data automatically (S80, S106).

5.4.3 | Technology

From the technology's perspective, we classified the practices into two strategies in terms of their functions: speed up or harden the DevSecOps processes, as illustrated in Figure 13. Technologies enable people to properly execute DevSecOps processes, remove/relieve the need for the manual security activities to increase the delivery velocity and decrease the attack surface to harden the pipeline.

Automation is the most popular strategy in DevSecOps mentioned in the vast majority of the articles (177 out of 215) and also one of the pillars of DevOps [14]. As shown in Figure 13, *Automation* represents a set of automation-related practices in the DevSecOps processes. For instance, automation of recurring

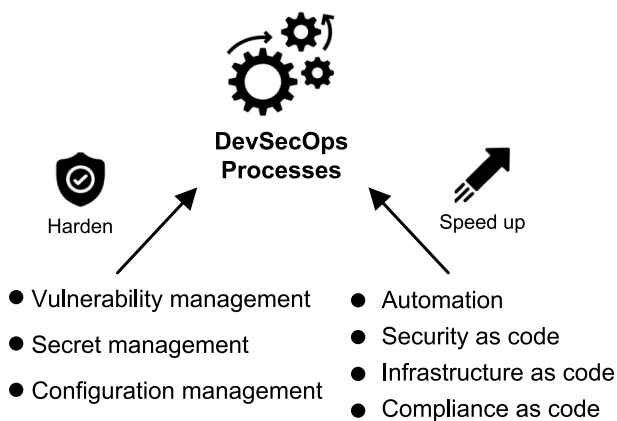


FIGURE 13 Technology-based DevSecOps practices.

security tasks is conducive to deliver security at developer's speed, and some practitioners even intent to automate everything (S51, S103). In addition, security as code, infrastructure as code and compliance as code ensure that any code deployed is secure and compliant, and that any deviation from this can be spotted early and fixed quickly (S121).

While speeding up the security delivery, some technical practices are being applied to address security challenges in the DevOps pipeline. Secrets in an Information Security environment include all the private information a team should know (e.g. a third party API). To establish a trusted connection, credentials, a certificate or an API token are necessary. Even with these precautions, however, handling secrets can be challenging, and can often become a source of error or even a security breach (S1). Secrets management can mitigate the risk of leaked credentials by making sure that accounts have only the privileges they need (S47). The management of configuration helps implement traceability of each code/configuration change (S118), thus making it easier to identify the root cause of an issue and any deviation from immutable artefacts. Besides, vulnerability management continuously collects testing results from the pipelines to assess and remediate vulnerabilities throughout the SDLC (S44) and it is responsible for protecting assets from known exploits and for identifying new threats in software.

6 | DISCUSSION

This section first shows our observation of the current situation of DevSecOps in industry. Then discusses our decision the process of GLR. Finally, we share the challenges to implement DevSecOps.

6.1 | Observation on DevSecOps in industry

6.1.1 | Enterprises accelerate the pace of practice

DevSecOps practice will no longer be limited to embedding security tools into DevOps platforms, but will become an independent integration platform, and DevSecOps will take no-aware security as the core goal in the future, reducing the interference of security tools to enterprise business production and operation as much as possible.

6.1.2 | Practice needs to fit enterprise attributes

For most enterprises, DevSecOps often means drastic changes. Not all A-parties are suitable for direct application of DevSecOps practice process, and the key security activities of different industries are also differentiated, so it is necessary to make further arguments according to their own organisational development goals, cultural characteristics and business scenarios, and gradually figure out the enterprise's own security capability system.

6.1.3 | Security is everyone's responsibility

Only when everyone participates in DevSecOps can security be ensured. In the process of implementing DevSecOps in mature enterprises, it is not difficult to find that the cultivation of enterprise security culture is always listed as the top priority. Security and various teams need to participate in the continuous construction and optimisation of the DevSecOps RD model, and continuously promote the theory and tool chain of DevSecOps forward.

6.2 | The choice of review method

After finding genuine reasons for undertaking a review, researchers need to decide which review method to use. Based on our study on GL and experience on conducting GLR, we find three concerns of GL in this decision process (Figure 14): *possibility of introducing GL to a review and then using it as evidence, as well as the methods of using GL as evidence.*

We first decide whether GL needs to be introduced to reviews. Three reasons, which motivated the inclusion of GL, were discovered in a tertiary study on MLRs in SE [43]. Similarly, our previous work [32], which was based on more reviews and the opinions from SE experts, summarised five reasons for including GL in SE reviews. Due to the papers found in the academic literature retrieval stage can not provide effective evidence to answer our research questions, we consider introducing grey literature to support the research.

With a loose organisational structure, GL may sometimes not be used as evidence for reviews. For example, the contents in Stack Overflow, a community question and answer site used by many SE researchers (e.g. [44–46]), are not suitable for reviews if

the answers are mainly code, or if the answers are generally short. Therefore, we need to assess whether the content in GL is suitable for review when conducting the pilot research. As the results of the pilot study can meet the demand, we decided to use GL for the additional information out of academic literature.

The consistency of the evidence characteristics in GL and academic literature needs special attention for some RQs. In this study, to answer the questions about DevSecOps practice, the contents about practices were extracted and we found that GL has a higher level of abstraction whereas academic literature focuses more on specific technologies (e.g. the study on self-service cybersecurity monitoring [12]). Due to the inconsistency of their evidence characteristics, and the difficulty of the technology mentioned in the academic literature to be associated with DevSecOps practice, we tend to use GLR only to analyse and report.

6.3 | Challenges to implement DevSecOps

Similar to DevOps [47], DevSecOps also faces a number of challenges in order to be successfully implemented by an organisation. However, from the selected studies, only a handful of them (24 out of 215) describe the challenges to implement DevSecOps, which can generally be categorised into *internal* and *external* factors. In particular, culture, cost, and organisational structure are the three main internal factors.

Culture resistance is the most mentioned internal factor, accounting for 11 out of 24 (45.8%) of the relevant articles. DevSecOps aims to remove the barriers between the three different teams and foster an atmosphere of collaboration in an organisation. However, it is hard to achieve these DevSecOps visions in reality due to the distinct core targets across these teams. The development team prefers moving fast and changing frequently, whereas the security team is in favour of stability, which often leads to a conflict between them (S97). It is also a new burden for developers to bear possible security issues in mind when coding, which they rarely used to do before. Their workload will also increase significantly by learning extra security knowledge and skills.

High cost indicates the challenges due to economic reasons, which are mentioned in 6 out of 24 (25.0%) of the relevant articles. Many extra costs are the expense for the organisations who decide to transform DevOps into DevSecOps. The costs involved in changing the established DevOps pipelines are usually unacceptable for small companies. Besides, Bogana Dobran stated that *'The performance and security requirements of legacy resources create complications when folded into DevOps environments'* (S24). When integrating security practices in legacy facilities without security considerations, practitioners have to accept a relatively higher complexity and investment.

Rigorous organisational structure is discussed in five articles. Sarah Vonnegut pointed out that *'if the burden of not correctly securing DevOps environments isn't fully understood by the board, it's impossible to expect the organisational structure to change'* (S131). All the stakeholders need to be

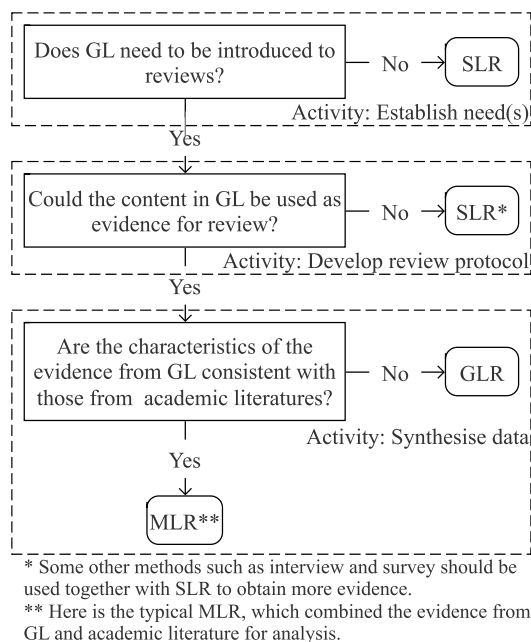


FIGURE 14 The decision process for review methods.

aware of the necessity of DevSecOps in order to support the shift to DevSecOps. DevSecOps promotes ‘*everyone is responsible for security*’, but it is difficult to define everyone's security boundary. Doug Drinkwater stated that ‘*companies need to decide which roles have responsibility for security*’ (S6).

There are several connections among these internal factors. A company's *culture* and *organisational structure* are mutually reinforcing each other. For example, organisational structure can be changed if security champions are appointed, and this kind of role can accelerate the transformation of culture. Establishing what kind of culture and organisational structure should take full account of cost factors as well.

In addition to the internal factors, we further identified three major *external factors*, which are experts, tools, and solutions.

Lack of DevSecOps experts is discussed in nine articles. Cybersecurity Ventures reported that ‘*there would be 3.5 million cybersecurity job openings by 2021*’ (S90). In this context, Chris Carlson, VP of product management at Qualys, pointed out that ‘*the number of security practitioners knowledgeable in DevSecOps is still low*’ (S6).

Lack of DevSecOps tools is explored in eight articles. Although many DevOps tools and security tools have been used in practice, DevSecOps requires new dedicated tools (S6). For example, the lack of visualisation tools makes it challenging to arrive at an informed decision on security in DevOps (S26).

Lack of mature DevSecOps solutions is indicated in five articles. Moving to DevSecOps may take a long time whilst there are still many phases that need to be inspected manually (S103) and for this reason some small companies are reluctant to try DevSecOps (S6).

Several links among external factors can also be identified. *DevSecOps experts* contribute to the integration of different *DevSecOps tools*, and they can help improve the existing solutions to fit their company. The number of DevSecOps experts needed and DevSecOps tools integrated are influenced by the chosen solutions.

Furthermore, there are intricate relationships among these internal and external factors. The *culture* within an organisation is profoundly affected by the employed experts and the adopted solutions. Inherent company culture can also in turn significantly impact the number of hired experts and the adopted solutions. *Cost* interacts with all external factors. Companies tend to consider the cost of their practices at all times. Salaries for security experts, the complexity of integrating security tools, and the loss of using new solutions are all the issues that companies need to consider if there is a limited budget before the official institutionalisation of DevSecOps. *Organisational structure* can be changed if required by the adopted solutions as well. Managers can in turn choose the appropriate solution based on the company's organisational structure.

7 | RELATED WORK

Håvard Myrbakken et al. [35] analysed 52 artefacts returned by Google search engine, which contained two academic research papers and 50 pieces of GL (e.g. white papers, blogs, and articles). In their research, DevSecOps was defined as a necessary

expansion of DevOps aiming at integrating security processes into DevOps life cycle through the collaboration of development, operations and security teams. Several DevSecOps characteristics were generated from the articles and explained from five aspects including culture, automation, measurement, sharing, and shift security to the left. Five practices were discovered from the articles as well: (1) threat modelling and risk assessments, (2) continuous testing, (3) monitoring and logging, (4) security as code, and (5) red-team and security drills. Three benefits including shifting security to the left, automating security, and security value as well as three challenges including keeping up with DevOps, organisational challenges, and tools and practices were identified from the articles.

In [13], by reviewing two selected academic papers as well as 11 grey articles from Google Search, Luis Prates et al. found nine relevant DevSecOps metrics reported by professionals: (1) defect density, (2) defect burn rate, (3) critical risk profiling, (4) top vulnerability types, (5) number of adversaries per application, (6) adversary return rate, (7) point of risk per device, (8) number of continuous delivery cycles per month, and (9) number of issues during red teaming drills. Considering the tendency of DevSecOps and the exploratory nature of their study, they also pointed out the needs for interviews and surveys with DevSecOps professionals.

Table 5 compares this work with the two related work on DevSecOps in terms of topics, research questions, methods, search strings, databases, sample size, and results. Myrbakken's work [35] gave us a glimpse of DevSecOps from four aspects: definition, characteristics, benefits, and challenges. Prates's work [13] further explored the metric-related issues in DevSecOps. In this study, we found in the pre-search that until the middle of 2019 the number of academic papers related to DevSecOps remained low and mostly had close collaboration with industry. As the absence of solid and comprehensive evidence from academic literature to answer our research questions (RQ2-RQ4), we conducted a GLR with a broad search term focussing on how practitioners understand DevSecOps and what types of practices are used in DevSecOps. With much more and up-to-date evidence from the grey literature, our work not only presents a comprehensive vision of the state-of-the-practice of DevSecOps, but also proposes two frameworks, that is, the DevSecOps capabilities and enablers model (Figure 10) and the DevSecOps practices framework (Figure 11), which can be used as important references for implementing DevSecOps.

8 | LIMITATION

We share our lessons in conducting GLR and summarise the limitations of this study.

8.1 | Lessons from exercising the grey literature review process

In this section, we review our experiences with the process of this GLR. When we conducted our review with reference to the guidelines [18], we encountered some difficulties in the

TABLE 5 Secondary studies on DevSecOps.

Article	Håvard Myrbakken et al. [35]	Luís Prates et al. [13]	This study
Topic	General DevSecOps	DevSecOps Metrics	General DevSecOps
RQs	RQ1: How does the literature define DevSecOps? RQ2: What are the characteristics of DevSecOps? RQ3: What are the main expected benefits and challenges of adopting DevSecOps? RQ4: Since it was first mentioned, how has DevSecOps evolved?	RQ: Which are the most relevant DevSecOps metrics?	RQ1: What are the impacts of DevOps on software security? RQ2: From what aspects do practitioners understand DevSecOps? RQ3: Which practices are associated with DevSecOps?
Methods	MLR	MLR	GLR
Search strings (Search terms)	("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND ("definition" OR "characteristics" OR "challenges" OR "benefits" OR "evolution")	DevSecOps, SecDevOps, Definition, Challenges, Metrics, Measuring, Adoption	DevSecOps OR SecDevOps OR DevOpsSec OR (DevOps AND Security) OR "Continuous Security"
Databases	Google Scholar, Google Search	Google Scholar, Google Search, IEEEExplore, ACM Digital Library, SpringerLink	Google Scholar, Google Search, IEEEExplore, ACM Digital Library, SpringerLink, ScienceDirect
Sample size	2 (academic) + 50 (grey - first 5 pages on Google Search)	2 (academic) + 11 (grey)	215 (three rounds of search)
Results	The definition of DevSecOps; Five DevSecOps principles; Five DevSecOps practices; Three benefits gained from DevSecOps and its practices; Three challenges an organisation faces from DevSecOps; The evolutionary tendency of DevSecOps.	Nine relevant DevSecOps metrics.	Study demographics of both academic and grey literature on DevSecOps (the tendency); Impacts of DevOps on software security (identification of security risks and opportunities in DevOps); DevSecOps capabilities model (the practitioners' understanding of DevSecOps); DevSecOps practices framework (three types of practices associated with DevSecOps).

phases of search and source selection. Our strategies may address these issues in the context of this review.

8.1.1 | Lessons learnt in the search phase

We performed a pre-search and a formal search with the same search string with Google to collect GL. Although our intention for pre-search was to understand the problem and further refine our research questions, we found that some high-quality sources were no longer available in the results of the later formal search, which was attributed to Google search engine. Our solution was to preserve the results of the pre-search and combine the articles from both the search processes. For future research with GL, we suggest routine searches for a given period (days or maybe a week) and integrating all the results as the final literature set to avoid the uncertainty of particular search engine at times.

8.1.2 | Lessons learnt in the source selection phase

As GL is more diverse and less formatted than academic literature, source selection can be particularly time-consuming and difficult. Our major challenge in this phase is that some articles contain promotional material of software products. At

first we decided to exclude all of these articles because we thought that such information may skew the quality and credibility of the articles. Nevertheless, we found some articles with high-quality evidence as well as advertisement. For example, we ever excluded the article (S15) because it recommends a SAST tool developed by the author's company in the end of article, but anything else before that is unbiased. Therefore, we double checked all of this kind of articles and finally decided to include an relevant article if its product teaser is less than a half of the article.

8.2 | Threats to validity

With reference to the SLR guidelines [48, 49] and the guidelines for including GL in SE research [18, 32, 50], we identified potential threats to the validity of our study and took actions to mitigate them.

Construct Validity is concerned with the correct operational measures for the studied concept [49]. Our goal is defined by the research questions, which are answered based on the categorisation schemes emerging from the evidence, which was finalised through iterations with reference to some existing DevOps or security models [36, 51].

The lack of evidence in grey articles is another threat to construct validity. On the one hand, because of the

unstructured GL, they are largely based on authors' experiences and opinions from practice, which could be regarded as evidence. On the other hand, some GL authors might cite other practitioners' statements. A serious limitation in grey articles, which could be hardly fixed, is that their authors rarely mention the sources of evidence or the systems they worked on. Despite this, the evidence in forms of opinions and experiences may also yield factual account and sound impression for the reason that the views of the practitioners, no matter whether repeated from others', can be essentially a status quo of the studied topic. A grey article could be also considered as an unstructured interview or questionnaire for research purpose, which is common in ethnographic research to obtain more real and contextual information [52].

Internal Validity is concerned with the causal relationship whereby certain conditions are believed to lead to other conditions [49]. Considering that inclusion and exclusion criteria may directly affect the quality of our data and further the answers to our research questions, we planned fine-grained procedures in order to secure the internal validity. We first designed a rigorous search strategy and search process, then applied the defined multi-step selection process independently by three researchers. During this process, we also updated the search strategy and selection process with necessary rework in certain scope. During the process, we always maintain uniform data extraction standards. Any disagreement was thoroughly discussed until a consensus reached.

Although GL could help us discover the state-of-the-practice of DevSecOps, it is difficult for researchers to evaluate GL [53, 54]. Our study analysed all the GL retrieved by Google in two search phases with the aim to offer a panoramic view of DevSecOps in industry. While we realise the quality differences among the included GL, there is no defined criteria to properly assess and report them [29, 55], which results in the conclusion of our pilot quality assessment with one fifth of included GL. Hence, we strongly call for the development of evaluation standards to be included in GL guidelines.

In order to ensure the accuracy and truthfulness of the survey participants, we choose to publish the questionnaire in the soft industry technology community and related technical conferences. We use the respondent's ability to obtain the survey report in the first time as an incentive, and this incentive method can also be better exclude developers who have no interest and experience in DevOps, and the collected data can be guaranteed to be of higher quality. We also ensured the authenticity of the questionnaire answers through the following measures: (1) Do a one-tailed test for the answering time of the questionnaire, that is, exclude those whose answering time is too short; (2) Set up a trap question to exclude those who fail the trap question.

External Validity is concerned with the domain generalised by the study's findings [49]. Besides the GL, we also surveyed the practitioners from Chinese or multinational companies and experts from the SE community. This means that our findings have universal significance in industry (especially in industry of China). Moreover, GL articles may

not report the internal gaps in org strategy. Some companies that practice DevSecOps do not have the culture of blogging and will not feature in the survey. It is true that such threats exist, but they do not invalidate our results.

Conclusion Validity is concerned with the repeatable operations such as data collection procedure [49]. We developed the protocol that defines the data extraction strategy to ensure the extracted data from the selected grey articles about DevSecOps. The review protocol was proposed by three student researchers, then reviewed and refined by their advisors and industry experts. The data extraction form was designed in the protocol to secure the consistent extraction of evidence for each research question. Crosscheck was done after the independent data extraction by three student researchers. Any divergence was discussed in regular meetings with all the authors.

9 | CONCLUSIONS

To achieve rapid value delivery, DevOps has been widely adopted in software industry. However, faster development and deployment cycles inevitably lead to more security concerns as the traditional security activities does not keep up with DevOps' pace and agility. DevSecOps was proposed to fill this gap by seamlessly intertwining development, operations, and security teams.

The study reported in this article contributes to the overall understanding of and insights into DevSecOps in practice by reviewing much more comprehensive and up-to-date grey literature than ever.

We articulate the impacts of DevOps on software security from the perspectives of security risks and opportunities. Based on the general consensus reached by practitioners that DevSecOps is an extension of DevOps, we categorise the understanding of DevSecOps in the selected articles from three core aspects of DevOps, that is, DevSecOps capabilities, cultural enablers, and technical enablers. We also summarise the existing typical DevSecOps practices in terms of people, process, and technology. To arouse the discussion on DevSecOps among practitioners and researchers, we also discuss the 'grey area' between DevOps and DevSecOps as well as the challenges of implementing DevSecOps in industry in terms of internal and external factors.

This work aimed at evoking greater enthusiasm for DevSecOps in academia and industry. As the concept emerging from software industry, the research on DevSecOps can offer the opportunity on reciprocal communication and collaboration between academics and industry. On the one hand, Tomas et al. [14] suggested a large-scale empirical study by interviews and surveys to be conducted to learn the state-of-the-practice of DevSecOps in industry. On the other hand, ethnographic methods, which can be used to explore the relationship between human, process, technology and environment [56], should be taken seriously in order to drive the investigation into DevSecOps in reality. We have always done this.

AUTHOR CONTRIBUTIONS

Xin Zhou: Conceptualisation; Methodology; Writing – original draft; Writing – review and editing. **Runfeng Mao:** Conceptualisation; Data curation; Writing – original draft; Writing – review and editing. **He Zhang:** Conceptualisation; Funding acquisition; Methodology. **Qiming Dai:** Formal analysis; Investigation; Writing – original draft. **Huang Huang:** Formal analysis; Investigation; Writing – original draft. **Haifeng Shen:** Resources; Validation; Writing – original draft. **Jingyue Li:** Resources; Validation; Visualisation. **Guoping Rong:** Resources; Validation.

ACKNOWLEDGEMENTS

This work is supported by the Innovation Project of State Key Laboratory for Novel Software Technology (Nanjing University) (KFKT2022A09, ZZKT2022A25), also jointly supported by the National Key Research and Development Program of China (No. 2019YFE0105500) and the Research Council of Norway (No. 309494), the National Natural Science Foundation of China (No. 62072227, No. 62202219), and the Key Research and Development Program of Jiangsu Province (No. BE2021002–2).

CONFLICT OF INTEREST STATEMENT

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 STATEMENT

Data available on request from the authors.

PERMISSION TO REPRODUCE MATERIALS FROM OTHER SOURCES

Upload a copy of our conference paper with the designation ‘Additional File for Review but NOT for publication’, and specify which conference our original article was presented at below: 20th International Conference on Software Quality, Reliability and Security (QRS2020).

ORCID

Xin Zhou  <https://orcid.org/0000-0002-3263-1275>

Huang Huang  <https://orcid.org/0000-0002-1296-4363>

REFERENCES

- Cohen, D., Lindvall, M., Costa, P.: An introduction to agile methods. *Adv. Comput.* 62, 1–66 (2004)
- Schwaber, K., Beedle, M.: *Agile Software Development with Scrum*, vol. 1. Prentice Hall, Upper Saddle River (2002)
- Beck, K.: *Extreme Programming Explained: Embrace Change*. Addison-Wesley (2000)
- Ahmad, M.O., Markkula, J., Oivo, M.: Kanban in software development: a systematic literature review. In: *Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 9–16. IEEE (2013)
- Ebert, C., et al.: Devops. *IEEE Software* 33(3), 94–100 (2016). <https://doi.org/10.1109/ms.2016.68>
- Lwakatare, L.E., et al.: Devops in practice: a multiple case study of five companies. *Inf. Software Technol.* 114, 217–230 (2019). <https://doi.org/10.1016/j.infsof.2019.06.010>
- Lwakatare, L.E., Kuvaja, P., Oivo, M.: Dimensions of devops. In: *Proceedings of the 2015 International Conference on Agile Software Development*, pp. 212–217. Springer (2015)
- Feitelson, D.G., Frachtenberg, E., Beck, K.L.: Development and deployment at Facebook. *IEEE Internet Comput.* 17(4), 8–17 (2013). <https://doi.org/10.1109/mic.2013.25>
- Rahman, A.A.U., Williams, L.: Software security in devops: synthesizing practitioners’ perceptions and practices. In: *Proceedings of the 2016 International Workshop on Continuous Software Evolution and Delivery*, pp. 70–76. IEEE (2016)
- MacDonald, N.: Devops Needs to Become Devopssec (2012). https://blogs.gartner.com/neil_macdonald/2012/01/17/devops-needs-to-become-devopssec/
- MacDonald, N., Head, I.: DevSecOps: How to Seamlessly Integrate Security into DevOps. Technical Report G00315283. Gartner (2016)
- Díaz, J., et al.: Self-service cybersecurity monitoring as enabler for devsecops. *IEEE Access* 7, 100283–100295 (2019). <https://doi.org/10.1109/access.2019.2930000>
- Prates, L., et al.: Devsecops metrics. In: *Information Systems: Research, Development, Applications, Education*, pp. 77–90. Springer (2019)
- Tomas, N., Li, J., Huang, H.: An empirical study on culture, automation, measurement, and sharing of devsecops. In: *Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services*, pp. 404–411. IEEE (2019)
- Soldani, J., Tamburri, D.A., Van Den Heuvel, W.J.: The pains and gains of microservices: a systematic grey literature review. *J. Syst. Software* 146, 215–232 (2018). <https://doi.org/10.1016/j.jss.2018.09.082>
- Banks, M.: Blog posts and tweets: the next frontier for grey literature. In: *Grey Literature in Library and Information Studies*. De Gruyter (2009)
- Salleh, N., Mendes, E., Grundy, J.: Empirical studies of pair programming for cs/se teaching in higher education: a systematic literature review. *IEEE Trans. Software Eng.* 37(4), 509–525 (2010). <https://doi.org/10.1109/tse.2010.59>
- Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Inf. Software Technol.* 106, 101–121 (2019). <https://doi.org/10.1016/j.infsof.2018.09.006>
- Mao, R., et al.: Preliminary findings about devsecops from grey literature. In: *Proceedings of the 20th International Conference on Software Quality, Reliability and Security*, pp. 450–457. IEEE (2020)
- Klijnsma, Y.: Inside the Magecart Breach of British Airways: How 22 Lines of Code Claimed 380,000 Victims (2018). <https://www.riskiq.com/blog/labs/magecart-british-airways-breach/>
- Kraemer, S., Carayon, P., Clem, J.: Human and organizational factors in computer and information security: pathways to vulnerabilities. *Comput. Secur.* 28(7), 509–520 (2009). <https://doi.org/10.1016/j.cose.2009.04.006>
- Mohan, V., Othmane, L.B.: Secdevops: is it a marketing buzzword?—mapping research on security in devops. In: *Proceedings of the 11th International Conference on Availability, Reliability and Security*, pp. 542–547. IEEE (2016)
- Sánchez-Gordón, M., Colomo-Palacios, R.: Security as culture: a systematic literature review of devsecops. In: *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pp. 266–269 (2020)
- Akbar, M.A., et al.: Toward successful devsecops in software development organizations: a decision-making framework. *Inf. Software Technol.* 147, 106894 (2022). <https://doi.org/10.1016/j.infsof.2022.106894>
- Rajapakse, R.N., et al.: Challenges and solutions when adopting devsecops: a systematic review. *Inf. Software Technol.* 141, 106700 (2022). <https://doi.org/10.1016/j.infsof.2021.106700>
- Schöpfel, J.: Observations on the future of grey literature. *Grey J.* 2, 67–76 (2006)
- Glass, R.L.: *Software Creativity 2.0. Developer.* Books* (2006)
- John, N.A.: The social logics of sharing. *Commun. Rev.* 16(3), 113–131 (2013). <https://doi.org/10.1080/10714421.2013.807119>
- Adams, R.J., Smart, P., Huff, A.S.: Shades of grey: guidelines for working with the grey literature in systematic reviews for management and

- organizational studies. *Int. J. Manag. Rev.* 19(4), 432–454 (2017). <https://doi.org/10.1111/ijmr.12102>
30. Kepes, S., et al.: Publication bias in the organizational sciences. *Organ. Res. Methods* 15(4), 624–662 (2012). <https://doi.org/10.1177/1094428112452760>
 31. Garousi, V., Felderer, M., Mäntylä, M.V.: The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, pp. 26. ACM (2016)
 32. Zhang, H., et al.: An evidence-based inquiry into the use of grey literature in software engineering. In: *Proceedings of the 42nd International Conference on Software Engineering*, pp. 1–13. ACM (2020)
 33. Modi, V., et al.: Impact evaluation of smoke-free mass media campaign on knowledge, attitude and behavior of the target audience in India. In: *Proceedings of the 2012 National Conference on Health Communication Marketing and Media* (2012)
 34. Brereton, P., et al.: Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Software* 80(4), 571–583 (2007). <https://doi.org/10.1016/j.jss.2006.07.009>
 35. Myrbakken, H., Colomo-Palacios, R.: Devsecops: a multivocal literature review. In: *Proceedings of the International Conference on Software Process Improvement and Capability Determination*, pp. 17–29. Springer (2017)
 36. Smeds, J., Nybom, K., Porres, I.: Devops: a definition and perceived adoption impediments. In: *Proceedings of the 2015 International Conference on Agile Software Development*, pp. 166–177. Springer (2015)
 37. de França, B.B.N., Jeronimo Junior, H., Travassos, G.H.: Characterizing devops by hearing multiple voices. In: *Proceedings of the 30th Brazilian Symposium on Software Engineering*, pp. 53–62. ACM (2016)
 38. Kao, D.Y.: Performing an apt investigation: using people-process-technology-strategy model in digital triage forensics. In: *2015 IEEE 39th Annual Computer Software and Applications Conference*, pp. 47–52. IEEE (2015)
 39. Prodan, M., Prodan, A., Purcarea, A.A.: Three new dimensions to people, process, technology improvement model. In: *New Contributions in Information Systems and Technologies*, pp. 481–490. Springer (2015)
 40. Gill, A.Q., et al.: Devops for information management systems. *VINE J. Inf. Knowl. Manag. Syst.* 48(1), 122–139 (2018). <https://doi.org/10.1108/vjikms-02-2017-0007>
 41. Ghaffari, F., Gharace, H., Arabsorkhi, A.: Cloud security issues based on people, process and technology model: a survey. In: *2019 5th International Conference on Web Research (ICWR)*, pp. 196–202. IEEE (2019)
 42. Bird, J.: *DevOpsSec: Delivering Secure Software through Continuous Delivery*. O'Reilly Media (2016)
 43. Neto, G.T.G., et al.: Multivocal literature reviews in software engineering: preliminary findings from a tertiary study. In: *Proceedings of the 13th International Symposium on Empirical Software Engineering and Measurement*, pp. 1–6. IEEE (2019)
 44. Zou, J., et al.: Which non-functional requirements do developers focus on? an empirical study on stack overflow using topic analysis. In: *Proceedings of the 12th Working Conference on Mining Software Repositories*, pp. 446–449. IEEE (2015)
 45. Lin, B., Serebrenik, A.: Recognizing gender of stack overflow users. In: *Proceedings of the 13th Working Conference on Mining Software Repositories*, pp. 425–429. ACM (2016)
 46. Low, J.F., Svetinovic, D.: Data analysis of social community reputation: good questions vs. good answers. In: *Proceedings of the 22nd International Conference on Industrial Engineering and Engineering Management*, pp. 1193–1197. IEEE (2015)
 47. Riungu-Kalliosari, L., et al.: Devops adoption benefits and challenges in practice: a case study. In: *International Conference on Product-Focused Software Process Improvement*, pp. 590–597. Springer (2016)
 48. Kitchenham, B., Charters, S.: *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. (version 2.3). Technical Report. Keele University and University of Durham (2007)
 49. Zhou, X., et al.: A map of threats to validity of systematic literature reviews in software engineering. In: *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, pp. 153–160. IEEE (2016)
 50. Mahood, Q., Van Eerd, D., Irvin, E.: Searching for grey literature for systematic reviews: challenges and benefits. *Res. Synth. Methods* 5(3), 221–234 (2014). <https://doi.org/10.1002/jrsm.1106>
 51. Sammy Migueles, J.S., Ware, M.: *Building Security in Maturity Model* (2019). <https://www.bsimm.com/content/dam/bsimm/reports/bsimm10.pdf>
 52. Fetterman, D.M.: *Ethnography: Step-By-Step*, vol. 17. Sage (2019)
 53. Al-Baik, O., Miller, J.: The kanban approach, between agility and lean-ness: a systematic review. *Empir. Software Eng.* 20(6), 1861–1897 (2015). <https://doi.org/10.1007/s10664-014-9340-x>
 54. Yasin, A., Hasnain, M.I.: On the Quality of Grey Literature and its Use in Information Synthesis during Systematic Literature Reviews (2012)
 55. Tyndall, J.: *Aacods Checklist*. Flinders University (2008)
 56. Zhang, H., et al.: Ethnographic research in software engineering: a critical review and checklist. In: *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 659–670. ACM (2019)

How to cite this article: Zhou, X., et al.: Revisit security in the era of DevOps: an evidence-based inquiry into DevSecOps industry. *IET Soft.* 17(4), 435–454 (2023). <https://doi.org/10.1049/sfw2.12132>