Preliminary Findings about DevSecOps from Grey Literature

Runfeng Mao, He Zhang, Qiming Dai, Huang Huang, Guoping Rong, Haifeng Shen, Lianping Chen, Kaixiang Lu
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
Peter Faber Business School, Australian Catholic University, Sydney, Australia
Huawei Technologies (Ireland) Ltd., Dublin, Ireland
Email: runfeng.mao, njudqm, njufrankhuang, luckydeerlkx@gmail.com, hezhang, ronggp@nju.edu.cn
haifeng.shen@acu.edu.au, lianping.chen@outlook.com

***Abstract*—*Context*: Emerging from the agile culture, DevOps particularly emphasizes development and deployment speed to achieve rapid value delivery, which however brings some security risks to the software development process. *DevSecOps* is an extension of DevOps, which is considered as a means to intertwine development, operation and security. Some companies with security concerns begin to take DevSecOps into consideration when it comes to the application of DevOps. *Objective*: The goal of this study is to report the state-of-the-practice of DevSecOps as well as calling for academia to pay more attention to DevSecOps. *Method*: Using Google search engine to collect articles on DevSecOps, we conducted a Grey Literature Review (GLR) on the selected articles. *Results*: Whilst there exists three major software security risks in DevOps, the establishment of DevOps pipeline provides opportunities for software security activities. Based on the preliminary consensus that DevSecOps is an extension of DevOps, it is observed that the interpretations of DevSecOps can be classified into three core aspects, which are: DevSecOps capabilities, cultural enablers, and technological enablers. Furthermore, to materialize the interpretations into daily software production activities, the recommended DevSecOps practices we obtain from Grey Literature (GL) can be categorized in terms of process, infrastructure and collaboration. *Conclusion*: Although DevSecOps is getting increasing attention by industry, it is still in its infancy and needs to be promoted by both academia and industry.**

***Index Terms*—DevSecOps; DevOps; Grey literature review; Empirical software engineering**

I. 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 realize and/or adapt business value [1]. As a result, various agile methodologies like Scrum [2], eXtreme Programming (XP) [3] and KanBan [4] have become pervasive for software development. By seamlessly spreading the agile culture across development and operations and by emphasizing software quality and collaboration between development and operation teams, DevOps [5] emerged as a philosophical shift towards evolving software at a continuous pace and streamlining all parts of the software lifecycle. It is crucial that software teams have ownership and responsibility to deploy software changes in DevOps [6], which allows the software to be delivered quickly [7]. At such a rapid rate of deployment and delivery (e.g. up to 500 times a day in Facebook [8]), the software might not undergo the security reviews [9]. In this context, Gartner fellow MacDonald pointed out that [10]:

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

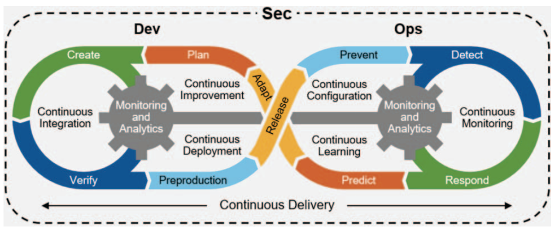

Fig. 1. DevSecOps in Gartner Report [11]

As shown in Figure 1, they described the new and updated services cycle through an iterative DevSecOps process in their following report [11].

Although DevSecOps has gained increasing attention in academia [12]–[14], like many other new topics (e.g., microservices [15]) in SE, industry is actually leading the way. Grey Literature (GL) is mainly produced by industrial practitioners and can serve as an important supplement to academic literature [16]. As there is very little academic literature on DevSecOps and GL is often used for emerging topics [17], to find the state-of-the-practice of DevSecOps, we conducted a Grey Literature Review (GLR) on DevSecOps following the current guidelines [18].

From the 141 identified GL, we analyzed the impacts of DevOps on software security. We identified three major challenges that DevOps brings to software security: 1)sacrifice of security for speed/agility; 2) afterthought in the process; and 3) environment risks. Meanwhile the centralized and standardized DevOps pipeline offers opportunities for software security. Although it is the preliminary consensus that DevSecOps is an extension of DevOps, the understandings of DevSecOps in industry could be categorized from three aspects: 1) DevSecOps Capabilities; 2) Culture Enablers; and 3) Technological enablers. Several typical DevSecOps practices were also discovered and classified in terms of the process, infrastructure and collaboration to materialize the three interpretations of DevSecOps into daily software production activities.

The rest of this paper is organized as follows. Section II introduces the background and related work about DevSecOps. Section III illustrates the research questions, methodology of our empirical study. Section IV presents our findings by analysing our data. Section V discusses results and limitations of our review. Section VI concludes this paper, by discussing findings, implications and directions for future work.

II. Background and Related Work

This section briefs the use of GL in SE, introduces the background of DevSecOps from both the perspectives of DevOps and software security and particularizes the previous empirical studies on DevSecOps.

A. Grey Literature in Software Engineering

Scientific information can be produced and published on platforms where SE practitioners share their experience, such as free online books and blogs [19]–[21].

Recently SE researchers have started paying more attention to GL. Shpilko et al. [22] proposed a model about GL followed Kepes's study [23] that divided the literature into four grey scales according to the difference in scope.

As SE is a practical (practitioner-oriented) field, it is crucial to balance research and practice. Our previous work [24] identified five reasons why SE researchers considered GL in their studies. Garousi et al. [25] also pointed out the significance of GL achieving the research-practice balance in SE research. The Multivocal Literature Reviews (MLR) guidelines [18] was proposed by them through the existing SLR guidelines, MLR guidelines and experience papers in other fields and their experience on conducting MLR as well.

B. Software Security in DevOps

Software security have always been a key concern of enterprises, especially when cyber crime is accelerating. If a company pays no attention to software security, security issues will often bring huge losses. For example, British Airways was attacked due to 22 lines of unsafe code, which leading to personal information leaks for approximately 380,000 customers in 2018 [26]. Kraemer's study [27] shows programmers tend to neglect security issues when they are affected by certain external factors such as time pressure and high workload. When software deployed and delivered in a certain rapid rate by adopting DevOps (e.g. up to 500 times a day in Facebook [8]), developers are more susceptible to these external factors, which may lead to unexpected mistakes. Once the changed software is deployed in production environment without undergoing sufficient security reviews, it is more likely to be of vulnerabilities and have a high risk of being attacked.

Hence, many organizations and practitioners attempt to integrate security into DevOps by adopting protection practices, such as providing security training for developers and adopting some traditional security activities [9]. All of these triggered the coining of a new term DevSecOps [28], which can help the organization in achieving better quality of software by bringing security principles within the DevOps process through all the software life cycle.

C. Previous Empirical Research on DevSecOps

Havard Myrbakken et al. [29] analyzed 52 artifacts, which came from Google and contained two academic research papers as well as fifty pieces of GL (e.g., white papers, blogs and articles). In their research, DevSecOps was defined as a necessary expansion of DevOps, which aimed at integrating security processes into DevOps life cycle by collaborating development, operation and security teams. Several DevSecOps characteristics were generated from the artifacts and explained from five aspects (culture, automation, measurement, sharing and shift security to the left). Five practices were discovered from the artifacts as well: 1) threat modeling 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, organizational challenges and tools & practices were pointed out from the artifacts.

Myrbakken's work gave us a glimpse of DevSecOps from four aspects: 1) definition; 2) characteristics; 3) benefits and 4) challenges. They used these four key words in their search string, which could lead to some GL being ignored. Our work obtained more comprehensive and updated GL, which could show the state-of-the-practice of DevSecOps.

III. Research Methodology

In this section, we elaborate our research design employed in this study with its processes as depicted in Figure 2.

A. Research Question

This study aims to address the following research questions:

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 in GL?

RQ1 is designed to explore how software security would be affected by implementing DevOps. RQ2 aims to present the practitioners' understanding on the concept of DevSecOps and its characteristics in SE. RQ3 steers our investigation of practices in support of DevSecOps in GL.

This study was undertaken from the mid of 2019 and followed the MLR guidelines in SE [18]. And the search team consists of three research students (one PhD candidates and two master by research student) and their supervisors.

B. Grey Literature Review

1) Search strategy: We used Google, which was used in many MLR studies [29], for our GL search. We retrieved all search results and extracted all the evidence we need. Our search string is as follows:

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

We carried out two search stages, a pre-search and a search. The interval between these two rounds of search is about

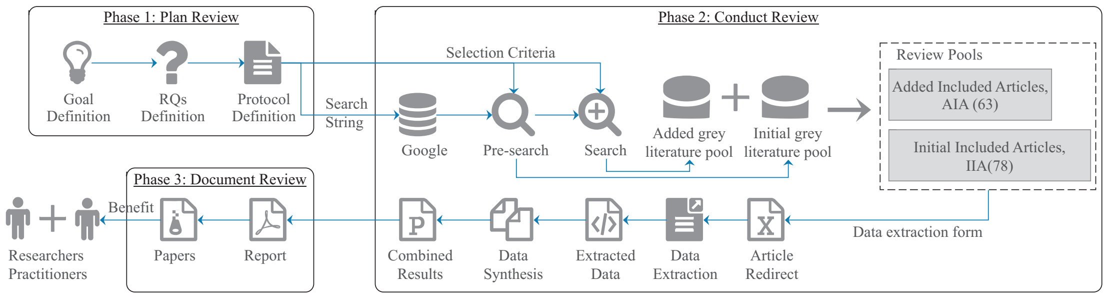

Fig. 2. The research process of this study

several months. The search process was almost the same except for the time difference for both searches. The pre-search was conducted to search all the relevant literature in the mid of 2019. We extracted all the data items listed in Table II from the initial results to depict the initial scope of DevSecOps from the practitioners' perspective and polish our protocol. According to the initial search results, each of us had a better understanding of the state-of-the-practice of DevSecOps, and we found that the initial results could answer most research questions. Hence, we did not change the defined protocol and retained the same search string. The other search was conducted in October 2019, aiming to investigate how much content will be added to the topic of DevSecOps in these several months of interval to reveal the practitioners' focus on DevSecOps and try to capture more latest related GL to strengthen our evidence.

2) Study selection : The articles were divided into three groups on average for review. The three student researchers (reviewers) independently reviewed two groups to ensure that each article would be reviewed by two different researchers.

TABLE I
INCLUSION AND EXCLUSION CRITERIA

	Inclusion criteria
IN-1	Written in English
IN-2	Article content related to DevSecOps
IN-3	The full text of GL is accessible
	Exclusion criteria
EX-1	Vendor product advertisement & job recruitment advertisement
EX-2	Books that we can not get the full text & Videos
EX-3	Product recommendation ratio greater than 50% in GL which are not obvious vendor product advertisement
EX-4	The corresponding new links when we pull down the original GL link
EX-5	Duplicated content

Due to the particularity of GL, there is a certain part of GL that tries to recommend products in the content of the GL, especially on the website of some companies, but these articles are not obvious product advertisements and the part before the recommendation of its product is indeed an objective statement of facts. In order to collect as much evidence as possible, for this part of articles, we set the first appearance of the product name as the beginning of product recommendation. Based on this, we excluded GL with product recommendation content greater than 50% by reading the full text of the article to ensure the objectivity of the evidence we would collect. Besides, some web pages will keep appearing new articles when they are pulled down, but the corresponding links will change accordingly. For this part of articles, we only selected the single first GL corresponding to the link that we had already obtained, and did not consider the new article obtained by pulling down and refreshing. We excluded the GL with exactly the same content as well because authors may publish their own articles on multiple websites.

The selection of GL for the pre-search was performed by carefully checking against the inclusion/exclusion criteria (listed in Table I). Individual selection results were later collectively cross-checked with other's output. Any disagreement was discussed in the meetings scheduled during this study. It was escalated to the research supervisors for advice and final decision if there was a disagreement that could not be solved in the meetings. We collected 174 GL from the pre-search in total, of which 78 were eligible by performing our inclusion and exclusion criteria. After the other search followed by the pre-search, we added 115 new GL into our existing GL pool. The same selection process used for the initial results was also conducted for these 115 GL, and 63 GL was added. We finally identified 141 (78+63) articles [1] for data extraction after the selection process for two different search stages.

TABLE II
DATA EXTRACTION ITEMS

RQ	Data item
-	Title
-	Year
-	The organization where the website belong
-	Terms related to "DevSecOps"
1	How DevOps affects software security?
2	The definitions of DevSecOps
2	The principles of DevSecOps
2	The characteristics of DevSecOps
3	Practices

3) Data extraction:

Before the extraction, we redirected articles that clearly stated that they were reproduced from another source. This means that the data we extracted was drawn from the original source rather than the reprinted one. Data extraction items (shown in Table II) were specified after the identification of research questions. The column "RQ" on the left represents the research questions that are expected to be answered with the data items extracted from GL on the right. The researchers read the full text of GL assigned to them and extracted

[1]The articles and their links are available at http://softeng.nju.edu.cn/tech-reports/TR-20-002-DevSecOps-EN.pdf

the data items independently. Meetings were frequently held to thoroughly discuss disagreements on the extracted items. If any disagreement was not resolved in the meeting, we would immediately consult our supervisors for advice and final decision. The extracted data was also later cross-checked together after we finished our extraction.

C. Data Synthesis and Analysis

We used both *quantitative* and *qualitative* methods to answer the research questions. We applied *thematic analysis* in combination with *narrative summaries* for data synthesis at major. *Coding* was carried out for *thematic analysis* in our study. Whilst *statistical analysis*, shown in the figures and tables, was used for the basic distribution of the selected articles, *descriptive statistics* was used to describe different views of practitioners from some DevSecOps aspects.

IV. Results

This section first describes the overall results form the literature review, and further answers the research questions by analyzing the data collected from the selected GL.

A. Study Demographic

We identified 141 articles published by October, 2019. Figure 3 shows the distribution of these included articles over years, and the shadow areas indicate the articles mentioning the word “DevSecOps”. Besides, there are eight articles without published date.

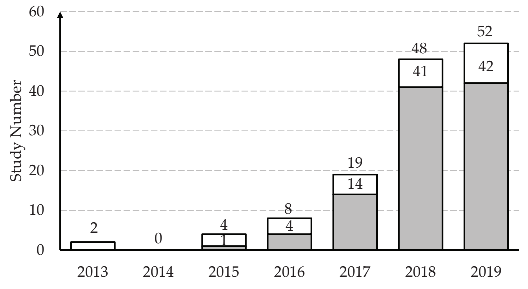

Fig. 3. Distribution of articles over years

From the figure 3, we can find the growing trend of articles about DevSecOps over the years, with exponential growth in recent years. For three consecutive years, the GL on DevSecOps has been roughly twice that of the previous year since 2015. And the number of related articles peaks in 2019 at 52. In particular, most articles (70.9%) were published during 2018 and 2019, which is nearly three times more than that before 2018. This indicates that more and more practitioners have been increasingly realizing the significance of DevSecOps and willing to adopt DevSecOps in practice. Among them, we also marked whether the word “DevSecOps” is mentioned in the article. From the result, we find that the trend of the number of articles mentioning DevSecOps over years is basically consistent with the distribution of the articles we collected. This also reveals the concept of DevSecOps has been increasingly accepted by practitioners since it appeared in 2012 [10].

As for why we still analyzed articles that did not mention the word “DevSecOps”, this was because some similar concepts appeared in these articles, such as DevOpsSec and SecDevOps. We do not intend to distinguish between these concepts, so the following analysis results are uniformly expressed using DevSecOps.

B. Impacts of DevOps on Software Security (RQ1)

Based on the collected evidence, we investigated the impacts of DevOps on software security which constitute the motivation of DevSecOps. We analyzed the impacts from two aspects: risks of security in DevOps and opportunities of security in DevOps. Furthermore, the security risks in DevOps are classified into three categories through thematic synthesis. As a result, we identified that 94 out of 141(66.7%) articles described DevOps’ impacts on software security and the majority of them focused on the security risks in DevOps (as shown in Figure 4). Note that one article may describe multiple impacts.

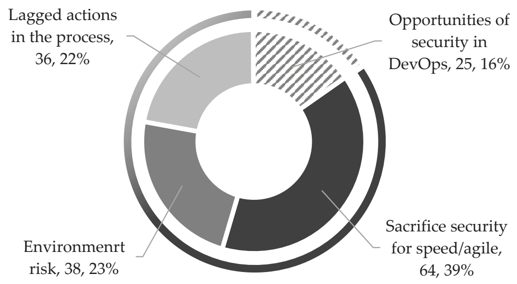

Fig. 4. Articles distribution on the impacts of DevOps on software security

1) Security risks in DevOps:

Sacrifice security for speed/agile. While many organizations have embraced this integrated approach to development and operations, they are often slow to include security within the DevOps framework. DevOps’ focus on speed often leaves security teams flat-footed and reactive. 64 of the selected articles pointed out that the DevOps practitioners reduce the priority of security because traditional security methods did not fit DevOps pipeline and were an inhibitor to DevOps agility. With cloud deployments and application development moving so rapidly, application features evolving daily, configurations changing and workloads shifting, there is no way for manual security process to keep up. Although the developers recognized the importance of security, they regarded security as the biggest hurdle to rapid application development. On the other hand, 10 articles highlighted some developers might use the same security credentials for multiple assets for convenience, adding more risks to data protection.

Afterthought in the process. The data sources pointed out that the security experts typically conducted the tests at the end of the software development lifecycle, which led to a situation where the security team was essentially out of the DevOps paradigm. Many companies have found that increasing the rate at which new iterations are released leads teams to bypass certain information security efforts. However, in a world where code changes frequently, attack surfaces and risk profiles can change just as quickly making security a critical concern for DevOps initiatives. The development team rarely has enough time to address all the issues before the product goes live which means that an insecure application lives somewhere on the internet. All issues associated with team's structural division increases the development cycle time, delaying delivery of valuable functionality or corrections, reducing collaboration, and increasing frustration and lacking of trust among teams. Therefore, a systematic approach is required to improve the whole organization, focusing on collaborative actions.

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 to this, the application of containers and microservices in DevOps make organizations to take the security consideration of these techniques into account.

2) Security opportunities in DevOps: DevOps advocates organizations to build a centralized, standardized delivery pipeline, which helps security team to get visibility into what's being built and gives them opportunities to inject various kinds of security activities into the pipeline. As discussed by Jaikumar Vijayan [S3] and Natasha Gupta [S65], high speed of DevOps is not achieved by cutting corners and skipping important steps, its environment is controlled and structured. Many of the practices that come with DevOps, such as automation, emphasis on testing, fast feedback loops, improved visibility, collaboration, consistent release practices, and more, are fertile ground for integrating security and audit capability as a built-in component of DevOps processes.

C. SE Practitioners' Understandings of DevSecOps (RQ2)

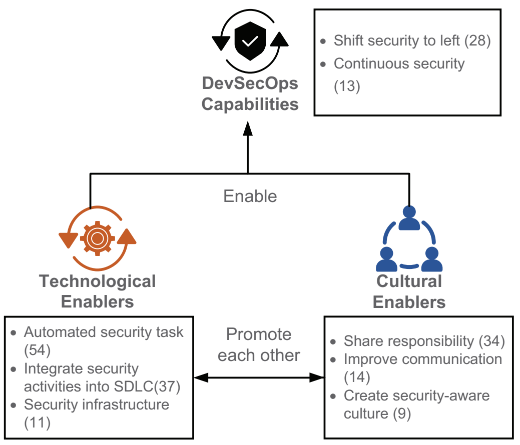
DevSecOps Capabilities
Shift security to left (28)
Continuous security (13)
Enable
Technological Enablers
Cultural Enablers
Promote each other
Automated security task (54)
Integrate security activities into SDLC(37)
Security infrastructure (11)
Share responsibility (34)
Improve communication (14)
Create security-aware culture (9)

Fig. 5. DevSecOps Capabilities and Enablers

Although there is no formal definition of DevSecOps both in academia and industry, it is a consensus that DevSecOps is an extension of DevOps [29]. Practitioners have different understandings of DevSecOps based on their own profession, DevOps practice maturity and the purpose of their article. In this context, we adopted the three core aspects of DevOps, namely *engineering capabilities*, *cultural enablers* and *technological enablers*, identified by Smeds et al. [30], to classify the multiple understandings of DevSecOps. We identified 76 articles from our dataset that provide detailed information about their understandings of DevSecOps, and categorized them according to the classification rules (Figure 5).

DevSecOps capabilities. Capabilities represent the processes that a organization should be able to carry out, while the enablers allow a fluent, flexible, and efficient way of working [30]. According to our results, the capabilities of DevSecOps include shift security to left and continuous security. Shift security to left is not only about introducing security activities into the early phase of development, but also integrating security into the entire DevOps lifecycle. Furthermore, continuous security is more about continual learning and continual improvement of projects and delivery security.

Cultural enablers. The cultural enablers list the traits that a DevSecOps team should exhibit. There are 34 out of 76 (44.7%) articles highlighted 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 of security and have a deeper understanding of how each discipline functions. The improvement of communication is also emphasized in 14 articles as a smooth communication through the project cycle facilitate cross-departmental collaboration. Based on collaboration, creating a security-aware culture is also a focus area of DevSecOps, as it encourage people to focus on security spontaneously.

Technological enablers. The results shows that DevSecOps stress the need for automating the security tasks since most of the practitioners regarded it as the technological enablers of DevSecOps. It is widely acknowledged that implementing automated security checks in the DevOps pipeline will substantially reduce the time and eventual cost of errors discovered using manual processes. Moreover, automation help organization to integrate security activities into SDLC (Software Development Life Cycle), without slowing it down and enable developers to improve code security without professional security knowledge. Security is also needed to be integrated into the infrastructure since the greater scale and more dynamic infrastructure enabled by containers have changed the software production environment.

D. Practices Associated with DevSecOps (RQ3)

While the understandings of SE practitioners reveal the fundamental ideas and values characterizing DevSecOps, the practices materialize them into daily software production activities [31]. Thus, we observed several recommended practices from the select GL, and mapped them to the cultural/technological enablers we identified in section IV-C. Furthermore, we identified commonalities and grouped them into three categories: Process, Infrastructure and Collaboration, to understand where and how to perform these practices. Figure 6 presents our classification results. The red square represents technological enabler while the blue square stand for cultural enabler. Every practice is mapped to one or more enablers.

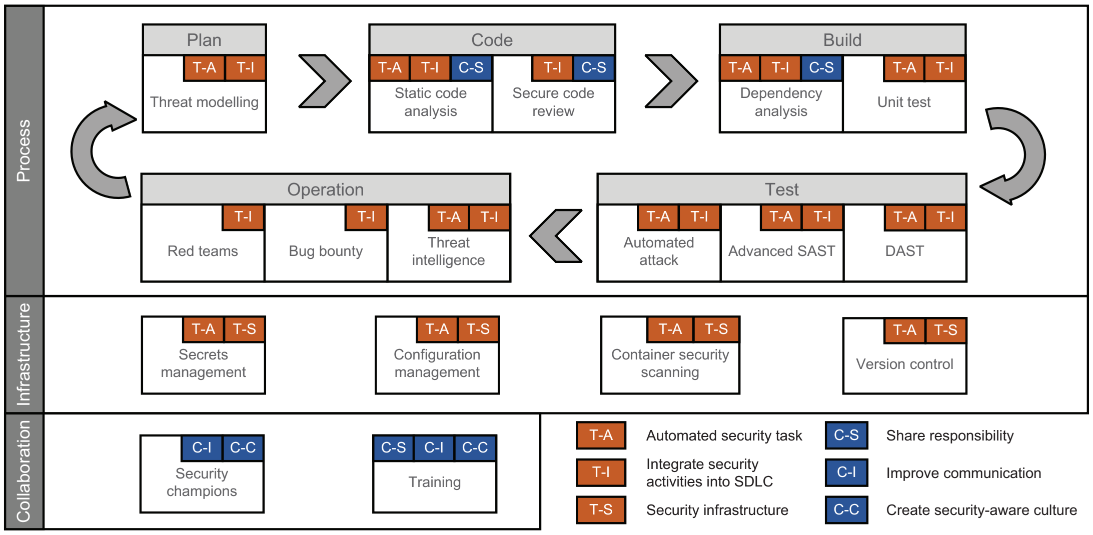

Fig. 6. Practices associated with DevSecOps

1) Process: Enterprise adopts DevSecOps practices to align and implement processes common to facilitate cooperation and achieve more secure development processes as an entirety.

Plan phase. Initially, the stakeholders plan the project regarding the type of software they need to develop, and make the rough picture regarding the development process. Threat modelling in plan phase ensures security is a consideration from the beginning of development, which categorizes potential threats, determines the possible outcome, and finally creates a proactive mitigation strategy results in a solid threat model [S24].However, threat modelling need to be automated because of its perceived slowness in DevOps [S15].

Code phase. In development phase, the developers code according to the requirements and then land the program into the source code repository. Practices like static code analysis, code reviews can be included to improve security without impacting developers' productivity [S5, 72]. It enables developers to find and fix common security issues before the code is committed into the source code repository.

Build phase. Once the code is committed to source repository, the build and basic automated testing of the application is performed to ensure that code is compilable and buildable at all times. Again, checks such as dependency analysis and unit tests can be added at this stage to enable the detection of critical and high security issues [S134]. If serious problems are found, the build should fail and send alert notifications.

Test phase. Test phase is triggered after a successful build by picking the generated artifact(s) and deploy it to staging and testing environments. All tests, including advanced SAST(Static Application Security Test) and DAST(Dynamic Application Security Test) [S60] are executed in this stage. Automated attacks [S53] can go even further and try to simulate attacks on running application, by executing basic set of targeted automated pen tests against the system as part of the automated test cycle.

Operation phase. After the systems are in production, automated security checks and monitoring feedback loops are essential to gain insights into the types of traffic that the applications are receiving and help identify patterns of malicious users. Red teams and Bug bounties [S118] in this phase can demonstrate what is wrong and provide the solution which creates a positive feedback loop between security teams and the developers, demonstrated by clear recommendations to improve the software quality. Furthermore, organization can address issues with cyber threat intelligence solutions which collect and process data automatically [S80, S106].

Processes are the key to the success of DevSecOps [S121]. Practitioners aim to create agreed and repeatable ways of working which are clearly documented to ensure transparency of the security towards the rest of the business.

2) Infrastructure: Infrastructure is an integral part of the software development which acts as a backbone for the whole system. The security assurances in infrastructure can ensure the whole pipeline runs smoothly.

Secrets management. 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, or a certificate, or an API token are necessary, but even with these precautions, 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 the accounts have only the privileges they need [S47].

Configuration management. Expressing the configuration of the running system in code allows for compliance with security policies and the elimination of manual errors through automated audit and remediation [S32].

Version control. It aims to manage versions of all changes to source code, executable images and tools used to create and test the software [S5]. The implement of it provides a foothold for security team to view into the entire system [S47].

Container security scanning. Since containers are the new building blocks of cloud native infrastructure [S22], and become a part of the attack surface that needs to be protected.

Conduct deep scanning of container images for vulnerabilities before run time facilitates minimize the attack surface and improve system stability [S63].

3) Collaboration: When the practitioners integrate security into their DevOps pipeline, people are still the greatest efficiency (or inefficiency) asset. Traditionally, the development, security, and operations teams are siloed. In DevOps, teams may still operate that way for a while; breaking down those traditional barriers can be the first and most important catalyst to DevSecOps practice. To identify and remedy the silos between different departments, some practitioners take some practices to create shared goals within DevSecOps teams, and drive a culture of innovation that consists of openness, transparency, ownership, and accountability.

Security champions. For security to be effective, enterprise needs to include security personnel as early as possible in the software delivery lifecycle. One way of doing this is by training security champions in the development team. Security Champions are a key element of the DevSecOps methodology, since they are the first step to create a cross-functional team focused on Application Security and Security Operations[S64].

Training. To foster and develop good security staff, organizations provide new hires with the appropriate training and tools they need to do their jobs well, and to contribute to the successful release of secure software[S121]. For software and IT engineers, organization will provide security-related training and equip them with the guidelines for setting routines to improve the security in the code phase[S92].

Proper training, a restructuring of teams and the appointment of security champions mean that security becomes less the function of a department and more a frame of mind that permeates the company.

V. Discussion

This Section discuss the 'grey area' of DevSecOps and the challenges of implementing DevSecOps.

A. DevOps vs. DevSecOps?

Mentioned in Section IV-A, the term **"DevSecOps"** we used in this study represents the concept that emphasizes the need to build a security foundation into DevOps initiative. It could be observed from our results that DevSecOps has aroused the concerns of DevOps practitioners and it has become increasingly acknowledged as a necessity since it was first mentioned by Neil MacDonald [10].

Although it is a consensus that DevSecOps is an extension of DevOps, the evidence from this study suggests that some practitioners do not subscribe to the concept. Margo Cronin, a senior solutions architect from Amazon, claimed that *"since security is naturally a built-in attribute of DevOps in her opinion"* [S11]. Some other similar cases were found to support that security is a natural component of DevOps and regard "DevSecOps" as a superfluous term [S42, S70]. Although some practitioners recognize the importance of including security in infancy stages of DevOps, they still doubt the necessity of a new acronym to tell a story for how to get development, operation and security working together [S69].

Apart form the controversy on the term, we observed that **"shift security to left"** and **"share responsibility"** are popular topics in DevSecOps, but they are indeed not a newcomer in SE. Gary McGraw proposed *"think about security early in the software lfecycle"* [32] and *"software security is everyone's job"* [33] a dozen years ago, and he also provided a set of security practices throughout the traditional waterfall model which is still a valuable reference for current security activities [32].

Compared with the traditional development model, although DevOps enables continuous and frequency deployment and delivery, there exists some security risks (cf. Section IV-B). In this context, the automation and efficiency of security practices might be the keys of DevSecOps.

B. Challenges of Implementing DevSecOps

Similar to DevOps [34], DevSecOps may not always be successful due to many factors. However, few articles (17/141) were identified as describing the challenges of implementing DevSecOps. The identified challenges could be classified from *internal* and *external* aspect.

Culture resistance, *high cost* and *solidified organizational structure* are the three main factors in internal aspects. There are several connections among these internal factors. Company *culture* and *organizational structure* are mutually reinforcing. For example, organizational 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 organizational structure should take full account of cost factors as well.

Three major *external factors* are identified from the selected articles, which are: *lack of DevSecOps experts, tools, and mature DevSecOps solutions*. 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.

There is a certain relationship between these internal factors and external factors and these factors influence each other. The *culture* within the organization is profoundly affected by employed experts and adopted solutions. Inherent company culture can also in turn determine the number of hired experts and the adoption of suitable 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 issues that companies need to consider if there is a limited budget before the official implementation of DevSecOps. *Organizational structure* can be changed if required by the adopted solutions as well. The company's managers can in turn choose the appropriate solution based on the company's organizational structure.

VI. Conclusion

Whilst the high speed of development and deployment, some security risks still exist in DevOps. In this context, DevSecOps, the concept evolved from DevOps, could make development, security and operation teams intertwined.

We illustrate the impact of DevOps on software security from both risks and opportunities aspects. Based on the preliminary consensus reached by practitioners that DevSecOps is an extension of DevOps, we categorize the understandings of the DevSecOps in the selected articles from three core aspects of DevOps, which are: DevSecOps capabilities, cultural enablers and technical enablers. We also summarize the existing typical DevSecOps practices in terms of process, infrastructure and collaboration. To arouse the discussion on DevSecOps among practitioners and researchers, we also discuss the 'grey area' of DevSecOps and the challenges of implementing DevSecOps in the industry from internal and external aspects.

This work is a preliminary study aimed at evoking greater enthusiasm for DevSecOps in academia and industry. As a practitioner-oriented field, the academic and industrial exchanges should be more frequent in SE. The research on DevSecOps can offer one such opportunity. Mentioned in [12], large-scale empirical study based on interviews and surveys should 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 [35], should be taken seriously to drive the investigation into DevSecOps in reality.

References

[1] D. Cohen, M. Lindvall, and P. Costa, "An introduction to agile methods," ser. Advances in computers. Elsevier, 2004, vol. 62, pp. 1–66.

[2] K. Schwaber and M. Beedle, *Agile software development with Scrum*. Prentice Hall Upper Saddle River, 2002, vol. 1.

[3] K. Beck, *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000.

[4] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," in *Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 2013, pp. 9–16.

[5] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *IEEE Software*, vol. 33, pp. 94–100, 2016.

[6] L. E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, and C. Lassenius, "Devops in practice: A multiple case study of five companies," *Information and Software Technology*, vol. 114, pp. 217–230, 2019.

[7] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "Dimensions of devops," in *Proceedings of the 16th International conference on agile software development*. Springer, 2015, pp. 212–217.

[8] D. G. Feitelson, E. Frachtenberg, and K. L. Beck, "Development and deployment at facebook," *IEEE Internet Computing*, vol. 17, pp. 8–17, 2013.

[9] A. A. U. Rahman and L. Williams, "Software security in devops: synthesizing practitioners' perceptions and practices," in *Proceedings of the 2016 International Workshop on Continuous Software Evolution and Delivery*. IEEE, 2016, pp. 70–76.

[10] N. MacDonald, "Devops needs to become devopssec," https://blogs.gartner.com/neil_macdonald/2012/01/17/devops-needs-to-become-devopssec/, 2012.

[11] N. MacDonald and I. Head, "Devsecops: How to seamlessly integrate security into devops," Gartner, Tech. Rep., 2016.

[12] N. Tomas, J. Li, and H. Huang, "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*. IEEE, 2019, pp. 404–411.

[13] L. Prates, J. Faustino, M. Silva, and R. Pereira, "Devsecops metrics," in *Information Systems: Research, Development, Applications, Education*. Springer, 2019, pp. 77–90.

[14] J. Díaz, J. E. Pérez, M. A. Lopez-Peña, G. A. Mena, and A. Yagüe, "Self-service cybersecurity monitoring as enabler for devsecops," *IEEE Access*, vol. 7, pp. 100 283–100 295, 2019.

[15] J. Soldani, D. A. Tamburri, and W.-J. Van Den Heuvel, "The pains and gains of microservices: A systematic grey literature review," *Journal of Systems and Software*, vol. 146, pp. 215–232, 2018.

[16] N. Salleh, E. Mendes, and J. Grundy, "Empirical studies of pair programming for cs/se teaching in higher education: A systematic literature review," *IEEE Transactions on Software Engineering*, vol. 37, pp. 509–525, 2010.

[17] M. Banks, "Blog posts and tweets: the next frontier for grey literature," in *Grey Literature in Library and Information Studies*. De Gruyter, 2009.

[18] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, pp. 101–121, 2019.

[19] J. Schöpfel, "Observations on the future of grey literature," *The Grey Journal*, vol. 2, pp. 67–76, 2006.

[20] R. L. Glass, *Software Creativity 2.0*. Developer.* Books, 2006.

[21] N. A. John, "The social logics of sharing," *The Communication Review*, vol. 16, pp. 113–131, 2013.

[22] R. J. Adams, P. Smart, and A. S. Huff, "Shades of grey: guidelines for working with the grey literature in systematic reviews for management and organizational studies," *International Journal of Management Reviews*, vol. 19, pp. 432–454, 2017.

[23] S. Kepes, G. C. Banks, M. McDaniel, and D. L. Whetzel, "Publication bias in the organizational sciences," *Organizational Research Methods*, vol. 15, pp. 624–662, 2012.

[24] H. Zhang, X. Zhou, X. Huang, H. Huang, and M. A. Babar, "An evidence-based inquiry into the use of grey literaturein software engineering," in *Proceedings of the 42nd International Conference on Software Engineering*. ACM, 2020, pp. 1422–1434.

[25] V. Garousi, M. Felderer, and M. V. Mäntylä, "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*. ACM, 2016, pp. 1–6.

[26] Y. Klijnsma, "Inside the magecart breach of british airways: How 22 lines of code claimed 380,000 victims," https://www.riskiq.com/blog/labs/magecart-british-airways-breach/, 2018.

[27] S. Kraemer, P. Carayon, and J. Clem, "Human and organizational factors in computer and information security: Pathways to vulnerabilities," *Computers and security*, vol. 28, pp. 509–520, 2009.

[28] V. Mohan and L. B. Othmane, "Secdevops: Is it a marketing buzzword?-mapping research on security in devops," in *Proceedings of the 11th International Conference on Availability, Reliability and Security*. IEEE, 2016, pp. 542–547.

[29] H. Myrbakken and R. Colomo-Palacios, "Devsecops: a multivocal literature review," in *Proceedings of the 17th International Conference on Software Process Improvement and Capability Determination*. Springer, 2017, pp. 17–29.

[30] J. Smeds, K. Nybom, and I. Porres, "Devops: a definition and perceived adoption impediments," in *Proceedings of the 16th International Conference on Agile Software Development*. Springer, 2015, pp. 166–177.

[31] B. B. N. de França, H. Jeronimo Junior, and G. H. Travassos, "Characterizing devops by hearing multiple voices," in *Proceedings of the 30th Brazilian Symposium on Software Engineering*. ACM, 2016, pp. 53–62.

[32] G. McGraw, "Software security," *IEEE Security and Privacy*, vol. 2, pp. 80–83, 2004.

[33] ——, *Software security: building security in*. Addison-Wesley Professional, 2006, vol. 1.

[34] L. Riungu-Kalliosaari, S. Mäkinen, L. E. Lwakatare, J. Tiihonen, and T. Männistö, "Devops adoption benefits and challenges in practice: a case study," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2016, pp. 590–597.

[35] H. Zhang, X. Huang, X. Zhou, H. Huang, and M. A. Babar, "Ethnographic research in software engineering: a critical review and checklist," in *Proceedings of the 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM, 2019, pp. 659–670.