DevSecOps: A Multivocal Literature Review

Håvard Myrbakken[1], Ricardo Colomo-Palacios[2],
haavam@hiof.no[1], ricardo.colomo-palacios@hiof.no[2]

Østfold University College, Remmen, 1757 Halden, Norway

Abstract. Involving security in DevOps has been a challenge because traditional security methods have been unable to keep up with DevOps' agility and speed. DevSecOps is the movement that works on developing and integrating modernized security methods that can keep up with DevOps. This study is meant to give an overview of what DevSecOps is, what implementing DevSecOps means, the benefits gained from DevSecOps and the challenges an organization faces when doing so. To that end, we conducted a multivocal literature review, where we reviewed a selection of grey literature. We found that implementing security that can keep up with DevOps is a challenge, but it can gain great benefits if done correctly.

Keywords: DevSecOps, DevOps, Security, Multivocal Literature Review

1 Introduction

In recent years a large part of software development companies have changed focus from developing software as a product (SaaP), where companies developed the software and delivered a finished product to a customer that then installed and ran it locally, to develop software as a service (SaaS), where software is centrally hosted on a cloud infrastructure and accessed through for example a web-browser [1], or other channels that delivers it directly to a customer's machine or device [2]. The use of it is then offered through licensing and subscriptions. With SaaS, the customers do not control the underlying cloud infrastructure or the application's functionality [1], as that is done by the provider. This gives the provider the opportunity to continuously improve and deliver their software without having to redistribute it to all their clients as they simply update the software on their own cloud infrastructure. This modern software engineering process of developing while continuously integrating and delivering software is complex. Continuous integration (CI) means to automatically integrate new code from several developers into the same version of the software and at the same time, check for errors [3]. Continuous Delivery (CD) means to deploy new software to production, with the differing factor from traditional software deployment being the frequency of deployment, which can happen multiple times every day [3]. "Continuous delivery enables businesses to reduce cycle time so as to get faster feedback from users, reduce the risk and cost of deployments, get better visibility into the delivery process itself, and manage the risks of software delivery more effectively.” [4]. These processes require a large number of tools and information systems [5]. These processes, tools and systems are often managed by independent operations teams [6]. Many challenges when implementing CI/CD resulted from lack of collaboration and communication between the operators and developers [2][3][6][7]. Attempts at overcoming these challenges have resulted in a concept, termed DevOps [2].

DevOps is described as the ”conceptual and operational merging of development and operations’ needs, teams, and technologies” [6]. This merging is meant to align the priorities of the development teams and operations teams so they work together towards a common goal of successful project execution [6] by cooperating on software development and deploying that software into production [2]. This can be done by involving operations in all development stages, by developers and operators collaborating to solve problems, make processes and products that can be automated, and agree on and develop metrics that everyone can make use of [8]. This reflects the four main principles of DevOps: culture, automation, measurement and sharing (CAMS) [4][2].

As DevOps has become more popular, many organizations are adopting the practices associated with it. However, a survey by the HPE Security Fortify team[9] from 2016 shows that while many believe that security should be a part of DevOps, security is not something many DevOps programs have included as part of their process. Gartner estimates in [10] that less than 20% of ”enterprise security architects have engaged with their DevOps initiatives to actively and systematically incorporate information security into their DevOps initiatives”. [10] points to management, developers, and operators viewing security as an inhibitor to the agility and speed required in DevOps practices, like CI and CD, as one reason for this.

The need for security in DevOps is met by DevSecOps. This concept is an attempt at creating and including modern security practices that can be incorporated in the fast and agile world of DevOps. It promotes an extension to DevOps’ goal of promoting collaboration between developers and operators by involving security experts from the start as well[11].

Since DevSecOps is a new trend, it is important to obtain an overview of the practices and experiences accumulated on the subject. There is not a lot of research on DevSecOps, but a search of available literature shows: [12] is a study that through Internet artefacts and a survey looks at practitioners experiences with DevSecOps and the practices they perform, [11] is a systematic mapping (SM) study on what is being researched in the field and it showed research was being conducted on the aspects of ”a definition, security best practices, compliance, process automation, tools for DevSecOps, software configuration, team collaboration, availability of activity data and information secrecy”. A search for systematic reviews or mappings on continuous processes (CI/CD) used in DevOps, resulted in a several results. Examples are: [13] examines the impact agile release engineering (ARE) and the continuous processes involved had on software engineering. [14] maps literature related to CD and provides an analysis of the benefits and challenges related to CD. [15] uses a literature review to show differences in how CI is done for different cases. None of the literature we found gives a collected overview of DevSecOps and what it is.

To the best of authors' knowledge, there is not a systematic literature review on DevSecOps or a large body of scientific work related to DevSecOps. This absence of works devoted to the topic lead us to the need to work on the topic using a tool like multivocal literature review, intended to bridge the gap between professional and scientific literature. By this mean, authors examine the concept of DevSecOps, how it has evolved since it was first introduced, and the challenges and benefits DevSecOps brings to an organization.

The rest of this paper is structured as a systematic literature review. In section 2 the methodology for the research is presented. In section 3 we present the results from our study. In section 4 we conclude on our paper, summarize the results and suggest future work.

2 Research Methodology

In this section, an overview of our research methodology is presented followed by an overview of the systematic approach used to gather relevant literature.

2.1 Multivocal Literature Review

After an initial search on literature to learn more on the topic of DevSecOps, we could not find a substantial body of academic research on the topic. We therefore decided to conduct a multivocal literature review (MLR). Multivocal literature is defined as all accessible literature on a topic [16]. This includes, but is not limited to: blogs, white papers, articles and academic literature. By using this variety of literature the results will give a more nuanced look at the topic, since it includes the voices and opinions of academics, practitioners, independent researchers, development firms and others with experience [16].

Previously published MLRs include but is not limited to: [17] is an MLR on automated software testing and the proposed guidelines from practitioners and researchers for when and what to automate. [18] is an MLR providing an overview of DevOps. [19] is an MLR on software test maturity assessments and test process improvement.

To the best of our knowledge, this is the first MLR on the topic although it is not the first for DevOps.

[20] points to the importance of MLRs in software engineering (SE) fields by stating that SE practitioners produces multivocal literature on a great scale, but that it is not published in academic forums. They mention however, that not including that literature in systematic reviews means researchers miss out on important current state-of-the-art practice in SE.

2.2 Research Questions

This MLR is conducted to obtain an understanding of what DevSecOps is, how it has evolved and the challenges and benefits of adopting such an approach. To specify the goal of this paper, four research questions were formulated:
Research Question 1: How does the literature define DevSecOps?
Research Question 2: What are the characteristics of DevSecOps?
Research Question 3: What are the main expected benefits and challenges of adopting DevSecOps?
Research Question 4: Since it was first mentioned, how has DevSecOps evolved?

2.3 Study protocol

The study protocol describes the systematic way we found the literature used in our study. This section lists the databases used in the search, what search strategy was used to find related literature, the inclusion and exclusion criteria used to find the most relevant literature, and the process in which we catalogued the literature.

Databases For this MLR we used Google's search engines to find relevant literature:

– Google Search (http://www.google.com/) to locate grey literature (white papers, blogs, articles etc.)
– Google Scholar (http://scholar.google.com/) to specifically locate available academic literature.

Google's search engines was chosen over more precise search engines (like Springer Link, ACM Digital Library, IEEE Explore etc.) because DevSecOps is a very new topic and very little academic research is available. We therefore knew beforehand that this literature review would rely mostly on the grey literature it would find, which Google's search engines would be able to locate.

Search Terms DevSecOps is a new term based on adding the term "SECurity" to DevOps which stands for "DEVeloper" and "OPeration". There is not a consensus in the field on the ordering of the words, so the search terms must cover all possible permutations. The search string must also be made to find relevant literature according to the RQs. The search string used is therefore as follows:

```
("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND
("definition" OR "characteristics" OR "challenges" OR "benefits"
OR "evolution").
```

Study Selection Once initial search results were retrieved, a procedure to exclude irrelevant papers were conducted using the following inclusion and exclusion criteria:

- Inclusion criteria:
  - Literature that explicitly discuss DevSecOps.
  - Literature that explicitly discuss DevOps and Security, particularly the challenges and benefits.
  - Literature discussing the present challenges to DevSecOps.
  - Literature discussing the benefits of DevSecOps.
  - Literature that discuss the definition of DevSecOps.
  - Literature published after 2014.
  - Include only the 5 first pages on Google Search.
- Exclusion criteria:
  - Literature that is inaccessible.
  - Results Google Search deems to similar to other results.
  - Vendors tool advertisements.

Search Procedure The process is as follows: First we perform an advanced search in Google Search and Google Scholar. To let Google's search engine put primary focus on the different RQs, the term will be split into 5 parts, each focusing on all permutations of DevSecOps and one of the words related to the RQs. The 5 search strings can be seen in 1. For each search we then read the literature systematically applying the remaining inclusion and exclusion criteria, selecting only relevant literature for the primary study. The process is visualized in figure 1:

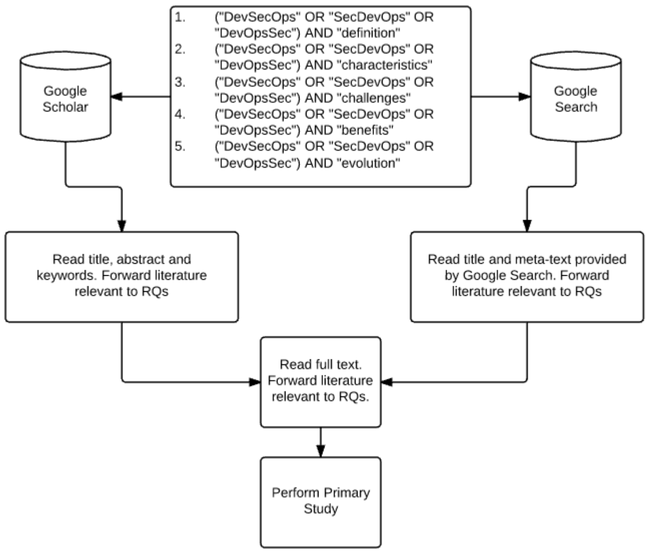
1. ("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND "definition"
2. ("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND "characteristics"
3. ("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND "challenges"
4. ("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND "benefits"
5. ("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND "evolution"
Google Scholar
Google Search
Read title, abstract and keywords. Forward literature relevant to RQs
Read title and meta-text provided by Google Search. Forward literature relevant to RQs
Read full text. Forward literature relevant to RQs.
Perform Primary Study

Fig. 1. An overview of the search process to find relevant literature for this study.

3 Results

In the following section we show the results from executing our search followed by our review of the literature in conjunction with our research questions.

3.1 Search Execution

The search was performed on during the first quarter of 2017. The initial search gave 284 results. After applying the inclusion and exclusion criteria 52 results were left. Table 1 summarizes the search:

Search Engine	Initial Results	Title, abstract, keywords and Meta text	Full text
Google Scholar	34	4	2
Google Search	250	62	50

Table 1. Summary of search results for primary study

3.2 RQ1 - how it is defined

In the literature we reviewed there seemed to be a consensus on what DevSecOps is seen as. DevSecOps is seen as a necessary expansion to DevOps, where the purpose is to integrate security controls and processes into the DevOps software development cycle [21] and that it is done by promoting the collaboration between security teams, development teams and operations teams [11].

3.3 RQ2 - DevSecOps characteristics

When reviewing the literature, the features that stood out as characterizing DevSecOps were the principles seen as the basis and reasoning for DevSecOps and the practices used when implementing security into their software development processes.

Principles: The principles that characterize DevSecOps are based on DevOps and the CAMS principles [4][22], culture, automation, measurement, and sharing, but with the addition of adding security from the start:

Culture: A DevOps culture promotes collaboration between development teams and operation teams [4], where they all accept that they are responsible for delivering software to an end-user[2]. DevSecOps means to include collaboration with the security team as well as promote a culture where operations and development also work on integrating security in their work [11] [23]. That means involving the security team from the planning stages, and making sure everyone agrees that security is everyone's responsibilities [24][25]. To get everyone to think security, practitioners points to creating a common mindset with regards to success by developing a set a metrics that everybody agrees on and can use [25], promote customer focus by creating an alignment of business and security strategies to ensure just right and enough security that everyone in an organization can support and implement [26][27].

Automation: In DevOps the automation of build, deployment, and testing is important to achieve rapid development, deployment [4][22], and feedback from end-users[2]. DevSecOps promotes a focus on automating security as well, to be able to keep up with the speed and scale achieved by DevOps. The aim should be 100% automation of security controls, where the controls can be deployed and managed without manual interference [28]. It is important to implement automatic security in a way that does not hinder DevOps' agility in any way, which can cause friction [29][27][10].

Measurement: In DevOps measurements include monitoring business metrics such as revenue and key performance indicators, like the effect new releases have on the stability of a system, in order to know the current state and finding out how to improve it[4][2]. DevSecOps promotes the use and development of metrics that track threats and vulnerabilities throughout the software development process [10]. Automatic security controls throughout the software development process means metrics are available to track threats and vulnerabilities in real-time and that allows the organization to verify how good an application is on demand [22].

Sharing: In DevOps developers and operators share knowledge, development tools, and techniques to manage the process [4] [2]. DevSecOps promotes the inclusion of the security team in the sharing promoted in a DevOps environment. By letting security teams know about the challenges faced by operators and developers, and vice versa, the security processes they develop will be improved [22].

Shift security to the left: In the traditional software development process, security is a step close to the end of the process. DevSecOps promotes a shift to the left for security, where it is to be included in every part of the software development process [23]. This means that security teams are involved from the very first planning step and is part of planning every iteration of the development cycle [29][30]. It also means security is there to help developers and operators on security considerations [28] [31] [24].

Practices: Several practices for DevSecOps were pointed to in the literature:

Threat modeling and risk assessments: Practicing secure DevOps means that organizations have to develop expertise and processes to best discover, protect against, and find solutions to threats and risks [32], preferably ahead of time [25]. Performing risk assessments from the first planning stage and continuously before every iteration is important as a way to prioritize risks, examine controls already in place and decide which are needed going forward [33] [21]. Threat modeling is another method where you attack your system on paper early in the development cycle to identify how an attack can occur and where it is most likely to happen [34].

Continuous testing: Automatic security controls at every part of the software development process is important for security assurance and allows tests to continuously scan code for changes [34] [32], continuously detect anomalies, and automatic rollback of code when needed [21] [24].

Monitoring and logging: When automating security controls throughout the software developing process it is important for those involved to be able to generate evidence on demand that controls are working and that they are effective [35]. To that end, it is important to monitor every part of the inventory and to log every resource [25] [33] [21].

Security as code: This means to define security policies, for example integration testing, and network configuration and access, and write scripted templates or configuration files that can be implemented into the development process from the start of the project. These codified security policies can then be activated automatically according to schedules or be activated by user (simple push of a button), and be stored in a central repository for reuse on new projects [36].

Red-Team and security drills: To stay ahead of possible attackers, practitioners of DevSecOps create a Red-Team that runs security drill on the deployed software. They have the task of finding and exploiting vulnerabilities in the system [25][34]. This not only helps to find security flaws, but improves measurements, and helps the organization find solutions [26]. The point of the Red-Team is to have people that never claim something can't possibly happen.

3.4 RQ3 - Benefits

The following section provides an overview, according to the literature, of the benefits gained from DevSecOps and its practices:

Shifting security to the left: By involving security experts from the start of the development process it is easier to plan and execute integration of security controls throughout the development process without causing delays or creating issues by implementing security controls after systems are running [29].

Automating security: This allows security controls to be fast, scalable and effective thus making it possible to keep a high pace for detecting errors, alerting about the errors, fixing the errors, finding countermeasures for future errors and forensics to identify why an error occurred [37]. This not only helps to lower risk and time spent on errors, but also makes it easier to understand risk and create policies and procedures [38]. The automation allows processes to be consistent and repeatable, with predictable outcomes for similar tests, it allows logging and documentation to be automatic [39] and letting security tests be run at the push of a button frees up developers time to write code instead of running tests [40]. This also reduces the risk for human error[39]. The ability to store security policy templates that is created during a development process in a central repository means that security teams don't need to manually configure every new environment when starting a new project which frees security experts from manual, repetitive and unproductive work[36].

Value: [41] [38] points to how security missteps can be very expensive and that it is cheaper to implement security from the start than to wait for something to happen. [38] points to a survey that mentions how high-performance organizations spend 22 % less time on unplanned work and rework. The ability to monitor and measure for security flaws early in the process ensures that bugs that prevent a delay in the deployment are caught and quantified [38]. This decreases the cost of making mistakes, finding them, and fixing them [36].

3.5 RQ3 - Challenges

The following section provides an overview of the challenges an organization faces from DevSecOps. The challenges are connected to the speed and agility needed not to slow down other DevOps practices, organizational changes, tools and practices:

Keeping up with DevOps: Using traditional, manual security methods heavily impairs the speed and agility of DevOps. This means security methods have to be more agile, and these agile security methods have to be understood by security teams and accepted by development teams [42] to make sure they contribute meaningfully to the DevOps movement without hampering their development speed and service delivery [43].

Organizational: Getting started with DevSecOps means the organization has to adopt change. Skills, culture, tools, processes, standards and practices must be considered as a possibility for implementing security [29]:

- There will be a need for skills in areas such as encryption and logging standards etc. [29].
- The organizational barriers between security teams and the rest of the organization must be broken down:
  - Developers and managers can be frustrated with the added time it takes to produce code, when adding security [33] [21]. Developers and operators think of security as a hindrance to their goals, which is to deliver functionality fast, while security teams are focused on making sure the functionality is secure and robust [44].
  - The security teams not being properly trained on tools developers and operators use, hinders them from being able to integrate security in a transparent and understandable way for other users, which would limit collaboration between teams [31].
  - Organizations see security as a costly activity, and not something that generates revenue [32] [35].
- There will be a need for new standards for security prevention, detection and response [29].

Tools and practices: The dynamic environment when practicing DevOps means that security functionality has to be available in tools that work on the right platforms. There is a lack of available tools[21]. Any security functionality not automated in the available tools will create friction in the DevOps cycle. The users need to be properly trained when using advanced tools. [38] points to developers that had difficulties writing secure code because they couldn't use the tools efficient enough to keep up with DevOps' speed.

3.6 RQ4 - The evolution of DevSecOps

The need for security to be integrated in DevOps was first mentioned in a blog by Neil MacDonald, a Gartner analyst, in a blogpost called "DevOps Needs to Become DevOpsSec" in 2012. DevSecOps has since become more and more acknowledged as a necessity. 2 shows the increase in number articles per year, which is evidence that awareness, recognition and use of DevSecOps is on the rise.

Year	Number
Unknown	7
2014	2
2015	8
2016	27
2017 (first quarter only)	8

Table 2. Overview of number of results per year

3.7 Limitations of results

This research is based on multivocal literature, and most of the material has not been subject to the rigorous peer-review academic research usually is. The literature has instead consisted of blogs, white-papers, industry reports and academic research.

DevSecOps is a very new term, and the term has not even been agreed upon. It varies between SecDevOps, DevSecOps, DevOpsSec, Secure DevOps, and Rugged DevOps. In this research paper we have consistently used DevSecOps (with exception to where I am referring to other sources and their titles). The fact that it is as new as it is, means the results from this MLR can become outdated as best practices change.

4 Conclusion

This MLR presents the research we did on DevSecOps to find out how DevSecOps can be defined, what doing DevSecOps means for an organization in regard to what principles and practices they should adhere to, what challenges they would face attempting to adopt DevSecOps, the benefits if it's done successfully and how it has evolved from the need to implement security in DevOps to what could seem like a movement on its own.

We used Google Scholar and Google Search to locate literature and after applying our inclusion and exclusion criteria, 52 artefacts were found to be relevant to our search terms. Only 2 of those were academic research papers. The remaining 50 consisted of white papers, blogs and articles.

We found that DevSecOps is defined by many as the integration of security processes and practices into DevOps environments, that DevSecOps promotes a set of principles meant to shift the mindsets of all participants in the software development process so everyone participates and do what they can to ensure security in the project and a set of practices that can ensure security in the project based on the idea of planning and implementing security from the start and as code.

We identified a set of challenges and benefits to implementing DevSecOps. The challenges we identified should not be seen as deterrents to implementing DevSecOps, but a symptom of its youth. As DevSecOps matures, better methods, practices, tools etc. can probably overcome them. The benefits we identified indicates it is maturing, by for example resulting in less unplanned work and a decrease in manual labour.

As future work, it would be interesting to conduct surveys on organizations to possibly expand this study's coverage on DevSecOps. It is also of interest to investigate this study's suggested practices: observing practices effects on the surrounding environments (development, operations, business, customers) to find best practices. A possibility would then be to investigate and propose possible architectures or frameworks for implementing DevSecOps, [45] for example looks at continuous software engineering while using a microservices architecture which could be an alternative for "security as code" in DevSecOps.

References

1. P. M. Mell and T. Grance. The nist definition of cloud computing. *Special Publications (NIST SP)-800-145*, (7), 9 2011. NIST Definitions on Cloud Computing.
2. B. Fitzgerald and K. J. Stol. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123:176 – 189, 2017.
3. R. B. Svensson G. G. Claps and A. Aurum. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software Technology*, 57:21 – 31, 2015.
4. J. Humble and M. Joanne. Why enterprises must adopt devops to enable continuous delivery. *The Journal of Information Technology Management*, (24), 7 2011.
5. J. Hernantes C. Ebert, G. Gallardo and N. Serrano. Devops. *IEEE Software*, 33(3):94–100, May 2016.
6. J. Yankel C. A. Cois and A. Connell. Modern devops: Optimizing software development through effective system interactions. In *2014 IEEE International Professional Communication Conference (IPCC)*, pages 1–7, Oct 2014.
7. M. Callanan and A. Spillane. Devops: Making it easy to do the right thing. *IEEE Software*, 33(3):53–59, May 2016.

8. D. Spinellis. Being a devops developer. *IEEE Software*, 33(3):4–5, May 2016.
9. Hewlett Packard Enterprise. Application security and devops. Technical report, Hewlett Packard Enterprise, 2016.
10. N. MacDonald and I. Head. DevSecOps: How to Seamlessly Integrate Security Into DevOps. Technical report, Gartner, 2016.
11. V. Mohan and L. B. Othmane. Secdevops: Is it a marketing buzzword? - mapping research on security in devops. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pages 542–547, Aug 2016.
12. A. Ashfaque U. Rahman and L. Williams. Software security in devops: Synthesizing practitioners' perceptions and practices. In *Proceedings of the International Workshop on Continuous Software Evolution and Delivery*, CSED '16, pages 70–76, New York, NY, USA, 2016. ACM.
13. M. Oivo T. Karvonen, W. Behutiye and P. Kuvaja. Systematic literature review on the impacts of agile release engineering practices. *Information and Software Technology*, 86:87 – 100, 2017.
14. L. E. Lwakatare S. Teppola T. Suomalainen J. Eskeli T. Karvonen P. Kuvaja J. M. Verner P. Rodríguez, A. Haghighatkhah and M. Oivo. Continuous deployment of software intensive products and services: A systematic mapping study. *Journal of Systems and Software*, 123:263 – 291, 2017.
15. D. Ståhl and J. Bosch. Modeling continuous integration practice differences in industry software development. *Journal of Systems and Software*, 87:48 – 59, 2014.
16. R. T. Ogawa and B. Malen. Towards rigor in reviews of multivocal literatures: Applying the exploratory case study method. *Review of Educational Research*, 61(3):265–286, 1991.
17. V. and M. V. Mäntylä. When and what to automate in software testing? a multivocal literature review. *Information and Software Technology*, 76:92 – 117, 2016.
18. H. J. Junior B. B. N. de França and G. H. Travassos. Characterizing devops by hearing multiple voices. In *Proceedings of the 30th Brazilian Symposium on Software Engineering*, SBES '16, pages 53–62, New York, NY, USA, 2016. ACM.
19. M. Felderer V. Garousi and T. Hacaloğlu. Software test maturity assessment and test process improvement: A multivocal literature review. *Information and Software Technology*, 85:16 – 42, 2017.
20. M. Felderer V. Garousi 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*, EASE '16, pages 26:1–26:6, New York, NY, USA, 2016. ACM.
21. D. Shackleford. A devsecops playbook. *SANS Institute InfoSec Reading Room*, 3 2016. A DevSecOps playbook.
22. S. Vonnegut. 4 keys to integrating security into devops. https://goo.gl/aZ0S3i, 2016.
23. S. Lietz. Shifting security to the left. https://goo.gl/sbheKS, 2016.
24. G. Bledsoe. Getting to devsecops: 5 best practices for integrating security into your devops". https://goo.gl/ZPzgxa, 2016.
25. F. Lim. Devsecops is the krav maga of security. https://goo.gl/BH4MS2, 2016.
26. S. Lietz. Principles of devsecops. https://goo.gl/N8zcXV, 2015.
27. T. Greene. What security teams need to know about devops. https://goo.gl/c8VOn4, 2016.
28. Anonymous User. Security breaks devops – here's how to fix it. https://goo.gl/Yr1jk3, 2015.

29. D. Shackleford. The devsecops approach to securing your code and your cloud. *SANS Institute InfoSec Reading Room*, 2 2017. A DevSecOps playbook.
30. C. Caum. Getting started with policy-driven development and devsecops. https://goo.gl/AevVcX, 2016.
31. Whitehat Security. Devops invites security to "join the party". https://goo.gl/spj0wK, 2016.
32. M. Hornbeek. Devops makes security assurance affordable. https://goo.gl/g0iKfZ, 2015.
33. K. Lindros. How to craft an effective devsecops process with your team. https://goo.gl/ppWtjx, 2016.
34. C. Romeo. The 3 most crucial security behaviors in devsecops. https://goo.gl/FJKuYQ, 2016.
35. A. Cureton. Building security into devops: Is devsecops the beginning of the future? https://goo.gl/Npv2Py, 2017.
36. J. McKay. How to use devsecops to smooth cloud deployment. https://goo.gl/vqoh4L, 2016.
37. Amazon Web Services. Introduction to devsecops on aws. https://goo.gl/wxl3YM, 2016.
38. R. Francis. 7 ways devops benefits cisos and their security programs. https://goo.gl/RxieGr, 2015.
39. A. Wallgreen. Devsecops: 9 ways devops and automation bolster security, compliance. https://goo.gl/RyA9QZ, 2015.
40. M. Rotenberg. 7 essential steps to devsecops success. https://goo.gl/JAOQlF, 2016.
41. F. Paul. Secdevops: Injecting security into devops processes. https://goo.gl/Eul2fn, 2015.
42. M. Rohr. Agile security and secdevops touch points. https://goo.gl/peuqpS, 2015.
43. M. Goldschmidt and M. McKinnon. Devsecops – agility with security. Technical report, Sense of Security, 2016.
44. M. Elder. Security considerations for devops adoption. https://goo.gl/b0CStP, 2014.
45. PM. Clarke. RV. O'Connor, P. Elger. Continuous software engineering—a microservices architecture perspective. *J Softw Evol Proc. 2017;e1866.*, 2017.