SecDevOps: Is It a Marketing Buzzword?

Mapping Research on Security in DevOps

Vaishnavi Mohan
Department of Computer Science
Technische Universität Darmstadt
Darmstadt, Germany
vaishnavi.mohan@stud.tu-darmstadt.de

Lotfi ben Othmane
Fraunhofer SIT
Darmstadt, Germany
lotfi.ben.othmane@sit.fraunhofer.de

***Abstract*—DevOps is changing the way organizations develop and deploy applications and service customers. Many organizations want to apply DevOps, but they are concerned by the security aspects of the produced software. This has triggered the creation of the terms SecDevOps and DevSecOps. These terms refer to incorporating security practices in a DevOps environment by promoting the collaboration between the development teams, the operations teams, and the security teams. This paper surveys the literature from academia and industry to identify the main aspects of this trend. The main aspects that we found are: definition, security best practices, compliance, process automation, tools for SecDevOps, software configuration, team collaboration, availability of activity data and information secrecy. Although the number of relevant publications is low, we believe that the terms are not buzzwords; they imply important challenges that the security and software communities shall address to help organizations develop secure software while applying DevOps processes.**

***Index Terms*—DevOps; SecDevOps; Security; Mapping Research; DevSecOps;**

I. Introduction

DevOps is a trending technology term. It refers to improving the performance of software development operations by involving the development team and the operations team in one process. This helps to increase the frequency of deployments, which helps to service the customers faster [1]. In fact, CA Technologies expects that 1254 out of 1425 organizations will adopt DevOps in the next five years [2].

DevOps helps organizations to [1]:

- improve the collaboration among the developers and operations teams,
- enhance the frequency and easiness of deployments,
- increase the flexibility in accommodating customer requirements,
- improve the quality of code due to developer collaboration.

Security is among the major concerns that limit the adoption of DevOps processes. This triggered the coining of the terms *SecDevOps* and *DevSecOps*. Both refer to incorporating security practices in the DevOps processes by promoting collaboration between the development teams, the operations teams, and the security teams. Other concerns include: reliable cross-team management process, recognized training methodology, availability of professionals for a complete DevOps implementation, awareness about security and compliance aspects, and balance of workload among the teams involved [1].

Security experts in the industry and academia have started investigating the security aspects of DevOps. Rahman et al. [2] surveyed the perceptions of DevOps practitioners towards security in DevOps. They identified the DevOps activities that potentially impact the security of the software and identified the security practices that organizations use to integrate security into DevOps. Our paper surveys the literature about the security aspects in DevOps from academia and industry. (We limited our search to presentations and papers published by OWASP AppSec and RSA for the case of industry publications.) It identifies the aspects that are being discussed in the SecDevOps or DevSecOps literature. The results could be a basis for researchers who want to investigate security problems related to DevOps.

The rest of this paper is organized as follows. Section II describes the research methodology, Section III describe the literature on security in DevOps and categorizes the relevant information extracted from the selected publications, Section IV summarizes the information extracted from the analyzed publications, and discusses the results and limitations of the study, and Section V concludes the paper.

II. Research Methodology

We used in this work the systematic mapping research method [9]. The systematic mapping research method is commonly used to survey the state of the art of research areas that are not yet mature [10]. Such studies help to summarize a particular area of research by categorizing the topics investigated in the relevant research publications. The following subsections describe the activities that we performed.

A. Definition of Research Questions

The main aim of the study is to identify the aspects that the literature related to SecDevOps or DevSecOps discusses. It addresses the question: *What are the aspects that the research community believes are related to SecDevOps and DevSecOps?*

B. Search for Primary Studies

We derived a set of keywords based on the defined research question. We did an initial search of the Web and we identified

Table I
SUMMARY OF THE IDENTIFIED PUBLICATIONS AND PRESENTATIONS.

No	Citation	Description
S1	[1]	The paper describes the important issues that should be considered when adopting DevOps and stresses the need to ensure that security and compliance are not compromised in such a development model.
S2	[2]	The paper enumerates the industrial perspectives on SecDevOps with an analysis of Internet artifacts and a survey with DevOps practitioners from 9 organizations.
S3	[3]	The paper showcases LiCShield framework for the protection of Linux containers and their workloads, that are an integral part of most cloud and DevOps environments.
S4	[4]	The paper showcases the IBM Cloud OpenStack Services offering. The importance of security in cloud environments, and therefore in DevOps is also highlighted.
S5	[5]	The paper introduces an engineering process to secure the subversion of deployment pipelines.
S6	[6]	The presentation describes the various levels of SecDevOps integration with an introduction to Security DevOps Maturity Model(SDOMM)
S7	[7]	The presentation stresses the importance that security has in DevOps adoption. The presentation enumerates process improvements and tools useful in securing DevOps.
S8	[8]	The presentation portrays the drawbacks of traditional security approaches and introduces the use of security stories (e.g., authentication story and code scan story) as an approach for SecDevOps.

Table II
SUMMARY OF THE SECDEVOPS ASPECTS IDENTIFIED BY THE MAPPING STUDY.

No	Aspects	Description
1.	Definition	definitions for the term SecDevOps and equivalent terms
2.	Security Best Practices	security and DevOps activities that are best suited for SecDevOps
3.	Compliance	privacy and compliance issues related to SecDevOps
4.	Process Automation	activities automation and its influence on DevOps processes
5.	Tools for SecDevOps	COTS and GOTS tools that can be used to integrate security into DevOps
6.	Software Configuration	SecDevOps configuration management needs and possible influence of software configurations on SecDevOps
7.	Team Collaboration	needs of SecDevOps model in terms of collaboration between the stakeholders
8.	Availability of Activity Data	abundance of the data that could be generated by SecDevOps processes and their potential use
9.	Information Secrecy	impact of cross-team collaboration and collected activities data on the development trade-secrecy

a set of terms that the community uses when discussing SecDevOps/DevSecOps topics. The terms are: SecDevOps, Secure DevOps, Security in DevOps, Safe DevOps, SecOps, (secure)AND(DevOps), and DevSecOps.

We used these keywords to search for the research papers that address the question. First, we searched the Google Scholar database, and the IEEE Xplore Digital Library for academic papers. We identified 66 research publications that match the research criteria. Second, we searched for presentations at OWASP and RSA conferences that also much the criteria. We identified 5 presentations. In general, talks at industry events are not reviewed/verified (and therefore not trustworthy), which is not the case for OWASP and RSA events.

C. Screening of Papers for Inclusion and Exclusion

Papers, that did not comply with the motivation of this research, are excluded from the study. In addition, papers that did not fit into at least one of the following criteria are excluded:

1) defines the term SecDevOps or Security in DevOps,
2) highlights the need for security in DevOps,
3) describes security concerns or protection mechanisms for DevOps components,
4) describes security practices that can be used to incorporate security into DevOps.

We found that only 5 publications out of the 66 and 3 presentations from OWASP AppSec and RSA conferences out the identified 5 presentations, are relevant to our research question. Again, the search for conference talks was not exhaustive and was confined to targeted conferences. Table I gives the list of the identified publications.

D. Keywords for the Classification

To identify research context, the various sections of the identified papers were condensed to a set of keywords. The keywords from all the papers were then grouped together to form aspects, which were used to classify the results from relevant publications. Those aspects were further refined based on common topics. The final set of aspects are: definition, security best practices, compliance, process automation, tools for SecDevOps, software configuration, team collaboration, availability of activity data and information secrecy. The description of each of the selected aspects is given below. Table II lists these aspects.

E. Data Extraction and Mapping Studies

We analyzed all selected papers and presentations to verify that they included ’concrete’ information related to each of the aspects identified in Table II. Then, we classified the selected publications and presentations into the aspects. We observed that most of the publications discuss multiple SecDevOps/DevSecOps aspects.

Table III summarizes the different aspects that the selected publications and presentation comprise.

Table III
CLASSIFICATION OF SECDEVOPS PUBLICATION INTO THE IDENTIFIED ASPECTS

Aspect	S1	S2	S3	S4	S5	S6	S7	S8
Definition		X		X				
Security Best Practices	X	X		X		X		X
Compliance	X					X	X	
Process Automation		X	X		X		X	X
Tools for SecDevOps	X					X	X	
Software Configuration		X			X			
Team Collaboration	X	X						
Availability of Activity Data	X					X		
Information Secrecy	X	X						

III. OVERVIEW OF THE CURRENT RESEARCH ASPECTS IN SECDEVOPS

We report in this section the treatment of the different aspects that we identified in Section II-E in each of the selected papers. Note that we will use, starting from this section, the term SecDevOps to refer also to DevSecOps, Secure DevOps, Security in DevOps, Safe DevOps, and SecOps.

A. Definition

Researchers from industry and academia agree that SecDevOps and DevSecOps imply integration of security practices in the DevOps processes. For example, Cash et al. [4] refer to the practice of incorporating quality security technologies into DevOps as SecDevOps. They consider that DevOps fits developing software for the cloud and consider that security in influencing enterprise decisions to utilize the cloud.

Rahman et al. [2] consider that DevSecOps, SecDevOps, SecOps, and RuggedOps are aliases for Security in DevOps. They consider that the terms refer to the integration of security principles in DevOps by promoting the collaboration between the security teams, the development teams and operations teams.

B. Security best practices

Farroha et al. [1] call for the integration into SecDevOps the following set of best practices: automating tests to detect non-compliance, tracking compliance breaches through automated reporting of violations, continuous monitoring and maintenance of a service catalog with tested and certified services. The paper also depicts successful strategies to maintain the incorporated security. Cash et al. [4] call for integrating into SecDevOps security scanning and configuration automation.

Rahman et al. [2] believes that DevOps activities impact security in 2 opposite ways: positive or negative. They believe that automation activities, such as automated monitoring, automated deployment pipeline, and automated testing contribute positively to the security of the software. However, selecting the wrong automated deployment tools, using the wrong software metrics and unsupervised collaboration contribute negatively towards the security of the software.

In addition, 5 out of 9 survey [2] respondents believe that the fast software deployment that DevOps processes support pushes organizations to overlook security tests, which leads to deploying vulnerable software. the majority of the survey respondents also believe that security policies, manual security tests, and security configuration are prevalent in DevOps organizations.

Schneider [6] describes the different stages of dynamic security scanning that can be used by organizations to integrate security into DevOps. The four levels of scanning are: (1) pre-authentication scanning, that involves scanning the public attack surface; (2) post-authentication scanning, that involves session maintenance, user role management, and logout and auto-relogin detection; (3) backend scanning of the various application layers independently; and (4) scanning workflows specific to the targeted application.

Vries [8] believes that security activities need to adopt concepts used in DevOps. The talk advocates the collaboration between the development team and the business owner of the software to set the security goals. The talk also calls for automating security tests and security scans in SecDevOps processes.

C. Compliance

Schneider [6] introduces the SecDevOps Maturity Model(SDOMM). The model is a manual to help projects achieve certain security aspects through automation in a continuous integration (CI) build chain. The model is useful for organizations willing to make the change to SecDevOps. It comprises four axes: (1) dynamic depth, the extent of dynamic scans in a CI chain, (2) static depth, the extent of static code analyses in a CI chain, (3) intensity, the impact of the executed attacks and (4) consolidation, the effectiveness of handling findings.

Storms [7] believes that DevOps fails to include security throughout the process and leaves it to the end. This is supported by CA technologies survey [11] that found that more than one-fourth of the surveyed companies are willing to adopt DevOps but state that security and compliance concerns stop them from doing so. The security problems due to DevOps that Storms lists include the high pace of deployments, the unclear access restrictions, and the lack of audit and control points.

Farroha et al. [1] suggest a set of requirements for compliance policies which include prohibiting unauthorized access, maintaining a log for accesses to sensitive data, and monitoring data operations. In addition, they illustrate how the collection of data could affect the policies for compliance.

D. Process automation

Vries [8] explains how the traditional security approach with a focus on documentation, manual processes and tools are unfit for continuous deployment environments and need to be replaced by a more modern approach to suit DevOps.

Rahman et al. [2] describe the impact of automated deployment on security. Automated deployment pipelines enable organizations to ship software changes at a rapid rate. Though, this increase in deployment speed is beneficial to the organizations, the speed might lead to overlooking the necessary security reviews and checks that need to be approved before delivery. If the security team is not a part of this rapid develop and deploy iteration, it might increase chances of production of vulnerable software. Their study also describes the security techniques prevalent among organizations to integrate security into DevOps. Use of automation activities like automated code review, automated monitoring, automated testing are popular automation activities for security integration in DevOps environments.

Storms [7] explains that the integration of security expertise in DevOps pipelines and processes, helps to enable the DevOps and Security teams work together more efficiently. His talk highlights the ways in which existing DevOps tools can be utilized to strengthen security. He also suggests process changes to move towards secure DevOps.

Matteti et al. [3] describe the need to secure Linux containers, which are considered a break-through for DevOps because of their contribution to simplifying automated deployments. Linux containers expose file systems, networks and kernels to attacks. These resources cannot be protected by the existing security measures that only protect specific applications rather than entire environments. Linux containers are considered more prone to attacks than VMs. Kernel exploits, attacks on shared Linux host resources, misconfiguration, side channels and data leakage are some vulnerabilities in Linux containers.

Matteti et al. describe also mechanisms to protect container environments. The security hardening mechanisms, including AppArmor and SELinux, and host based intrusion detection systems are not easy to be used within Linux container environments. The authors propose the LiCShield Framework that provides protection to hosts, by confining accesses of containers and container management daemons to perform only the operations observed in testing environments and restricting container operations, by tightening the internal noisy environments. The authors describe also the protection mechanisms available in Docker, a container management daemon. Docker is protected by AppArmor/SELinux profiles that secure the critical host locations from modifications. This security mechanism of Docker does not secure the container workloads, nor does it provide protection against the vulnerability of the Docker daemon itself. LiCShield Framework has been created to overcome these drawbacks.

Bass et al. [5] elucidate the vulnerabilities in a deployment pipeline. They describe three scenarios to subvert a deployment pipeline that range from deploying an invalid image, to deploying an image without performing all the necessary checks, to having an unprotected production environment. They also indicate the possible attacks on host and network security that might lead to subversion of the deployment pipeline. The authors also classify components involved in the deployment pipeline as trustworthy and untrustworthy. Their study portrays mechanisms to create trustworthiness in a deployment pipeline. Security testing, static analysis and formal verification are some techniques that may be used to secure a pipeline. The authors propose a method to secure the pipeline by restricting the attack surface of the code base. This involves restricting the reach of parts of the code base from critical parts. The approach suggests that untrustworthy components need to communicate via trustworthy components to reach sensitive parts of the pipeline. The paper also describes a step-by-step process of how the hardening of a deployment pipeline can be performed. The result of the process is re-architecture of untrustworthy components to have restricted access or converting them into trustworthy components.

Table IV
TOOLS FOR INTEGRATING SECURITY IN DEVOPS [1]

Category	Tool
Security Tools	Snorby threat stack Tripwire Snort
Monitoring/Alerting tools	New Relic Nagios Icinga Graphite Ganglia Cacti PagerDuty Sensu
Logging Tools	PaperTrail Logstash loggly Splunk SumoLogic

E. Tools for SecDevOps

Farroha et al. [1] identified a set of tools that could be integrated to DevOps to support security, monitoring, and logging. Table IV lists these tools. Also, Schneider [6] provides some examples of open source tools that enable security in DevOps environments. Table V lists these tools.

Storms [7] showed how security features available in open source software, such as Git, Chef and Jenkins can be utilized to include security into the development and deployment processes. He also suggests the use of a set of monitoring and logging tools, which are included in Table V.

F. Software configuration

Rahman et al. [2] suggest that the relationship between the use of automation activities in DevOps environments and its influence on the software quality could be a useful line of research for DevOps enthusiasts. They also suggest that the relationship between collaboration, the use of security activities and security practices as a scope for future research.

Table V
SECURITY TOOLS USEFUL IN DEVOPS ENVIRONMENTS - INDUSTRY PERSPECTIVE

Category	Tool
Scanning Tools	Arachni Scanner OWASP ZAP
Security Frameworks	Gauntlt
Results consolidation tools	ThreadFix OWASP Code Pulse
Monitoring tools	New Relic Pager Duty Boundry Pingdom
Logging tools	Splunk SumoLogic

Bass et al. [5] propose studying the fetching of code from third party libraries for the vulnerabilities fixes, and the security of the cloud where the image is deployed.

G. Team collaboration

Farroha et al. [1] advocate the importance of involving software stakeholders to build a secure system. The rights to protect sensitive data and ensure compliance need to be granted to stakeholders to enable security.

Rahman et al. [2] reviewed the literature about SecDevOps and found that most artifacts propose enforcing the collaboration among the security team, the development team, and the operations team for a better integration of security principles into DevOps. In addition, they found that the literature suggests training the developers to build security tools, which could then be integrated into SecDevOps processes. Rahman et al. [2] also conducted a survey with DevOps practicing enterprises. The survey revealed the enterprises' awareness towards collaboration and indicated that at least 7 out of the 9 surveyed organizations observed at least a moderate level of collaboration amongst the development team, the operations team, and the security team.

H. Availability of activity data

Farroha et al. [1] suggest how the characteristics of data in today's era of Big Data affects the policies for compliance. They enumerate 9 characteristics of Big Data: volume, variety, velocity, veracity, validity, volatility, visualization, vulnerability, and value. They believe that these characteristics influence the formulation of compliance policies. In addition, the availability of data generated by SecDevOps processes and the easy and instant access to this data raises concerns about the privacy of the developers and about the secrecy of this information.

Schneider [6] lists a set of tools that support developing secure software. The data from these tools could be utilized to identify false positives, to develop custom logic, to flag unstable builds, to classify the severity of attacks on builds, and to identify trends in focused areas. The data could be utilized to improve security measures.

I. Information secrecy

Rahman et al. [2] believe that collaboration among teams, if left unsupervised, would lead to deterioration of the system's security. When teams collaborate closely, the information exchanged is not restricted and this might be a threat to the system's security. Well-defined policies regarding information exchange across teams should be in place to prevent security threats due to collaboration.

Policies for data acquisition and data protection are critical for the industry. In addition, the amount of Personally Identifiable Information(PII) allowable to be collected for analysis is one of the major privacy concerns that need to be addressed. SecDevOps allows to collect data related to the development activities. Farroha et al. [1] believe that the increase in the volume of this type of data, their diversity, and their availability is alarming. They propose setting information governance policies to control the use of such data.

IV. DISCUSSION

This section summarizes the findings of the paper, discusses the impact of these findings and the limitations of the work.

A. Summary

We reviewed 5 peer-reviewed papers and 3 presentations from AppSec and RSA conferences related to SecDevOps. We observe from the literature survey that there is a trend in the industry in adopting DevOps processes. This enthusiasm is manifested by the trend of producing tools (often as open sources) that automate security activities that needs to be integrated in SecDevOps process. This will help organizations to adopt security practices in an agile way and to make security techniques more usable.

The aspects discussed in the reviewed papers are: definition, security best practices, compliance, process automation, tools for SecDevOps, software configuration, team collaboration, availability of activity data, and information secrecy. The variety of these aspects demonstrates that SecDevOps is not only about process automation and integrating security activities in DevOps processes, it concerns also the factors that support that integration (i.e., tools, teams collaboration, and configuration management), and side effects of such integration, i.e., availability of activity data and information secrecy. The variety of these aspects suggest, despite the low number of surveyed papers, that SecDevOps is not a buzzword, or at least it is not anymore. It refers a topic that is starting to have its own merit: need, aspects, and potential foundations.

B. Impact of the results

DevOps is promoting frequent software deployments, which challenge the adoption of security activities in the process. This is pushing for developing tools that automate security activities to streamline SecDevOps processes. This work demonstrates that SecDevOps is more than integrating security best practices in DevOps and automating security activities. SecDevOps implies also, for example, strengthening the collaboration between the development teams, the operation teams, and the security teams.

The aspects of SecDevOps that we identified in this paper could be used as a research map to address the challenges related to the adoption of SecDevOps. The result could be a basis for researchers who want to work on open problems affecting the security of DevOps organizations, and provide them with an overview of the current state of research in SecDevOps.

We observed from the study that collaboration among the development team, the security team and the operation team is crucial to the success of SecDevOps. However, collaboration implies sharing information, which may threaten e.g., the trade secrets of organizations. Organizations need to define policies for sharing information among the participants in SecDevOps processes to ensure that critical information is not leaked during collaboration.

The integration of development processes to operation processes and increased collaboration among the development team, the operation team, and the security team would allow to collect data about the process. The data could be used, for example, to identify weaknesses, assess the performance of the teams, and check compliance with standards. The data could be also be used to develop tools that support the development team in their tasks, such as tools to mine patterns for false positives. Unfortunately, such data may include information private to the process participants. Use of such data may lead to violation of their privacy. Research to ensure privacy of developers and secrecy of information (such as trade secret) should be performed.

C. Limitations of the study

This study has several limitations. First, we identified only 5 relevant publications out of 66 publications found by the search. We cannot assure the completeness in the set of relevant publications related to SecDevOps, some may have been missed. This also applies to presentations in RSA and APPSEC conferences. The number of relevant publications is low because the field is relatively young.

Second, the extraction of data from the papers and presentations may be flawed due to human perception and understanding of the text. We were careful in the coding, we read the text several times, and we performed peer-reviews to reduce the impact of this threat. The extracted data were grouped into SecDevOps aspects (see Table III). The two authors discussed the data and the aspects classification to have more objective opinions. However, we acknowledge that we may have missed some aspects.

V. Conclusion

We surveyed in this paper publications and presentations related to SecDevOps. We found that the community is currently investigating the following aspects: definition, security best practices, compliance, process automation, tools for SecDevOps, software configuration, team collaboration, availability of activity data and information secrecy. The variety of these aspects demonstrates that SecDevOps is not a buzzword. We believe that the scope of research on SecDevOps should move from integrating security activities into DevOps processes towards addressing related concerns, such as collaboration, information secrecy, and learning from available process data.

We believe that organizations adopting SecDevOps should at least harden their deployment environment, implement supervised collaborations between teams, restrict the access of untrustworthy components of a DevOps environment, and ensure security practices are followed during the execution of DevOps activities.

As future work, we are performing a case study on automating a manual waterfall-based deployment process into a secure DevOps process for a leading software organization. Organizations willing to make the change to DevOps definitely need to consider the impacts on the security of their software and adapt essential security measures to implement secure DevOps.

Acknowledgment

This work was supported, in part, by the Hessian LOEWE excellence initiative within CASED.

References

[1] B. Farroha and D. Farroha, “A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment,” in *Proc. of the 2014 IEEE Military Communications Conference*, (Baltimore, MD, USA), pp. 288–293, Oct 2014.

[2] A. A. U. Rahman and L. Williams, “Software Security in DevOps: Synthesizing Practitioners’ Perceptions and Practices,” in *Proc. of the 2016 International Workshop on Continuous Software Evolution and Delivery (CSED)*, (Austin, TX, USA), May 2016.

[3] M. Mattetti, A. Shulman-Peleg, Y. Allouche, A. Corradi, S. Dolev, and L. Foschini, “Securing the infrastructure and the workloads of linux containers,” in *Proc. of the 2015 IEEE Conference on Communications and Network Security (CNS)*, (Florence, Italy), pp. 559–567, Sept 2015.

[4] S. Cash, V. Jain, L. Jiang, A. Karve, J. Kidambi, M. Lyons, T. Mathews, S. Mullen, M. Mulsow, and N. Patel, “Managed infrastructure with IBM Cloud OpenStack Services,” *IBM Journal of Research and Development*, vol. 60, pp. 6:1–6:12, March 2016.

[5] L. Bass, R. Holz, P. Rimba, A. B. Tran, and L. Zhu, “Securing a Deployment Pipeline,” in *Proc. of the Third International Workshop on Release Engineering*, (Florence, Italy), pp. 4–7, 2015.

[6] C. Schneider, “Security DevOps - staying secure in agile projects,” in *OWASP AppSec Europe*, (Amsterdam, Netherlands), 2015.

[7] A. Storms, “How security can be the next force multiplier in devops,” in *RSAConference*, (San Francisco, USA), 2015.

[8] S. de Vries, “Continuous Security Testing In a DevOps World,” in *OWASP AppSec Europe*, (Cambridge, UK), 2014.

[9] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic Mapping Studies in Software Engineering,” in *Proc. of the 12th International Conference on Evaluation and Assessment in Software Engineering*, (Bari, Italy), pp. 68–77, 2008.

[10] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[11] CA Technologies, “Devops: The Worst-Kept Secret to Winning in the Application Economy.” http://rewrite.ca.com/us/articles/devops/research-report--devops-the-worst-kept-secret-to-winning-in-the-\application-economy.html. Accessed: 2016-06-25.