Department of Science and Information Technology

DevSecOps Metrics: Learning from Academics and Professionals

Luís Filipe Bispo Prates

Dissertation submitted in partial fulfilment of the requirement for the Degree of

Master in Telecommunications and Computer Engineering

Supervisor:
Dr. Rúben Filipe de Sousa Pereira,
Assistant Professor ISCTE-IUL

September 2019

Acknowledgments

I express my gratitude to Professor Rúben Pereira, for the opportunity of doing my thesis with him and most of all I want to thank, his posture of openness and support, making it easy to work.
I also want to thank my wife, Lina, for her unconditional support. It was crucial to have her support and understanding to complete this work.

Resumo

Ao longo dos anos, várias são as abordagens que tem sido adotadas como processo de desenvolvimento de Software, tais como o modelo em Cascata e o desenvolvimento Ágil, mais recentemente o termo DevOps foi introduzido, refere-se a uma abordagem que junta elementos da equipa de desenvolvimento e operações na mesma equipa, de modo a que exista uma coloboração mais próxima e partilha de conhecimento entre estes elementos, com o intuito de se atingir entregas do Software em desenvolvimento com tempos menores, com mais frequência e qualidade.

DevSecOps é uma abordagem ao processo de desenvolvimento de Software emergente que junta elementos da equipa de segurança à equipa de DevOps, trazendo práticas de segurança para o ciclo de desenvolvimento de Software.

As práticas de segurança são cada vez mais importantes no ciclo de desenvolvimento de software pois visam a evitar violações de dados e verificar o cumprimento da lei. Mais, ganharam extrema importância para as organizações visto que as mesmas têm por obrigação a proteção de dados dos seus clientes.

Este estudo pretende identificar métricas, que podem ser utilizadas pelas equipas de modo a medir a eficiência da implementação de DevSecOps nas suas organizações.

Para identificar essas métricas, este estudo foi realizado usando como metodologia de investigação uma Ciência de Design, esta metodologia caracteriza-se por ser uma pesquisa orientada a resultados, tendo sido escolhida, com o objetivo de produzir um artefacto, contendo, as métricas para DevSecOps mais relevantes.

Foi possível identificar 9 métricas para DevSecOps, sugeridas por profissionais e académicos da área estando estas listadas no artefacto produzido por este estudo.

Mais, foram conduzidas entrevistas com os profissionais de DevSecOps com o intuito de avaliar a utilidade das métricas. Com a ajuda das entrevistas, foi possível identificar as métricas utilizadas pelos profissionais e determinar as mais úteis e relevantes. Os entrevistados sugeriram 3 métricas adicionais perfazendo assim 12 métricas incluídas neste documento.

Palavras chave: DevOps, DevSecOps, Métricas DevSecOps, SecDevOps, Ciência de Design

Abstract

DevSecOps is an emerging paradigm that breaks the Security team silo into the DevOps team, adding security practices to the Software Development Lifecycle (SDL) from inception. Security practices, in SDL, are important to avoid data breaches, guarantee compliance with the law and for organizations, it is an obligation to protect customer data. This study aims to identify metrics teams can use to measure the effectiveness of DevSecOps implementation inside organizations. To that end, this study was conducted using a Design Science Research (DSR) as its research methodology, with the intent of producing an artefact containing the most relevant DevSecOps metrics. A total of nine DevSecOps metrics purposed by professionals and academics were identified and listed on the artefact produced by this study. Interviews were conducted with DevSecOps professionals as a method of evaluating if the identified metrics were useful. Through the interviews, it was possible to identify the metrics that are being used by professionals and which are the most useful. Interviewees purposed three additional metrics. This study identifies a total of twelve metrics that can be used to measure effectiveness in DevSecOps.

Keywords: DevOps, DevSecOps, DevSecOps Metrics, SecDevOps, Design Science Research

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Acronyms

ALM	–	Application Lifecycle Management
CD	–	Continuous Delivery
CI	–	Continuous Integration
CMMI	–	Capability Maturity Model Integration
DevOps	–	Software Development and Information Technology Operations
DevSecOps	–	Software Development, Information Technology Security and Information Technology Operations
DSR	–	Design Science Research
IT	–	Information Technology
MLR	–	Multivocal Literature Review
QRI	–	Qualitative Research Interview
RM	–	Research Methodology
RQ	–	Research Question
SDL	–	Software Development Lifecycle
SLR	–	Systematic Literature Review
URL	–	Uniform Resource Locator

1 – Introduction

Nowadays there is a trending approach within Information Technology (IT) called DevOps that from a high-level perspective is defined has the merging of the Development team and Operations team into one. This approach has proven productivity gains and DevOps professionals feel their work has more impact and it is recognized by all the organization (Silva, Faustino, Pereira, & Mira Da Silva, 2018). DevOps increases both deployment frequency and the pace by which companies can serve their customers without compromising the quality of deliveries (Mohan & Othmane, 2016) and accomplishes this by taking advantage of automated development, deployment, and infrastructure (Smeds, Nybom, & Porres, 2015) processes supported by engineering practices such as Continuous Integration (CI) and Continuous Delivery (CD).

DevOps has indeed influenced software development but faster development cycles and the increase of deployments that DevOps promises in conjunction with new engineering practices and tools may compromise security. Security concerns related with DevOps are discussed on (Rahman & Williams, 2016) other research focuses on security on CI/CD pipelines (Bass, Holz, Rimba, Tran, & Zhu, 2015). From these researches, the term DevSecOps and other aliases were coined (Mohan & Othmane, 2016). DevSecOps is defined as the integration of security practices into DevOps (Myrbakken & Colomo-Palacios, 2017). This term is still recent but already is consider has topic having its merit (Mohan & Othmane, 2016).

This research aims to study the scientific developments on DevSecOps and elicit a set of metrics grounded on professional and academics viewpoints, so organizations can monitor DevSecOps. Metrics are important to improve the rigor of measurement in both Software Engineering and Information Systems fields and proposing such measures opens a debate for better understanding of the topic under discussion (Fenton & Bieman, 2015).

To achieve the aim of this study a Design Science Research (DSR) was developed to produce an artefact with relevant metrics for DevSecOps.

The rest of this document is organized as such. Chapter 2 gives theoretical background on DevOps and DevSecOps, Chapter 3 describes the research methodology (RM), Chapter 4 describes the Design and Development phase of the selected RM, Chapter 5 evaluates the findings of Chapter 4, Chapter 6 concludes the paper.

1.1. Motivation

Metrics are important to improve the rigor of measurement in both Software Engineering and Information systems fields and proposing such measures opens a debate for better understanding of the topic under discussion (Fenton & Bieman, 2015). One of the principles found in DevOps and DevSecOps is measuring. DevSecOps encourages the development of metrics that track threats and vulnerabilities throughout the software development lifecycle. Applying automatic security controls to the software development process provides development teams with metrics capable of tracking threats and vulnerabilities, allowing the organization with insights on the quality of software being developed (Myrbakken & Colomo-Palacios, 2017).

1.2. Definition of Objectives

Based on what was described before it was established the importance of having metrics has a way to better understand a topic under discussion and for that reason, this study aims to identify the most relevant DevSecOps metrics. Therefore, this work aims to obtain information about which metrics associated with DevSecOps are already identified by academics and professionals and the value they bring to development teams and organizations.

2 – Theoretical Background

2.1. DevOps

DevOps literature shows that defining the term has been hard. DevOps most typical description is Development plus Operations, but this description is not enough to explain DevOps (Smeds, Nybom, & Porres, 2015). Roche provides a good summary of the different viewpoints of what is DevOps. For some, it is a specific job that requires development and IT operational skills for others DevOps is more than that having also cultural aspects (Roche, 2013) (Walls, 2013). Those who think that the term is more than a specific job defend the existence of four perspectives: collaboration, automation, sharing and measurement (Bang, Chung, Choh, & Dupuis, 2013) (Lucy Ellen Lwakatare, 2015).

DevOps is a set of engineering practices influenced by cultural aspects and supported by technological enablers (Smeds, Nybom, & Porres, 2015). DevOps capabilities according to current literature (Smeds, Nybom, & Porres, 2015) (Virmani, 2015) are listed in Table 1.

Table 1 - DevOps Capabilities

DevOps Capabilities
Continuous planning
Continuous integration and testing
Continuous release and deployment
Continuous infrastructure monitoring and optimization
Collaborative and continuous development
Continuous user behavior monitoring and feedback
Service failure recovery without delay

DevOps appearance is a direct result of applying to the IT value stream the principles and lessons learned from Lean, Theory of Constraints, the Toyota Production System and Agile (Kim, Humble, Debois, & Willis, 2016). Agile principles point out the need of having small and self-motivated teams, that incrementally deliver small batches of working software (Beck, Martin, Hunt, & Fowler, 2001). Lean principles give emphasize on how to create value for the customer by creating flow and pull and quality at the source (Womack & Daniel, 1996).

In summary, DevOps principles are derived from Lean and Agile. DevOps is as a completely new organizational mindset that replaces siloed units with cross-functional teams. DevOps achieves this by taking advantage of automated development, deployment, and infrastructure

and enables teams to continuous work and deliver operational features (Ebert, Gallardo, Hernantes, & Serrano, 2016). Figure 1 represents the DevOps flow.

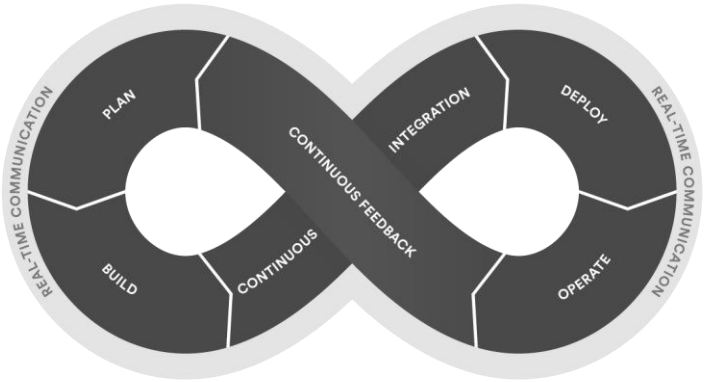

Figure 1 - DevOps flow *(Altassian DevOps, 2019)*

2.2. DevSecOps

The same way that we can say DevOps is Development and Operations merged we can say that DevSecOps is Development, Security and Operations merged. DevSecOps is defined in literature as the integration of security processes and practices into DevOps environments and seen as a necessary expansion to DevOps (Myrbakken & Colomo-Palacios, 2017).

The terms “DevSecOps”, “SecDevOps”, “SecOps”, “RuggedOps”, “Security in Continuous Delivery”, and “Security in Continuous Deployment” are all aliases to DevSecOps (Rahman & Williams, 2016). In current literature is already possible to find a set of practices for DevSecOps (Myrbakken & Colomo-Palacios, 2017). Continuous Testing, Security as Code, Threat modelling, Risk analysis, Monitoring and logging and Red Team security drills. Continuous Testing is the practice of having automatic security controls throughout the software development lifecycle, continuously detecting for defects in code changes with the possibility of an automatic rollback if necessary (Virmani, 2015) (Myrbakken & Colomo-Palacios, 2017). Security as Code is the practice of having security policies like network configurations codified integrated with software development lifecycle (Myrbakken & Colomo-Palacios, 2017). Monitoring and logging practices are the practice of observing various quality parameters associated with the implemented controls and measure their effectiveness (Myrbakken & Colomo-Palacios, 2017) (Virmani, 2015). Threat Modeling is the activity attacking your system on paper and using this information to identify, describe, and categorize threats to your system (Rahman & Williams, 2016) (Myrbakken & Colomo-Palacios, 2017). Risk Analysis is the activity of creating security design specifications from the first planning and before every iteration (Rahman & Williams, 2016) (Myrbakken &

Colomo-Palacios, 2017). Red Team security drills is the practice of creating a proactive team that performs a malicious attack on deployed software with the intent of finding and exploiting vulnerabilities, finding security flaws and helping the organization find solutions (Myrbakken & Colomo-Palacios, 2017) (Ray, Vemuri, & Kantubhukta, 2005).

The two main benefits of DevSecOps are having fast and scalable security controls by Automating Security and having security controls since the beginning of the development process by Shifting Security to Left, this means bringing security experts involved from the beginning to plan and integrate security controls (Myrbakken & Colomo-Palacios, 2017) but also to share knowledge with other team elements making them more security-aware.

3 – Research Methodology

This section, the research methodologies adopted for this study are introduced and explained. Design Science Research (DSR) is the main methodology for this study since the study intends to produce an artefact with new knowledge related to DevSecOps. It was also necessary to use Multivocal literature review (MLR), through the MLR it is possible to collect data on DevSecOps from both academic literature and grey literature. Interviews are also used to evaluate the findings of the information obtained from the MLR.

3.1. Design Science Research

As established before, DSR is the main research methodology for this study. DSR is defined as a rigorous process to design artifacts to solve observed problems, to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences (Hevner, March, & Park, 2004).
DSR, is composed by the following six activities (Peffers, Tuunanen, Tuure, Rothenberger, & Chatterjee, 2007).

Problem Identification and Motivation – Consists of defining the research problem and motivation for a solution.

Definition of the Objectives – Defining objectives for a solution based on the problem definition.

Design and Development – Activity where the artefact is created, and a research contribution is embedded.

Demonstration – Consists of showing the artefact ability to solve on or more instances of the problem. For this study this activity will not be considered.

Evaluation – Consists of evaluating how well the created artefact supports a solution to the problem.

Communication – In this activity is where the artefact is shared and its purpose, rigor of design, and effectiveness are presented to relevant audiences such as practicing professionals.

Figure *2*, represents DSR activities flow, with each activity already containing a description for this study.

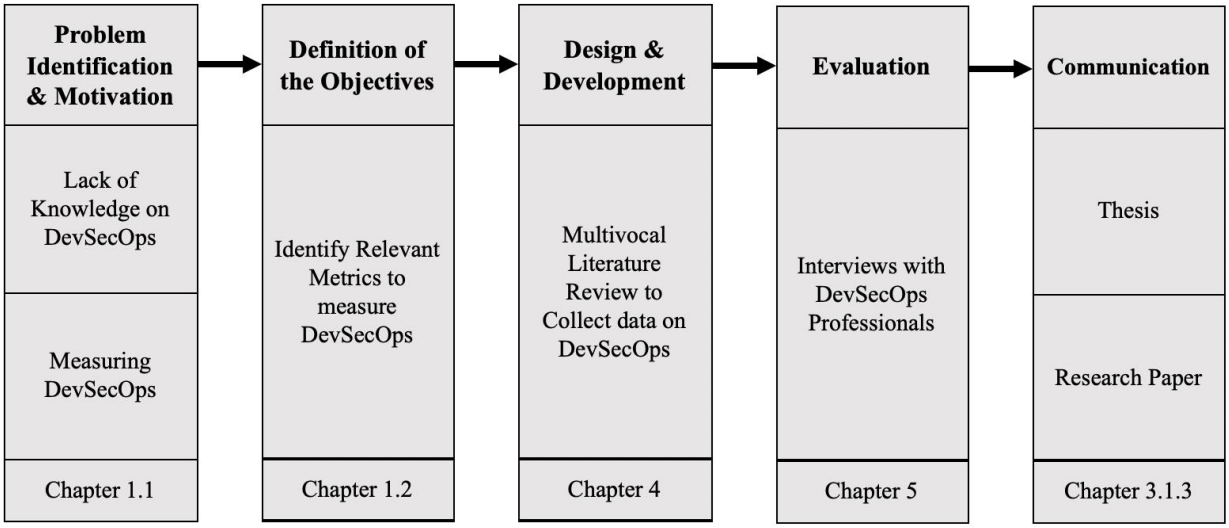

Figure 2 - DSR Activities Flow

The problem identification and motivation as one of the DSR activities is already introduced and established in chapter 1.1.

The definition of objectives as one of the DSR activities is already covered in chapter 1.2.

3.1.1. Design & Development

This activity of the DSR will be executed through an MLR. There are no guidelines to perform an MLR since MLR is a form of Systematic Literature Review (SLR) the MLR is planned as an SLR but including “grey literature”. MLR is very important for a practitioner-oriented field such as Software Engineering, because helps synthesizing and combining both the state-of-the art and practice (Garousi, Michael Felderer, & Mäntylä, 2016) . Other studies also used MLR to establish ground work for DevSecOps (Myrbakken & Colomo-Palacios, 2017), there is also other examples of applying MLR in Software engineer. (Calderón, Ruiz, & O’Connor, 2017) (Sánchez-Gordón & Colomo-Palacios, 2018).

SLR is a type of literature review that is used to identify, evaluate and interpreting all available research relevant to a specific question (Kitchenham, 2004). Kitchenham procedures for performing systematic reviews (Kitchenham, 2004) will be adopted by the authors. Figure 3 details how this research steps map the three phases proposed by Kitchenham (Kitchenham, 2004).

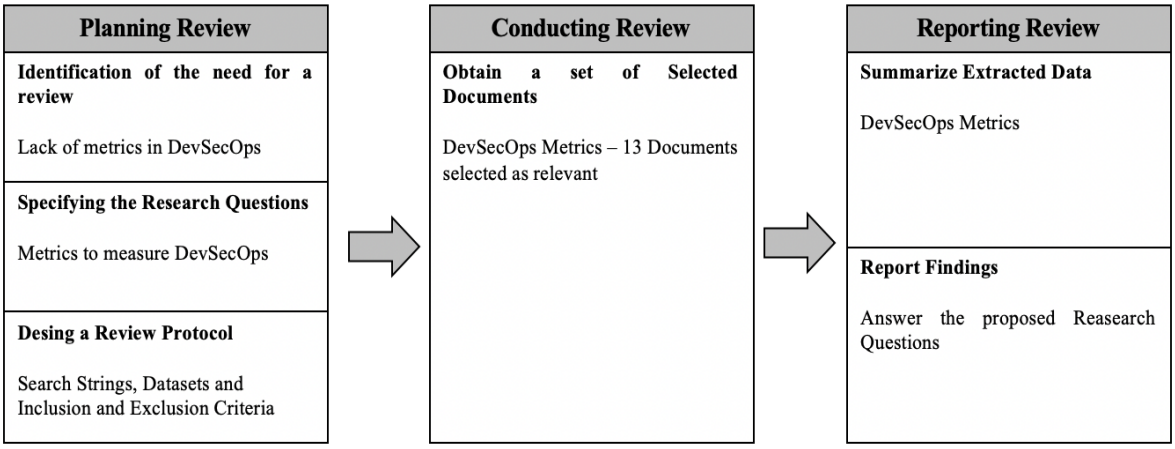

Figure 3 – MLR Review Steps

Planning Review – this phase consists in three steps. First step is identifying the need and motivation for the review, second step is specifying the research questions that are going to be addressed and answered by the review. Although this are steps for the MLR it is already established on chapter 3.3.1. Final step designing a review protocol with the constraints that are going to be applied in the review, this protocol is presented on chapter 4.1.

Conducting Review – this phase consists in applying the designed review protocol. This phase is detailed on chapter 4.2.

Reporting Review – final phase of the review is summarizing the extracted data from the selected literature and report findings. This phase is presented on chapter 4.3.

3.1.2. Evaluation

To evaluate these study findings a Qualitative Research Interview (QRI) approach is used in collaboration with DevSecOps professionals working in the Portuguese IT sector.
This type of interview is a professional conversation based on daily life to obtain descriptions of the life world of the interviewees for interpreting the meaning of the described phenomena (Kvale, 1996).
Interviewing has been used through human history in various forms and for different purposes, for instance, Ancient Egyptians used interviewing for demographic studies (Babbie, 1992) and

Thucydides interviewed veterans of the Peloponnesian Wars to write its historical account (Kvale, 1996).

Interviews may take different question formats, ranging from more open questions (unstructured) to closed questions (structured), were the first is used as an attempt to understand and probe complex topics by demonstrating greater interest in the interviewee's point of view (Bryman, 2012) and the second reflects the researcher's concern (Bryman, 2012) represented by a limited set of preestablished questions giving a low margin of response variation (Fontana & Frey, 1994).

QRI has two main types of interview unstructured and semi-structured (Bryman, 2012).

For this study, the type of interview chosen was semi-structured, this means that interviews were conducted using an interview guide, but still the interviewees had a lot of room to reply has they saw fit. With the semi-structured approach, it is also expected that questions that are not present on the purposed questionnaire. The questions that will appear during interviews are based on things that the interviewee said, and the interviewer decided to follow-up.

3.1.3. Communication

Has previously established, this activity consists of specifying how the produced artefact is going to be shared. The artefact produced by this study is intended to be shared through the submission of research paper as well as the final document of this thesis and its public defense.

Part of this research was already published on Information Systems: Research, Development, Applications, Education. SIGSAND/PLAIS 2019. Lecture Notes in Business Information Processing Volume 359. Springer. (Prates, Faustino, Silva, & Pereira, 2019)

4 – Design & Development

This section details the execution of the design and development phase of the DSR to produce an artefact. It will be done through the application of the MLR. The review protocol of MLR is present here. Conducting and reporting review phases of the MLR are also presented in this chapter. The motivation for this work is presented, followed by the Research Question (RQ) this study intended to address and answer. Finally, the Review Protocol is proposed.

4.1. Review Protocol

The first stage of the review protocol is literature search, a search string must be defined and applied in the chosen data sources with the intent of retrieving the highest possible number of studies related with the proposed research questions.

The **search string** is a set of keywords related to DevSecOps. Search terms used in this research are presented in Table 2.

Table 2 - Search Terms

Main Term	Keywords
DevSecOps or SecDevOps	Definition, Challenges, Metrics, Measuring, Adoption

The chosen academic data sources for the this MLR are four well-known academic databases.

- IEEEXplore (www.ieeexplore.ieee.org/Xplore/)
- ACM Digital Library (www.portal.acm.org/dl.cfm)
- SpringerLink (www.springerlink.com/)
- Google Scholar (https://scholar.google.com/)

For searching grey literature Google Search (www.google.com) was chosen as the database.

Inclusion and exclusion criteria are applied to literature from both data sources. Criteria is presented in Table 3.

After applying the inclusion and exclusion criteria, remaining documents are read with the intent of obtaining the final selection of studies and at this point it is possible to conduct the review. The review protocol is represented in Figure 4.

Table 3 - Inclusion and Exclusion Criteria

Inclusion Criteria	**Exclusion Criteria**
Written in English	Not Written in English
Publication Date after 2013, inclusive	Publication date before 2013
Scientific papers in conferences or Journals, Blogs	Inaccessible Literature
Explicit discusses DevSecOps	Duplicated
Limit results to first 3 pages of Google Search	Vendor Tool Advertisement
	Unidentified Author
	No Publication date

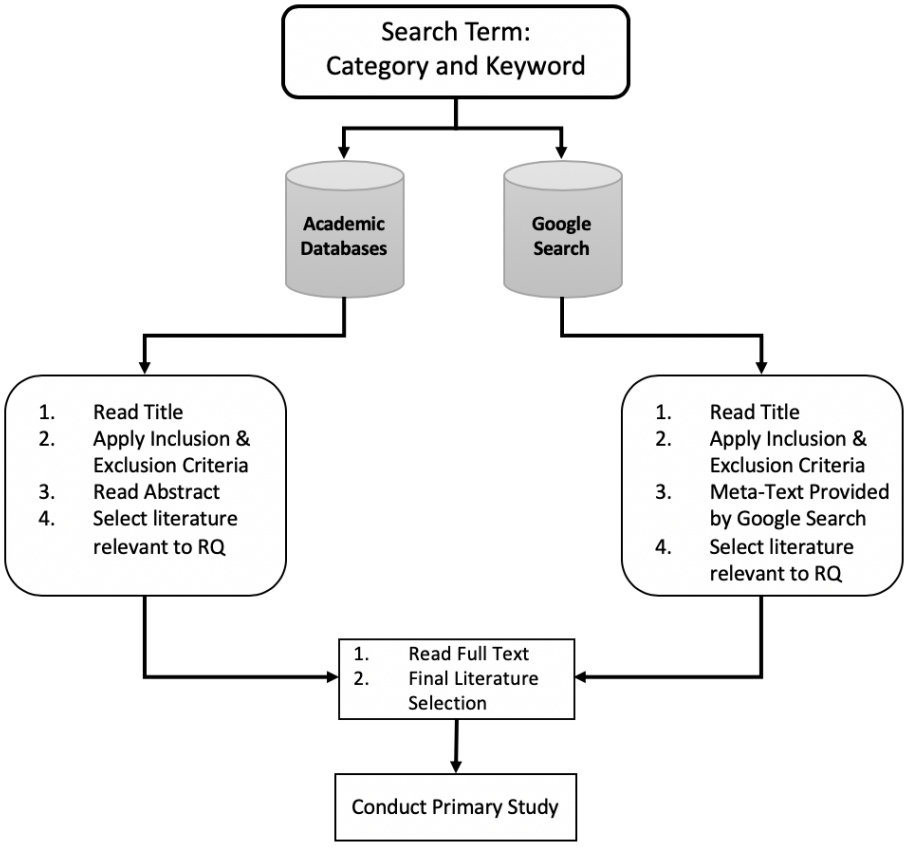

Figure 4 - Review Protocol adapted from *(Myrbakken & Colomo-Palacios, 2017)*.

4.2. Conducting the Review

This section corresponds to the second phase of the MLR and consists of applying the previously defined review protocol.

The first step was to run the search string composed by the search terms defined in Table *2* . After running the search terms on the selected data sources 558 articles were obtained. Distribution of articles by category is illustrated in Figure 5 and by database illustrated in Figure 6. The searches on the data sources only considered articles published after 2013.

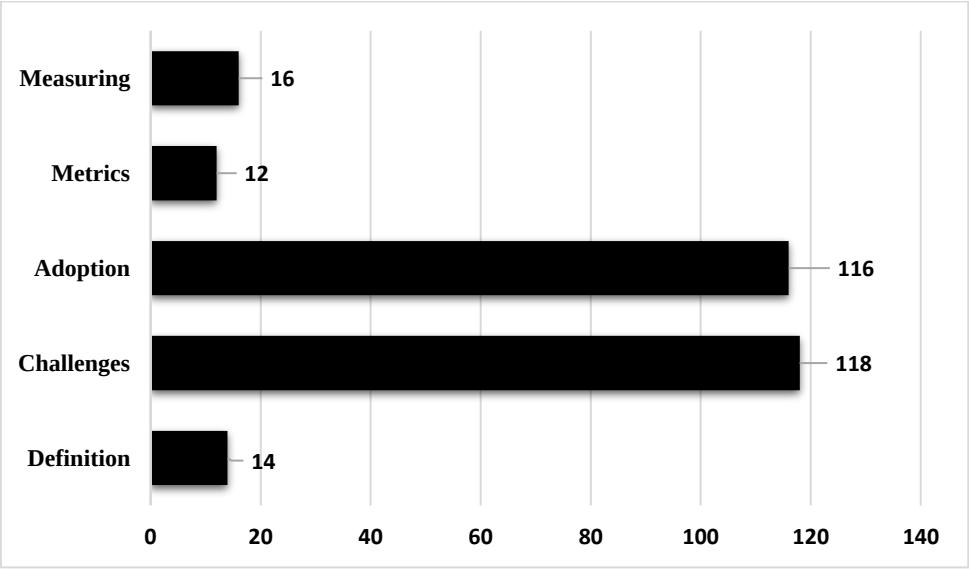

Figure 5 - Distribution of articles by Search Term.

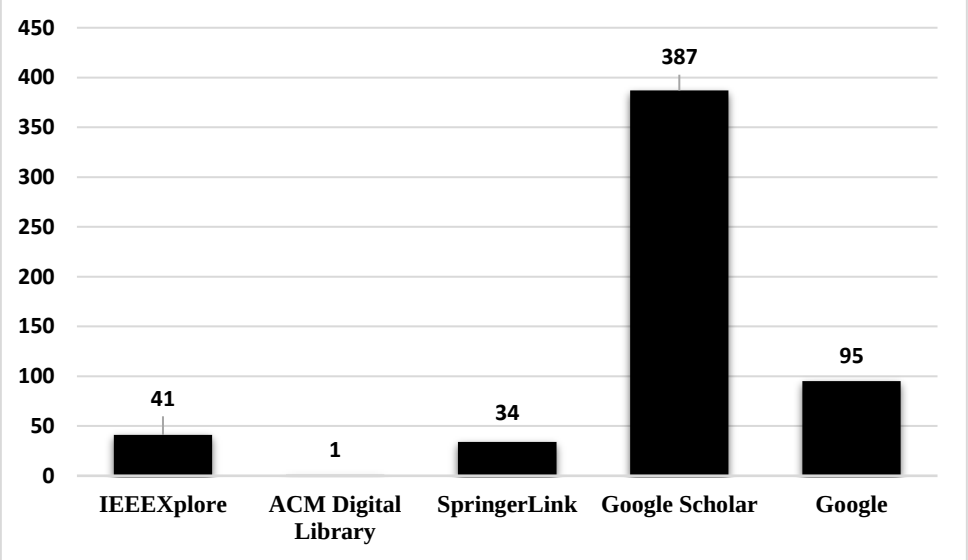

Figure 6 - Distribution of articles by Database.

Next step of the review protocol is applying the inclusion and exclusion criteria.

4.2.1. Academic Databases

The first step is ensuring that there are no duplicated articles. Removing the duplicates consists of a two-step approach.

1. Remove Duplicates from articles retrieve from the same database.
2. Remove Duplicates between the four academic databases.

Studies information exported from each data source were in different formats. Table 4 shows the export format from each academic data source.

Table 4 - Academic Databases Export Format

Data source	Format
ACM	type, id, author, editor, advisor, note, title, pages, article_no, num_pages, keywords, doi, journal, issue_date, volume, issue_no, description, month, year, issn, booktitle, acronym, edition, isbn, conf_loc, publisher, publisher_loc
IEEE	Document Title, Authors, Author Affiliations, Publication Title, Date Added To Xplore, Publication_Year, Volume, Issue, Start Page, End Page, Abstract, ISSN, ISBNs, DOI, Funding Information, PDF Link, Author Keywords, IEEE Terms, INSPEC Controlled Terms, INSPEC Non-Controlled Terms, Mesh_Terms, Article Citation Count, Reference Count, Copyright Year, License, Online Date, Issue Date, Meeting Date, Publisher, Document Identifier
SpringerLink	Item Title, Publication Title, Book Series Title, Journal Volume, Journal Issue, Item DOI, Authors, Publication Year, URL, Content Type
Google Scholar	Title, Publication, Authors, Year

To ensure that the removal of duplicated studies is accurate, a database schema was created on PostgreSQL and a Table with the following attributes Title, Publication, Authors, Year were included since this is enough to identify a duplicated study. Insertion scripts that converted from the original format to the new database format were created for each data source, except for Google Scholar that already respected the desired format. After removing duplicated articles and applying the remaining items on the inclusion and exclusion criteria a total of 40 studies from academic databases were flagged as relevant to the research question. Table 5 details the number of academic articles remaining after each phase.

Table 5 - Academic articles remaining after each phase.

Phase	Number of Articles
Duplicated	62
Read Title	51
Inclusion & Exclusion Criteria	49
Read Abstract	40
Full-Text Read and Final Selection	2

4.2.2. Grey Literature

The approach to filtering the grey literature is like the one used on the academic databases. The first step is removing the duplicated, this was achieved by filtering the duplicated uniform resource locator (URL) in Excel. After removing the duplicated articles, inclusion and exclusion criteria is applied a total of 56 were flagged as relevant to the research question. Table 6 details number of grey literature articles remaining after each phase.

Table 6 - Grey Literature Articles remaining after each phase.

Phase	Number of Articles
Duplicated	234
Read Title	92
Inclusion & Exclusion Criteria	65
Meta Text Provided by Google	56
Full-Text Read and Final Selection	11

4.2.3. Final selection of studies

From the pool of literature flagged as possible relevant to the research question, all texts were read to further decide the document's relevance, and a total of 15 were obtained as relevant to our study.

4.2.4. Data Extraction Analysis

As can be seen in Figure 7, both in academic data sources and grey literature sources the interest on the topic rose considerably after 2017.

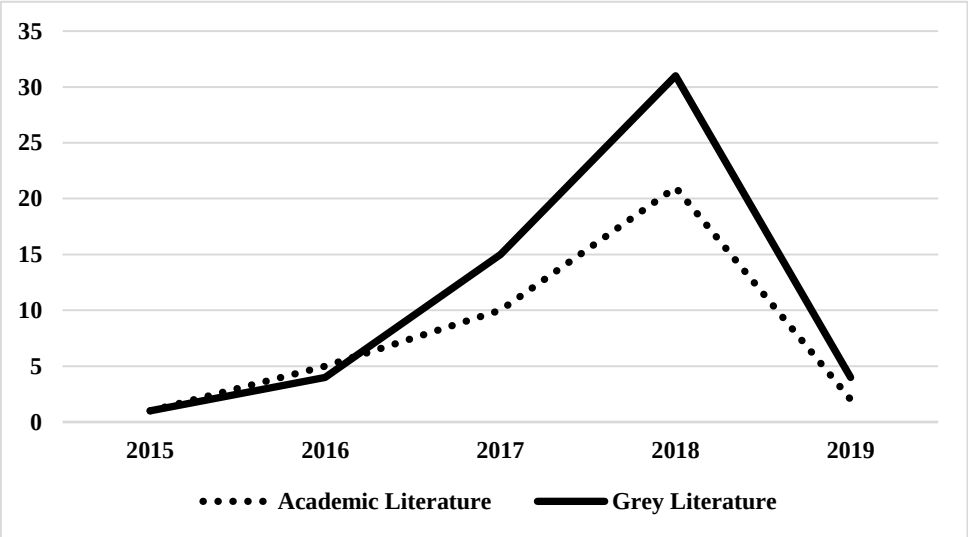

Figure 7 - Academic and Grey Literature articles flagged as relevant by year.

The year 2019 has fewer studies because this review only took into consideration the studies until the beginning of the same year.

4.3. Reporting the Review

This MLR phase presents the research done on DevSecOps to identify metrics. Google Scholar, Google Search, IEEE Explore, Springer and ACM Library were used to locate literature.
After applying the inclusion and exclusion criteria, 13 articles were found to be relevant to the defined search terms. Only 2 of those were academic research papers. The remaining 11 consisted of blogs and articles. Based on the literature review, 9 relevant metrics were reported by professionals. Table 7 lists and describes the identified metrics.

Table 7 - DevSecOps Metrics

Metric	References	Description	Goal
Defect Density	(Chickowski, 2018) (Humphrey, 2018) (Jerbi, 2017) (Hsu, 2018)	This metric can be defined has the number of confirmed defects detected in software/component during a defined period of development/operation divided by the size of the software/component.	Helps security teams and developers negotiate reasonable goals to reduce defect density over time.
Defect Burn Rate	(Chickowski, 2018) (Crouch, 2017) (Casey, 2018) (Jerbi, 2017)	Indicates how quickly the team is addressing defects.	Measuring development team productivity solving defects.
Critical Risk Profiling	(Chickowski, 2018) (Woodward, 2018) (Vijayan, s.d.) (Raynaud, 2017) (Hsu, 2018) (Paule, 2018)	Is the relation between issue criticality and the value of that vulnerability to possible attackers	The goal of this is metric is help prioritize the order development teams should address issues.
Top Vulnerability Types	(Chickowski, 2018) (Jose, 2018) (Paule, 2018)	Lists the top vulnerability types and the most recurring ones.	Helps planning training provided to developers accordingly and capacitate them with knowledge to handle and mitigate returning vulnerabilities.
Number of Adversaries per Application	(Chickowski, 2018) (Paule, 2018)	Identifies how many adversaries an application might have this metric is associated with the practice of Threat Modelling and Risk Analysis.	The goal is to identify the applications inside an organization that are more exposed to possible attacks and prepare accordingly.
Adversary Return Rate	(Chickowski, 2018)	Measures how often an adversary will use the same strategy and procedures.	Helps define appropriate training to better handle these attacks.
Point of Risk Per Device	(Humphrey, 2018)	Tracks the number of vulnerabilities per server.	Helps prioritize these vulnerabilities according to their criticality giving special attention to the ones that are most exposed to attack from the internet.
Number of Continuous Delivery Cycles Per Month	(Humphrey, 2018) (Crouch, 2017) (Casey, 2018) (Rao, 2017)	Number of successful deploys to production per month.	Measures how quickly code changes can be deployed to production.
Number of issues during Red Teaming Drills	(Chris Romeo, s.d.) (Raynaud, 2017)	Is the number of found issues and fixed by Red Team.	Measures Red Team Effectiveness.

5 – Evaluation

In this Chapter, the evaluation method for the artefact produced in Chapter 4, is presented. Previously it was established that interviews, would be used, therefore, the Interview process and results are presented in this chapter.

5.1. Interviews

A Qualitative Research Interview (QRI) approach is used in collaboration with DevSecOps professionals working in the Portuguese IT sector.
The interviewees for this study were obtained through professional network connections, DevOps meetings and Linkedin's social network.
Interviews were done face-to-face or through remote communication. Each Interview was recorded for later revision. A total of 5 professionals were interviewed for this study.

5.2. Questionnaire

Table 8 lists the support questions used for the interviews conducted for this study.

Table 8 - Interviews Questionnaire

Question
Age
Years of Experience with IT
Current Position
Years of Experience with DevOps
Years of Experience with DevSecOps
Do you consider the metrics presented useful? Which ones?
Use any metric from the presented list? Which ones?
Do you suggest any more metrics besides the ones that are in the list? Which ones?
What is the most valuable indicator to have a Production Sign-Off?
Do you consider important to have metrics?

The average age of the interviewees is 31,4 years. Table 9 gives an overview of the interviewees.

Table 9 - Overview of Interviewees

Interviewee	Age	Current Position	Years of Experience in IT	Years of Experience with DevOps	Years of Experience with DevSecOps
A	29	Quality Assurance Engineer	5	2	1
B	34	Senior DevOps Engineer	10	5	3
C	31	DevOps Engineer	7	3	1
D	33	Senior Software Engineer	6	3	2
E	30	Security Engineer	6	2	2

Figure 8 shows the average experience of interviewees in the IT Sector and working in a DevOps and DevSecOps environment.

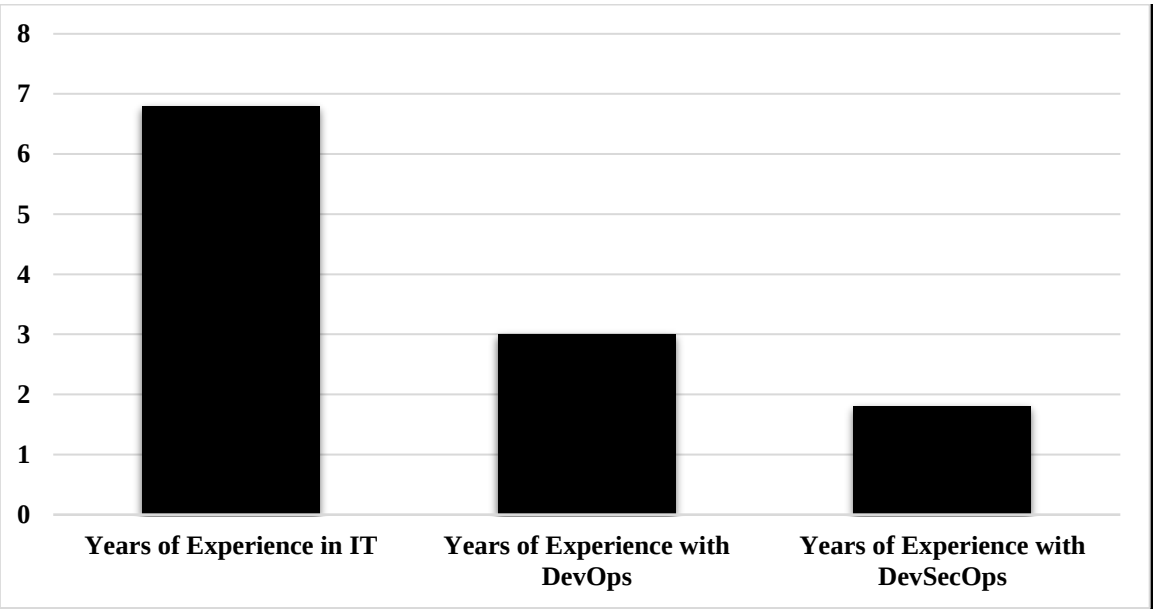

Figure 8 - Interviewees Experience Overview.

5.3. Results

This section contains the compiled results of the interviews.
The interviewees considered the metrics presented in Table 7 useful and considered the metrics in Table 10 as the most useful.

Table 10 - Most useful metrics according to interviewees

Metric
Defect Density
Top Vulnerability Types
Adversary Return Rate

The reason to consider Defect Density useful, was that allows them to know which components contain defects related to security and from there plan work solve those issues. Top Vulnerability Types and Adversary Return Rate were considered useful because helped prevent know security issues and define the appropriate training.
From the metrics in Table 7, at least one or more of interviewee reported using one of the metrics identified in Table 11.

Table 11 - Metrics reported as used by interviewee's

Metric	**Interviewee Who Reported**
Defect Density	A, B, C
Defect Burn Rate	A, B, C
Top Vulnerability Types	A, B, D, E
Number of Continuous Delivery Cycles per Month	B, C, D
Adversary Return Rate	B, C, E

Interviewees purposed additional metrics, Table 12 list metrics purposed by one or more of the interviewees.

Table 12 - Metrics purposed by interviewees

Metric	Description	Interviewee Who Purposed
Lead Time	The time that takes since works start on an item and having it deployed oi production	B, C
Automated Security Test % Pass	Percentage of automated security tests passed	A, E
Defect Escape Rate	Rate of issues found in production vs not found in quality environments	A, B

Lead Time, according to the interviewees, is important, because provides them with an estimate on how much time their team is taking to have new features in production, serving has an indicator of their effectiveness and as a time baseline to be improved.

Automated Security Test Pass percentage is important, according to the interviewees that purposed it, because it gives immediate feedback if any previous implemented security measure is failing.

Defect Escape Rate was considered important, because allows professionals to evaluate why defects are reaching Production Environment's and with that metric they can improve their tests to catch defects before reaching Production.

All interviewees consider that it is very important to have metrics and without them, they're unable to defend their work, prevent security breaches and monitor the current security status of projects.

Moreover, some comments from the interviewees (B, D, E) were also very interesting. Most of them consider that their organizations do not mind much about security and that security is often a non-functional requirement that usually everybody assumes it is handled until there is some security breach or a special request from a customer. For instance, interviewee D stated "Most of the organizations I worked for only had security related tasks after Customers during Acceptance Testing hire an external Security Team to evaluate the software security and report of that team was negative", Interviewee E mentioned that "It is often assumed that security is there by default, until something goes

wrong" and Interviewee B said that "As a DevOps professional it is only in his last organization that he started to have to worry about having mechanisms to test application security in their CI/CD pipelines, instead of having Security teams at the end of Software Development lifecycle".

6 – Conclusion

This study presents an explorative research on DevSecOps to identify metrics associated with DevSecOps that can be used to measure its effectiveness. DevSecOps is a recent topic has it was established earlier it is expected to continue to grow. It was difficult to find information regarding DevSecOps metrics special in academic literature. Even so, it was possible to obtain the following findings:

- A total of 9 DevSecOps Metrics was identified in the literature review.
- All interviewees considered important to have metrics to allow them to optimize their work and defend it.
- Interviewed Professionals consider Defect Density, Top Vulnerability Types and Adversary Return Rate as the most useful metrics.
- Interviewees purposed additional metrics Lead Time, Defect Escape Rate and Automate Security Test Pass Percentage.
- Interviewees reported using 5 of the 9 metrics presented, Defect Density, Defect Burn Rate, Top Vulnerability Types, Number of Continuous Delivery Cycles per Month and Adversary Return Rate.
- Interviewees commented that usually in their collective experience organizations do not mind much about security until something bad happens.

For professionals, the authors believe that the findings from this study can be already used as starting set of metrics to take in consideration when adopting DevSecOps, some of the metrics are easy to implement and can quickly help professionals measure their progress. As for academics, authors believe that the findings can be considered enough material has a basis study for further exploration in measuring DevSecOps, but for them the metrics do not have much practical meaning without having a project to test them.

6.1. Research Limitations

This study has two main threats to its validity. The first threat is that the literature dedicated to DevSecOps obtained in Chapter 4 is still scarce. Moreover, the literature deemed relevant with insights on metrics for DevSecOps is mostly from blogs, tech conferences.

The second threat is that interviews were conducted only with 5 professionals, it is still a new area meaning that a lot of organizations did not yet adopt DevSecOps and therefore professionals working in a DevSecOps environment are still scarce.
However, the results from this study are considered valid by the authors and of value as a basis for further research.

6.2. Future Work

Since DevSecOps is a trending topic and this study had an exploratory nature, further researches may continue by increasing the number of interviews and surveys with DevSecOps professionals to tune and complement the proposed metrics as well as what is the outcome of each one. Plus, it would also be interesting to understand what mechanisms and policies could be implemented to mitigate the security issues that the presented metrics are intended to measure.

Bibliography

Altassian DevOps. (2019). (Altassian) Obtido em 16 de 3 de 2019, de https://www.atlassian.com/devops

Babbie, E. (1992). *The Practice of Social Research.* Belmont,CA: Wadsworth Publishing.

Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. D. (2013). A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps. *2nd annual conference on Research in information technology.* New York.

Bass, L., Holz, R., Rimba, P., Tran, A. B., & Zhu, L. (2015). Securing a deployment pipeline. *Third International Workshop on Release Engineering.* New Jersey.

Beck, K., Martin, R., Hunt, A., & Fowler, M. (2001). *Agile Manifesto.*

Bryman, A. (2012). *Social Research Methods.* Oxford: Oxford University Press.

Calderón, A., Ruiz, M., & O'Connor, R. (2017). A multivocal literature review on serious games for software process standards education. *Computer Standards & Interfaces*.

Carter, K. (2017). Francois Raynaud on DevSecOps. *IEEE Software, 34*(5), 93-96.

Casey, K. (19 de June de 2018). *Enterprisers Project*. Obtido em 26 de March de 2019, de https://enterprisersproject.com/article/2018/6/how-build-strong-devsecops-culture-5-tips?page=1

Chickowski, E. (2018, May 1). *Seven Winning DevSecOps Metrics Security Should Track.* (Bitdefender) Retrieved March 25, 2019, from https://businessinsights.bitdefender.com/seven-winning-devsecops-metrics-security-should-track

Chris Romeo. (s.d.). *Techbeacon*. (Microfocus) Obtido em 3 de March de 2019, de https://techbeacon.com/devops/3-most-crucial-security-behaviors-devsecops

Crouch, A. (13 de December de 2017). *https://www.agileconnection.com*. (Agile Connection) Obtido em 26 de March de 2019, de https://www.agileconnection.com/article/devsecops-incorporate-security-devops-reduce-software-risk

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*, 94-100.

Elmore, R. F. (1991). Comment on "Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method.". *Review of Educational Research*(61), 293-297.

Fenton, N., & Bieman, J. (2015). *Software Metrics.* Boca Raton: CRC Press.

Fontana, A., & Frey, J. (1994). Interviewing: The Art of Science. Em *The Handbook of Qaulitative Research* (pp. 361-376). Tousand Oaks, California: Sage Publications.

Garousi, V., Michael Felderer, M., & Mäntylä, M. V. (2016). The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. *20th International Conference on Evaluation and Assessment in Software Engineering (EASE '16).* New York.

Gruhn, V., Hannebauer, C., & John, C. (2013). Security of public continuous integration services. *9th International Symposium on Open Collaboration.* New York.

Hevner, A., March, S., & Park, J. (2004). Design research in information systems research. *MIS Quarterly*, 75–105.

Hsu, T. (2018). *Hands-On Security in DevOps.* Birmingham: Pack Publishing.

Humphrey, A. (16 de January de 2018). *Diving into DevSecOps: Measuring Effectiveness & Success*. (Armor) Obtido em 29 de March de 2019, de https://www.armor.com/blog/diving-devsecops-measuring-effectiveness-success/

Jerbi, A. (13 de November de 2017). *InfoWorld*. Obtido em 25 de March de 2019, de https://www.infoworld.com/article/3237046/kpis-for-managing-and-optimizing-devsecops-success.html

Jose, F. (3 de July de 2018). *Effective DevSecops*. Obtido em 3 de April de 2019, de https://medium.com/@fabiojose/effective-devsecops-f22dd023c5cd

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook.* IT Revolution Press.

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews, Keele University Technical Report TR/SE-0401*. Keele: Keele University.

Kvale, S. (1996). *Interview Views: An Introduction to Qualitative Research Interviewing.* California: Thousand Oaks.

Lucy Ellen Lwakatare, P. K. (2015). Dimensions of DevOps. Em *Agile Processes in Software Engineering and Extreme Programming* (Vol. 212). Springer.

Mohan, V., & Othmane, L. B. (2016). SecDevOps: Is It a Marketing Buzzword? - Mapping Research on Security in DevOps. *11th International Conference on Availability, Reliability and Security (ARES).* Salzburg.

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: A Multivocal Literature Review. Em *Software Process Improvement and Capability Determination.* Springer.

Paule, C. (2018). *Securing DevOps — Detection of vulnerabilities in CD pipelines.* Stuttgart: University of Stuttgart.

Peffers, K., Tuunanen, Tuure, Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *J. Manage. Inf. Syst.*, 45-77.

Prates, L., Faustino, J., Silva, M., & Pereira, R. (2019). DevSecOps Metrics. *SIGSAND/PLAIS 2019. Lecture Notes in Business Information Processing.* Springer, Cham.

Rahman, A. A., & Williams, L. (2016). Software security in DevOps: synthesizing practitioners perceptions and practices. *International Workshop on Continuous Software Evolution and Delivery.* New York.

Rao, M. (6 de July de 2017). *Synopsys.* Obtido em 2 de April de 2019, de https://www.synopsys.com/blogs/software-security/devsecops-pipeline-checklist/

Ray, H. T., Vemuri, R., & Kantubhukta, H. R. (2005). Toward an automated attack model for red teams. *IEEE Security & Privacy, 3*(4), 18-25.

Raynaud, F. (June de 2017). *DevSecCon.* Obtido em 31 de March de 2019, de https://www.devseccon.com/wp-content/uploads/2017/07/DevSecOps-whitepaper.pdf

Roche, J. (2013). Adopting DevOps Practices in Quality Assurance. *Communications of the ACM, 56*(11), 8-20.

Sánchez-Gordón, M., & Colomo-Palacios, R. (2018). A Multivocal Literature Review on the use of DevOps for e-Learning systems. *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 883-888). Salamanca: ACM.

Silva, M., Faustino, J., Pereira, R., & Mira Da Silva, M. (2018). Productivity Gains of DevOps Adoption in an IT Team: A Case Study. *Designing Digitalization.* Lund.

Smeds, J., Nybom, K., & Porres, I. (2015). DevOps: A Definition and Perceived Adoption Impediments. *Lecture Notes in Business Information Processing, 212*.

Vijayan, J. (s.d.). *TechBeacon*. Obtido em 1 de April de 2019, de https://techbeacon.com/security/6-devsecops-best-practices-automate-early-often

Virmani, M. (2015). Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery. *INTECH 2015.* Pontevedra.

Walls, M. (2013). *Building a DevOps Culture.* O'Reilly Media.

Womack, J., & Daniel, J. (1996). *Lean Thinking: Banish Waste and Create Wealth In Your Corporation.* New York: Free Press.

Woodward, S. (18 de September de 2018). *BrightTalk*. Obtido em 27 de March de 2019, de https://www.brighttalk.com/webcast/499/333412/devsecops-metrics-approaches-in-2018