DevSecOps Metrics

Luís Prates[1], João Faustino[1], Miguel Silva[1] and Rúben Pereira[2],

[1] Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal

{lfbps,joao_faustino,miguel_angelo_silva}@icste-iul.pt

[2] ISTAR-IUL, Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal

{ruben.filipe.pereira}@iscte-iul.pt

Abstract. DevSecOps is an emerging paradigm that breaks the Security Team Silo into the DevOps Methodology and adds security practices to the Software Development Cycle (SDL). Security practices in SDL are important to avoid data breaches, guarantee compliance with the law and is an obligation to protect customers data. This study aims to identify metrics teams can use to measure the effectiveness of DevSecOps methodology implementation inside organizations. To that end, we performed a Multivocal Literature Review (MLR), where we reviewed a selection of grey literature. Several metrics purposed by professionals to monitor DevSecOps were identified and listed.

Keywords: DevOps, DevSecOps, DevSecOps Metrics, SecDevOps, Multivocal Literature Review

1 Introduction

Nowadays there is a trending methodology 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 methodology has proven productivity gains and DevOps professionals feel their work has more impact and it's recognized by all the organization [1]. DevOps increases both deployment frequency and the pace by which companies can serve their customers without compromising the quality of deliveries [2] . DevOps has indeed influenced software development but faster development cycles and increase of deployments that DevOps promises in conjunction with new engineering practices and tools may compromise security and this is discussed on research related with security aspects of DevOps [3] other research focus on security on CI/CD pipeline [4] from these researches the term DevSecOps and other aliases were coined [2]. DevSecOps is defined as the integration of security practices into DevOps [5]. This term is still recent but already is consider has topic having its own merit [2].

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 [6].

Since DevSecOps is a very recent topic the research methodology selected for this study is a MLR. MLR is a kind of Systematic Literature Review (SLR) [7] and is useful when trying to close the gap between academic research and professional practice [8].

The rest of this document is organized as such. Chapter 2 gives theoretical background on DevOps and DevSecOps, Chapter 3 describes the research methodology, Chapter 4 describes the literature review plan, Chapter 5 summarizes the information extracted from the analyzed publications, and discusses the results and limitations of the study, and Chapter 6 reports the findings and Chapter 7 concludes the paper.

2 Theoretical Background Review

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 [9]. Roche provides a good summary on 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 [10]. Those who think that the term is more than a specific job defend the existence of four perspectives: collaboration, automation, sharing and measurement [11] [12]. DevOps is not only culture aspects it is also a set of engineering practices influenced by cultural aspects and supported by technological enablers [9]. DevOps capabilities are 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 [9] [13] .

DevOps is a complete 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 [14].

2.2 DevSecOps

The same way that we can say DevOps is Development and Operations merged together we can say that DevSecOps is Development, Security and Operations merged together. DevSecOps is defined in literature as the integration of security processes and practices into DevOps environments and seen as a necessary expansion to DevOps [5].

The terms “DevSecOps”, “SecDevOps”, “SecOps”, “RuggedOps”, “Security in Continuous Delivery”, and “Security in Continuous Deployment” are all aliases to DevSecOps [3]. In current literature is already possible to find a set of practices for DevSecOps [5]. 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 automatic rollback if necessary [13] [5]. Security as Code is the practice of having security policies like network configurations codified integrated with software development lifecycle [5]. Monitoring and logging practices is observing various quality parameters associated with the implemented controls and measure their effectiveness [5] [13]. Threat Modeling is the activity attacking your system on paper and using this information to identify, describe, and categorize threats to your system [3] [5]. Risk Analysis is the activity of creating security design specifications from the first planning and before every iteration [3] [5]. 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 [5] [15].

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 [5] but also to share knowledge with other team elements making them more security aware.

3 Research Methodology

This study follows a MLR methodology. A MLR is a form of a SLR which includes grey literature in addition to the published (formal) literature [7].

MLR in Software Engineering (SE) is not usual [7] and there are no guidelines to perform a MLR, since MLR is a form of SLR the review is planned as SLR but including "grey literature".

SLR is a type of literature review that is used to identify, evaluate and interpreting all available research relevant to a specific question [16]. Kitchenham's procedures for performing systematic reviews will be adopted by the authors. ***Error! Reference source not found.****Fig. 1* details how this research steps maps to the three phases proposed by Kitchenham [16].

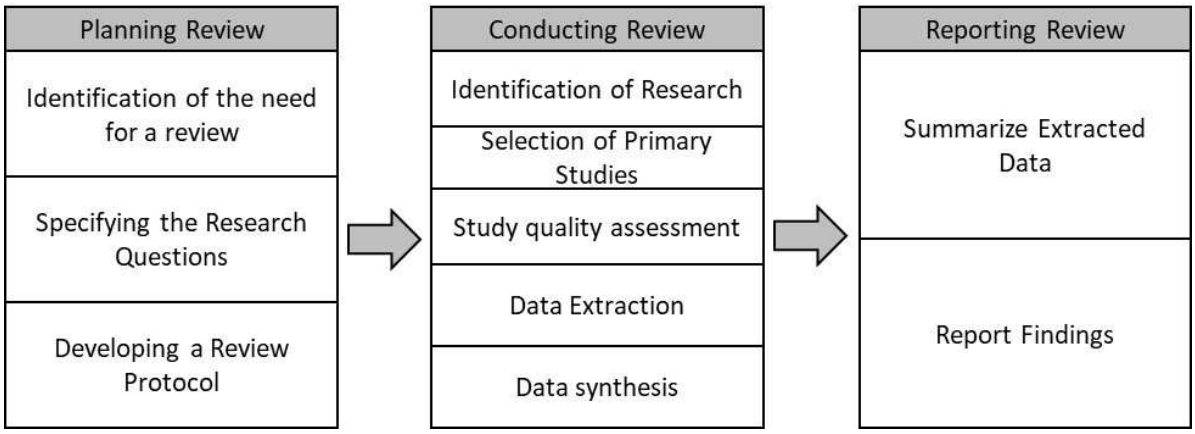

Fig. 1. MLR Steps

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. Final step designing a review protocol with the constraints that are going to be applied in the review. This phase is presented in Section 4.

Conducting Review - this phase consists in applying the designed review protocol. This phase is presented in Section 5.

Reporting Review – final phase of the review is summarizing the extracted data from the selected literature and report findings. This phase is presented in Section 6.

4 Planning the Review

This section details the first phase of the SLR. Motivation for this work is presented, followed by the Research Question this study intent to address and answer. Finally, Review Protocol is proposed.

4.1 Motivation

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 [6]. One of the principles found in DevOps and DevSecOps is measuring. DevSecOps encourages 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 [5].

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.

4.2 Research Questions

Based on what was described before it was established the importance of having metrics has way to better understand a topic under discussion for that reason the research aims to answer the following Research Question (RQ).

RQ: Which are the most relevant DevSecOps metrics.

4.3 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 1.

Table 1. Search Terms

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

The chosen academic data sources for the this MLR are three 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.

Inclusion and exclusion criteria is applied to literature from both data sources. Criteria is presented in Table 2.

Table 2. 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

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's possible to conduct the review. The review protocol is represented in *Fig. 2*.

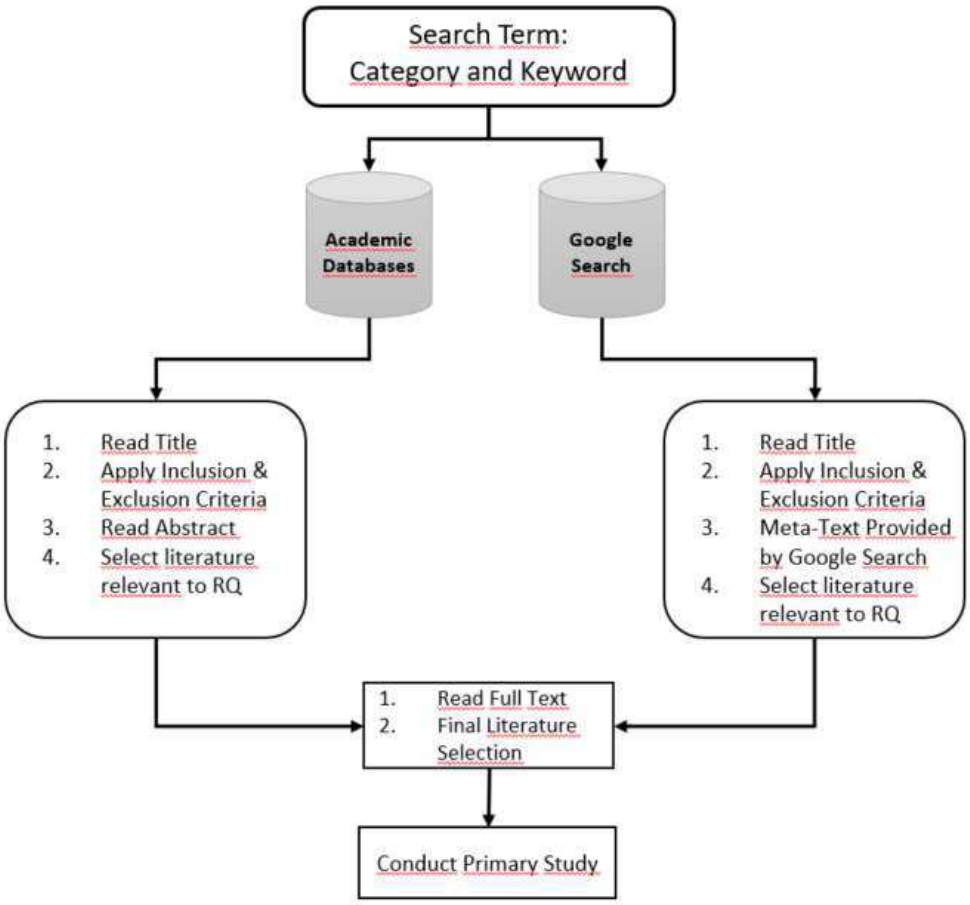

Fig. 2. Review Protocol adapted from [5]

5 – Conducting the Review

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

5.1. Selection of Studies

First step was to run the search string composed by the search terms defined on Table 1. After running the search terms on the selected data sources 558 articles were obtained. Distribution of articles by category is illustrated on *Fig. 3* and by database illustrated on *Fig. 4* The searches on the data sources only considered articles published after 2013.

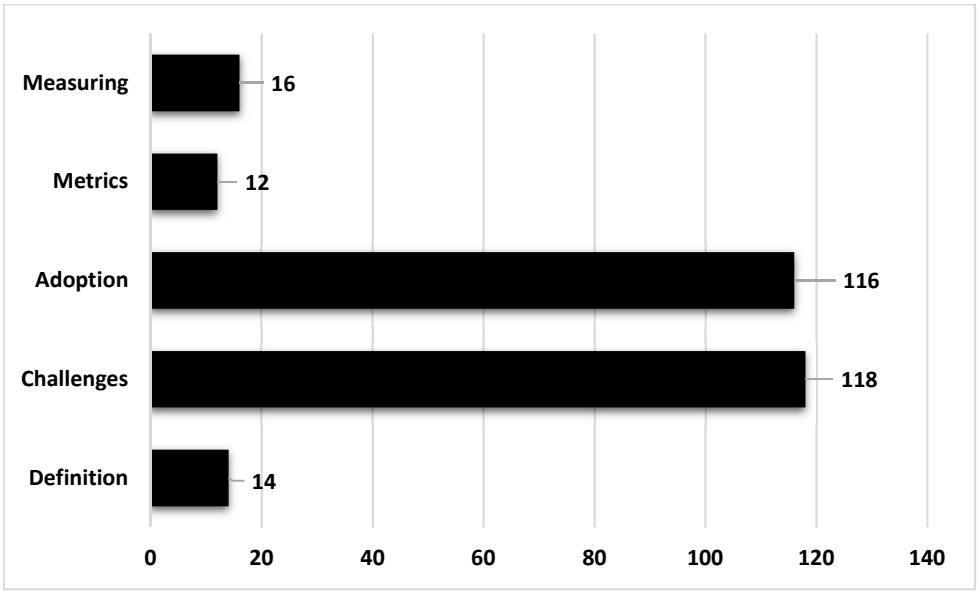

Fig. 3. Distribution of articles by Search Term

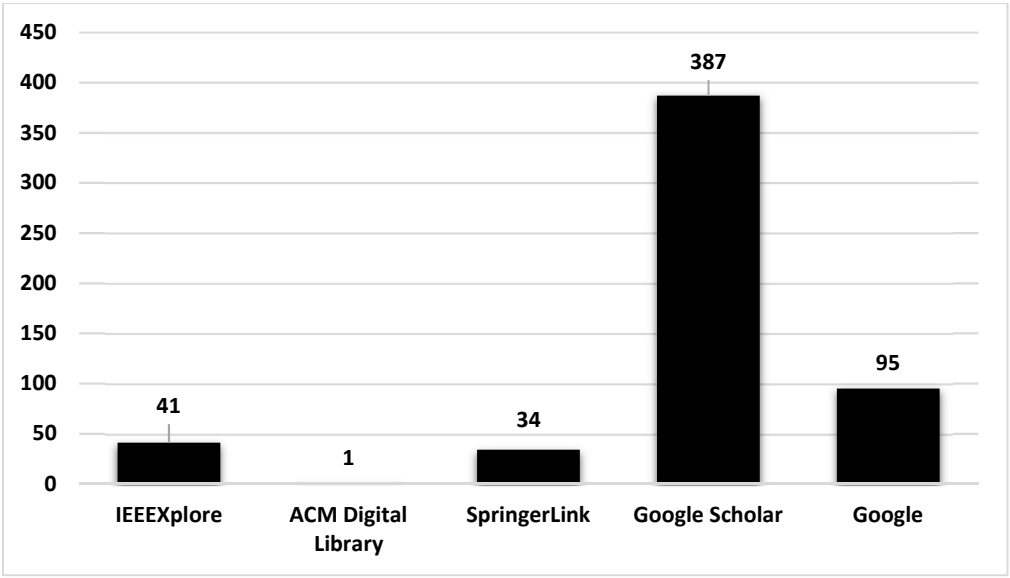

Fig. 4. Distribution of articles by Database

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

5.1.1. Academic Databases. First step is ensuring that there is not duplicated articles. Removing the duplicates consists on a two-step approach.

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

Studies information exported from each data source were on different formats. Table 3 shows the export format from each academic data source.

Table 3. 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 are sufficient 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 4 details number of academic articles remaining after each phase.

Table 4. 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

5.1.2. Grey Literature. The approach to filtering the grey literature is like the one used on the academic databases. First step is removing the duplicated, this was achieved by filtering duplicated URL's on Excel. After removing the duplicated articles, inclusion and exclusion criteria is applied a total of 56 were flagged as relevant to research question. Table 5 details number of grey literature articles remaining after each phase.

Table 5. 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

5.1.3. 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 13 were obtained as relevant to our study.

5.2. Data Extraction Analysis

Based on the obtained artefacts from this MLR there is little literature related with DevSecOps and in particularly on literature related in how organizations can measure the efficiency of DevSecOps implementations.

Even so we can verify that the topic has been gaining interest as it can be seen in *Fig. 5,* both in academic and grey literature data sources the interest on the topic rose considerably after 2017.

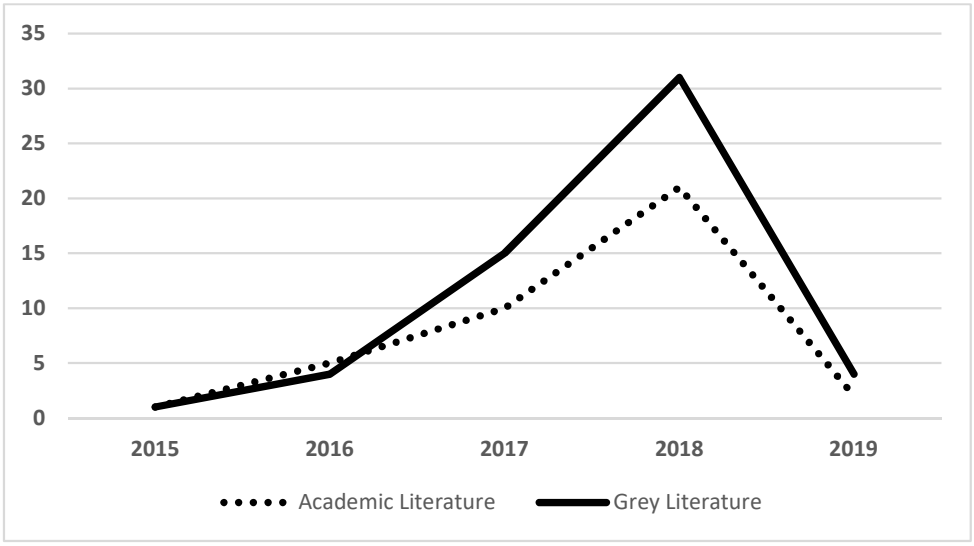

Fig. 5. Academic and Grey Literature articles flagged as relevant by year

The year 2019 has less studies because this review only took into consideration the studies until the 10th of April of the same year.

Final selection of studies only contained 2 academic articles and much of the literature use to answer the research questions is based on Blogs and articles from industry professionals. Table 6 summarizes number of articles based on literature source.

Table 6. Final number of articles by literature source.

Literature Source	Number of Articles
Academic	2
Grey	11

6 Reporting the Review

This MLR phase presents the research done on DevSecOps to identify its metrics. We used Google Scholar, Google Search, IEEE Explore, Springer and ACM Library to locate literature and after applying our inclusion and exclusion criteria, 15 articles were found to be relevant to our search terms. Only 2 of those were academic research papers. The remaining 11 consisted of blogs and articles. Based on the literature review found e 9 relevant metrics were reported by professionals. Table 7 lists and describes identified metrics.

7 Conclusion and Future Work

This MLR presents the research done 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 very hard to find information regarding metrics associated with DevSecOps special in academic literature. Even so it was possible to identify a total 9 metrics as indicators of DevSecOps effectiveness. This topic is expected to grow and for that reason it's study should continue; this study serves as the initial support for further studies.

This study for academics may serve has the basis for further research into DevSecOps metrics or other related metrics. For Professionals this study summarizes the principal metrics for measuring DevSecOps effectiveness in one document.

Since DevSecOps is a trending topic and this study had an exploratory nature, further researches may continue the study performing 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. The authors are already pursuing this investigation line.

Table 7. DevSecOps Metrics

Metric	References	Description	Goal	Measuring
Defect Density	[17] [18] [19] [20]	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 density is measured by dividing the total number of confirmed defects by the total line of codes of all the modules in the new release. Ideal is to have the lowest density value possible.
Defect Burn Rate	[17][19] [21] [19]	Indicates how quickly the team is addressing defects.	Measuring development team productivity solving defects.	Take the total number of defects found in development and divided it by the sum of defects found in development and production and multiplied by 100. If the rate has a high value it means the team is being effective.
Critical Risk Profiling	[17] [22] [23] [24] [25] [26]	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.	Vulnerability should be associated with a score for a criticality and another that defines the value of that vulnerability to attackers. Vulnerabilities that have high score in both criticality and value should be handle first. Having vulnerabilities with low score on criticality and value is a good indicator.
Top Vulnerability Types	[17] [20] [27]	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.	Keeping track of most recurring vulnerabilities related with software development (example OWASP Top 10). A good indicator is to have the lowest number of vulnerabilities without a mitigation plan.
Number of Adversaries per Application	[17] [26]	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.	Team exercise where the objective is to think how many adversaries they think an application as and register those findings.
Adversary Return Rate	[17]	Measures how often an adversary will use the same strategy and procedures.	Helps define appropriate training to better handle these attacks.	Measure is done by counting the number of times adversaries use the same attacking strategy and compiling into a ranking that visible for every team member. Ideal is to have a plan to handle each attacking strategy.
Point of Risk Per Device	[18]	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.	Identify and keep track of unpatched vulnerabilities per server. The number of vulnerabilities should tend to zero.
Number of Continuous Delivery Cycles Per Month	[18] [19] [21] [26]	Number of successful deploys to production per month.	Measuring how quickly code changes can be deployed to production.	This metric is measured by counting the number of attempts to deploy versus the number of successful attempts. A positive value is to have the highest number of successful attempts.
Number of issues during Red Teaming Drills	[21] [26]	Number of found issues and fixed by Red Team.	Measuring Red Team Effectiveness.	Measured by counting the number of defects found and fixed by the Red Team.

References

[1] M. Silva, J. Faustino, R. Pereira and M. Mira Da Silva, "Productivity Gains of DevOps Adoption in an IT Team: A Case Study," in *Designing Digitalization*, Lund, 2018.

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

[3] A. A. U. Rahman and L. Williams, "Software security in DevOps: synthesizing practitioners perceptions and practices," in *International Workshop on Continuous Software Evolution and Delivery*, New York, 2016.

[4] L. Bass, R. Holz, P. Rimba, A. B. Tran and L. Zhu, "Securing a deployment pipeline," in *Third International Workshop on Release Engineering*, New Jersey, 2015.

[5] M. H. and C.-P. R., "DevSecOps: A Multivocal Literature Review," in *Software Process Improvement and Capability Determination*, Springer, 2017.

[6] N. Fenton and J. Bieman, Software Metrics, Boca Raton: CRC Press, 2015.

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

[8] R. F. Elmore, "Comment on "Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method."," *Review of Educational Research,* no. 61, pp. 293-297, 1991.

[9] S. J., N. K. and P. I., "DevOps: A Definition and Perceived Adoption Impediments.," *Lecture Notes in Business Information Processing,* vol. 212, 2015.

[10] J. Roche, "Adopting DevOps Practices in Quality Assurance," *Communications of the ACM,* vol. 56, no. 11, pp. 8-20, 2013.

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

[12] P. K. a. M. O. Lucy Ellen Lwakatare, "Dimensions of DevOps," in *Agile Processes in Software Engineering and Extreme Programming*, vol. 212, Springer, 2015.

[13] M. Virmani, "Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery," in *INTECH 2015*, Pontevedra, 2015.

[14] C. Ebert, G. Gallardo, J. Hernantes and N. Serrano, "DevOps," *IEEE Software,* pp. 94-100, 2016.

[15] R. V. a. H. R. K. H. T. Ray, "Toward an automated attack model for red teams," *IEEE Security & Privacy,* vol. 3, no. 4, pp. 18-25, 2005.

[16] B. Kitchenham, "Procedures for Performing Systematic Reviews, Keele University Technical Report TR/SE-0401," Keele University, Keele, 2004.

[17] E. Chickowski, "Seven Winning DevSecOps Metrics Security Should Track," Bitdefender, 1 May 2018. [Online]. Available: https://businessinsights.bitdefender.com/seven-winning-devsecops-metrics-security-should-track. [Accessed 25 March 2019].

[18] A. Humphrey, "Diving into DevSecOps: Measuring Effectiveness & Success," Armor, 16 January 2018. [Online]. Available: https://www.armor.com/blog/diving-devsecops-measuring-effectiveness-success/. [Accessed 29 March 2019].

[19] A. Jerbi, "InfoWorld," 13 November 2017. [Online]. Available: https://www.infoworld.com/article/3237046/kpis-for-managing-and-optimizing-devsecops-success.html. [Accessed 25 March 2019].

[20] T. Hsu, Hands-On Security in DevOps, Birmingham: Pack Publishing, 2018.

[21] A. Crouch, "https://www.agileconnection.com," Agile Connection, 13 December 2017. [Online]. Available: https://www.agileconnection.com/article/devsecops-incorporate-security-devops-reduce-software-risk. [Accessed 26 March 2019].

[22] K. Casey, "Enterprisers Project," 19 June 2018. [Online]. Available: https://enterprisersproject.com/article/2018/6/how-build-strong-devsecops-culture-5-tips?page=1. [Accessed 26 March 2019].

[23] S. Woodward, "BrightTalk," 18 September 2018. [Online]. Available: https://www.brighttalk.com/webcast/499/333412/devsecops-metrics-approaches-in-2018. [Accessed 27 March 2019].

[24] J. Vijayan, "TechBeacon," [Online]. Available: https://techbeacon.com/security/6-devsecops-best-practices-automate-early-often. [Accessed 1 April 2019].

[25] F. Raynaud, "DevSecCon," June 2017. [Online]. Available: https://www.devseccon.com/wp-content/uploads/2017/07/DevSecOps-whitepaper.pdf. [Accessed 31 March 2019].

[26] C. Paule, "Securing DevOps — Detection of vulnerabilities in CD pipelines," University of Stuttgart, Stuttgart, 2018.

[27] F. Jose, "Effective DevSecops," 3 July 2018. [Online]. Available: https://medium.com/@fabiojose/effective-devsecops-f22dd023c5cd. [Accessed 3 April 2019].

[28] M. Rao, "Synopsys," 6 July 2017. [Online]. Available: https://www.synopsys.com/blogs/software-security/devsecops-pipeline-checklist/. [Accessed 2 April 2019].

[29] Chris Romeo, "Techbeacon," Microfocus, [Online]. Available: https://techbeacon.com/devops/3-most-crucial-security-behaviors-devsecops. [Accessed 3 March 2019].