2020 International Conference on Computational Science and Computational Intelligence (CSCI)

Methods of Implementation, Maturity Models and Definition of Roles in DevOps Frameworks: A Systematic Mapping

Luciano de Aguiar Monteiro, Domingos Sávio M. Pessoa Monteiro, Washington Henrique Carvalho Almeida, Anderson Cavalcanti de Lima and Ioram Schechtman Sette

Center of Advanced Studies and Systems of Recife
Recife, Brazil
E-mail: {lam, dsmpm, whca, iss}@cesar.school
Higher Education Institute iCEV
Piauí, Brazil
E-mail: {anderson.lima}@grupocev.com

***Abstract*— The software development industry has been evolving and has produced new development standards and models for providing services. Agile methodologies have reached their plateau with DevOps, which has increased software quality and hastened its delivery. However, a gap in terms of its formalization and unresolved issues concerning its adoption have become relevant. This study seeks, by using a Systematic Mapping, to elucidate existing gaps in this area such as methods for implementing DevOps, the use of maturity models and defining the roles of those who participate in this process. This Systematic Method was applied to five databases (ACM Digital Library, IEEE Xplore, Science Direct, Springer Link and Scopus), during which automatic searches are conducted with a view to answering the research questions. 32 studies returned answers to these. 11 maturity models were found; however, only two of them had been validated by companies, one of which only applied to IBM software.**

Keywords-DevOps; Software Engineer; Continuous Integration; Continuous Delivery.

I. INTRODUCTION

DevOps is a set of continuous delivery practices aimed at shortening software release deadlines, increasing efficiency, reducing the time between versions and maintaining the quality of applications [1]. This approach focuses on how to prompt the convergence of standards between the **Dev**elopment teams and the **Op**erations teams and seeks to improve cooperation between both teams - hence the origin of the term.

The adoption of Agile Methodologies and DevOps continues to grow, boosted by the need for speed, agility and flexibility, as evidenced in the 2019 World Quality Report [2], in which 99% of the interviewees claimed to be using DevOps in at least some of their businesses.

Implementing DevOps has been becoming more difficult due to the lack of efforts made to formalize concepts and to adopt processes. Despite much research, it is still unclear how this rich information could be harnessed, in an organized and structured way so as to adopt DevOps more thoroughly [3]. Consequently, there is a need to analyze maturity models.

The lack of defining the roles that different members of DevOps teams need to play causes a divergence of ideas in how best to implement and adopt DevOps. [4] explain that the still undefined assignment of responsibilities in the world of DevOps makes it difficult to associate a function to each DevOps tool.

The main goal of this study is to conduct a Systematic Mapping in order to understand how best to implement the DevOps Model in the Software Development Process; to identify if there is a maturity model that is followed and validated; and to identify the roles of the participants in this process.

The remainder of this article is structured as follows: in Section 2 we present the theoretical framework; in Section 3 we describe the process of systematic mapping; in Section 4 we indicate a general view of the results from this study which were obtained in each stage of the process described in section 5 and summarize these results; and, finally, in Section 6 we draw some conclusions and make recommendations for future works.

II. THEORETICAL FRAMEWORK

The software development cycle added a framework in the process, called DevOps practices. This first and foremost led to automating actions, such as making measurements, promoting a collaborative culture, and using tools throughout the development process, and specifically sought to improve the quality of software and to deliver faster [5].

A. Infrastructure as Code

Automating the flow of the software production line is a premise that has been followed by companies that have adopted DevOps. It is achieved by using configuration management processes and the automatic provision of environments.

Infrastructure as Code (IaC) is a technique in which the environment required for production or testing is created from tools that use configuration files. These configuration

978-1-7281-7624-6/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCI51800.2020.00327

management tools such as Puppet-Ansible and Chef are the main resources for implementing IaC strategies [6].

B. Version Control System

The basic assumption in the DevOps environment is the concept of the willingness to share. Therefore, the initial software production line must be set up with this primary aim, sharing its code, so the entire team can have a broad idea of what is being done and developed. Thus, all of the developers in this project make their codes available in a source-code version control system, such as Git.

Incidentally, [7] asserts that the tools for version control play a central role in the large amount of DevOps tools, in which developers write a source-code and share them in these repositories, thereby creating a Continuous Integration pipeline.

C. Continuous Integration (CI) and Continuous Delivery (CD)

CI is a DevOps practice in which the members of a development team regularly share their work and automate the building process, testing and validation [8]. When an alteration is detected in the environment, the CI server activates the processes for building and testing the software, thereby releasing reports for actions related to the tests. Thereafter, if there are no failures in the tests, the production environment can also be made available by the CI service, in a process called "Continuous Deployment", the main aim of which is to identify errors and software defects quickly, and to have these corrected quickly.

The continuity of a development cycle is called Continuous Delivery (CD). It is an approach to software engineering in which the teams continue producing valuable systems in short cycles, making sure they can be safely released at any moment [9]. Consequently, clients can have new functionalities that are safely, continuously and automatically implemented in the system. The practice of CD is achieved by exercising discipline and automating delivery, which includes creating, testing implementing the software [10].

D. DevOps Environment

A software production line that uses DevOps, has a well-defined automation cycle, which begins by merging the developers' source-code with the code version control system. When the CI server identifies the conclusion of the commits, it performs the due tests and, if necessary, it releases feedback to the developers.

III. The Process of Systematic Mapping

This section describes the protocols used for conducting this research with a view to addressing the main issue, by using Systematic Mapping (SM) [11].

The SM process was conducted as per [12], in five essential stages, namely: (1) set the research questions; (2) search for relevant articles; (3) sort the articles found; (4) select key-words, by using the abstracts; and (5) map the data.

A. Research Questions

The following questions (RQ) RQ1 to RQ5 were used as an instrument to guide this Systematic Mapping:

RQ1. How is DevOps being applied to the Software Development Process?

RQ2. How is DevOps implemented in the Software Development Process?

RQ3. Does the Software Development Process have maturity models for implementing DevOps?

RQ4. Are there well-defined roles in the Software Development Process that can use DevOps?

RQ5. How is the applicability of DevOps assessed in the Software Development Process?

B. Search Strategy

The search strategy that was adopted to find relevant studies included making automatic searches in the databases. The searches were focused on the period from 2015 to May 2020.

The electronic databases used for the Systematic Mapping were: ACM Digital Library; IEEE xPlore; Science Direct; Scopus; and Springer Link.

C. Search Terms

As we sought to find studies focused on the subject of this research, we defined the key-words for the search terms that, according to [13], must be developed based on the central terms identified in each research question.

In this context, the keywords from the selected control articles were applied to the exploratory analysis as guidelines for this stage, by means of Boolean operators, for example "AND" and "OR", as described below:

(devops) AND (strategy OR model OR deployment OR problems OR solutions)

The same Search Term was applied in all the databases, returning 666 papers from ACM, 230 papers from IEEE, 328 papers from Science Direct, 503 papers from Scopus and 1298 papers from Springer. The stages of the research process and the selection for Systematic Mapping are represented in Fig. 1

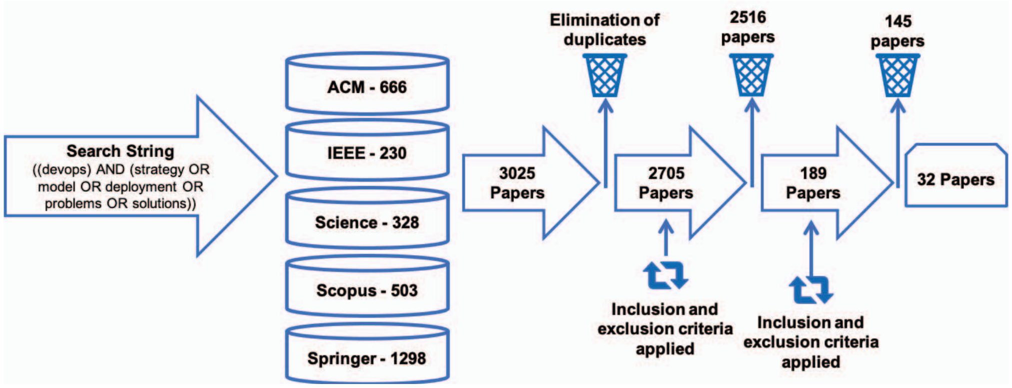

Figure 1. Results of the search and selection process.

D. Criteria for Selecting Studies

After defining the search terms and databases, the criteria for inclusion were drawn up, such that they can identify a correlation between of the studied selected and the questions stipulated in the Systematic Review. The criteria for inclusion are listed in Table 1.

TABLE 1. CRITERIA FOR INCLUSION.

CI	*Criteria*
1	Studies that address any of the research questions
2	Papers that tackle DevOps practices in the Software development cycle
3	Studies that propose or report the use of any DevOps tools
4	Studies that propose or report any DevOps implementation models

We also drew up criteria for exclusion, shown in Table 2 in order to discard studies that were not related to the research in any way.

TABLE 2. CRITERIA FOR EXCLUSION.

CE	*Criteria*
1	The study does not have an abstract
2	The study was not published in English
3	The study is an earlier version of another already published paper that is a candidate for selection
4	Article not accessible
5	Paper was published as a short paper or poster
6	The article does not relate to DevOps
7	The article does not tackle practices, implementation techniques nor models for adopting DevOps

E. Selection of Studies

The studies were selected using the criteria for inclusion in Table 1 and the criteria for exclusion in Table 2. To do so, the reviewers conducted a paired analysis. In case of divergence, a third reviewer mediated the final decision.

The selection occurred in two distinct steps, in which the reviewers, during the first step, assessed the title of the article and the abstract, by applying the inclusion and exclusion criteria. In the last step, the reviewers assessed the full content of the article. The 189 studies initially identified yielded 16 conflicting evaluations, which were then resolved by a third reviewer who made an independent assessment, only 32 studies addressed research questions RQ1 to RQ5.

F. Data Extraction

During the data extraction phase, three main activities were undertaken: the studies were classified based on their field of research, a thematic synthesis was written [14] and evidence of the research questions being addressed was sought (RQ1 to RQ5).

After coding and an initial analysis, the studies were sorted into the following themes: DevOps Grounding; DevOps Practices; Implementation Models; Roles and Competences.

G. Quality Criteria

Considering that the primary studies came from a wide variety of types of research, a generic set of questions was used as the basis to evaluate the quality of the studies [15], as shown in Table 3. The answers to the questions were classified on a separate scale, in which the answer "Yes" scored 1, "No" scored 0 and "Partly" scored 0.50, the maximum score being 8 points.

TABLE 3. QUALITY CRITERIA.

QC	*Criteria*
1	Does the study report clear and unequivocal discoveries based on evidence and argument?
2	Is there a detailed description of the review process?
3	Is the research objective clearly defined?
4	Were the participants' roles and competences defined in the DevOps cycle?
5	Did the study tackle DevOps in the Software Development Process in a conclusive way?
6	Did the study use any Maturity Models for the application of DevOps?
7	Did the studies show any evidence of validating the application of DevOps in the software development cycle?
8	Is there enough evidence from which to draw conclusions?

IV. RESULT OF THE SYSTEMATIC MAPPING

The Systematic Mapping resulted in 32 studies being selected. This section presents several aspects of the papers found. The facts include the year of publication, the thematic synthesis and the kinds of studies selected.

Fig. 2 shows the number of studies per year sorted according to the database searched. Springer was the base with the highest number of studies selected, there being a total of 13 such studies.

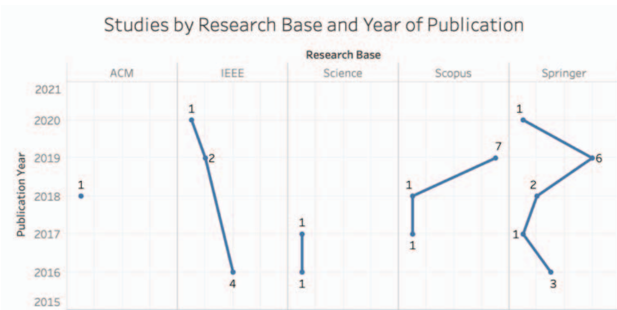

Figure 2. Search Base sorted by year.

During the data extraction phase, one of the aspects catalogued was the kind of study, as shown in Fig. 3. Most of papers yielded by the SM were Bibliographical studies.

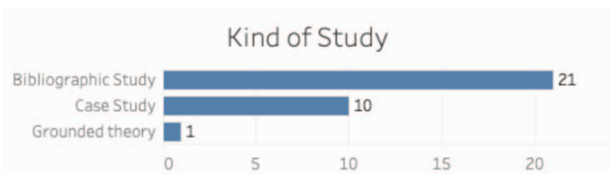

Figure 3. Kind of Styde.

We wrote a thematic synthesis of the studies and also sorted them according to their research areas, as shown in Fig. 4. It is important to point out that a study can belong to more than one theme.

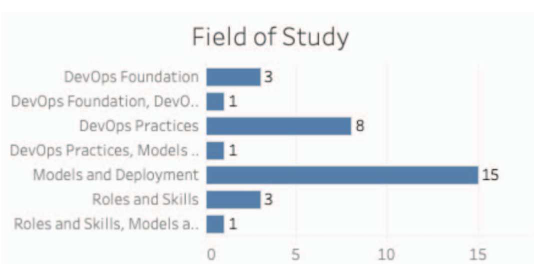

Figure 4. Field of Study

V. DISCUSSION

This section provides a detailed discussion on the results of this systematic mapping study. The subsections give a synthesis of the evaluation of the quality of the selected studies, as well as the answers to the research questions RQ1 to RQ5.

A. Criteria for Quality

Table 4 shows the results of the selected studies in terms of Criteria for Quality.

TABLE 4. PAPERS INCLUDED.

Study Ref.	*Title*	*Score*
[16]	DevOps and Continuous Delivery	8
[17]	DevOps competences and maturity for organizations producing software	8
[18]	From Agile to DevOps: Smart Skills and Collaborations	7,5
[19]	A DevOps Implementation Framework for Large Agile-Based Financial Organizations	7
[20]	A survey of DevOps maturity models	7
[21]	Integrating development and operations in cross-functional teams — Toward a DevOps competency model	7
[3]	Adopting DevOps in the real world: A theory, a model, and a case study	6,5
[22]	Continuous deployment at Facebook and OANDA	6
[23]	User Stories to User Reality: A DevOps Approach for the Cloud	6
[24]	Accelerating Towards DevOps	6
[25]	DevOps	6
[26]	DevOps: Foundations and Using them in Data Center	6
[6]	Empowering Continuous Delivery in Software Development: The DevOps Strategy	6
[27]	DevOps in practice: A multiple case study of five companies	6
[28]	Modeling and measuring attributes that influence DevOps implementation in an enterprise using structural equation modeling	6
[29]	Business model innovation using modern DevOps	5,5
[30]	CMMI guided process improvement for DevOps projects: An exploratory case study	5,5
[31]	DevOps culture and its impact on cloud delivery and software development	5,5
[32]	Devops, A New Approach to Cloud Development & Testing	5,5
[33]	Continuous Architecture and Continuous Delivery	5,5
[34]	DevOps for IT Service Reliability and Availability	5,5
[35]	DevOps Foundations	5,5
[36]	Comparison of DevOps maturity models	5,5
[37]	DevOps in Practice – A Preliminary Analysis of Two Multinational Companies	5,5
[38]	DevOps meets dynamic orchestration	5,5
[39]	Overcoming Challenges with Continuous Integration and Deployment Pipelines: An Experience Report from a Small Company	5
[40]	Exploiting DevOps practices for dependable and secure continuous delivery pipelines	5
[40]	Software Quality Improvement Practices in Continuous Integration	5
[41]	DevOps Finetuning	4,5
[42]	Continuous Delivery	4
[43]	DevOps for containerized applications	3,5
[44]	DevOps movement of enterprise agile breakdown silos create collaboration, increase quality, and application speed	3,5

B. Answer to RQ1 Research Question - How is DevOps being applied to the Software Development Process?

RQ1 question sought to highlight how DevOps is being applied during the software development process. Eleven studies addressed this question.

The first definition addressing RQ1 in the papers was how the work was conducted. [37] affirms that it is an approach that is mainly adopted by start-ups, in which there are no separate functions defined for developers. This pattern of responsibilities in operations is fully shared.

However, [43] explains how DevOps is applied throughout the software development process. This includes pointing out that the application undergoes an initial design phase and the development starts with the source-code updates, which are stored in a source-code management system. Paper [45] detailed that Jenkins is customized according to needs, by means of various plug-ins, and it is largely employed in Continuous Integration.

Study [46] approaches canary deployments and its application. The system can be monitored in order to detect errors or other kinds of degradation. If there are errors, redirection to instances of the canary can be disabled and they will be reverted to the previous version. If there are no mistakes, a continued update can be activated, manually or automatically.

Authors [24][31][23] highlight that highly favorable practices to the successful adoption of DevOps are: Safety throughout the pipeline; Effective adoption of a cloud environment; Microservices; API-oriented solutions to facilitate integrations; Tests at the beginning and at the end of the process; Adoption of automation; Configuration of KPI keys for critical recognition of process patterns; Optimization plan for instantaneous failover and system and software incremental scale; Tools provide automation that complements the collaborative culture and mature processes.

[27] considered Infrastructure as Code to be a facilitator for the automatic deployment of the software in the destination environment by the software development teams. [29] complemented that Docker, Container and orchestrators promote growth in business in this environment.

[38] ratified that the adoption of Container technology as a light and scalable solution for the challenges of implementation, which isolates the application and its dependencies in isolated units, regardless of their programming language and execution platforms, eliminates the need for collaboration between Dev and Ops in execution time.

C. Answer to RQ2 Research Question – How is DevOps implemented in the Software Development Process?

Question RQ2 sought to answer how the implementation of DevOps occurs in development companies. Eight studies addressed this question.

Studies [3][44] recommend that the following aspects must be considered for the successful adoption of DevOps: Evaluation of the correct DevOps strategy for the business model; Identification of the maturity of DevOps in the main development processes and IT operations; Establishment of automation standards throughout the continuous pipeline; Establishment of successful measurements and metrics for DevOps; and, Sharing and transparency of the actions.

In order to implement DevOps in a Cloud [33], the following Roadmap was presented: Involve all the teams (operations, testers and DBAs) from the beginning; Implement feedback loops; Enable monitoring and alert metrics (telemetry); Limit Work in Progress; Automate the implementation; Implement continuous tests; and, Plan hybrid clouds.

Table 5 consolidates the set of studies that address RQ2 questions and that tackle how to implement DevOps practices, according to the description presented in the table.

TABLE 5. PAPERS THAT ANSWERED RQ2 ACCORDING TO THE THEME.

Ref.	***DevOps Practice***	***Description***
[6]	IaC	Used to solve problems related to the manual configuration management process by using automatic provisioning and configuration, being able to create automation logic for different tasks.
[47]	CI	This routine practice for the process of integrating developer code can detect errors quickly and avoid compilation flaws. Selecting the appropriate tool that fits according to the requirements would be the main challenge for CI.
[39]	CD	The compilation and release process combined with the origin control, so that any alterations in these files lead to an automatic update in the compilation and release infrastructure.
[22] [47] [42]	Conti-nuous Deploy-ment	Developers must be involved in the entire implementation cycle, each change goes through a test pipeline and, if it is qualified or passes the tests, it is automatically implemented in the development process.

D. Answer to RQ3 Research Question - Does the Software Development Process have maturity models for implementing DevOps?

SM searched for an answer to RQ3, and identified four primary studies that met the requirements.

Maturity Models are guiding instruments that can quantify the level of knowledge about a certain process or performance of an activity. [36] [20][48] identified a total of 11 (eleven) DevOps Maturity Models, which are listed in Table 6:

TABLE 6. DESCRIPTION OF MATURITY MODELS.

Maturity Model	***No. Level***	***Description***
IBM DevOps Model	5	Describes a complete analysis of adopting the IBM DevOps approach to promote continuous software delivery.
Mohamed's Model	5	It is based on CMMI and helps to improve efficiency, increases visibility and minimizes significant risks, such as idle time during the implementation.
Capgemini DevOps Maturity Model	5	Maturity model that allows companies to identify the current level of maturity. This model has five maturity levels that measure three dimensions: people, processes and tools.
Bucena DevOps Maturity Model	5	It is based on the CMM approach, which consists of 5 maturity levels. Each of these levels has four dimensions: technology, process, people and culture.

Maturity Model	*No. Level*	*Description*
Eficode Maturity	-	It has five dimensions: organization and culture, environments and release, builds and continuous integration, quality assurance, and visibility and reporting. It creates and integrates continuously, assurance of quality and visibility and reports. The model defines four maturity levels
Hewlett Packard Enterprise DevOps	5	This model is based on CMMI and was designed to cover the entire lifecycle of an application. It is used for measuring dimensions of the process, automation and collaboration.
Feijter Maturity Model	-	It includes areas of focus that allow software companies to mature in a refined way. The Feijeter model includes sixty-three resources (represented as letters in the model) which are distributed over ten resource levels.
BTopham Maturity Model	-	The model is based on CMMI and designed to cover the entire lifecycle of an application or service for large enterprises. This model lacks clarity and specific characteristics.
Samer I. Maturity Model	5	The model has five maturity levels identical to those of the BTophan Maturity Model. It defines slightly different aspects as follows: communication, automation, quality and governance.
Focus Area Maturity Model	-	This is a nontraditional model due to its focus area architecture. It describes sixteen focus areas that are logically arranged in three groups: 1) culture and communication, 2) product, process and quality, 3) basis.
The DevOps Maturity Model	-	It includes focus areas and allows an SPO to mature in a refined way. It contrasts with the CMM model, which allows organizations to mature in a generic way

It must be considered that, out of the 11 (eleven) Maturity Models that were presented, only the Hewlett Packard Enterprise and IBM DevOps models were applied and validated by the institutes that developed them, whereas other models are validated by an independent organization which applies them [20]. The DevOps Maturity Model [36] conducted a case study in which the model was applied. However, as there was a small number of participants, it was unable to attest to its effectiveness.

E. *Answer to RQ4 Research Question – Are there well-defined roles in the Software Development Process that can use DevOps?*

The software development cycle with DevOps has experienced a change in paradigm and an evolution of agile methodologies. Hence it needs to detail the roles of participants and identify their competences. This is the aim of RQ4 question, in which five papers addressed this research question.

Studies [36][41][21] indicated that the competences required from a DevOps professional are: To be a good communicator; To have abilities related to the entire lifecycle of the software development process, including, among others, agile project management, management of and of automation of versions; Aspects of software engineering (design, development, maintenance, testing and evaluation) and software operations; Social and relationship competences, as they are part of constructing a successful work relationship inside a multifunctional team and with other interested parties.

With regard to the roles assigned to the participants in the software development cycle with DevOps, two studies addressed this topic. A comparison of the studies is given in Table 7.

TABLE 7. DEVOPS SKILLS.

[49]	[16]	*Description*
Product management	Assess	Input from business stakeholders (for new services) and service-level agreement assessments (for existing services)
	Design	Development and Business take the lead in this stage, with Operations available to provide input as needed
Development	Develop	Development is the lead during this stage and is ultimately responsible for building software that meets the needs of the business
Test/QA	Test	Development and operations are the leads during this stage, as final preparations are made for unit testing, integration testing, and release to production
Architecture	Deploy/ Release	This is the traditional DevOps handoff stage, but in this scenario, the handoff is a change in lead roles versus a turnover of responsibility
Operation	Manage	During this stage, infrastructure, systems, and application management tools monitor production environments and applications

F. *Answer to RQ5 Research Question – How is the applicability of DevOps assessed in the Software Development Process?*

Regarding this research question, five studies answered how to assess the applicability of DevOps through metrics.

The assessment of the good execution of the software development cycle is identified by means of metrics collected from along the development pipeline. [30][25][34][35][50] present a set of metrics that can be used to assess the applicability of DevOps in the Software Development Process: Delivery time; Frequency of distribution; Percentage of errors in alterations; Recovery time (from an error in the development environment); A culture of collecting and using measures throughout the organization; A repository at the organizational level to store measurement data; A KPI kit to assess performance; A standardized method to report both raw and analyzed metrics; Code quality metrics (defects, security vulnerabilities, technical debt.

VI. CONCLUSION AND FUTURE STUDIES

This article presented a result of a Systematic Mapping of the Literature that initially identified 3025 studies, of which 189 were potentially relevant. However, only 32 studies addressed research questions RQ1 to RQ5. As to the question researched, five studies tackled how to evaluate the applicability of DevOps by using metrics.

The studies showed how DevOps is highlighting the following issues: pipeline security, effective adoption of a cloud environment, adoption of microservices, infrastructure as code, use of container solutions, and tools to automate the development pipeline. Regarding best practices, DevOps was found to have: Infrastructure as Code, Continuous Integration, Continuous Delivery, Continuous Deployment.

In the area of Environmental Management, studies also highlighted: Involvement of the entire team, enabling monitoring metrics, limiting work in progress, automation, and continuous testing.

As for the Maturity Model, 11 were found. However, only two were validated by companies, and one of them is applied only in IBM solutions. The roles of the actors involved in the process were found in 02 studies, with different names, but performing the same function. As to competences, knowledge of technology and interpersonal relationships.

The mapping of the literature carried out in the present study can support teams or software engineers in the process of implementing and conducting DevOps in the systems development cycle. This is the main contribution of this paper.

As to a future line of research, we will undertake a comparative analysis of the DevOps maturity models found, propose a new, more adherent model and conduct a case study to validate it.

References

[1] D. Taibi, V. Lenarduzzi, and C. Pahl, "Continuous architecting with microservices and DevOps: A systematic mapping study," in *Communications in Computer and Information Science*, 2019, vol. 1073, pp. 126–151.

[2] M. Buenen and A. Walgude, "World Quality Report 2018–19," *Capgemini*, vol. 1, pp. 1–30, 2019.

[3] W. P. Luz, G. Pinto, and R. Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study," *J. Syst. Softw.*, vol. 157, 2019.

[4] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A survey of DevOps concepts and challenges," *ACM Computing Surveys*, vol. 52, no. 6. Association for Computing Machinery, 01-Nov-2019.

[5] K. Nybom, J. Smeds, and I. Porres, "On the impact of mixing responsibilities between Devs and Ops," in *Lecture Notes in Business Information Processing*, vol. 251, 2016, pp. 131–143.

[6] C. Siebra *et al.*, "Empowering Continuous Delivery in Software Development: The DevOps Strategy," in *Communications in Computer and Information Science*, 2019, vol. 1077, pp. 247–265.

[7] F. Erich, "DevOps is simply interaction between development and operations," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11350 LNCS, pp. 89–99.

[8] S. A. I. B. S. Arachchi and I. Perera, "Continuous integration and continuous delivery pipeline automation for agile software project management," *MERCon 2018 - 4th Int. Multidiscip. Moratuwa Eng. Res. Conf.*, pp. 156–161, 2018.

[9] E. Laukkanen, T. O. A. Lehtinen, J. Itkonen, M. Paasivaara, and C. Lassenius, "Bottom-up Adoption of Continuous Delivery in a Stage-Gate Managed Software Organization," *Proc. 10th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas. - ESEM '16*, pp. 1–10, 2016.

[10] J. Humble and D. Farley, "Continuous Delivery Huge Benefits, but Challenges Too," *IEEE Softw.*, p. 497, 2015.

[11] R. Wendler, "The maturity of maturity model research: A systematic mapping study," *Inf. Softw. Technol.*, vol. 54, no. 12, pp. 1317–1339, 2012.

[12] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," *12Th Int. Conf. Eval. Assess. Softw. Eng.*, vol. 17, p. 10, 2008.

[13] B. Kitchenham, "Procedures for performing systematic reviews," *Dep. Comput. Sci. Keele Univ. Natl. ICT, Aust. Ltd*, p. 33, 2015.

[14] D. S. Cruzes and T. Dybå, "Recommended steps for thematic synthesis in software engineering," *Int. Symp. Empir. Softw. Eng. Meas.*, no. 7491, pp. 275–284, 2011.

[15] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Inf. Softw. Technol.*, vol. 55, no. 12, pp. 2049–2075, 2013.

[16] R. Sturm, C. Pollard, and J. Craig, "DevOps and Continuous Delivery," in *Application Performance Management (APM) in the Digital Enterprise*, Elsevier, 2017, pp. 121–135.

[17] R. de Feijter, S. Overbeek, R. van Vliet, E. Jagroep, and S. Brinkkemper, "DevOps competences and maturity for software producing organizations," in *Lecture Notes in Business Information Processing*, 2018, vol. 318, pp. 244–259.

[18] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, "From Agile to DevOps: Smart Skills and Collaborations," *Inf. Syst. Front.*, Mar. 2019.

[19] A. D. Nagarajan and S. J. Overbeek, "A DevOps Implementation Framework for Large Agile-Based Financial Organizations," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11229 LNCS, pp. 172–188.

[20] M. Zarour, N. Alhammad, M. Alenezi, and K. Alsarayrah, "A research on DevOps maturity models," *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 4854–4862, 2019.

[21] A. Wiedemann, M. Wiesche, and H. Krcmar, "Integrating development and operations in cross-functional teams — Toward a DevOps competency model," in *SIGMIS-CPR 2019 - Proceedings of the 2019 Computers and People Research Conference*, 2019, pp. 14–19.

[22] T. Savor, M. Douglas, M. Gentili, L. Williams, K. Beck, and M. Stumm, "Continuous deployment at Facebook and OANDA," in *Proceedings - International Conference on Software Engineering*, 2016, pp. 21–30.

[23] R. Punjabi and R. Bajaj, "User stories to user reality: A DevOps approach for the cloud," in *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 - Proceedings*, 2017, pp. 658–662.

[24] S. M. Farooqui and S. M. Farooqui, "Accelerating Towards DevOps," in *Enterprise DevOps Framework*, Apress, 2018, pp. 95–105.

[25] A. Davis and A. Davis, "DevOps," in *Mastering Salesforce DevOps*, Apress, 2019, pp. 27–64.

[26] M. Muñoz and O. Díaz, "DevOps: Foundations and Its Utilization in Data Center," 2017, pp. 205–225.

[27] L. E. Lwakatare *et al.*, "DevOps in practice: A multiple case study of five companies," *Inf. Softw. Technol.*, vol. 114, no. April, pp. 217–230, 2019.

[28] V. Gupta, P. K. Kapur, and D. Kumar, "Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling," *Inf. Softw. Technol.*, vol. 92, pp. 75–91, Dec. 2017.

[29] D. K. Koilada, "Business model innovation using modern DevOps," in *2019 IEEE Technology Engineering Management Conference (TEMSCON)*, 2019, pp. 1–6.

[30] G. Rong, H. Zhang, and D. Shao, "CMMI Guided Process Improvement for DevOps Projects: An Exploratory Case Study," in *Proceedings of the International Conference on Software and Systems Process*, 2016, pp. 76–85.

[31] M. Rajkumar, A. K. Pole, V. S. Adige, and P. Mahanta, "DevOps culture and its impact on cloud delivery and software development," *Proc. - 2016 Int. Conf. Adv. Comput. Commun. Autom. ICACCA 2016*, 2016.

[32] V. Baskaran, S. Singh, V. Reddy, and S. Mohandas, "Digital assurance for oil and gas 4.0: Role, implementation and case studies," in *Society of Petroleum Engineers - SPE/IATMI Asia Pacific Oil and Gas Conference and Exhibition 2019, APOG 2019*, 2019.

[33] M. Erder and P. Pureur, "Continuous Architecture and Continuous Delivery," in *Continuous Architecture*, Elsevier, 2016, pp. 103–129.

[34] M. Kuruba, "DevOps for IT Service Reliability and Availability," 2020, pp. 149–183.

[35] A. Ravichandran, K. Taylor, P. Waterhouse, A. Ravichandran, K. Taylor, and P. Waterhouse, "DevOps Foundations," in *DevOps for Digital Leaders*, A. Ravichandran, K. Taylor, and P. Waterhouse, Eds. Berkeley, CA: Apress, 2016, pp. 27–47.

[36] M. Gasparaite and S. Ragaišis, "Comparison of DevOps maturity models," *CEUR Workshop Proc.*, vol. 2470, pp. 65–69, 2019.

[37] J. Díaz, J. E. Perez, A. Yague, A. Villegas, and A. de Antona, "DevOps in Practice – A Preliminary Analysis of Two Multinational Companies," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11915 LNCS, pp. 323–330.

[38] K. Bahadori and T. Vardanega, "Devops meets dynamic orchestration," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11350 LNCS, 2019, pp. 142–154.

[39] V. Debroy and S. Miller, "Overcoming Challenges with Continuous Integration and Deployment Pipelines: An Experience Report from a Small Company," *IEEE Softw.*, vol. 37, no. 3, pp. 21–29, May 2020.

[40] İ. Keskin Kaynak, E. Çilden, and S. Aydin, "Software Quality Improvement Practices in Continuous Integration," in *Communications in Computer and Information Science*, 2019, vol. 1060, pp. 507–517.

[41] A. Ravichandran, K. Taylor, P. Waterhouse, A. Ravichandran, K. Taylor, and P. Waterhouse, "DevOps Finetuning," in *DevOps for Digital Leaders*, Apress, 2016, pp. 151–169.

[42] M. Lenz and M. Lenz, "Continuous Delivery," in *Python Continuous Integration and Delivery*, Apress, 2019, pp. 53–66.

[43] A. S. Biener and A. C. Crawford, "DevOps for containerized applications," in *Advances in Intelligent Systems and Computing*, vol. 787, 2019, pp. 35–44.

[44] F. Colavita, "DevOps movement of enterprise agile breakdown silos, create collaboration, increase quality, and application speed," in *Communications in Computer and Information Science*, vol. 422, 2016, pp. 203–213.

[45] İ. Keskin Kaynak, E. Çilden, and S. Aydin, "Software Quality Improvement Practices in Continuous Integration," in *Communications in Computer and Information Science*, 2019, vol. 1060, pp. 507–517.

[46] T. F. Düllmann, C. Paule, and A. Van Hoorn, "Exploiting DevOps practices for dependable and secure continuous delivery pipelines," in *Proceedings - International Conference on Software Engineering*, 2018, pp. 27–30.

[47] P. Agrawal and N. Rawat, "DevOps, A New Approach To Cloud Development & Testing," in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2020, vol. 1, pp. 1–4.

[48] M. Gasparaite and S. Ragaišis, "Comparison of DevOps maturity models," in *CEUR Workshop Proceedings*, 2019, vol. 2470, pp. 65–69.

[49] R. de Feijter, S. Overbeek, R. van Vliet, E. Jagroep, and S. Brinkkemper, "DevOps competences and maturity for software producing organizations," in *Lecture Notes in Business Information Processing*, 2018, vol. 318, pp. 244–259.

[50] V. Gupta, P. K. Kapur, and D. Kumar, "Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling," *Inf. Softw. Technol.*, vol. 92, pp. 75–91, 2017.