DevOps Maturity Model

Yasser Rossano Manuel Zacarias
yasser.zacarias@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

January 2021

Abstract

The expanding pace of business competitiveness and environment dynamic imposes rapid changes to organizations, in many areas including Information Systems (IS), which is mainly responsible to ensure the structural division of software development and system operation. Although the team structural division, there is a need to maintain cohesion between them. DevOps is a collaborative and multidisciplinary effort in software development to bridge the gap between the Development and Operation teams. However, many organizations struggle with soft aspects of DevOps and also in breaking the barrier that can be created by other parts of the organization to get on board. Ideally for successful DevOps implementation collaboration and cooperation are needed, especially regarding management support.

The research methodology used throughout this research was Design Science Research. Additionally, to research method, Systematic Literature Review was performed in which the following results were obtained: DevOps concepts (32), processes (9), practices (33), and roles (20). As a result of this, a Process Reference Model (PRM) and Process Assessment Model (PAM) was developed and were the foundation for the developed maturity model. A maturity model created was grounded on PRM and PAM that provided a reliable tool for organizations to evaluate the maturity and provide guidance to achieve higher levels of DevOps maturity. The artefact was demonstrated and evaluated in one Organization. Therefore, the results point out that the proposed maturity model is a valuable instrument for the organization.

Keywords: DevOps, Maturity Model, Process Assessment Model, Process Reference Model.

1. Introduction

For decades organizations have been looking for new ways to improve their software development processes to keep up with business and market demands [1][2].In past decade DevOps, a new approach originated in the context of agile software development movement combining development and operations [3][4], but focusing on development, quality assurance and operations aspects [5], allowed the integration of development and operations teams to achieve fast high-quality releases [6][7][8], while agile practices are mainly focused on rapid interactive development side aspects of IS and little attention is given to the operations. However, successful implementation of DevOps benefits greatly from taking agile approach of software development process.

The approach to DevOps whose can be seen as a cultural movement that aligns people, process, and technology with a common objective of increasing value and eliminate waste using some associated technologies [8] has strongly developed over decades which is now widely adopted in software engineering companies. This results in DevOps being an integral part of Software Development Lifecycle (SDLC). According to [9][10]asserted that DevOps was "a conceptual framework which aims at befitting IS development by integrating development and operations in various ways".

Maturity Model (MM) "offer organizations a simple but effective possibility to measure the quality of their processes"[11], and it has been used as a tool to assess the effectiveness of organizational processes, capabilities or business intelligence.

Although the benefits that DevOps can bring to development and operations, it might not be always successful. This approach is not a simple straightforward task and involves the consideration of a set of challenges such as organizational, communication patterns, process, and technical. According to [12][13][14] organizational challenges may refer to organizational culture, enterprise data models, IT operating models, reward models, and risk allocation. Also the assessment methods for the maturity models where the literature lacks detailed description, that prescribe how to assess the DevOps adop-

tion process for organizations, so they can improve their maturity incrementally [15].

The document consists of creating a process reference and process assessment model which together form the DevOps maturity model based on ISO 3300xx family. Perform a SLR represents the first step to create a process reference model which allow to identifying those processes that are closely related to DevOps. After that, process assessment model allow to assess the capability maturity level of each process found by a Company and then provide criteria and characteristics that need to be fulfilled to reach a particular maturity level. Once the foundation of the maturity components are built and established, the maturity model will be tested by realizing a self-assessment to the company adopting DevOps, in order to validate the proposal. Therefore, this proposal represents a different approach to DevOps maturity model and it is structured following processes identified in the literature.

1.1. Research Problem

The DevOps phenomenon is gathering pace as more organizations seek to leverage the benefits it can potentially bring to software engineering functions. This approach helps deliver value faster and continuously, reducing problems due to miscommunication between team members and accelerating problem resolution. It raises the questions about the point of the time in which companies needs to analyze the techniques and approaches followed by them, regarding the implementation of DevOps, and assess where they are standing in terms of adoption, what capabilities they need to acquire, and what tools can support along this process in order to improve upon their performance.

MM has been used to assess adopted practices, processes, capabilities (Capability Maturity Model) or the business intelligence in the organization. A Maturity Model for DevOps could be a very useful metric to evaluate and analyze the tools, techniques, and approaches followed by an organization concerning DevOps.

Succinctly, the problem that we aim to address **is the lack of DevOps maturity model based on the ISO standard to guide companies to assess the maturity of their DevOps adoption, by helping them assess their current DevOps maturity and move it to a next maturity level.** This resulted in the following research objectives for this study:

RO1.Propose a DevOps maturity model to assess the current state of DevOps adoption in an organization.

RO2.Provide guidance to achieve higher levels of DevOps maturity.

In order to design an effective and comprehensive DevOps maturity model, it is necessary to research the design context. Thus, major question was raised to address the objective of this research.

RQ1. What is a suitable DevOps Maturity framework for assessing the maturity of a DevOps environment in any organization?

2. Research Method

For conducting and guide this research the Design Science Research Methodology (DSRM), provides a process model for doing research in Information Systems and other applied resource disciplines, as well as a mental model for reviewers to evaluate researchers. [16].

DSRM comprises the following steps:

Problem and motivation identification – In this section, we identify the importance and purpose of the research problem. Therefore, we identified one problem that motivated this literature. From the problem statement in chapter 1, it became clear a need exist for DevOps maturity models and frameworks, as well as more empirical studies addressing the subject DevOps in general.

Define the objectives for a solution – A maturity model will be created as described in section 9 to address the identified problem. The solution will consist of a literature review and experts validation of the maturity model.

Design and development – The design of the model consists of three steps. First, identification of the DevOps processes by the literature review (section 4). Secondly, the process reference model is created based on the identified DevOps processes. Finally, process assessment is created according to the existent PRM as it is described in section 9.

Demonstration - The newly created DevOps maturity model is used for self -ssessment in Company A, for measuring the maturity of the respective DevOps implementation. This will provide insights into the applicability of the model.

Evaluation - Observe and measure how well an artefact supports a solution to the problem, comparing the objectives to the results observed from the use of the artefact in the demonstration.

Communication - The results of this study is communicated in two manners; this research report and the article derived from them.

3. Background

3.1. DevOps

The first usage of DevOps, a combination of development and operation, stems from a presentation during the 2008 Agile conference by Debois and Shafers [17]. Even though this is more than a decade ago it seems that research on DevOps is still in its infancy [18]. Therefore, studies tried to define DevOps:

"DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions while guaranteeing their correctness and reliability."[19]

"A set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production while ensuring high quality." [20]

3.2. Collaboration and Communication

The basic DevOps aspect of collaboration is the implied collaboration between the Development and Operation side of software engineering. Ghantous and Gill literature research of 2017, researched this more deeply and showed that collaboration and communication is the most frequently mentioned conceptual element to describe DevOps reaching 23 percents in overall of articles [14]. Lwakatare splits this up in two main practices [17]: "Increasing the scope of responsibilities" and "intensifying cooperation and involvement in each other daily work."

3.3. Automation

Automation underlines most of the practices that constitute DevOps. According to the assumption made by Humble in regard to DevOps: "achieving both frequent, reliable deployments and a stable production environment" [17]. This can be achieved by creating a continuous delivery process which consists of continuous planning, integration, deployment, testing, and monitoring [21].

3.4. Culture

The Changeover to a new culture can be difficult, however, Shamow described the focus on the change of culture in companies is a necessity for adopting DevOps [22]. Some of those changes, as described by Shamow, are the importance for people inside the company to know the seriousness of bypassing the DevOps teams in crises, to not worry about specific tools and to provide full transparency between groups.

3.5. Monitoring

Monitoring is multipurpose, it can be used by developers to make sure that the deploying software is performing correctly and detect problems early on, but also to prevent problems from arising. For example, by monitoring the physical capabilities of the system (CPU, memory, hard disk space) with effective tools, this can prevent system crashes or applications getting too slow by adding enough resources before those issues occur.

3.6. Measurement

Measurement is very important for evaluating the success of both the development and operation teams. This can be achieved by monitoring high-level and low-level metrics.

3.7. Maturity Model

A maturity model is a conceptual model that consists of a sequence of discrete maturity levels for a class of processes in one or more business domains. It represents an anticipated, desire or typical evolutionary path for these processes. They are commonly used as an instrument to conceptualize and measure the maturity of an organization or a process regarding some specific target state [23].

3.8. Capability Maturity Model Integration

The Capability Maturity Model Integration (CMMI) model is an exception to most maturity models, as much empirical evidence on this model exists and the CMMI has been used as a framework for many other models. The CMMI-based process improvement resulted in better project performance and higher quality products through cost reduction, better scheduling of requirements, better quality products, higher customer satisfaction and higher return of investment as described by Goldenson and Gibson based on their research in 35 organizations [24][25].

3.9. Process Reference Model

Reference Model is an abstract framework for understanding significant relationships among the entities of some environment that enables the development of specific architectures using consistent standards or specifications supporting that environment. it describes for a certain application domain a set of processes. thus, each process is described by its purpose and the associated process outcomes.

3.10. Process Assessment Model (PAM)

Relates to one or more Process Reference Models. A PAM holds all details to determine process maturity.

With the objective of performing an assessment, the document ISO/IEC 33002 defines the minimum set of requirements needed to achieve objectives, and consistent results, to form a structure for the assessment of process and the application of process assessment

- facilitates self-assessment;
- provides a basis for use in process improvement and capability determination;
- produces a process rating;
- addresses the ability of the process to achieve its purpose;
- is applicable across all application domains and sizes of organization;
- can provide an objective benchmark between organizations.

4. Systematic Literature Review

As described in section Research Methodology and according to DSRM develop and design phase, it was chosen to start a literature review on the existent DevOps concepts, practices, processes, and roles to create a comprehensive overview of the context of DevOps processes.

This section aims to present the SLR approach [26][27], by following the proposed guidelines to identify, analyze, and interpret all available and relevant literature published in the context of DevOps concepts, processes, practices, and roles. The SLR method comprises three consecutive stages: planning, conducting, and reporting. Figure 1 illustrates the phases and activities comprised by the SLR.

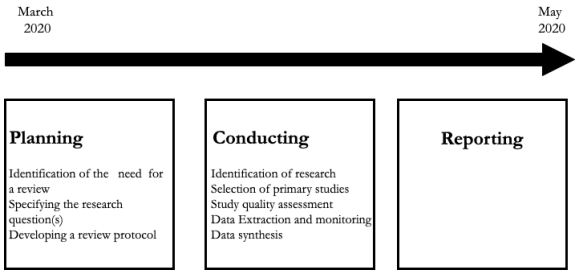

Figure 1: Research method phases and activities [26]

In the initial phase, planning review, the first step consists of identifying the need for a review. This section focuses on the remaining planning phase, which comprises the definition of the research questions and the development of the review protocol, which outlines the procedures for the conducting phase that includes the search process; the establishment of inclusion and exclusion criteria; quality assessment instrument, and the data extraction and synthesis strategy.

4.1. Research Questions

- RQ1: What are the DevOps concepts?
- RQ2: What are the DevOps processes?
- RQ3: What are the DevOps practices?
- RQ4: What are the DevOps roles?

4.2. Search Process

The review comprises the following well-known academic databases: ACM Library, AIS Library, IEEEXplore, Google Scholar, Research Gate, Semantic Scholar, EBSCO.

The search conducted aims to find all literature, having a focus on academic studies. The papers returned were obtained by applying the search string to the title and abstract in each digital library.

4.3. Inclusion and Exclusion criteria

For this section, the inclusion and exclusion criteria were defined to filter the literature and identify the most relevant ones for this work.

- Article not including key elements of search string
- Empirical study included
- Clearly describes DevOps and its related concepts
- Focusses on the processes, practices, and roles

4.4. Quality Assessment

For each selected paper a quality assessment was conducted to appraise the relevance and quality of their content.

The quality assessment of the selected articles was based on the four questions defined in our research. The scoring procedure was based on [26], where each question could have the possible answers: Yes (Y) =1 if the article answers to the questions with a 100 of certainty, partially (P) = 0.5 if the article does not fully answer the questions, and No (N) = 0 if the does not answer to any questions.

4.5. Findings

This section describes the results obtained by conducting the SLR. The selection process was conducted based on a set of steps described in figure 2.

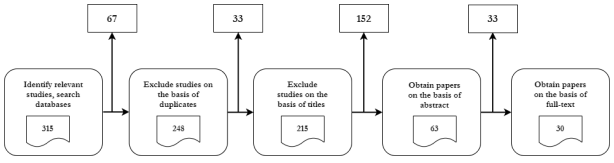

Figure 2: Research method phases and activities [26]

The search strategy applied at stage 1 of the review resulted in 315 studies including the possible set of duplicates. At stage 2, duplicates studies were removed automatically and then reviewed manually to ensure that no duplicates were left behind, allowing to reduce the number of studies to 248. At stage 3, reviews were made to exclude studies based on the titles, resulting in 215. At stage 4, reviews of all 219 abstracts and as result of this process 63 studies were selected. At the five and final stages, 63 studies were evaluated in detail focusing on the full text, some of the studies were excluded ending up with 30 primary studies.

As it is possible to notice in Figure 3, most of the selected papers were from ACM Digital Library, IEEE Xplore, Research Gate, followed by Springer

Link. After that, AIS Electronic Library, Science Direct and Semantic Scholar with the same number of papers respectively.

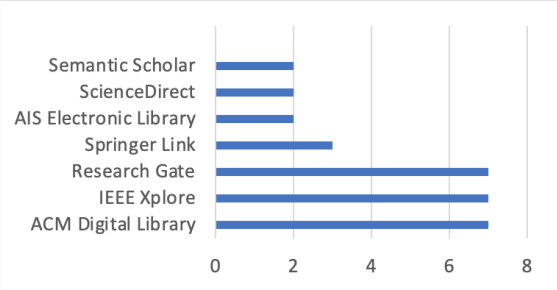

Figure 3: Data sources of the selected papers

From Figure 4 is possible to observe the distribution of the papers selected over the years and it shows that 2017 is the year from which more papers were selected for this research, this might indicate that in this particular year there was an increase of research in the field of DevOps.

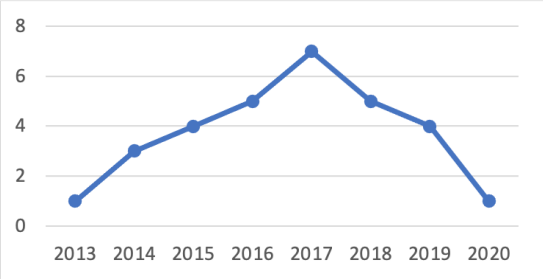

Figure 4: Distribution of the selected papers over the years

5. Concepts

For the concepts, studies explicitly using the term DevOps concepts, are shown in table 4.4, which presents a set of concepts and establish the comparison of the often-mentioned concepts in the studied literature. Therefore, we can highlight Continuous Delivery, Automation, Automated Pipeline, Communication and Collaboration, and Knowledge Sharing as most mentioned.

6. Processes

The identified processes present a vast scope of what could involve a DevOps adoption. When analyzing the identified processes, it was possible to perceive that there is no standard definition for DevOps process according to [28], but a few different versions and implementations have been provided by the literature, in which the process is composed of stages that encompass software development and operation.

It is important to highlight the processes that stood out for the number of times they were cited by articles as Plan, Test, Release, Deploy, and Monitor.

7. Practices

To analyze the practices proposed as DevOps practices in the literature we attempted to identify those which have been explicitly presented as DevOps practices. Which is possible to highlight Continuous Delivery, Continuous Integration, Continuous Deployment, Continuous Monitoring, and Continuous Testing as the most cited.

8. Roles

From the carefully selected and reviewed literature, it was possible to identify DevOps related roles which are performed by different people in DevOps process (stages), however, it does not mean that they are all roles covering this subject. Therefore, it is important to highlight that roles vary according to the organizational structures. as the most frequently cited roles, DevOps Engineer, Developer, IT Manager, Network Administrator, and Cloud Engineer.

9. DevOps Maturity Model Proposal

9.1. objectives

The main goal of this proposed solution is to develop a maturity model to assess the current state of DevOps adoption in the organization and provide instruction to achieve a higher level of DevOps maturity.

In order to achieve this objective two tasks must be completed:

- The creation of a Process Reference Model (PRM)
- The creation of a Process Assessment Model (PAM)

9.2. PRM for DevOps

The previous sections provide the necessary material to create a first version of the DevOps maturity model. In this case, it was chosen to use ISO/IEC framework as a starting point, based on the arguments given in the objectives section.

9.3. PAM for DevOps

PRM are always related to PAM which holds all indicators to determine the maturity of the processes of the reference model. In this section, we will cover the processes identified and detailed regarding the process assessment model adopted and what are the conditions to achieve a certain maturity level for each process.

The study resulted in the following processes that are grouped into two major action fields Figure 6.

Process ID	SOM.03
Title	Monitor
Context	Phase takes place parallel to the Operate phase and consists of data collection, analysis, and feedback to the start of the pipeline and to other phases as needed.
Purpose	This activity the software in production is monitored by mainly sysadmin, operators and other managing the project.
Outcomes	As result of successful implementation of this process: 1. Applications performance are monitored. 2. Services performance are monitored. 3. Application log is monitored. 4. Code integrity is verified. 5. User activity monitored for improper behaviour of the system. 6. Information about issues from a software release in production is identified and collected. 7. Data collected and analysed, and feedback is provided.

Figure 5: PRM for Monitor

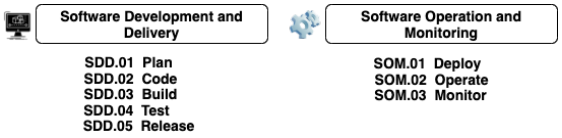

Figure 6: Process Reference Model (PRM) including DevOps processes identified from our previous research effort and grouped by action fields. Adapted from [29]

It is important to emphasize that PAM is a two-dimensional model concerning the process dimension and capability dimension. The representation of what constitutes the process dimension of the PAM for eight processes are presented in the following figure.

Process ID	SDD.01
Title	Plan
Context	This step is usually dealt with by project managers in collaboration with the team and exploiting project management tools.
Purpose	Activity planning and task scheduling for the current release.
Outcomes	As result of successful implementation of this process: 1. Product requirement collected and documented. 2. Stakeholders are identified. 3. Project milestones are defined. 4. User experience is designed. 5. Scope defined. 6. Features are planned.
Base Practices	SDD.01.1 – Identify and collect product requirement [Outcome 1] SDD.01.2 – Identify Stakeholders [Outcome 2] SDD.01.3 – Set project milestones [Outcome 3] SDD.01.4 – Design user experience [Outcome 4] SDD.01.5 – Set scope[Outcome 5] SDD.01.6 – Plan features [Outcome 6]
Inputs	
Document product requirements [Outcome 1] Envisioning software [Outcome 4] Task management [Outcome 3] Schedules [Outcome 2]	
Outputs	
Product requirements specification document [Outcome 1] Vision of the project [Outcome 4]	

Figure 7: PAM for Monitor

9.4. Maturity Model

According to the information provided by the PRM and PAM, we create the first version of the DevOps maturity model. In this case, it was chosen to use the ISO/IEC 330xx family standard. Therefore, the implementation of the model will help improve operational efficiency and increase the visibility of the processes. The strength of this model is that using tools of ISO/IEC helped on the identification and create the reference of the DevOps processes that were used as the base for the following phases.

9.5. Maturity Levels

The aim of maturity levels is to classify organisations according to their ability to control their various processes. Therefore, they are defined on a six-point ordinal scale that enables maturity to be assessed from the bottom of the scale, Incomplete, through to the top end of the scale, Innovating. The scale represents increasing maturity of the implemented process, from failing to achieve the process purpose through to continually improving and able to respond to organizational change.

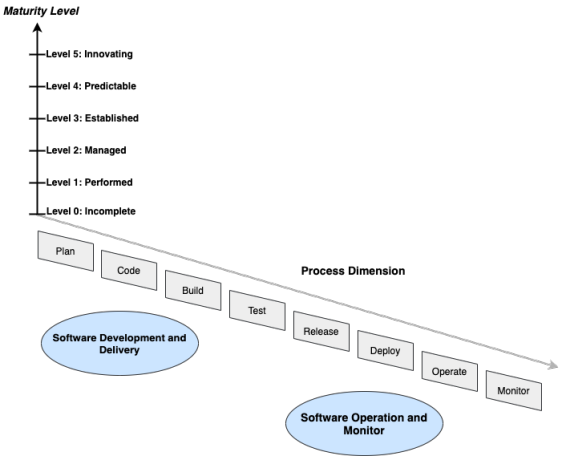

Figure 8: Maturity Model

10. Demonstration

The objective of the DevOps maturity model is to be applicable in real scenarios, thus it is important to interview experts and practitioners in the DevOps field in which processes outcomes they think are necessary and increase value for successful implementation of DevOps in the organization.

Regarding the assessment, we met with Company A's DevOps team in order to perform the assessment following PAM addressing all the processes identified. Although PAM is carried out taking into account process best practices, inputs, and outputs, the assessment only took into account the process purpose and outcomes as it was the first interaction with the processes.

The interview taken with the experts followed a questionnaire, but we could divert if something came up that had to be explored further. The interview was done in one round comprised of questions about DevOps and processes purpose and outcomes associated with different stages of DevOps processes. The data acquired would be very important further to enrich the existent model, although it was not possible to perform more rounds of interviews.

The demonstration counted on the assessment of

plan, code, build, test, release, deploy, operate, and monitor. However, for this section we brought only one example to give an overview of how the assessment was performed.

10.1. Results

The meeting took about one hour and fifteen minutes, it started with a brief presentation of all participants gathering some general data, like the background and years of experience of the interviewee. Then a presentation of the framework was made known, detailing high-level steps taken to achieve it.

The next step following up the meeting was framework assessment with two members of DevOps team of the Company A, and it was done assessing each of processes purposes and outcomes and evaluating them accordingly to the process attribute rating scale defined by ISO/IEC 330xx family. Figure 9 show the result obtained from the interview.

		Level			
	Outcomes	Not Achieved	Partially achieved	Largely achieved	Fully achieved
Plan	1. Product requirement collected and documented				X
	2. Stakeholders are identified			X	
	3. Project milestones are defined			X	
	4. User experience is designed				X
	5. Scope defined			X	
	6. Features are planned				X
Monitor	1. Applications performance are monitored			X	
	2. Services performance are monitored			X	
	3. Application log is monitored			X	
	4. Code integrity is verified				X
	5. User activity monitored for improper behaviour of the system				X
	6. Information about issues from a software release in production is identified and collected				X
	7. Data collected and analysed, and feedback is provided			X	

Figure 9: Process Assessment Result

The first process that was assessed in the interview was the Plan that is composed of six outcomes, where we can observe the highest level obtained was F and the lower was L. According to the interviewee, the lowest levels are related to the non-fulfilment of some activities that involve the outcome by all teams involved in the process. The interviewee explained this as;

"Not all features are planned at this stage and not all project milestones are defined"

The second process regarding DevOps is the Operate, which belongs to the major field of software operation and monitoring. it is composed by seven outcomes of which four obtained L as rating and the other three obtained F. Although it could be considered as a good, improvements have to be made to allow the fulfilment of all process outcomes. as the interviewee said:

"Although we have this outcome implemented, we are not mature enough to assure the complete fulfilment of them. we still have to improve it"

11. Conclusions

To conduct this research, we followed DSRM process, that comprises 6 phases of development. First, we identified the problem, the lack of DevOps maturity models to guide companies to assess the maturity of their DevOps adoption. The main objectives propose a DevOps maturity model to assess the current state of DevOps adoption in an organization and provide guidance to achieve higher levels of DevOps maturity. To address this problem, a literature review was conducted addressing DevOps concepts, processes, practices, and roles and as result of this, we found 32 concepts, 9 processes, 33 practices, and 20 roles as it can be observed in tables 4.4, 4.6, appendix, and 4.7. From this stands out the concepts of automation, knowledge sharing, continuous delivery, infrastructure as a code, communication and collaboration as the most frequently cited ones; Plan, Test, Release, Deploy, and Monitor for the processes; Continuous Delivery, Continuous Integration, Continuous Deployment, Continuous Monitoring, and Continuous Testing for practices; and finally, as the most frequently cited roles, DevOps Engineer, Developer, IT Manager, Network Administrator, and Cloud Engineer.

The SLR also allowed us to explore and bring to this research the relation between DevOps processes and some practices, as shown in figures 4.5, and 4.6, as well as the relationship between practices and roles (figure 4.7). Thus, the identified DevOps processes were the foundation for PRM and consequently served as the base for PAM.

Once the processes were identified through SLR, thus constituting our PRM, we moved to PAM where we had the processes Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor with their respective details following the procedure and structure of ISO/IEC 330xx family standard.

11.1. Limitations

Regarding limitations, it was not possible to gather enough information and present a robust conclusion regarding specific topics, such as Outcomes, since DevOps is a recent subject. The current research cannot avoid biases since sources of literature written in other languages were excluded and certain unavailability to find many studies addressing all DevOps related subjects in electronic databases. Since DevOps is recent, there are not a lot of experts in this area.

This research was meant to fill a gap that was found during the literature review(chapter **??**). However, the resulting model does not achieve fully its purpose, thus it was not possible to do more

than one interaction that could have provided more inputs to improve the existing outcomes and to the evaluation model.

The lack of literature addressing the DevOps processes made it very difficult to identify the base practices, inputs, and outputs of each process. We did our best to consult all results from the used search keys, however, some were inaccessible due to access restrictions. Certainly, it is not guaranteed that the chosen search keys, sufficiently represented the goal to retrieve all literature available on DevOps processes.

11.2. Future Work

This study has provided information about the DevOps maturity model and upon this creates a comprehensive model based on ISO/IEC. However, this research is not complete and can be taken a further look at. These opportunities will be described in this section.

The development of a tool that would allow the assessment to be made and that was flexible and adaptable to Companies to evaluate their maturity.

The DevOps maturity model created in this research can be used by other researchers to further build upon. This can be achieved by doing more research, which will not only contribute for DevOps maturity model but allow more researchers to use ISO/IEC a basis to these models.

Demonstrate and evaluate the framework to a substantially larger number of companies, ideally in companies that differ in size and industry, with the ultimate goal of being able to benchmark effectively. Conclusions, future work and some final remarks...

Acknowledgements

This thesis is the conclusion of two and a half years of studying computer science and engineering MSc. at the Instituto Superior Técnico. It has been an amazing time of personal growth, both inside and outside of the Instituto environment. This thesis has taken more than 6 months to complete, however, I would not have been able to manage this without the help of the other people.

To my two mothers Luísa Manuel and Ana Madaleno, I am very thankful for granting me this desired opportunity and because for believe in myself when I thought I was not able.

I would like to acknowledge my dissertation supervisors Prof. André Vasconcelos and Prof. Miguel Mira da Silva for their insight, patience, support and sharing of knowledge that has made this Thesis possible.

I would like also to thank my parents for their friendship, encouragement and caring over all these years, for always being there for me through thick and thin and without whom this project would not be possible. I would also like to thank my aunts, uncles and cousins for their understanding and support throughout all these years.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

References

[1] Barry W Boehm. A spiral model of software development and enhancement. *Computer*, (5):61–72, 1988.

[2] Adriana Herden, Pedro Porfirio Muniz Farias, and Adriano Bessa Albuquerque. An agile approach to improve process-oriented software development. In *Computer Science On-line Conference*, pages 413–424. Springer, 2016.

[3] G Bou Ghantous and Asif Gill. Devops: Concepts, practices, tools, benefits and challenges. *PACIS2017*, 2017.

[4] J.M. Radstaak. Developing a devops maturity model: A validated model to evaluate the maturity of devops in organizations. Master's thesis, University of Twente, 2019.

[5] Viral Gupta, Parmod Kumar Kapur, and Deepak Kumar. Modeling and measuring attributes influencing devops implementation in an enterprise using structural equation modeling. *Information and Software Technology*, 92:75–91, 2017.

[6] Jan Waller, Nils C Ehmke, and Wilhelm Hasselbring. Including performance benchmarks into continuous integration to enable devops. *ACM SIGSOFT Software Engineering Notes*, 40(2):1–4, 2015.

[7] Johannes Wettinger. Concepts for integrating devops methodologies with model-driven cloud management based on tosca. Master's thesis, University of Stuttgart, 2012.

[8] Avita Katal, Vinayak Bajoria, and Susheela Dahiya. Devops: Bridging the gap between development and operations. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1–7. IEEE, 2019.

[9] Floris Erich, Chintan Amrit, and Maya Daneva. A mapping study on cooperation between information system development and operations. In *International Conference on Product-Focused Software Process Improvement*, pages 277–280. Springer, 2014.

[10] Jason Sharp and Jeffry Babb. Is information systems late to the party? the current state of devops research in the association for information systems elibrary. 2018.

[11] Roy Wendler. The maturity of maturity model research: A systematic mapping study. *Information and software technology*, 54(12):1317–1339, 2012.

[12] Lianping Chen. Continuous delivery: Huge benefits, but challenges too. *IEEE Software*, 32(2):50–54, 2015.

[13] Gerry Gerard Claps, Richard Berntsson Svensson, and Aybüke Aurum. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software technology*, 57:21–31, 2015.

[14] Wolfgang Gottesheim. Challenges, benefits and best practices of performance focused devops. In *Proceedings of the 4th International Workshop on Large-Scale Testing*, page 3. ACM, 2015.

[15] Mohammad Zarour, Norah Alhammad, Mamdouh Alenezi, and Khalid Alsarayrah. A research on devops maturity models. 8:4854, 2019.

[16] Bob Aiello and Leslie Sachs. *Agile application lifecycle management: Using DevOps to drive process improvement*. Addison-Wesley Professional, 2016.

[17] Lucy Ellen Lwakatare, Pasi Kuvaja, and Markku Oivo. An exploratory study of devops extending the dimensions of devops with practices. *ICSEA 2016*, 104, 2016.

[18] Pilar Rodríguez, Alireza Haghighatkhah, Lucy Ellen Lwakatare, Susanna Teppola, Tanja Suomalainen, Juho Eskeli, Teemu Karvonen, Pasi Kuvaja, June M Verner, and Markku Oivo. Continuous deployment of software intensive products and services: A systematic mapping study. *Journal of Systems and Software*, 123:263–291, 2017.

[19] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. A survey of devops concepts and challenges. *ACM Computing Surveys (CSUR)*, 52(6):1–35, 2019.

[20] Leah Riungu-Kalliosaari, Simo Mäkinen, Lucy Ellen Lwakatare, Juha Tiihonen, and Tomi Männistö. Devops adoption benefits and challenges in practice: a case study. In *17th International Conference on Product-Focused Software Process Improvement*, pages 590–597. Springer, 2016.

[21] Manish Virmani. Understanding devops & bridging the gap from continuous integration to continuous delivery. In *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)*, pages 78–82. IEEE, 2015.

[22] Eric Shamow. Devops at advance internet: How we got in the door. *IT Journal*, 14:14–16, 2011.

[23] Andreas Schumacher, Selim Erol, and Wilfried Sihn. A maturity model for assessing industry 4.0 readiness and maturity of manufacturing enterprises. *Procedia Cirp*, 52:161–166, 2016.

[24] Dennis R. Goldenson and Diane L Gibson. Demonstrating the impact and benefits of cmmi ® : An update and preliminary results. 2003.

[25] D Gibson, D Goldenson, and Keith Kost. Performance results of cmmi-based process improvement. software engineering institute, 2006.

[26] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. 2007.

[27] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages xiii–xxiii, 2002.

[28] Antonio Capizzi, Salvatore Distefano, and Manuel Mazzara. From devops to devdataops: Data management in devops processes. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pages 52–62. Springer, 2019.

[29] Henner Gimpel, Sabiölla Hosseini, Rocco Xaver Richard Huber, Laura Probst, Maximilian Röglinger, and Ulrich Faisst. Structuring digital transformation: A framework of action fields and its application at zeiss. *J. Inf. Technol. Theory Appl.*, 19(1):3, 2018.