A maturity model for DevOps

Daniel Teixeira, Rúben Pereira*
and Telmo Henriques

Instituto Universitário de Lisboa (ISCTE-IUL),
Department of Information Science and Technology,
Lisbon, 1649-026, Portugal
Email: dstas@iscte-iul.pt
Email: ruben.filipe.pereira@iscte-iul.pt
Email: telmo_antonio_henriques@iscte-iul.pt
*Corresponding author

Miguel Mira da Silva

Universidade de Lisboa – Instituto Superior Técnico (IST),
Department of Computer Science and Engineering,
Lisbon, 1049-001, Portugal
Email: mms@tecnico.ulisboa.pt

João Faustino and Miguel Silva

Instituto Universitário de Lisboa (ISCTE-IUL),
Department of Information Science and Technology,
Lisbon, 1649-026, Portugal
Email: joao_pedro_faustino@iscte-iul.pt
Email: marsa@iscte-iul.pt

Abstract: Nowadays, businesses aim to respond to customer needs at unprecedented speed. Thus, many companies are rushing to the DevOps movement. DevOps is the combination of Development and Operations and a new way of thinking in the software engineering domain. However, no common understanding of what it means has yet been achieved. Also, no adoption models or fine-grained maturity models to assist DevOps maturation and implementation were identified. Therefore, this research attempt to fill these gaps. A systematic literature review is performed to identify the determining factors contributing to the implementation of DevOps, including the main capabilities and areas with which it evolves. Then, two sets of interviews with DevOps experts were performed and their experience used to build the DevOps Maturity Model. The DevOps maturity model was then developed grounded on scientific and professional viewpoints. Once developed the Maturity Model was demonstrated in a real organisation.

Keywords: DevOps; maturity model; CMMI; practices; capabilities; areas; systematic literature review; interviews; definition; design science research.

Reference to this paper should be made as follows: Teixeira, D., Pereira, R., Henriques, T., Mira da Silva, M., Faustino, J. and Silva, M. (2020) 'A maturity model for DevOps', *Int. J. Agile Systems and Management*, Vol. 13, No. 4, pp.464–511.

Biographical notes: Daniel Teixeira is an information systems consultant. He graduated as MSc in Computer Engineering at ISCTE. He has been working in several projects in industry. Besides his professional work, he is very enthusiastic about trending information systems topics. So far, his research interests rely on DevOps and maturity models.

Rúben Pereira is an Assistant Professor at ISCTE. He has a PhD in information systems at Instituto Superior Técnico where he also graduated as a Master's in Computer Engineering and Computer Science. He has been a consultant in several industries, such as services, banking, telecommunications, and e-commerce, among others. He is the author of several scientific papers in the area of IT Services Management and IT Governance, covering the main IT Frameworks like ITIL and COBIT. His areas of scientific interest extend to information technology risk management, BPM, continuous improvement and innovation, process optimisation, digital transformation, DEVOPS, RPA, among others.

Telmo Henriques is currently an Invited Assistant Professor at ISCTE-IUL and a Research Associate at ISTAR-IUL. He is an Associate Editor of the *Journal of Global Information Management* and of the *Information Resource Management Journal*, having also collaborated, as a member of the Editorial Advisory Board, on the *IGI Encyclopedia of Organizational Knowledge, Administration, and Technology* and the *IGI Encyclopedia of Information Science and Technology*. His past long professional management experience, within the areas of management consulting and banking, has involved the assumption of main responsibilities on ICT consultancy, information systems methodologies, quality management and organisational change & development.

Miguel Mira da Silva is currently Associate Professor of Information Systems in the University of Lisbon, leader of the research group 'Digital Transformation' at INOV, and coordinator of the MISE online MSc. He has a PhD in Computing Science from the University of Glasgow and an MSc in Management from the London Business School. Miguel created five companies, published four books and 200 research papers, managed dozens of research and consulting projects, graduated nine PhDs and 150 MSc students, and created a MOOC about digital transformation. His current interests include digital transformation, IT governance, and online learning.

João Faustino is an IT Consultant. He is also a PhD student in Information Science and Technology at ISCTE-IUL where he also graduated as MSc in Computer Engineering. Most of his career, he has worked as a consultant on the Financial Services industry, on application management teams for both corrective and enhancement maintenance of Core Systems for Insurance companies. Besides his professional work, which is mainly related with Oracle Technologies, he is very enthusiastic about open source technologies, data science, mobile development and process optimisation. Most of his research is about IT service management and DevOps.

Miguel Silva is the head of a Software Engineering team in one of the biggest European engineering company. He has MSc in Computer Engineering at ISCTE and he achieved his BSc graduation at FCT-UNL also in Computer Engineering. He has been working since 2009 in the ITSM area in the telecommunications industry, and he is responsible to manage a DevOps team that manages and implements one of the biggest Service Now instances in the world. His main interest is related with agile methodologies (SCRUM, DevOps, etc).

1 Introduction

In recent years, the advancements on DevOps area have facilitated a lot of new growth opportunity for software companies (Nidagundi and Novickis, 2017) as it improves the way how a business delivers value to its customers, suppliers, and partners, it is an essential business process, not just an IT capability (Katal et al., 2019). This is one of the main reasons why the DevOps' adoption is growing and is a new tendency in business and IT alignment (Bucena and Kirikova, 2017). DevOps allows a business to maximise the speed of delivery of a product or service, from the initial idea to production release and all the way up to customer feedback to improvements based on that feedback (Koilada, 2019).

Businesses today need to respond to customer needs at unprecedented speed. Driven by this need for speed, many companies are then rushing to the DevOps movement and implementing Continuous Delivery (Chen, 2018).

The growth opportunities for DevOps continue to increase. Ovum, a market-leading data, research and consulting company, sees plenty of evolution potential in DevOps as there is potential for improved integration with Application Lifecycle Management on the dev side and improved integration with operations and IT business services (Azoff, 2016). According with the 2018 State of DevOps Report has been registered a steady increase in survey responses from people on DevOps teams, from just 16% in 2014 to 29% in 2018 (Velasquez et al., 2018).

The adoption of DevOps drives a challenging cultural shift towards collaboration and knowledge-sharing between SD, quality control and operations (Colomo-Palacios et al., 2018). The tremendous growth in demand for DevOps has, however, led to the appearance of new needs. For instance, many companies find it difficult to understand what DevOps is and what advantages it will have (St et al., 2017).

Many companies miss the maturity of the concept – with no clear definition of DevOps and its practices, no clear goals available and a lack of understanding about development workflow phases and responsibilities (Bucena and Kirikova, 2017). There is both a lack of understanding around DevOps and a clear definition of what it is (Lwakatare et al., 2015). Therefore, organisations are not sure how to effectively implement DevOps capabilities (Chen, 2019). Plus, complexity is evolving since DevOps security concerns start to be raised (Prates et al., 2019).

The disruptive nature of the changes required to adopt DevOps leads to organisational and business stress. While Zhu et al. (2016) consider the organisational strains as being standard for new technologies, for Bucena and Kirikova (2017) the adoption of DevOps is not trivial and can require complex changes in an enterprise's process, organisation and workflow. To succeed in adopting DevOps, the enterprises should understand the different aspects that are related to the DevOps approach and have a well-thought-out strategy. They should start the adoption process with a clear idea of what actions should be performed, how they should be prioritised, what tools could support these actions, and how to measure the success of the adoption process (Bucena and Kirikova, 2017). Moreover, the way an organisation is structured may influence DevOps' adoption, for example, when discussing communication, common goals and practices, decision making, and systems thinking within the organisation (Smeds et al., 2015b).

Whereas DevOps benefits are widely discussed regarding DevOps culture and available tools, it makes sense to exist a Maturity Model (MM) for DevOps approaches. A MM is a widely used technique that has proven valuable for assessing business processes or certain aspects of organisations, as it represents a path towards an increasingly organised and systematic way of doing business (Proenca, 2016). They also allow for a better positioning of the organisation and help find better solutions for change (Becker et al., 2009). Moreover, MM are an important tool for business-IT alignment (Pereira and Da Silva, 2011; Aguiar et al., 2018).

According to the literature, both areas and capabilities play an important role in DevOps adoption and maturation. Therefore, this study aims to: Develop a MM for DevOps (RO1). To achieve this objective, it may be necessary to identify both DevOps capabilities (RQ1.1) and DevOps areas (RO1.2). Plus, the two sets of interviews with DevOps experts were performed to tune the final artefact.

2 Theoretical background

2.1 DevOps

A good cooperation between IT Development and IT Operation teams is viewed to be crucial in order to ensure successful deployment and operations of IT systems (Tessem and Iden, 2008). However, for historical reasons, most IT organisations are characterised by clear boundaries between these two teams, which have very different goals, mindsets and cultures (Swanson and Beath, 1990; Gazivoda, 2018).

According to Sharma (2014), many organisations are not successful with software projects and their failures are related to the challenges in product development and delivery. Despite this, many companies find that the development and delivery of software applications are crucial to their business, and that only 25% of companies consider their teams to be efficient (Sharma, 2014). This gap in efficiency leads to many losses of business opportunities. This demonstrates that even a disruptive methodology cannot be perfect for every project.

In order to address the problems between the development and operations teams a new agile approach appeared, namely DevOps. DevOps has been heralded as a novel paradigm to overcome the traditional boundaries between IT Development (Dev) and IT Operations (Ops) teams (Nielsen et al., 2017) aiming to improve collaboration between these teams. It represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach (Jabbari et al., 2016). However, DevOps is not only influenced by cultural aspects as also supported by technological enablers (Smeds et al., 2015a).

According to Riungu-Kalliosaari et al. (2016), DevOps is a set of practices intended to reduce the time between making a change to a system and this change being placed into normal production, while ensuring high quality. The main goal associated with this concept is to avoid common problems when operations and developers are kept as separated teams (Bezemer et al., 2018).

To sum up, DevOps replaces siloed units with cross-functional teams so organisations may leverage automated development, deployment, and infrastructure and enables teams to continuous work and deliver operational features (Ebert et al., 2016).

2.2 Maturity model

MM's are commonly used as an instrument to conceptualise and measure maturity of an organisation or a process regarding some specific target state (Schumacher et al., 2016). Further, MM intended for a prescriptive purpose of use include good or best practices which is helpful to provide practical guidance (Röglinger et al., 2018). They refer that maturity not only implies a potential for growth in capability, but also focuses on richness and consistency regarding execution. In this regard, Andersen and Jessen (2003) define maturity as the quality or state of being mature. The maturity concept must be related to a state in which organisations are in perfect conditions to achieve their goals (Berssaneti et al., 2012).

Two approaches for implementing MMs exist. With a top-down approach, such as proposed by Becker et al. (2009) a fixed number of maturity stages or levels is specified first and further corroborated with characteristics (typically in form of specific assessment items) that support the initial assumptions about how maturity evolves. On the other hand, when using a bottom-up approach, such as suggested by Lahrmann et al. (2011), distinct characteristics are determined first and clustered in a second step into maturity levels to induce a more general view of the different steps of maturity evolution. This research follows the top-down MM approach proposed by Becker et al. (2009).

2.3 CMMI

Capability maturity model integration (CMMI) (and its predecessor CMM) is a framework intended to cover many software engineering best practices and can be used for SPI. CMMI is most well known in its 'staged' representation, which has five maturity levels. To reach a maturity level, a company must satisfy the goals of the process areas for that and all lower levels. The expected capacity of an organisation that operates in a more mature way depends directly on your ability to perform, control, and improve performance in one or more areas of implementation of the model practices (Barbosa et al., 2007).

CMMI evokes barriers in some because of the processes involved in certification. However, CMMI at its core is not a methodology but rather a set of principles. In the case of CMMI, the set of principles focuses on maturation of a SD process. CMMI is concerned with defining metrics and practices to ensure continuous improvement (Chrissis et al., 2010). The goal of CMMI is not just to support a minimum set of standards to achieve to a particular level, but to enable increasing improvement in organisational processes. CMMI's approach is based on MM. It supports both a staged approach and a continuous model for improvement. It provides several key process areas at different levels. Maturity levels are those that are related to the path which helps organisations to apply improvements to a set of related processes by incrementally addressing successive sets of process areas and goes through 1 to 5.

3 Related work

Since this research aims to study DevOps' maturity, it is mandatory to search literature where it is possible to study other proposals for DevOps' MMs. However, given that DevOps is a new term and concept recently introduced, the author decided to extend the scope of the study to SDMMs. To do that, the author performed a literature review.

A literature review may be helpful distinguishing what has been done from what needs to be done, discovering important variables relevant to the topic, synthesising and gaining a new perspective or identifying relationships between ideas and practice (Hart, 1998). An effective review creates a solid foundation for advancing knowledge. It facilitates theory development, closes areas where a plethora of research exists, and uncovers areas where research is needed (Webster and Watson, 2002). For easier understanding of the peers, as well as to add more scientific rigor to our research, the author decided to follow the concept centric approach proposed by Webster and Watson (2002).

To perform the literature, review the authors have searched and consulted the following digital repositories: IEEExplore, ACM, Research Gate and it was also used the search engine of Google Scholar.

This research was carried out between September of 2018 and January of 2019. The keywords used to perform this research were: 'DevOps maturity model', 'DevOps maturity', 'Software Development Projects maturity model', 'Software development projects maturity', 'Scrum maturity model' and 'Scrum maturity'.

In this section, the main findings regarding Software Development Processes (SDP), Scrum and DevOps MMs are presented (Table 1). Plus, Table 2 details these studies characteristics, while Table 3 contains all the studies mapped with the corresponding maturity vectors found by the authors in the proposed MMs. Together, these three tables explain how this research increases the body of knowledge.

Table 1 SDP, and DevOps MMs

ID	*Author*	*MMs*			*Model*	*Maturity levels*	*Dimension*
		DevOps	*Scrum*	*SDP*			
S.1	Mohamed (2015)	X			CMMI	5	4
S.2	Bucena and Kirikova (2017)	X			Not defined	5	4
S.3	Yin (2011)		X		CMMI	5	Not defined
S.4	Srivastava et al. (2017)		X		Not defined		Not defined
S.5	Kawamoto and De Almeida (2017)		X		CMMI	Not defined	Not defined
S.6	Baskarada et al. (2005)			X	CMMI	5	Not defined
S.7	Patel and Ramachandran (2009)			X	CMMI	5	Not defined
S.8	Buglione (2011)			X	CMMI	5	4
S.9	Santana et al. (2013)			X	CMMI	5	Not defined
S.10	Fontana et al. (2015)			X	CMMI	Not defined	6
S.11	Stojanov et al. (2015)			X	Not defined	5	5

Since DevOps is a recent theme and there are not a lot of dedicated maturity studies in literature (Rong et al., 2016a). So, the authors have decided to include agile and scrum MMs.

Both Scrum and DevOps have in common to broaden the usage of Agile practices to operations to streamline the entire software delivery process in a holistic way (Hüttermann, 2012; Bang et al., 2013). Table 1 presents all the MMs for SDP, Scrum and DevOps found among the literature.

From the analysis of Table 1 some conclusions can be withdrawal. The low number of DevOps MMs that has been found indicate that few studies exist deep studying DevOps. The number of studies on SDP is greater than for scrum and DevOps. One of the main reasons for this is that most of the SDP uses Agile methodology, which in turn is the basis for both DevOps and Scrum so it is expected that there exist more studies about this theme than for the others.

CMMI seems to be the basis of these models since it was used in 73% of these studies. It was not explicit any of the vectors that constitutes the Scrum' MMs and, apart from one study, the same happened to the number of levels used. This is justified by the fact that CMMI is a well-known methodology used to develop and refine an organisation's SD process (Farkas and Walsh, 2002). CMMI is an approach to improve processes that provides elements that are essential for an effective process. It brings together best practices that address development and maintenance activities, thus covering the entire lifecycle of a product from conception to delivery and maintenance (Chrissis et al., 2010). It has been also included a vector named 'Dimension' that represents the number of vectors that were represented in model. From Table 1, it is possible to see that the study with less dimensions had four and on the opposite side, the study with more dimensions has six. This helps the authors to put into perspective the number of dimensions used in other MM, to understand the number of dimensions that should be used in this study.

Studies' characteristics are better detailed in Table 2 where vectors are used for proper analysis, such as the year in which the model was developed, which MM was based on, if it follows Becker's top-down approach, if the author justified the vectors used, whether they comply with the Design Science Research (DSR) steps and if any demonstration of the model was performed in practice.

Overall, two MMs for DevOps were identified in literature. However, as one can see in Table 2, both MMs lack the use of structured methods in the design process which may raise doubts on their scientific rigor. For instance, only one is based on CMMI and none adopts Becker theory or DSR to build the MM.

Moreover, Table 3 lists and synthesises the related work and identifies what vectors were used to design each analysed MM. By doing it, the authors aimed to identify the main vectors that were applied on those case studies and understand the reasons behind those.

For a better understanding, the studies have been grouped by approach. A vector can be written on a different way depending on its context, so the authors have grouped these vectors by the meaning of the vector. Table 3 shows the vectors grouped by study.

Through the analysis of Table 3, it can be devised that several MM exist in the literature (11 studies). In six of these studies, the authors did not specify the vectors that would be used. Although DevOps studies are less than agile studies, some agile MMs use the same vectors defined by the DevOps MMs. This may be due to the fact that, first, DevOps and agile keep a close relationship and, secondly, DevOps is a recent topic and there is not much information available about it (Hussain et al., 2017). On agile studies, with some exceptions, it appears that each author defined most of their vectors.

Focusing on DevOps studies, there are no common characteristics present among the two models found. This also proves that the field needs further developments to reach more consensus and completeness. Each author decided to establish their own vectors based on what they thought best defines the characteristics and that could help define the maturity of DevOps in the context of their studies.

Table 2 MMs characteristics

ID	*Year*	*Proposed MM*	*Based MM*	*Becker's top-down approach*	*Vectors validation*	*DSR*	*Demonstration*
S.1	2015	DevOps	CMMI	Not used	Not validated	Not used	Not applied
S.2	2017	DevOps	Not defined	Not used	Validated	Not used	Applied
S.3	2011	Scrum	CMMI	Not used	Not defined	Not used	Not applied
S.4	2017	Scrum	Not defined	Not used	Not defined	Not used	Not applied
S.5	2017	Scrum	CMMI	Not used	Not defined	Not used	Not applied
S.6	2005	SDP	CMMI	Not used	Not defined	Not used	Not applied
S.7	2009	SDP	CMMI	Not used	Not defined	Not used	Not applied
S.8	2011	SDP	CMMI	Not used	Not Validated	Not used	Applied
S.9	2013	SDP	CMMI	Not used	Not defined	Not used	Not applied
S.10	2015	SDP	Not defined	Not used	Validated	Not used	Applied
S.11	2015	SDP	CMMI	Not used	Validated	Not used	Not applied

Table 3 Vectors used in the MMs from related work

Vector	*DevOps*		*Scrum*			*Agile*					
	S.1	*S.2*	*S.3*	*S.4*	*S.5*	*S.6*	*S.7*	*S.8*	*S.9*	*S.10*	*S.11*
Culture		X									
Collaboration	X									X	
Process		X									
Quality	X										
Automation	X										
Governance	X										
Technology		X									
People		X								X	X
General								X			
Sustained Success								X			
Organisation's Environment								X			
Interested parties, needs and expectations								X			
Embrace Change to Deliver Customer Value										X	

Table 3 Vectors used in the MMs from related work (continued)

Vector	*DevOps*		*Scrum*			*Agile*					
	S.1	*S.2*	*S.3*	*S.4*	*S.5*	*S.6*	*S.7*	*S.8*	*S.9*	*S.10*	*S.11*
Plan and Deliver Software Frequently										X	
Technical Excellence										X	
Practices											X
Deliveries											X
Requirements											X
Product											X
Customer											X

To Mohamed (2015), the keys to successful adoption of DevOps are quality, automation, collaboration, and governance/process, while claiming that, together, these fundamental elements can unify the traditional IT silos to enable agility across the end-to-end application life cycle. On the other hand, Bucena and Kirikova (2017) DevOps MM was developed on the basis of analysis of related work and includes five levels of maturity with respect to the four enterprise areas, namely, technology, process, people, and culture. No surprises with the absence of DevOps as possible vectors to assess DevOps maturity.

With the lack of consensus among the studies as well as the absence of both the use of rigorous methods/methodologies in the design process and DevOps capabilities as vectors of maturity assessment, the design of a new MM for DevOps can be faced as an opportunity and a step forward on the perspective of associated mature practices.

4 Research methodology

4.1 Design science research

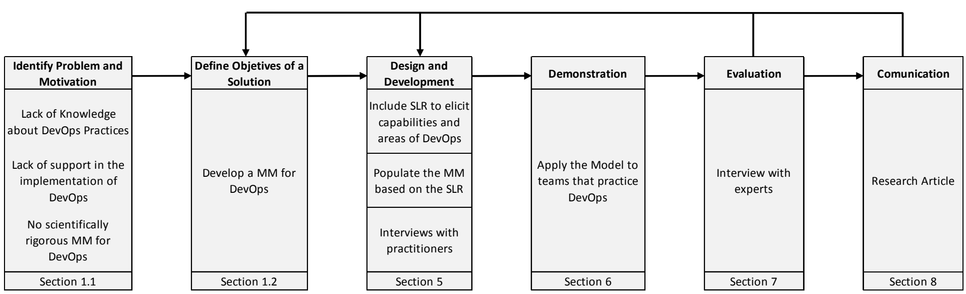
For the development of the proposed DevOps MM, it was applied the design science research methodology (DSRM) presented by Peffers et al. (2006) and the seven guidelines for DSR proposed by Hevner et al. (2004). DSR approach was selected since this research aims at solving practical problems by creating and evaluating IT artefacts intended to solve identified organisational problems (Hevner et al., 2004). DSR was recently indicated as the main methodology to develop MM (Pereira and Serrano, 2020), thus reinforcing its choice in this study.

IT artefacts are broadly defined as constructs (i.e., vocabulary and symbols), models (i.e., abstractions and representations), methods (i.e., algorithms and practices), and instantiations (i.e., implemented and prototype systems) (Hevner et al., 2004). According to Becker et al. (2009) and Mettler (2009), it can be assumed that the development of MMs falls within the application area for the guidelines by Hevner et al. (2004).

According to Peffers et al. (2006), the DSRM consists of six activities (i.e., steps). Figure 1 presents our applied techniques and performed activities in each DSRM step. In order to achieve rigorous as well as relevant research results, we draw upon the following DSRM steps, whereby the paper is structured accordingly:

- *Problem identification and motivation*: In the first section, it was specified the problem, provided practical relevance and justified the value of a solution. Additionally, based on problem scope, research questions were derived guiding this research.
- *Define the objectives for a solution*: The second section provides objectives of the intended collaboration MM. Based on a literature review, design recommendations in MM design and assessment will be identified and suggestions for circumvention will be proposed.
- *Design and development*: This activity is present in Section 5 and describes the MM development. Based on a literature review the MM will be designed and iteratively developed according to the requirements of MM construction (Becker et al., 2009).
- *Demonstration*: By means of an application test with three participant organisations the applicability and usability of the artefact was demonstrated. The utility of the MM will be further validated DevOps experts.
- *Evaluation*: According to Hevner et al. (2004), the artefact will be evaluated in terms of quality, utility and efficacy which cannot be demonstrated fully in this research.
- *Communication*: Communicate the problem, the importance, the utility, the rigor and the effectiveness of its design.

Figure 1 Applied DSR guidelines



4.2 Systematic literature review

One of the major tools used in other domains to support an evidence-based paradigm is the generation of Systematic Literature Reviews (SLR), which is used to aggregate the experiences gained from a range of different studies in order to answer a specific research question (Khan et al., 2004).

A SLR is a literature review method that aims to address a problem by identifying, evaluating, integrating all relevant findings, and interpreting research on research topics to answer research questions based on the stages used in SLR (Siddaway, 2014). The process of addressing the problem of lack of knowledge aims to identify the relationships and gaps in the existing literature. The identification process is used to describe directions for future research, because it consists of the process of formulating a general statement or an overarching conceptualisation, commenting on, evaluating, extending, or developing theory from existing literature (Siddaway, 2014).

This research follows Kitchenham procedures for SLR (Kitchenham, 2004), complemented by the concept centric approach from Webster and Watson (2002).

4.3 Semi-structured individual interviews and email interviews

The interview study reported here was carried out with DevOps practitioners Professionals from all over the world. The study took place as a qualitative interview study in the tradition of the qualitative research interview.

Semi-structured interviews are characterised by the use of a script consisting of closed or open predefined questions (Rijo, 2008). They are suitable when the research wants to validate several hypotheses but also to know the fieldwork and to explore new ones (Pozzebon, 2006). Particularly, they enable the interviewee to discuss the subject matter without being too attached to the formulated inquiry (Manzini, 2004). They also facilitate the interviewer to have clear support following the questions (Manzini, 2004). Moreover, they ensure to authors that their hypotheses or assumptions will be broadly covered by the conversation (Minayo, 2004).

Qualitative research has become essential to the humanities over the past twenty years (Ratislavová and Ratislav, 2014). Synchronous and asynchronous interviews and virtual focus groups are the most common methods (Ratislavová and Ratislav, 2014). The use of Email Interview can be employed quickly, conveniently, and inexpensively and can generate high-quality data when handled carefully. While a mixed mode interviewing strategy should always be considered when possible, semi-structured email interviewing can be a viable alternative to the face-to-face and telephone interviews, especially when time, financial constraints, or geographical boundaries are barriers to an investigation (Meho, 2006).

5 Design and development

To design the artefact, the author followed the steps listed below:

Step 1: Identify which are the main DevOps capabilities

Method(ology): SLR

Step 2: Identify which are the areas that most relate with DevOps.

Method(ology):SLR

Step 3: Identify the main practices of each DevOps capability

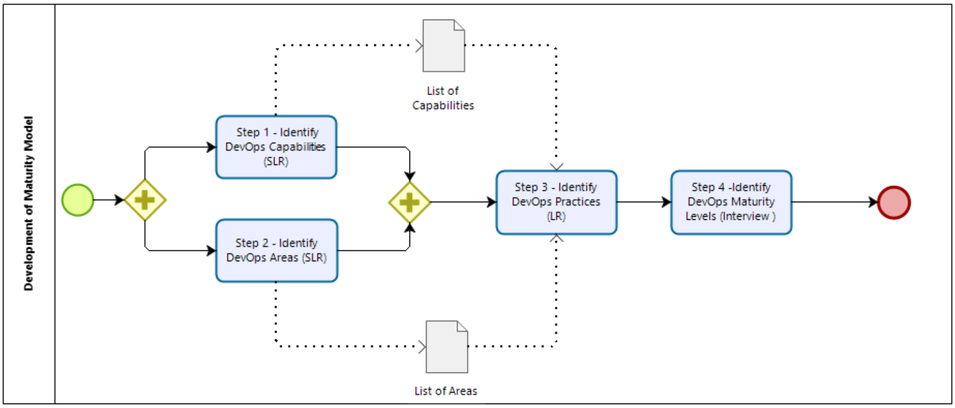
Method(ology): Literature Review

Step 4: Identify the maturity level of each DevOps practice

Method(ology): Interview

For a better understanding of the Design and Development's phase, the authors built the workflow (Figure 2) of the four previously described steps.

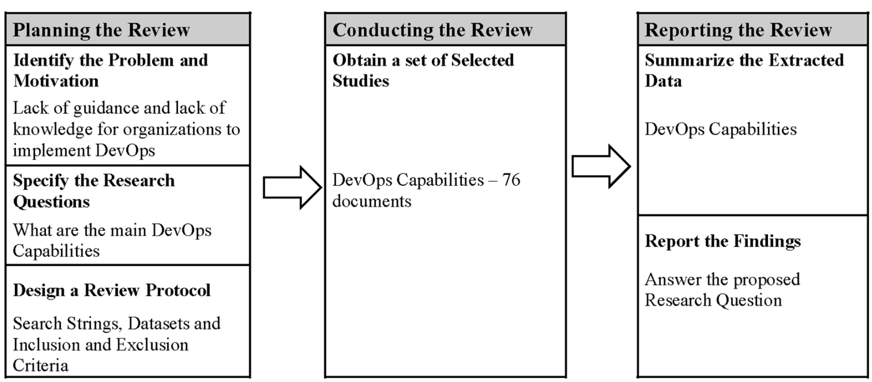
Figure 2 Workflow of the design and development's phase (see online version for colours)



5.1 *Step 1 (Capabilities)*

Figure 3 details the SLR phases adopted in Step 1. The SLR was chosen as a starting point to develop our Research Methodology since we wanted to summarise the existing evidence regarding DevOps' capabilities, with the aim of answering the proposed Research Objectives.

Figure 3 SLR methodology for DevOps' capabilities



The search string which was used and respective datasets are listed below.

- *Search String*: DevOps AND (Capability OR Capabilities OR Practice)
- *Datasets*: Google Scholar, ScienceDirect, IEEEXplore, ACM.

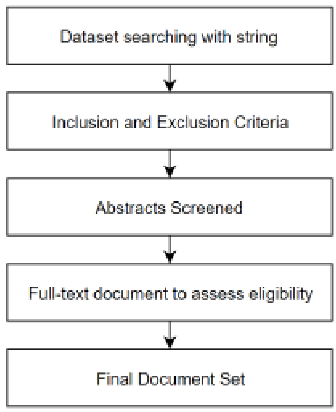
After that, inclusion and exclusion criteria must be applied to filter the obtained documents. Our criteria are presented in Table 4.

Afterwards, the first set of documents is obtained. Then, in a first phase, the abstracts were screened to decide their relevance to the research. Finally, these documents were red in order to obtain the final selection of studies to perform the review. The review protocol is illustrated in Figure 4.

Table 4 Inclusion and exclusion criteria for DevOps' capabilities

Inclusion criteria	*Exclusion criteria*
Written in English or Portuguese	Not written in English or Portuguese
Scientific papers in conferences or journals and books	Non-Free documents nor Master Thesis
Title relevance regarding DevOps	No title relevance DevOps

Figure 4 Review protocol for DevOps' capabilities



For a better understanding, as well as to add more scientific rigor to the research, the authors decided to follow the centric approach proposed by Webster and Watson (2002).

After applying the review protocol, 76 relevant studies were obtained for our research. Table 5 lists all the DevOps capabilities that were found, with its respective scientific references that support each capability.

Table 5 DevOps capabilities SLR

ID	*Capabilities*	*Reference*	*No. of references*
C1	Continuous Integration	Yin et al. (2004), Debois (2011), Hüttermann (2012), Fitzgerald and Stol (2014), Chen et al. (2015), De Bayser et al. (2015), Jabbari et al. (2016), Mohan and Ben Othmane (2016), Moore et al. (2016), Shahin et al. (2016), Soni (2016), Stoneham et al. (2016), Croker and Hering (2016), de França et al. (2016), Bucena and Kirikova (2017), Laukkarinen et al. (2017, 2018), Molto et al. (2017), Palihawadana et al. (2017), Pang and Hindle (2017), Punjabi and Bajaj (2017), Rubasinghe et al. (2017, 2018), Sharma (2017a), Shivakumar (2017), Snyder and Curtis (2017), Vassallo et al. (2017), Bai et al. (2018), Düllmann et al. (2018), Kuusinen et al. (2018), Lewerentz et al. (2018), Mackey (2018), Mansfield-Devine (2018), Marijan et al. (2018), Rahman et al. (2018), Rodríguez et al. (2018), Cleveland et al. (2018), Steffens et al. (2018), Tuma et al. (2018), Colomo-Palacios et al. (2018), Wiesche (2018), Wongkampoo and Kiattisin (2018), Xia et al. (2018), Zhu and Bayley (2018) and Debroy et al. (2018)	44

Table 5 DevOps capabilities SLR (continued)

ID	*Capabilities*	*Reference*	*No. of references*
C2	Continuous Deployment	Yin et al. (2004), Debois (2011), Hüttermann (2012), Fitzgerald and Stol (2014), Chen et al. (2015), Farshchi et al. (2015), Fördős and Cesarini (2016), Jabbari et al. (2016), Karapantelakis et al. (2016), Mohan and Ben Othmane (2016), Rana and Staron (2016), Shahin et al. (2016), Soni (2016), Stoneham et al. (2016), Ur Rahman and Williams (2016b), Ali et al. (2017), Bass (2017), Palihawadana et al. (2017), Pang and Hindle (2017), Perera et al. (2017a), Punjabi and Bajaj (2017), Rubasinghe et al. (2017), Sharma (2017a), Bucena and Kirikova (2017), Shivakumar (2017), Kuusinen et al. (2018), Laukkarinen et al. (2018), Mackey (2018), Mansfield-Devine (2018), Bhattacharjee et al. (2018), Rahman et al. (2018), Rubasinghe et al. (2018), Steffens et al. (2018), Tuma et al. (2018), Wiesche (2018), Xia et al. (2018), Zhu and Bayley (2018), Cleveland et al. (2018), Debroy et al. (2018) and Düllmann et al. (2018)	38
C3	Continuous Monitoring	Yin et al. (2004), Hüttermann (2012), Roche (2013), Fitzgerald and Stol (2014), Chen et al. (2015), John et al. (2015), Rana and Staron (2016), Soni (2016), Ur Rahman and Williams (2016b), de França et al. (2016), Hanappi et al. (2016), Karapantelakis et al. (2016), Bucena and Kirikova (2017), Li et al. (2017), Pang and Hindle (2017), Perera et al. (2017a), Rubasinghe et al. (2017), Rufino et al. (2017), Sharma (2017a), Shivakumar (2017), Snyder and Curtis (2017), Vassallo et al. (2017), Bai et al. (2018), Kuusinen et al. (2018), Steffens et al. (2018) and Düllmann et al. (2018)	25
C4	Continuous Testing	Yin et al. (2004), Hüttermann (2012), Roche (2013), Fitzgerald and Stol (2014), Chen et al. (2015), Shahin et al. (2016), Croker and Hering (2016), Soni (2016), Stoneham et al. (2016), Jabbari et al. (2016), Bucena and Kirikova (2017), Palihawadana et al. (2017), Pang and Hindle (2017), Punjabi and Bajaj (2017), Sharma (2017a), Shivakumar (2017), Snyder and Curtis (2017), St et al. (2017), Vassallo et al. (2017), Feijter et al. (2017), Murugesan (2017), Nielsen et al. (2017), Rubasinghe et al. (2018), Samarawickrama and Perera (2018), Silva et al. (2018), Wiesche (2018) and Kuusinen et al. (2018)	26
C5	Feedback Loops between Dev and Ops	Yin et al. (2004), Hüttermann (2012), Roche (2013), John et al. (2015), Stoneham et al. (2016), Hanappi et al. (2016), Jabbari et al. (2016), Bucena and Kirikova (2017), Feijter et al. (2017), Nielsen et al. (2017), Pang and Hindle (2017), Sharma (2017a), St et al. (2017), Murugesan (2017), Silva et al. (2018), Wongkampoo and Kiattisin (2018), Debroy et al. (2018), Kuusinen et al. (2018) and Mikkonen et al. (2018)	18
C6	Infrastructure as code	Yin et al. (2004), Hüttermann (2012), De Bayser et al. (2015), Rana and Staron (2016), Shahin et al. (2016), de França et al. (2016), Fördős and Cesarini (2016), Jabbari et al. (2016), Bucena and Kirikova (2017), Sharma (2017a), Jimenez et al. (2017), Bhattacharjee et al. (2018), Rahman et al. (2018), Steffens et al. (2018), Debroy et al. (2018) and Düllmann et al. (2018)	14

Table 5 DevOps capabilities SLR (continued)

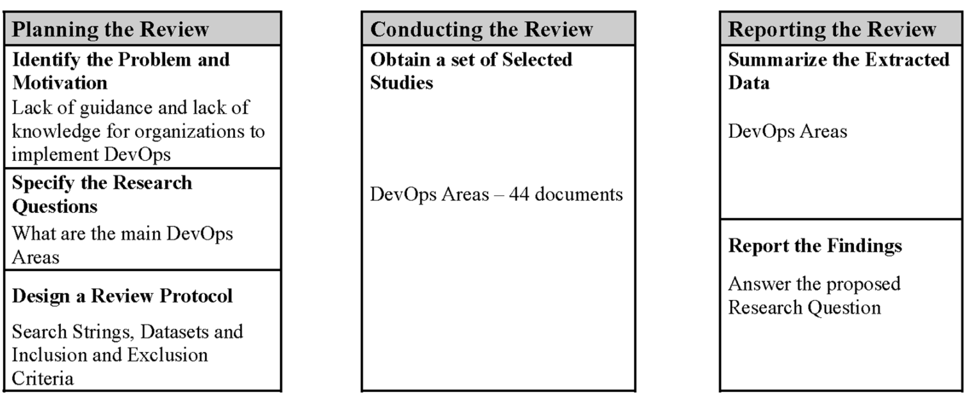
ID	*Capabilities*	*Reference*	*No. of references*
C7	Change Management	Debois (2011), Hüttermann (2012), Mohamed (2015), Science and Sciences (2015), Jabbari et al. (2016), Abdelkebir et al. (2017), Rubasinghe et al. (2017), Sharma (2017b) and Zhu and Bayley (2018)	9
C8	Continuous planning	Hüttermann (2012), Fitzgerald and Stol (2014), Jabbari et al. (2016), Ur Rahman and Williams (2016a), Pang and Hindle (2017), Sharma (2017b) and Kuusinen et al. (2018)	7
C9	Prototyping application	Hüttermann (2012), Fitzgerald and Stol (2014), De Bayser et al. (2015), Jabbari et al. (2016), Sharma (2017b) and Cleveland et al. (2018)	6
C10	Process Standardisation	Hüttermann (2012), Roche (2013), Jabbari et al. (2016), Rana and Staron (2016) and Sharma (2017b)	5
C11	Stakeholder Participation	Hüttermann (2012), Jabbari et al. (2016) and Sharma (2017b)	3
C12	Shift Left	Hüttermann (2012), Feijter et al. (2017) and Sharma (2017b)	3

5.2 Step 2 (Areas)

The three SLR phases, described in Section 4.1 are represented in Figure 5, and were specifically adapted to this section purpose.

We have chosen SLR as Research Methodology since it was intended to summarise the existing evidence regarding DevOps' areas, with the aim of answering the proposed Research Question.

Figure 5 SLR methodology for DevOps areas



The search string which was used and respective datasets are listed below.

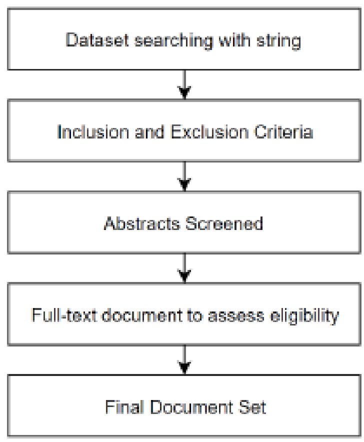
- *Search String*: DevOps AND (Area, Principles, View, Dimensions and Perspective)
- *Datasets*: Google Scholar, ScienceDirect, IEEEXplore, ACM.

After that, inclusion and exclusion criteria must be applied to filter the obtained documents. Our criteria are presented in Table 6.

Table 6 Inclusion and exclusion criteria for DevOps areas

Inclusion criteria	*Exclusion criteria*
Written in English or Portuguese	Not written in English or Portuguese
Scientific papers in conferences or journals and books	Non-Free documents nor Master Thesis
Title relevance regarding DevOps	No title relevance DevOps

Figure 6 Review protocol for DevOps areas



Afterwards, the first set of documents is obtained. Then, in a first phase, the abstracts must be screened to decide their relevance to the research. Finally, these documents are read in order to obtain the final selection of studies to perform the review. The review protocol is illustrated in Figure 6.

For a better understanding, as well as to add more scientific rigor to our research, the authors decided to follow the centric approach proposed by Webster and Watson (2002).

After applying the review protocol, 44 relevant studies were obtained for our research. Table 7 lists all the DevOps capabilities that were found, with its respective scientific references that support each capability.

5.3 Step 3 (DevOps practices)

Having analysed Table 5, and considering that there is a considerable gap between C6 and C7, the authors have decided to identify all the practices for each capability from C1 and C6. Since that the information regarding these capabilities are spread in a lot of studies, each capability's practices will be synthesised by grouping it by Area.

After analysing the descriptions of the areas from C1 to C12, the authors have concluded that some areas identify themselves with other areas. Considering that it would be complex to detail all the practices of all these areas, and since there are areas that cover other areas, the authors have decided to group some Areas. Thus, Technology will include Automation, Culture includes Sharing and Process includes Measurement.

Table 7 DevOps areas SLR

ID	*Area*	*References*	*No. of references*
A1	Culture	Debois (2011), Hüttermann (2012), Bang et al. (2013), Erich et al. (2014a), Sharma and Coyne (2015), Smeds et al. (2015b), Rana and Staron (2016), de França et al. (2016), Diel et al. (2016), Jabbari et al. (2016), Bucena and Kirikova (2017), Nielsen et al. (2017), Perera et al. (2017b), Gupta et al. (2017), Silva et al. (2018) and Colomo-Palacios et al. (2018)	16
A2	Measurement	Debois (2011), Hüttermann (2012), Bang et al. (2013), Erich et al. (2014a), Smeds et al. (2015b), Rana and Staron (2016), de França et al. (2016), Jabbari et al. (2016), Perera et al. (2017b), Gupta et al. (2017), Nielsen et al. (2017), Colomo-Palacios et al. (2018), Silva et al. (2018) and Luz et al. (2018)	14
A3	Sharing	Debois (2011), Hüttermann (2012), Bang et al. (2013), Erich et al. (2014a), Smeds et al. (2015b), Rana and Staron (2016), de França et al. (2016), Jabbari et al. (2016), Perera et al. (2017b), Gupta et al. (2017), Nielsen et al. (2017), Colomo-Palacios et al. (2018), Silva et al. (2018) and Luz et al. (2018)	14
A4	Automation	Debois (2011), Hüttermann (2012), Bang et al. (2013), Erich et al. (2014a), Smeds et al. (2015b), Mohamed (2015), Rana and Staron (2016), de França et al. (2016), Jabbari et al. (2016), Nielsen et al. (2017), Perera et al. (2017b), Gupta et al. (2017), Colomo-Palacios et al. (2018), Silva et al. (2018) and Luz et al. (2018)	14
A5	Technology	Hüttermann (2012), McCarthy et al. (2015), Sharma and Coyne (2015), Diel et al. (2016), Abdelkebir et al. (2017), Bucena and Kirikova (2017), Hussain et al. (2017), Sturm et al. (2017), Gazivoda (2018) and Silva et al. (2018)	10
A6	People	Hüttermann (2012), McCarthy et al. (2015), Sharma and Coyne (2015), Abdelkebir et al. (2017), Bucena and Kirikova (2017), Hussain et al. (2017), Sturm et al. (2017), Gazivoda (2018) and Silva et al. (2018)	9
A7	Process	Hüttermann (2012), McCarthy et al. (2015), Sharma and Coyne (2015), Abdelkebir et al. (2017), Bucena and Kirikova (2017), Hussain et al. (2017), Sturm et al. (2017), Gazivoda (2018) and Silva et al. (2018)	9
A8	Quality	Erich et al. (2014a), Mohamed (2015) and Luz et al. (2018)	3
A9	Collaboration	Mohamed (2015) and Luz et al. (2018)	2
A10	Diy Deployments	Debois (2011)	1
A11	Agility	Luz et al. (2018)	1
A12	Resilience	Luz et al. (2018)	1
A13	Transparency	Luz et al. (2018)	1
A14	Services	Erich et al. (2014a)	1
A15	Structures	Erich et al. (2014a)	1
A16	Standards	Erich et al. (2014a)	1
A17	Governance	Mohamed (2015)	1

This leave us with the four main areas: Culture, Technology, People and Process. In order to study the practices from the Capabilities in a determined areas, all the documents that were used in the SLR of the Capabilities and the Areas were analysed.

Table 8 presents all the Continuous Deployment practices found for DevOps capability, ordered by area. The rest of the capabilities can be seen in Appendix (Tables 14–18).

5.4 Step 4 (maturity levels)

The results of each conducted interview iteration are presented, followed by the associated emerging final MM for DevOps.

5.4.1 First iteration

To perform the first round of interviews, 15 DevOps professionals were interviewed.

The LinkedIn database was used to find the interviewees. Overall, 87 invites were made to DevOps experts and 33 were accepted. In this list of 33 contacts, only 15 responded to the interview.

In this research, it was considered the position of the possible participant, always willing to interview professionals with higher positions than DevOps developers. Interviewees information can be seen in Table 9.

Although some of the DevOps capabilities already exists, the term DevOps was born in 2011. The average age of the 15 interviewed is 39.4 years, while the average experience in DevOps is 5.6 years. Since DevOps was born nine years ago, 5.6 years in average of experience means that the interviewed have been working in this area during more than half of its existence as a practice. Plus, 13 out of the 15 interviewees work in the IT sector.

The same interviewer conducted all the 15 interviews ensuring that the same interview guides and protocol were used throughout the interviews. The first, second, third, fourth and last interviews were conducted in the participants' workplace, while the rest were carried out by Skype. The interview was semi-structured and aimed at exploring practitioners' experiences with DevOps practices. All the 15 interviews were conducted between March and June 2019.

The authors have interviewed DevOps practitioners according to a preset script which included semi-structured open-ended questions. The interview guideline addressed topics such as the expert's background, expert's team and company information, DevOps practices and observations about it.

Grounded on maturity levels classification, and since all organisation are at level 1 (ad-hoc) by default, the authors have only asked the interviewees to associate the practices with levels 2, 3, 4 and 5. The distribution of the practices by levels is presented in Table 10.

5.4.2 Second iteration

All the 15 interviewees from the first iteration were asked to participate in a second round. From those, 13 accepted to participate. The objective of this phase was to breakdown the practices that had the same number of votes to more than one level of maturity and try to reach consensus on all practices. Therefore, the participant had a chance to choose between the most voted levels of the first phase in each of the enlisted practices.

Table 8 CD practices

	Continuous Deployment	
	Practice	*Author*
People	–	–
Process	Orchestrated deployments	Sharma and Coyne (2015)
	Track which version is deployed	
	Manage the configurations of the environments of all the stages	
	Manage the software components that get deployed	
	Manage the middleware components and middleware configurations that need to be updated	
	Manage the database components that need to be changed	
	Manage the configuration changes to the environments to which these components are to be deployed	
	Release working software any time, any place	Duvall et al. (2007)
	Label a repository's assets	
	Produce a clean environment	
	Label each build	
	Create build feedback reports	
	Possess capability to roll back release	
	Multiple deployments to production	Mohamed (2016)
	Deploy a new release whenever one is needed	
Technology	Development and production share a homogenous infrastructure	Ebert et al. (2016)
	Configuration management tools	
	Automated deployment of software to different environments	Nielsen et al. (2017)
	Deployments should include the automated provisioning of all environments	Debois (2011)
	Automated deployment	Nielsen et al. (2017)
	Continuous deployment	
Culture	Early and frequent involvement of operations staff in the planning stages of major new releases	Debois (2011)

All the interviews were conducted by email. The interviews were semi-structured and aimed at exploring practitioners' experiences with DevOps practices. All the 13 interviews were conducted between June and August 2019.

DevOps practitioners were interviewed according to a preset script which included semi-structured open-ended questions. The interview guideline addressed topics such as DevOps practices and observations about it. Since no relevant conclusions could be drawn from the first iteration, in this second phase the authors changed the possible answers for the DevOps practices maturity levels to the most voted levels from the first phase. This was held since there were many maturity levels for each practice.

Table 9 Interviewees details

ID	*Role*	*Age*	*DevOps experience (Years)*	*Industry*	*First iteration*	*Second iteration*
I1	Head of DevOps Transformation	41	6	Software development	X	X
I2	Solution Architect	46	8	Software development	X	X
I3	Senior Manager	41	8	Software development	X	X
I4	Senior DevOps Engineer / Team Lead	26	3	Software development	X	X
I5	Head of Agile and DevOps Transformation	38	3	Software development	X	X
I6	DevOps Manager/Evangelist	42	3	Finance	X	X
I7	Lead DevOps specialist	39	3	Healthcare	X	X
I8	DevOps Architect	38	8	Software development	X	X
I9	DevOps Operations Lead	40	3	Software development	X	X
I10	DevOps Engineer	33	4	Software development	X	X
I11	Managing Director	48	8	Software development	X	X
I12	Senior Developer	38	6	Software development	X	X
I13	Lead DevOps specialist	45	8	Software development	X	
I14	Senior Manager	39	7	Software development	X	
I15	IT Development T. Leader – Applications	37	6	Software development	X	X
Average		*39.4*	*5.6*			

Table 10 Distribution of the number of practices per level from first iteration

Level	*Frequency*
Level 2	31
Level 3	50
Level 4	19
Level 5	9

Grounded on maturity levels classification, and since all organisation are at level 1 (ad-hoc) by default, the authors only asked the interviewees to associate the practices with the most voted levels for each practice from the first phase. The distribution of the practices by levels and the difference from the first iteration are presented in Table 11.

Table 11 Distribution of the number of practices per level from second iteration

Level	*Frequency*	*Difference*
Level 2	10	–21
Level 3	54	+4
Level 4	27	+8
Level 5	18	+9

Analysing Table 11, one of the most relevant difference between the two phases is the migration of some level two responses to the other levels. There is a clear increase of level 5 votes. On the other hand, level 3 continues to be the most voted level.

Only about one third of the previous level two votes remained. Although none of the participants said anything about this, it seems that, since each participant had the chance to choose from the most voted level from the first iteration, they considered a higher level since that it was a possibility. Also, since that two from the first iteration interview did not answer this issue, it may have had an influence on this result.

The most voted levels are concentrated in two levels: three and four. The participants only considered 18 practices to belong to a much higher maturity level (level 5). Since level three is one of the most basic level, it had a much higher number of practices.

5.4.3 Maturity model

Although it is a single model, for its better comprehension, it was divided into six parts, one for each capability. Even though the interviewees had the chance to add or remove practices from the initial list, none of them did. This means that the initial list of DevOps practices remained unchanged through all these interview phases. Although every participant had the chance to remove a practice and/or add an observation, there were only few cases where it happened. However, since it was not coherent nor consistent among the participants, those removed practices and observations were not taken in consideration.

Each MM table is divided by areas (People, Process, Technology and Culture) in which are presented the respective practices. Table 12 present the MM for Continuous Deployment. The rest of the MM can be seen in Appendix (Tables 19–24). According these tables that, together, integrate the MM for DevOps, an analysis has been made.

Observing Table 19, it is possible to devise that there is only one practice from level 2. Level 3 is the level with more practices and level 4 and level 5 almost have the same number of practices. If we look to the practices per area, since the author was not able to find any practice associated with this area and the interviewees did not add any, People does not have any practice. On the other hand, Process seems to be the area with more practices, since it has at least one at each level.

In both Tables 19 and 20, people area does not have any associated practice. On the other hand, level 2 is more populated than it was in Table 19. Level 3 is the level with more practices, while Process continues to be the area of DevOps with more practices. Technology has at least a practice per level.

In the Continuous Monitoring (Table 21) it is possible to see the first practice for the People's area and is the only practice for the level 2 on this table. Process and Technology have practices from the level 3 to level 5.

Table 22 is the people's area contains more practices than the tables before. There are three People practices and they are all in level 3. Culture is the most completed area in this table, since it has practices in every level. Level 5 only has one practice.

Table 23 is the one with less practices. The author could not identify more practices from the literature and the interviewees did not add any. Level 3 is the most populated level and there is only on practice that does not belong to this level. Technology is the Area with most practices. On the other hand, there is no practice in People's area.

Last but not least, Table 24 presents all the practices from Feedback Loops capability. There was not found any practice in level 2. Level 3 only have practices for the Process area, while level 4 contains practices for People, Process and Culture. Culture seems to be an area where all its practices are from a greater maturity, since three out of four practices presented in this area belong to level 5. The level with more practices is level 4.

After analysing all the tables that contained the MM for DevOps, a last analysis must be conducted. The preliminary list for the MM was conducted by the author, through a literature review. Although the fact that all the interviewees had the chance to add or remove any practices they want, none of them did. This result in some capabilities with less practices than others, and some areas with just few practices. If any of them had less than four practices, it means that there will be levels with no practices.

People is the area with less practices from the four. On the other hand, Process, followed by Technology are the areas with more practices. Level 3 is the level with most practices while level 2 is the one with less practices. This may be due to the lack of literature about this theme.

6 Demonstration

In order to demonstrate the artefact, two teams fully compliant with DevOps were assessed. Then, an interview was held with DevOps teams where the proposed MM was tested. The objective is to demonstrate that the MM fulfils the purpose it was designed to applying it in a professional environment. Since not all capabilities or areas have practices, only the capabilities/areas with at least one practice have been considered to assess team's maturity. According with CMMI, which has been previously presented, a level can only be reached if all the practices from that level are executed.

6.1 First demonstration

The first team assessed operates in the services sector, in the field of Cloud and DevOps consulting. The person responsible to conduct this demonstration is the DevOps Operations Lead with three years of experience in DevOps. Figure 7 shows the maturity of the DevOps in this team.

Figure 7 shows the maturity of the first team. As it evidences, the most matured capability is the Feedback Loops, followed by CI.

At level 4, Feedback Loops has a maturity level almost all areas at level 5, if it was not by the People's area. This means that the team has all the practices implemented for Culture and Processes, and a big part of the People's practices. Looking to the CI, Technology is at its maximum, level 5. Culture is the next area with more maturity and Process is at the end.

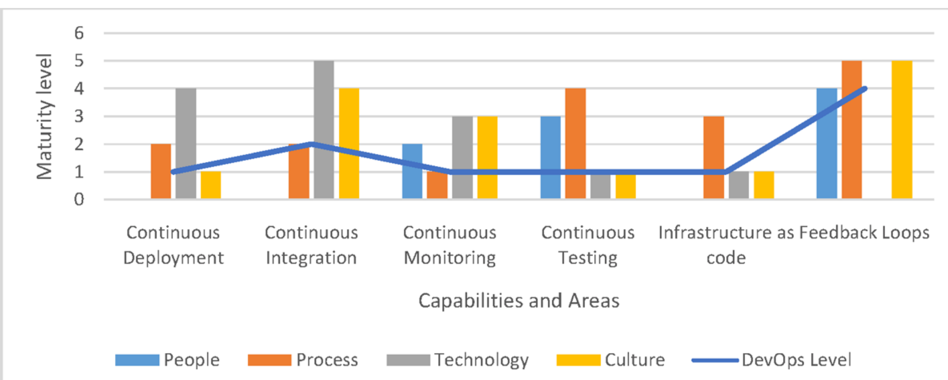
Table 12 CDMM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous deployment	People				
	Process	CD9 Label a repository's assets	CD2 Track which version is deployed CD3 Manage the configurations of the environments of all the stages CD4 Manage the software components that get deployed CD5 Manage the middleware components and middleware configurations that need to be updated CD6 Manage the database components that need to be changed CD10 Produce a clean environment CD11 Label each build CD12 Create build feedback Reports CD14 Deploy a new release whenever one is needed CD17 Automated deployment CD18 Continuous deployment	CD1 Orchestrated deployments CD16 Deployments should include the automated provisioning of all environments	CD1 Orchestrated deployments CD7 Manage the configuration changes to the environments to which these components are to be deployed CD8 Release working software any time, any place CD15 Multiple deployments to production
	Technology	–	CD19 Development and production share a homogenous infrastructure CD20 Configuration management tools	CD21 Automated deployment of software to different environments	–

Table 12 CDMM (continued)

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous deployment	Culture	–	CD22 Team must provide overall visibility into your application release activities and timing to all major stakeholders CD25 Unite the two teams that worked independently to work at tighter integration CD26 Both development and operations personnel should share the same knowledge management resources CD27 Testers and operations personnel would be able to self- service deployments of the required version of the system to their environments on demand CD28 Early and frequent involvement of operations staff in the planning stages of major new releases	CD24 Team must be able to speed lead times and make more frequent application deployments at the pace demanded by the business	CD23 Teams must be able to provide self-service, on-demand provisioning and management of cloud environments and infrastructure resources

Figure 7 First demonstration maturity (see online version for colours)



Looking to the other capabilities, they all are at level 2. Continuous Monitoring has 3 areas at level 3 and seems to be the next most maturated capability.

In a more general view, the most maturated capability is Feedback Loops. The most maturated area is Process.

6.2 *Second demonstration*

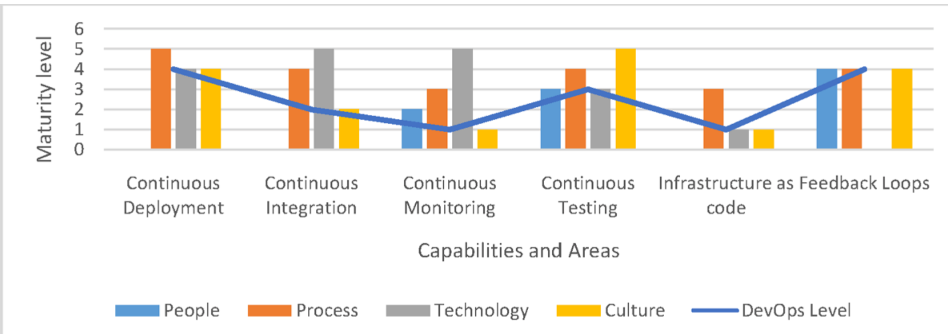
The second team is from the SD industry. The person responsible to conduct this demonstration is the Senior Manager with eight years of experience in DevOps. Figure 8 shows the maturity of the DevOps in this team. The Y axis represent the maturity level and the X axis represent the DevOps capabilities and areas.

Looking at this figure, it is perceptible that this team has, in general, a much higher maturity than the previous one. Two capabilities at level 4 and one in level 3. CD, Feedback Loops are the most matured capabilities while Infrastructure as a Code is the less matured one.

Looking to the CD graphic, one of the areas reached level 5, while the others are at level 4. Feedback loops has all its areas with similar maturity levels. Continuous Testing has one area in level 5, one in level 4 and the others in level 3.

CI, although it has 1 area in level 5 and another one in level 4, it is only in the maturity level 2, due to its lack of culture maturity. Continuous Monitoring has the same problem: although it has 1 area in level 5, one in level 4 and another in level 3, its maturity is only 2. The most immature capability is Infrastructure as a Code. On the three areas evaluated, only one is above level 2.

Figure 8 Second demonstration maturity (see online version for colours)



7 Evaluation and communication

Following the Pries-Heje et al. (2008) approach, in which the authors present the importance of an ex ante perspective, with the evaluation occurring both prior to the construction of an artefact IS, and an ex post evaluation, that is, evaluations that take place after the artefact has been built. Plus, Venable identifies two main forms for the DSRM evaluation (Venable, 2006):

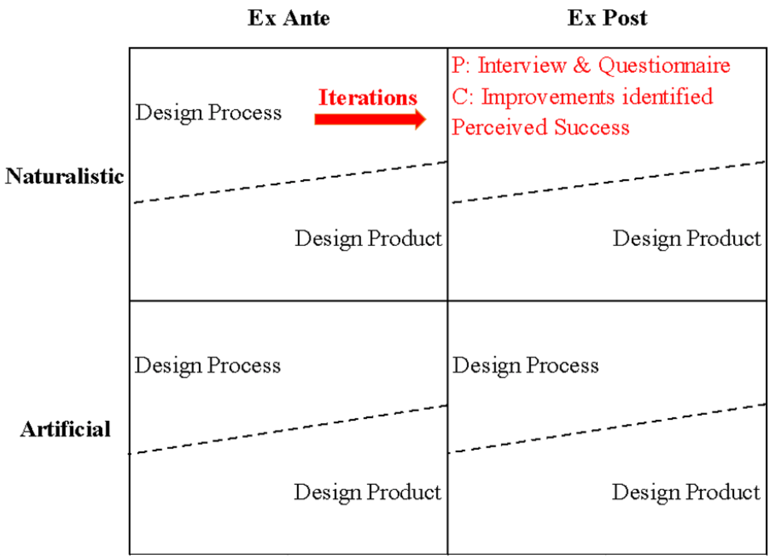
- *artificial evaluation* is evaluating a solution technology in a contrived, non-real way
- *naturalistic evaluation* enables the authors to explore how well or poorly a solution technology works in its real environment – the organisation.

Furthermore, an additional dichotomy is incorporated into the Pries-Hege's framework, which is comprised of the design product and design process. Using the definition of Dubin for each aspect of design theory (Dubin, 1976):

- *design product* is "a plan of something to be done or produced"
- *design process* is "to so plan and proportion the parts of a machine or structure that all requirements will be satisfied"

By distinguishing all these concepts, it is possible to map the objectives of evaluation and what is more accurately adapted to the artefact constructed in order to prove the utility, effectiveness and other criteria, as shown in Figure 9. This framework for the DSRM evaluation is supposed to facilitate the answer to the following questions – 'What' is evaluated, 'When' to evaluate, and 'How' to evaluate. Figure 9, helps us to answer these questions by providing a high-level perspective, also considering that "P summarises the essential characteristics of the evaluation Process, while C indicates the evaluation criteria" (Pries-Heje et al., 2008).

Figure 9 Strategic DSRM evaluation framework (see online version for colours)



Source: Adapted from Pries-Heje et al. (2008)

However, further details are needed to answer these questions and several decisions need to be made. This non-compliance is fulfilled with the proposed framework by Venable et al. (2012) that is intended to be a complement to the strategic DSRM evaluation framework mentioned above, providing for example a guide on how to select evaluation methods.

The DSRM evaluation method selection framework suggests possible evaluation methods. For the current study, Survey was selected, in a form of interviews and questionnaires.

Concerning research communication, a part of this research is presented by one paper and the whole research is represented by this document. The authors will now show the evaluation that was given by the demonstration inquires, where the constructed MM was applied by DevOps practitioners in its teams. The authors asked the participant to evaluate the proposed MM: the inquired person had the chance to say anything he wanted about this MM, if it was useful, complete or applicable in real life cases.

This first evaluation corresponds to the First demonstration case, where the participant of 40 years old and three years of experience on the DevOps field applied the MM in his team. The second evaluation is from the SD industry, where the participant is responsible to conduct this demonstration is the Senior Manager with eight years of experience in DevOps. The following was stated (Table 13):

The participants evaluated the MM positively as it can be seen in evidenced by their feedback. On the first case, the participant said that it is a valuable work and it can be a good help for the DevOps implementation. The participant also said that as a service provider, some practices can be hard to get through because they are a true challenge to implement.

The second participant in the evaluation stated that this MM is a useful tool to know the maturity of DevOps in a team. The fact that the MM was build based on the literature and improved with DevOps practitioners, gives this research more credibility. Although the participant considers this MM complete, for him, it could get better if all the Areas had at least one practice, so it can measure the maturity of all the DevOps.

Taking these two evaluations in consideration, the feedback received is positive. Both participants thought this is a useful tool to measure the DevOps adoption. By the feedback, it is possible to perceive that this MM is applicable in real cases. The suggestion of improving the model to have at least one practice in each area is shared by the authors. However, it was not possible to find in the literature studies that deeply explore DevOps and the people interviewed for the construction of this MM did not add any practice.

Table 13 Evaluations of the MM applicability

ID	*Evaluation*
E1	"You produced such valuable work. This list can act as a service menu for a DevOps process and culture implementation and at the same time this will help the person in charge of the DevOps transformation keep the focus on what should be delivered to the stakeholders.
	As a service provider, I cannot deny the difficulty to address some targets of your work with my clients. For example, when you are working to transform an ITIL organisation to an Agile/DevOps organisation, people tend to refrain the changes and points as the "Share the feedback freely without blame" are a true challenge to be implemented.
	For me, decide which parts of your practices should or not be implemented is a matter to balance the client needs, the size of the client organisation and keep the process as simple as possible."
E2	"It is hard to find DevOps practices in the existent literature. It is even harder to understand what is important and what is the correct order to implement, so the team has solids basis.
	This work provides an interesting set of DevOps practices, divided by the most important capabilities. It is even better because I can have a vision by area. Applying this MM to our team gave me insight into what should be implemented and in what order. Knowing that this was made with interviews to DevOps practitioners give me more confidence in using this model as basis to future team improvements decisions, as I can rely on this research.
	This is a useful tool if you want to know the maturity of your team in DevOps. Although I believe that it is a complete tool, I would consider it more complete if it has more practices. At least, if every capability and every area had at least one practice."

8 Conclusions

In this research two SLR were conducted to respond to the call by authors and practitioners for a deeper theoretical and practical understanding of DevOps capabilities and areas that could work as determinant factors and contribute to the implementation of

DevOps. Then, a total of 28 interviews were performed with DevOps practitioners. With their experience, the interviewees helped to assign a specific maturity level for each DevOps practice. At the end of the previous steps, the proposed MM for DevOps was then completed. Grounded on the previous sections one may argue that all the proposed Research Objectives were achieved:

- Regarding RO1.1, the main DevOps areas were elicited and described, and they specifically include culture, measurement, sharing, automation, technology, people and process.
- Concerning RO1.2, the main DevOps capabilities have been also identified and detailed. The elicited capabilities include CI, CD, continuous testing, feedback loops between Dev and Ops and infrastructure as code.
- After these sub-objectives are met, a MM for DevOps was built. It was sustained on the previous main areas and main capabilities. It was developed a new DevOps MM based on CMMI MM to enable assessing any organisation working model/state against DevOps model.

Regarding this, the main objectives that this research proposed were hit. Despite this, it was possible to conclude the following set of insights:

- Both DevOps practitioners and scientific studies continue to increase since 2015. This study also identified some relationships between the DevOps areas and capabilities based on the analysis of Figure 7. The documents that focus on the DevOps culture are most likely to relate it to all of the main capabilities found. On the other hand, it is more difficult to find a document that relates Technology, People and Process with the main capabilities.
- The capabilities of CI and CD are the more investigated in the literature. The areas that most relate with them are Culture, Sharing and Automation. These three areas are the most referred DevOps areas in the literature. Processes seems to be the area that less influences the capabilities, while Infrastructure as Code is the capability which the fewest studies tend to relate with DevOps.
- This research has brought contributions to the academic and scientific community by exploring a field that had not yet been explored and proposing a novel artefact. It has also improved the knowledge base and endeavoured to lay down new bases for further research.
- This research is a new systematised contribution to knowledge, through the identification of patterns that have been recognised in the literature –and that, as such, corresponds to a new level of knowledge in the approach to the topic. This research also provides some contributions for professionals and practitioners. In the absence of studies exploring the DevOps main capabilities and DevOps areas, and even the relationship between them, this research brings new insights on how and why practitioners should adopt DevOps practices and which areas they have to change or, at least, keep in mind as being relevant for an effective adoption of DevOps.

- Based on these findings, and using the summarised information provided in this work as a starting point, the authors deepened the identified DevOps areas and capabilities to be an a priori and open model, which was the target of this research project – which aimed to test and refine this systematised view (in the form of a MM), having not only implications for existing scientific knowledge but also being useful for organisational practices of DevOps.

8.1 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 fully avoid biases since it has excluded literature sources written in other languages or unavailable in electronic databases. Since DevOps is recent, there are not a lot of experts in this area. This limited the interviews on each phase (Prates et al., 2019).

8.2 Future work

In the future, research should be carried out into the most referenced capabilities, CI and CD and the most referenced areas, Culture, Sharing and Automation, as they seem to be essential in the DevOps movement. Also, it would be interesting to deeply explore the relationship between CI and Culture, Sharing and Automation, as these areas seem to relate the most with the main capability found among this literature review. It is also expected that the proposed MM needs to be evolved in the future. As any other research this process is normal. But in this case is even more evident since DevOps is recent and more organisations are implementing it and academics investigating it.

References

Abdelkebir, S., Maleh, Y. and Belaissaoui, M. (2017) 'An agile framework for ITS management in organizations', *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems – ICCWCS'17*, pp.1–8, doi: 10.1145/3167486.3167556.

Aguiar, J., Pereira, R., Vasconcelos, J. and Bianchi, I. (2018) 'An overlapless incident management maturity model for multi-framework assessment (ITIL, COBIT, CMMI-SVC)', *Interdisciplinary Journal of Information, Knowledge, and Management*, doi: 10.28945/4083.

Ali, E., Caputo, A. and Lawless, S. (2017) 'Entity attribute ranking using learning to rank', *CEUR Workshop Proceedings*, Vol. 1883, pp.19–24, doi: 10.1145/1235.

Andersen, E.S. and Jessen, S.A. (2003) 'Project maturity in organisations', *International Journal of Project Management*, Vol. 21, No. 6, pp.457–461, doi: 10.1016/S0263-7863(02)00088-1.

Azoff, M. (2016) 'Ovum decision matrix: selecting a DevOps release management how enterprises can improve their software application delivery', *Ovum*, pp.1–8.

Bai, X., Li, M., Pei, D., Li, S. and Ye, D. (2018) 'Continuous delivery of personalized assessment and feedback in agile software engineering projects', *Proceedings – International Conference on Software Engineering*, ACM, Part F1373, pp.58–67, doi: 10.1145/3183377.3183387.

Bang, S., Chung, S. Choh, Y. and Dupuis, M. (2013) 'A grounded theory analysis of modern web applications', *Proceedings of the 2nd Annual Conference on Research in Information Technology – RIIT '13*, p.61, doi: 10.1145/2512209.2512229.

Barbosa, D.F., Furtado, E.S. and Gomes, A.S. (2007) 'Uma proposta de institucionalização da usabilidade alinhada com práticas do modelo CMMI e foco nas necessidades da organização', *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems*, pp.45–48, doi: 10.1145/1298023.1298060.

Baskarada, S., Gao, J. and Koronios, A. (2005) 'Agile maturity model approach to assessing and enhancing the quality of asset information in engineering asset management information systems', *Proceedings of the 9th International Conference on Business Information Systems*, pp.66–68.

Bass, L. (2017) 'The software architect and DevOps', *IEEE Software*, Vol. 35, No. 1, pp.8–10, doi: 10.1109/MS.2017.4541051.

Becker, J., Knackstedt, R. and Pöppelbuß, J. (2009) 'Developing maturity models for IT management', *Business & Information Systems Engineering*, Vol. 1, No. 3, pp.213–222, doi: 10.1007/s12599-009-0044-5.

Berssaneti, F.T., De Carvalho, M.M. and Muscat, A.R.N. (2012) 'Impacto dos modelos de referência e maturidade no gerenciamento de projetos: estudo exploratório em projetos de tecnologia da informação', *Production*, Vol. 22, No. 3, pp.404–435, doi: 10.1590/S0103-65132012005000027.

Bezemer, C., Eismann, S. Ferme, V., Grohmann, J., Heinrich, R., Jamshidi, P., Shang, W., Hoorn, A., Villaviencio, M., Walter, J. and Willnecker, F. (2018) *How is Performance Addressed in DevOps? A Survey on Industrial Practices*, arXiv:1808.06915v1 [cs.SE], August, doi: 10.1145/3297663.3309672.

Bhattacharjee, A., Barve, Y., Gokhale, A. and Kuroda, T. (2018) '(WIP) CloudCAMP: automating the deployment and management of cloud services', *2018 IEEE International Conference on Services Computing (SCC)*, IEEE, pp.237–240, doi: 10.1109/SCC.2018.00038.

Bucena, I. and Kirikova, M. (2017) 'Simplifying the devops adoption process', *CEUR Workshop Proceedings*, pp.32–35.

Buglione, L. (2011) 'Light maturity models (LMM)', *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement – Profes '11*, Vol. 5, No. Lmm, p.57, doi: 10.1145/2181101.2181115.

Chen, B. (2019) 'Improving the software logging practices in DevOps', *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, IEEE, pp.194–197, doi: 10.1109/icse-companion.2019.00080.

Chen, H.M., Kazman, R., Haziyev, S., Kropov, V. and Chtchourov, D. (2015) 'Architectural support for DevOps in a neo-metropolis BDaaS platform', *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, January, pp.25–30, doi: 10.1109/SRDSW.2015.14.

Chen, L. (2018) 'Microservices: architecting for continuous delivery and DevOps', *Proceedings – 2018 IEEE 15th International Conference on Software Architecture, ICSA 2018*, pp.39–46, doi: 10.1109/ICSA.2018.00013.

Chrissis, M.B., Konrad, M.D. and Shrum, S. (2010) *CMMI for Development*, Version 1.3, Carnegie Mellon University, doi: CMU/SEI-2010-TR-033 ESC-TR-2010-033.

Clevand, S.B., Dooley, R., Perry, D., Stubbs, J., Fonner, J.M. and Jacobs, G.A. (2018) 'Building science gateway infrastructure in the middle of the pacific and beyond', *Proceedings of the Practice and Experience on Advanced Research Computing – PEARC '18*, Vol. Ci, pp.1–8, doi: 10.1145/3219104.3219151.

Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P. and Larrucea, X. (2018) 'A case analysis of enabling continuous software deployment through knowledge management', *International Journal of Information Management*, Elsevier, Vol. 40, October, pp.186–189, doi: 10.1016/j.ijinfomgt.2017.11.005.

Croker, M. and Hering, M. (2016) *DevOps: Delivering at the Speed of Today's Business DevOps: A Matter of Survival in the Digital Age.*

De Bayser, M., Azevedo, L.G. and Cerqueira, R. (2015) 'ResearchOps: the case for DevOps in scientific applications', *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, Vol. 59, No. 3, pp.1398–1404, doi: 10.1109/INM.2015.7140503.

de França, B.B.N., Jeronimo, H. and Travassos, G.H. (2016) 'Characterizing DevOps by hearing multiple voices', *Proceedings of the 30th Brazilian Symposium on Software Engineering – SBES '16*, pp.53–62, doi: 10.1145/2973839.2973845.

Debois, P. (2011) 'DevOps: a software revolution in the making', *Cutter IT Journel*, Vol. 24, No. 8, p.34, doi: 10.1016/B978-0-08-025947-5.50004-2.

Debroy, V., Miller, S. and Brimble, L. (2018) 'Building lean continuous integration and delivery pipelines by applying DevOps principles: a case study at Varidesk', *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering – ESEC/FSE 2018*, pp.851–856, doi: 10.1145/3236024.3275528.

Diel, E., Marczak, S. and Cruzes, D.S. (2016) 'Communication challenges and strategies in distributed DevOps', *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, IEEE, pp.24–28, doi: 10.1109/ICGSE.2016.28.

Dubin, R. (1976) 'Theory building in applied areas', *Handbook of Industrial and Organizational Psychology*, pp.13–16.

Düllmann, T.F., Paule, C. and Van Hoorn, A. (2018) 'Exploiting devops practices for dependable and secure continuous delivery pipelines', *Proceedings – International Conference on Software Engineering*, Part F1378, pp.27–30, doi: 10.1145/3194760.3194763.

Duvall, P.M., Matyas, S. and Glover, A. (2007) *Continuous Integration: Improving Software Quality and Reducing Risk*, Addison-Wesley signature series, Boston, USA.

Ebert, C., Gallardo, G., Hernantes, J. and Serrano, N. (2016) 'DevOps', *IEEE Software*, Vol. 33, No. 3, pp.94–100, doi: 10.1109/MS.2016.68.

Erich, F., Amrit, C. and Daneva, M. (2014a) 'Cooperation between information system development and operations', *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement – ESEM '14*, pp.1–1, doi: 10.1145/2652524.2652598.

Erich, F., Amrit, C. and Daneva, M. (2014b) *DevOps Literature Review*, University of Twente, October, p.27, doi: 10.1007/978-3-319-13835-0.

Farkas, A. and Walsh, C. (2002) 'A perspective of the common criteria in modern IT business', *3rd International Common Criteria Conference*, May, pp.1–8, Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.1073&rep=rep1&type=pdf

Farshchi, M., Schneider, J., Weber, I. and Grundy, J. (2015) 'Experience report: anomaly detection of cloud application operations using log and cloud metric correlation analysis', *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, pp.24–34, doi: 10.1109/ISSRE.2015.7381796.

Feijter, R., Vliet, R., Jagroep, E., Overbeek, S. and Brinkkemper, S. (2017) *Towards the Adoption of DevOps in Software Product Organizations: A Maturity Model Approach*, Technical Report UU-CS-2017-009, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands, May, pp.1–173.

Feitelson, D.G., Frachtenberg, E. and Beck, K.L. (2013) 'Development and deployment at facebook', *IEEE Internet Computing*, Vol. 17, No. 4, pp.8–17, doi: 10.1109/MIC.2013.25.

Fitzgerald, B. and Stol, K-J. (2014) 'Continuous software engineering and beyond: trends and challenges', *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering – RCoSE 2014*, pp.1–9, doi: 10.1145/2593812.2593813.

Fontana, R.M., Junior, V.M., Reinehr, S. and Malucelli, A. (2015) 'Progressive outcomes: a framework for maturing in agile software development', *Journal of Systems and Software*. Elsevier Ltd., Vol. 102, pp.88–108, doi: 10.1016/j.jss.2014.12.032.

Fördős, V. and Cesarini, F. (2016) 'CRDTs for the configuration of distributed Erlang systems', *Proceedings of the 15th International Workshop on Erlang – Erlang 2016*, pp.42–53, doi: 10.1145/2975969.2975974.

Gazivoda, M. (2018) *Application of DevOps Approach in Developing Business Intelligence System in Bank*, June, pp.11–14.

Gupta, V., Kapur, P.K. and Kumar, D. (2017) 'Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling', *Information and Software Technology*, Elsevier B.V., Vol. 92, pp.75–91, doi: 10.1016/j.infsof.2017.07.010.

Hanappi, O., Hummer, W. and Dustdar, S. (2016) 'Asserting reliable convergence for configuration management scripts', *ACM SIGPLAN Notices*, Vol. 51, No. 10, pp.328–343. doi: 10.1145/3022671.2984000.

Hart, C. (1998) 'Doing a literature review: releasing the social science research imagination', *SAGE: London*, Vol. 1, No. 1, pp.1–25, doi: 10.1080/01422419908228843.

Hevner, A.R., March, S.T., Park, J. and Ram, S. (2004) 'Design science in information systems research', *MIS Quarterly*, doi: 10.2307/25148625.

Humble, J. and Farley, D. (2011) *Continuous Delivery*, Addison-Wesley Signature Series, doi: 10.1017/CBO9781107415324.004.

Hussain, W., Clear, T. and MacDonell, S. (2017) 'Emerging trends for global DevOps: a New Zealand perspective', *Proceedings – 2017 IEEE 12th International Conference on Global Software Engineering, ICGSE 2017*, IEEE, pp.21–30, doi: 10.1109/ICGSE.2017.16.

Hüttermann, M. (2012) 'Introducing DevOps', *DevOps for Developers (Part I)*, pp.15–31, doi: 10.1007/978-1-4302-4570-4_2.

Jabbari, R., Ali, N., Petersen, K. and Tanveer, B. (2016) 'What is DevOps?', *Proceedings of the Scientific Workshop Proceedings of XP2016 on – XP '16 Workshops*, pp.1–11, doi: 10.1145/2962695.2962707.

Jimenez, I., Sevilla, M., Watkins, N., Maltzahn, C., Lofstead, J., Mohror, K., Arpaci-Dusseau, A.C. and Arpaci-Dusseau, R. (2017) 'The popper convention: Making reproducible systems evaluation practical', *Proceedings – 2017 IEEE 31st International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2017*, pp.1561–1570, doi: 10.1109/IPDPSW.2017.157.

John, W., Meirosu, C., Pechenot, B., Sköldström, P., Kreuger, P. and Steinert, R. (2015) 'Scalable software defined monitoring for service provider DevOps', *Proceedings – European Workshop on Software Defined Networks, EWSDN*, pp.61–66, doi: 10.1109/EWSDN.2015.62.

Karapantelakis, A., Liang, H., Wang, K., Vandikas, K., Inam, R., Fersman, E., Mulas-Viela, I., Seyvet, N. and Giannokostas, V. (2016) 'DevOps for IoT applications using cellular networks and cloud', *Proceedings – 2016 IEEE 4th International Conference on Future Internet of Things and Cloud, FiCloud 2016*, pp.340–347, doi: 10.1109/FiCloud.2016.55.

Katal, A., Bajoria, V. and Dahiya, S. (2019) 'DevOps: bridging the gap between development and operations', *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp.1–7, doi: 10.1109/iccmc.2019.8819631.

Kawamoto, S. and De Almeida, J.R. (2017) 'Scrum-DR: an extension of the scrum framework adherent to the capability maturity model using design rationale techniques', *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2017 – Proceedings*, January, pp.1–7, doi: 10.1109/CHILECON.2017.8229530.

Khan, K.S., Kunz, R., Kleijnen, J. and Antes, G. (2004) 'Systematic reviews to support evidence-based medicine: how to review and apply findings of healthcare research', *Postgraduate Medical Journal*, pp.53–62, doi: 10.1016/S1036-7314(00)70624-2.

Kitchenham, B. (2004) *Procedures for Performing Systematic Reviews*, Joint Technical Report, Computer Science Department, Keele University, doi: 10.1.1.122.3308.

Koilada, D.K. (2019) 'Business model innovation using modern DevOps', *2019 IEEE Technology & Engineering Management Conference (TEMSCON)*, IEEE, (Cd), pp.1–6, doi: 10.1109/temscon.2019.8813557.

Kuusinem, K., Balakumar, V., Jepsen, S.C., Larsen, S.H., Lemqvist, T.A., Muric, A., Nielsen, A. and Vestergaard, O. (2018) 'A large agile organization on its journey towards DevOps', *Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Prague, pp.60–63. doi: 10.1109/SEAA.2018.00019.

Lahrmann, G., Marx, F., Mettler, T., Winter, R. and Wortmann, F. (2011) 'Inductive design of maturity models: applying the Rasch algorithm for design science research', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, doi: 10.1007/978-3-642-20633-7_13.

Laukkarinen, T., Kuusinen, K. and Mikkonen, T. (2017) 'DevOps in regulated software development: case medical devices', *Proceedings – 2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Results Track, ICSE-NIER 2017*, pp.15–18, doi: 10.1109/ICSE-NIER.2017.20.

Laukkarinen, T., Kuusinen, K. and Mikkonen, T. (2018) 'Regulated software meets DevOps', *Information and Software Technology*, Elsevier B.V., Vol. 97, pp.176–178, doi: 10.1016/j.infsof.2018.01.011.

Lewerentz, M., Bluhm, T., Daher, R., Dumke, S., Grahl, M., Grün, M., Holtz, A., Krom, J., Kühner, G., Laqua, H., Riemann, H., Spring, A., Werner, A. and W7-X Team (2018) 'Implementing DevOps practices at the control and data acquisition system of an experimental fusion device', *Fusion Engineering and Design*, Elsevier, November, pp.0–1, doi: 10.1016/j.fusengdes.2018.11.022.

Li, Z., Zhang, Y. and Liu, Y. (2017) 'Towards a full-stack devops environment (platform-as-a-service) for cloud-hosted applications', *Tsinghua Science and Technology*, Vol. 22, No. 1, pp.1–9, doi: 10.1109/TST.2017.7830891.

Luz, W.P., Pinto, G. and Bonifácio, R. (2018) 'Building a collaborative culture', *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement – ESEM '18*, ACM Press, New York, USA, pp.1–10, doi: 10.1145/3239235.3240299.

Lwakatare, L.E., Kuvaja, P. and Oivo, M. (2015) 'Dimensions of devOps', *Lecture Notes in Business Information Processing*, pp.212–217, doi: 10.1007/978-3-319-18612-2_19.

Mackey, T. (2018) 'Building open source security into agile application builds', *Network Security*, Elsevier Ltd, Vol. 2018, No. 4, pp.5–8, doi: 10.1016/S1353-4858(18)30032-1.

Mansfield-Devine, S. (2018) 'DevOps: finding room for security', *Network Security*, Elsevier Ltd, Vol. 2018, No. 7, pp.15–20, doi: 10.1016/S1353-4858(18)30070-9.

Manzini, E. (2004) 'Entrevista semi-estruturada: análise de objetivos e de roteiros', *Seminário internacional sobre pesquisa e estudos qualitativos*, doi: 10.1590/S0036-36342005000100012.

Marijan, D., Liaaen, M. and Sen, S. (2018) 'DevOps improvements for reduced cycle times with integrated test optimizations for continuous integration', *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, pp.22–27, doi: 10.1109/COMPSAC.2018.00012.

McCarthy, M.A. *et al.* (2015) 'Composable DevOps: automated ontology based DevOps maturity analysis', *Proceedings – 2015 IEEE International Conference on Services Computing, SCC 2015*, pp.600–607, doi: 10.1109/SCC.2015.87.

Meho, L.I. (2006) 'E-mail interviewing in qualitative research: a methodological discussion', *Journal of the American Society for Information Science and Technology*, Vol. 57, No. 2004, pp.1284–1295, doi: 10.1002/asi.

Mettler, T. (2009) *A Design Science Research Perspective on Maturity Models in Information Systems*, Technical Report: BE IWI/HNE/03, Universiteit St. Gallen, doi: 10.2174/97816080506351100101.

Mikkonen, T., Lassenius, C., Männistö, T., Oivo, M. and Järvinen, J. (2018) 'Continuous and collaborative technology transfer: software engineering research with real-time industry impact', *Information and Software Technology*, Elsevier, Vol. 95, October, pp.34–45, doi: 10.1016/j.infsof.2017.10.013.

Minayo, M.C.S. (2004) 'O desafio do conhecimento: pesquisa qualitativa em saúde', *Saúde em debate*.

Mohamed, S. I. (2016) 'DevOps maturity calculator DOMC-value oriented approach', *International Journal of Engineering Research & Science*, Vol. 2, No. 2, pp.2395–6992.

Mohamed, S.I. (2015) 'DevOps shifting software engineering strategy value based perspective', *IOSR Journal of Computer Engineering Ver. IV*, Vol. 17, No. 2, pp.2278–661, doi: 10.9790/0661-17245157.

Mohan, V. and Ben Othmane, L. (2016) 'SecDevOps: Is it a marketing buzzword? Mapping research on security in DevOps', *Proceedings – 2016 11th International Conference on Availability, Reliability and Security, ARES 2016*, pp.542–547, doi: 10.1109/ARES.2016.92.

Moltó, G., Caballer, M., Pérez, A., Alfonso, C. and Blanquer, I. (2017) 'Coherent application delivery on hybrid distributed computing infrastructures of virtual machines and docker containers', *Proceedings – 2017 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2017*, pp.486–490, doi: 10.1109/PDP.2017.29.

Moore, J., Kortuem, G., Smith, A., Chowdhury, N., Cavero, D. and Gooch, D. (2016) 'DevOps for the Urban IoT', *Proceedings of the Second International Conference on IoT in Urban Space – Urb-IoT '16*, pp.78–81, doi: 10.1145/2962735.2962747.

Murugesan, S. (2017) 'Opening statement', *Cutter IT Journal*, Vol. 30, No. 3, pp.3–5, doi: 10.1111/j.1467-8268.1991.tb00046.x.

Nidagundi, P. and Novickis, L. (2017) 'Towards utilization of lean canvas in the Devops software', *Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference*, Vol. 2, p.107, doi: 10.17770/etr2017vol2.2522.

Nielsen, P.A., Winkler, T.J. and Nørbjerg, J. (2017) 'Closing the IT development-operations gap: the devops knowledge sharing framework', *CEUR Workshop Proceedings*, Vol. 1898, pp.3–9.

Palihawadana, S., Wijeweera, C.H., Sanjitha, M.G., Liyanage, V.K., Perera, I. and Meedeniya, D.A. (2017) 'Tool support for traceability management of software artefacts with DevOps practices', *3rd International Moratuwa Engineering Research Conference, MERCon 2017*, pp.129–134, doi: 10.1109/MERCon.2017.7980469.

Pang, C. and Hindle, A. (2017) 'Continuous maintenance', *Proceedings – 2016 IEEE International Conference on Software Maintenance and Evolution, ICSME 2016*, pp.458–462, doi: 10.1109/ICSME.2016.45.

Patel, C. and Ramachandran, M. (2009) 'Agile Maturity Model (AMM): a software process improvement framework for agile software development practices', *Int. J. of Software Engineering, IJSE*, Vol. 2, No. I, pp.3–28, doi: 10.4304/jsw.4.5.422-435.

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V. and Bragge, J. (2006) 'The design science research process: a model for producing and presenting information systems research', *The Proceedings of Design Research in Information Systems and Technology DESRIST 2006*, pp.12–14, doi: 10.2753/MIS0742-1222240302.

Pereira, R. and Da Silva, M.M. (2011) 'A maturity model for implementing ITIL V3 in practice', *Proceedings – IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, doi: 10.1109/EDOCW.2011.30.

Pereira, R. and Serrano, J. (2020) 'A review of methods used on IT maturity models development: a systematic literature review and a critical analysis', *Journal of Information Technology*, p.026839621988687, doi: 10.1177/0268396219886874.

Perera, P., Bandara, M. and Perera, I. (2017a) 'Evaluating the impact of DevOps practice in Sri Lankan software development organizations', *16th International Conference on Advances in ICT for Emerging Regions, ICTer 2016 – Conference Proceedings*, December, pp.281–287, doi: 10.1109/ICTER.2016.7829932.

Perera, P., Silva, R. and Perera, I. (2017b) 'Improve software quality through practicing DevOps', *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, IEEE, pp.1–6, doi: 10.1109/ICTER.2017.8257807.

Pozzebon, M. (2006) 'Conducting and evaluating critical interpretive research: examining criteria as a key component in building a research tradition', *Information Systems Research*, doi: 10.1007/1-4020-8095-6_16.

Prates, L., Faustino, J., Silva, M. and Pereira, R. (2019) *DevSecOps Metrics*, doi: 10.1007/978-3-030-29608-7_7.

Pries-Heje, J., Baskerville, R. and Venable, J.R. (2008) 'Association for information systems ais electronic library (AISeL) strategies for design science research evaluation', *European Conference on Information Systems*, p.87, Available at: http://aisel.aisnet.org/ecis2008

Proenca, D. (2016) 'Methods and techniques for maturity assessment', *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pp.1–4, doi: 10.1109/CISTI.2016.7521483.

Punjabi, R. and Bajaj, R. (2017) 'User stories to user reality: a devops approach for the cloud', *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 – Proceedings*, pp.658–662, doi: 10.1109/RTEICT.2016.7807905.

Rahman, A., Mahdavi-Hezaveh, R. and Williams, L. (2018) 'A systematic mapping study of infrastructure as code research', *Information and Software Technology*, Elsevier B.V., November, pp.2–7, doi: 10.1016/j.infsof.2018.12.004.

Rana, R. and Staron, M. (2016) 'First international workshop on emerging trends in devops and infrastructure', *Proceedings of the Scientific Workshop Proceedings of XP2016 on – XP '16 Workshops*, pp.1–3, doi: 10.1145/2962695.2962706.

Ratislavová, K. and Ratislav, J. (2014) 'Asynchronous email interview as a qualitative research method in the humanities', *Human Affairs*, Vol. 4, No. 24, pp.452–460, doi: 10.2478/s13374-014-0240-y.

Rijo, R.P.C.L. (2008) *Framework para a Gestão de Projectos de Sistemas de Informação de Contact Centers*, February, p.326.

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L.E., Tiihonen, J. and Männistö, T. (2016) 'DevOps adoption benefits and challenges in practice: a case study', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp.590–597, doi: 10.1007/978-3-319-49094-6_44.

Roche, J. (2013) 'Adopting DevOps practices in quality assurance', *Communications of the ACM*, Vol. 56, No. 11, pp.38–43, doi: 10.1145/2524713.2524721.

Rodríguez, P., Mäntylä, M., Oivo, M., Lwakatare, L.E., Seppänen, P. and Kuvaja, P. (2018) 'Advances in using agile and lean processes for software development', *Advances in Computers*, doi: 10.1016/bs.adcom.2018.03.014.

Röglinger, M., Schwindenhammer, L. and Stelzl, K. (2018) 'How to put organizational ambidexterity into practice – towards a maturity model', *BPM: International Conference on Business Process Management – Lectures Notes in Business Information Processing.*

Rong, G., Zhang, H. and Shao, D. (2016a) 'CMMI guided process improvement for DevOps projects', *Proceedings of the International Workshop on Software and Systems Process – ICSSP '16*, ACM Press, New York, USA, pp.76–85, doi: 10.1145/2904354.2904372.

Rong, G., Zhang, H. and Shao, D. (2016b) 'CMMI guided process improvement for DevOps projects', *Proceedings of the International Workshop on Software and Systems Process – ICSSP '16*, ACM Press, New York, USA, pp.76–85, doi: 10.1145/2904354.2904372.

Rubasinghe, I., Meedeniya, D. and Perera, I. (2018) 'Automated inter-artefact traceability establishment for devops practice', *Proceedings of the IEEE/ACIS 17th International Conference on Computer and Information Science*, pp.211–216, doi: 10.1109/ICIS.2018.8466414.

Rubasinghe, I.D., Meedeniya, D.A. and Perera, I. (2017) 'Towards traceability management in continuous integration with SAT-analyzer', *Proceedings of the 3rd International Conference on Communication and Information Processing*, pp.77–81, doi: 10.1145/3162957.3162985.

Rufino, J., Alam, M. and Ferreira, J. (2017) 'Monitoring V2X applications using DevOps and docker', *2017 International Smart Cities Conference, ISC2 2017*, pp.2–9, doi: 10.1109/ISC2.2017.8090868.

Samarawickrama, S.S. and Perera, I. (2018) 'Continuous scrum: a framework to enhance scrum with DevOps', *17th International Conference on Advances in ICT for Emerging Regions, ICTer 2017 – Proceedings*, pp.19–25, doi: 10.1109/ICTER.2017.8257808.

Santana, F., Soares, F. and Meira, L. (2013) 'An agile maturity model for software development organizations', *Proceedings of the Eighth International Conference on Software Engineering Advances (ICSEA'13)*, pp.324–328, Available at: http://s3.amazonaws.com/academia.edu.documents/31692670/ICSEA-AgileMM_v2.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1492019091&Signature=%2B72RILftao361ieM7EnKqlsWA38%3D&response-content-disposition=inline%3Bfilename%3DAn_Agile_Maturity_Model_for_

Schumacher, A., Erol, S. and Sihn, W. (2016) 'A maturity model for assessing industry 4.0 readiness and maturity of manufacturing enterprises', *Procedia CIRP*, The Author(s), Vol. 52, pp.161–166, doi: 10.1016/j.procir.2016.07.040.

Science, C. and Sciences, N. (2015) 'Full-scale software engineering FsSE 2015', *Full-scale Software Engineering FsSE*, pp.31–36.

Science, M. (2016) *Research Article Goal Oriented DevOps Transformation Framework – Metric Phased Approach*, Samer I. Mohamed Modern Science and Arts University, Faculty of Engineering, Electrical and Communication Department.

Senapathi, M., Buchan, J. and Osman, H. (2018) 'DevOps Capabilities, practices, and challenges', *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 – EASE'18*, pp.57–67, doi: 10.1145/3210459.3210465.

Shahin, M., Babar, M. A. and Zhu, L. (2016) 'The intersection of continuous deployment and architecting process', *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement – ESEM '16*, pp.1–10, doi: 10.1145/2961111.2962587.

Sharma, S. (2017a) 'DevOps plays for driving innovation', *The DevOps Adoption Playbook*, pp.189–260, doi: 10.1002/9781119310778.ch5.

Sharma, S. (2017b) *The DevOps Adoption Playbook*, John Wiley & Sons, Inc., Indianapolis, Indiana, doi: 10.1002/9781119310778.

Sharma, S. and Coyne, B. (2015) *DevOps for Dummies*, John Wiley & Sons, Inc., doi: 10.1017/CBO9781107415324.004.

Shivakumar, S.K. (2017) *DevOps for Digital Enterprises*, White Paper, Infosys, Available at: https://www.infosys.com/digital/insights/documents/devops-digital-enterprises.pdf

Siddaway, A. (2014) *What is a Systematic Literature Review and How Do I Do One?*, University of Sterling, Sterling, UK.

Silva., M., Fautino, J., Pereira, R. and Mira da Silva, M. (2018) 'Productivity gains of DevOps adoption in an IT team: a case study', *27th International Conference on Information Systems Development*, pp.1–10, Available at: https://repositorio.iscte-iul.pt/handle/10071/16388

Smeds, J., Nybom, K. and Porres, I. (2015a) 'DevOps: a definition and perceived adoption impediments', *Lecture Notes in Business Information Processing*, doi: 10.1007/978-3-319-18612-2_14.

Smeds, J., Nybom, K. and Porres, I. (2015b) 'DevOps: a definition and perceived adoption impediments', in Lassenius, C., Dingsøyr, T. and Paasivaara, M. (Eds.): *Lecture Notes in Business Information Processing*, Springer International Publishing (Lecture Notes in Business Information Processing), Cham, pp.166–177, doi: 10.1007/978-3-319-18612-2_14.

Snyder, B. and Curtis, B. (2017) 'Using analytics to guide improvement during an agile-DevOps transformation', *IEEE Software*, Vol. 35, No. 1, pp.78–83, doi: 10.1109/MS.2017.4541032.

Soni, M. (2016) 'End to end automation on cloud with build pipeline: the case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery', *Proceedings – 2015 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2015*, pp.85–89, doi: 10.1109/CCEM.2015.29.

Srivastava, A., Bhardwaj, S. and Saraswat, S. (2017) 'SCRUM model for agile methodology', *Proceeding – IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, January, pp.864–869, doi: 10.1109/CCAA.2017.8229928.

St, D., Ab, E. and Bosch, J. (2017) 'Continuous practices and DevOps: beyond the buzz, what does it all mean?', *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA) Continuous*, pp.440–448, doi: 10.1109/SEAA.2017.78.

Steffens, A., Lichter, H. and Döring, J.S. (2018) 'Designing a next-generation continuous software delivery system', *Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering – RCoSE '18*, pp.1–7, doi: 10.1145/3194760.3194768.

Stojanov, I., Turetken, O. and Trienekens, J.J.M. (2015) 'A maturity model for scaling agile development', *Proceedings – 41st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2015*, pp.446–453, doi: 10.1109/SEAA.2015.29.

Stoneham, J., Thrasher, P., Potts, T., Mickman, H. and DeArdo, C. (2016) 'DevOps case studies', *DevOps Enterprise Forum 2016*, p.46.

Sturm, R., Pollard, C. and Craig, J. (2017) 'DevOps and continuous delivery', *Application Performance Management (APM) in the Digital Enterprise*. Elsevier, pp.121–135, doi: 10.1016/B978-0-12-804018-8.00010-3.

Swanson, E.B. and Beath, C.M. (1990) 'Departmentalization in software development and maintenance', *Communications of the ACM*, pp.4–5, doi: 10.1145/78973.78976.

Tessem, B. and Iden, J. (2008) 'Cooperation between developers and operations in software engineering projects', *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering – CHASE '08*, pp.54–66, doi: 10.1145/1370114.1370141.

Tuma, K., Calikli, G. and Scandariato, R. (2018) 'Threat analysis of software systems: a systematic literature review', *Journal of Systems and Software*, Elsevier Inc., Vol. 144, pp.275–294, doi: 10.1016/j.jss.2018.06.073.

Ur Rahman, A.A. and Williams, L. (2016a) 'Security practices in DevOps', *Proceedings of the Symposium and Bootcamp on the Science of Security – HotSos '16*, pp.109–111, doi: 10.1145/2898375.2898383.

Ur Rahman, A.A. and Williams, L. (2016b) 'Software security in DevOps', *Proceedings of the International Workshop on Continuous Software Evolution and Delivery – CSED '16*, pp.70–76, doi: 10.1145/2896941.2896946.

Vassallo, C., Zampetti, F., Romano, D., Beller, M., Panichella, A., Penta, M. and Zaidman, A. (2017) 'Continuous delivery practices in a large financial organization', *Proceedings – 2016 IEEE International Conference on Software Maintenance and Evolution, ICSME 2016*, pp.519–528, doi: 10.1109/ICSME.2016.72.

Velasquez, N.F., Mann, A., Brown, A., Stahnke, M. and Kersten, N. (2018) *State of DevOps Report 2018*, Puppetlabs, doi: 10.1016/S0022-3913(12)00047-9.

Venable, J., Pries-Heje, J. and Baskerville, R. (2012) 'A comprehensive framework for evaluation in design science research', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp.2–20, doi: 10.1007/978-3-642-29863-9_31.

Venable, J.R. (2006) 'A framework for design science research activities', *Emerging Trends and Challenges in Information Technology Management*, Vol. 2, pp.184–187, doi: 10.4018/978-1-59904-019-6.ch044.

Webster, J. and Watson, R.T. (2002) 'Analyzing the past to prepare for the future: writing a literature review', *MIS Quarterly*, Vol. 26, No. 2, pp.xiii–xxiii, doi: 10.1.1.104.6570.

Wiesche, M. (2018) 'Are you ready for devops? required skill set for devops teams DevOps teams', *Twenty-Sixth European Conference on Information Systems (ECIS2018)*, pp.12–14.

Wongkampoo, S. and Kiattisin, S. (2018) 'Atom-task precondition technique to optimize large scale GUI testing time based on parallel scheduling algorithm', *ICSEC 2017 – 21st International Computer Science and Engineering Conference 2017, Proceeding*, Vol. 6, pp.229–232, doi: 10.1109/ICSEC.2017.8443913.

Xia, C., Zhang, Y., Wang, L., Coleman, S. and Liu, Y. (2018) 'Microservice-based cloud robotics system for intelligent space', *Robotics and Autonomous Systems*, Elsevier B.V., Vol. 110, pp.139–150, doi: 10.1016/j.robot.2018.10.001.

Yin, A.P.G. (2011) 'Scrum Maturity Model. Dissertacao para obtencao do Grau de Mestre em Engenharia Informática e de Computadores', *Universidade Técnica de Lisboa*, p.165.

Yin, X., Zhang, J. and Wang, X. (2004) 'Summary for policymakers', in Intergovernmental Panel on Climate Change (Ed.): *Climate Change 2013 – The Physical Science Basis*, Cambridge University Press, Cambridge, pp.1–30, doi: 10.1017/CBO9781107415324.004.

Zhu, H. and Bayley, I. (2018) 'If docker is the answer, what is the question?', *Proceedings – 12th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2018 and 9th International Workshop on Joint Cloud Computing, JCC 2018*, pp.152–163, doi: 10.1109/SOSE.2018.00027.

Zhu, L., Bass, L. and Champlin-Scharff, G. (2016) 'DevOps and its practices', *IEEE Software*, Vol. 33, No. 3, pp.32–34, doi: 10.1109/MS.2016.81.

Appendix

Table 14 CI practices

	Continuous integration	
	Practice	*Author*
People	–	–
Process	Automation of tasks	Nielsen et al. (2017)
	Provision of virtualised hardware resources via scripts (instead of doing manual configuration work)	
	Developers should make use of continuous integration, that is branch-out and merge- back their work with the software mainline (the trunk) several times a day, in order to discover integration risks as early as possible	
	Continuous integration cycles to include also software release	de França et al. (2016)
	Continuous feedback loop	
	Enable rapid automated regression testing of code changes	Marijan et al. (2018)
	Test in a clone of the production environment	Sharma (2017a)
	Make it easy for anyone to get the latest executable	
Technology	Use of cloud services	Nielsen et al. (2017)
	Tools interoperability for unifying force across diverse teams, skills, technology languages, and methodologies	
	Version Control	Humble and Farley (2011)
	An Automated Build	
	Use build servers	Sharma (2017a)
	Maintain a single-source repository	
	Automate the build	
Culture	Collaboration between teams	Luz et al. (2018)
	Development and QA teams perform unit and integration testing	Sturm et al. (2017)
	Operations participates in integration and load testing to assess operational readiness	
	Agreement of the team	Humble and Farley (2011)
	Make sure everyone can see what is happening	Sharma (2017a)

Table 15 Continuous monitoring practices

	Continuous monitoring	
	Practice	*Author*
People	Analysis skills	Wiesche (2018)
Process	Define some useful measurement metrics	Nielsen et al. (2017)
	Ensure continuous feedback provided through the monitoring process and the users	

Table 15 Continuous monitoring practices (continued)

	Continuous monitoring	
	Practice	*Author*
Process	Application monitoring	Sharma (2017b)
	System monitoring	
	Application user behaviour	
	User sentiment	
	Delivery pipeline metrics	
	Systems are monitored after deployment	Zhu et al. (2016)
	• Instrumenting your applications and your infrastructure so you can collect the data you need	Humble and Farley (2011)
	• Storing the data so it can easily be retrieved for analysis	
	• Creating dashboards which aggregate the data and present it in a format suitable for operations and for the business	
	• Setting up notifications so that people can find out about the events they care about	
Technology	Analytics can be used to integrate the system and infrastructure performance data with customer usage behaviour	Lwakatare et al. (2015)
	Not just gather this data but also run analytics on it	Sharma (2017b)
	Basic services such as dashboards	Senapathi et al. (2018)
	Use a Realtime User Monitoring tool	Erich et al. (2014b)
	APIs or services	Humble and Farley (2011)
	The application should use to notify the operations team of its state	
Culture	Collaboration between developers and operations so that the systems are designed to expose relevant information	Lwakatare et al. (2015)

Table 16 Continuous testing practices

	Continuous testing	
	Practice	*Author*
People	Understand test automation functions	Wiesche (2018)
	Automate tests	
	Understand functionalities for test management	
Process	Script-based testing early and throughout the software delivery process	Nielsen et al. (2017)
	Shorten later testing cycles	
	Ensure continuous feedback on quality	
	Testing earlier and continuously across the life cycle	Sharma and Coyne (2015)
	High test coverage of high-risk areas	Marijan et al. (2018)
	Integrate testing activities as closely as possible with coding	Fitzgerald and Stol (2014)

Table 16 Continuous testing practices (continued)

	Continuous testing	
	Practice	*Author*
Technology	Virtualisation to simulate the production environments	Silva et al. (2018)
	Test case generation	Vassallo et al. (2017)
Culture	Both IT Development and IT Operations should carry out quality assurance and be responsible for test automation	Nielsen et al. (2017)
	Each developer should take personal responsibility for their code and write the test cases	De Bayser et al. (2015)
	Testing on real users at scale	Feitelson et al. (2013)
	Driving development with tests	Vassallo et al. (2017)
	TDD is a development practice that starts with writing tests before you write any code	Perera et al. (2017b)
	BDD encourages working with the business stakeholder to describe the desired business functionality of the application	
	ATDD builds on TDD and BDD, and it is involved in finding scenarios from the end user perspective	
	Testing/quality team is connected with Development team early in the development cycle to create the required test cases	Mohamed (2015)

Table 17 Infrastructure as a code practices

	Infrastructure as code	
	Practice	*Author*
People	–	–
Process	Versioning environments	Mohamed (2016)
Technology	Entire infrastructure in a common language	Luz et al. (2018)
	Automate server	Sharma and Coyne (2015)
	Generic tools	
	Application or middleware-centric tools	
	Environment and deployment tools	
Culture	Everyone knows how the execution environment of an application is provided and managed	Luz et al. (2018)

Table 18 Feedback loops practices

	Feedback loops between Dev and Ops	
	Practice	*Author*
People	Feedback ability, in both directions – so, to give feedback but also to accept it	Wiesche (2018)
Process	Shorten later testing cycles to ensure continuous feedback	Nielsen et al. (2017)
	Ensure continuous feedback provided through the monitoring process and the users	

Table 18 Feedback loops practices (continued)

	Feedback loops between Dev and Ops	
	Practice	*Author*
Process	The frequency of integration is also important in that it should be regular enough to ensure quick feedback to developers	Fitzgerald and Stol (2014)
	Mechanisms to involve users in the development process and collect user feedback from deliveries as early as possible	Rodríguez et al. (2018)
	Techniques need to be nonintrusive so that users are not stressed with continuous feedback requests	
	Short feedback loops	
	Feedback loops strategy	Science (2016)
	The measurement results should be provided to not only the operation people, but also the development people	Rong et al. (2016b)
	Any change, of whatever kind, needs to trigger the feedback process	Humble and Farley (2011)
	The feedback must be delivered as soon as possible	
	The delivery team must receive feedback and then act on it	
Technology	–	–
Culture	Share feedback freely without blame	Perera et al. (2017a)
	High focus on requirements	Nielsen et al. (2017)
	Management through close relationship with the users to determine their needs and quickly react on their feedback	
	Keeping a constant feedback about the current state of the system	Rodríguez et al. (2018)

Table 19 CDMM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous deployment	People	–	–	–	–
	Process	CD9 Label a repository's assets	CD2 Track which version is deployed CD3 Manage the configurations of the environments of all the stages CD4 Manage the software components that get deployed CD5 Manage the middleware components and middleware configurations that need to be updated CD6 Manage the database components that need to be changed	CD1 Orchestrated deployments CD16 Deployments should include the automated provisioning of all environments	CD1 Orchestrated deployments CD7 Manage the configuration changes to the environments to which these components are to be deployed CD8 Release working software any time, any place CD15 Multiple deployments to production

Table 19 CDMM (continued)

		Level 2	Level 3	Level 4	Level 5
Continuous deployment	Process		CD10 Produce a clean environment CD11 Label each build CD12 Create build feedback Reports CD14 Deploy a new release whenever one is needed CD17 Automated deployment CD18 Continuous deployment		
	Technology	–	CD19 Development and production share a homogenous infrastructure CD20 Configuration management tools	CD21 Automated deployment of software to different environments	–
	Culture	–	CD22 Team must provide overall visibility into your application release activities and timing to all major stakeholders CD25 Unite the two teams that worked independently to work at tighter integration CD26 Both development and operations personnel should share the same knowledge management resources CD27 Testers and operations personnel would be able to self-service deployments of the required version of the system to their environments on demand CD28 Early and frequent involvement of operations staff in the planning stages of major new releases	CD24 Team must be able to speed lead times and make more frequent application deployments at the pace demanded by the business	CD23 Teams must be able to provide self-service, on-demand provisioning and management of cloud environments and infrastructure resources

Table 20 CI MM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous integration	People	–	–	–	–
	Process	CI8 Make it easy for anyone to get the latest executable	CI1 Automation of tasks CI2 Provision of virtualised hardware resources via scripts (instead of doing manual configuration work) CI3 Developers should make use of continuous integration, that is branch-out and merge-back their work with the software mainline (the trunk) several times a day, in order to discover integration risks as early as possible CI5 Continuous integration cycles to include also software release CI6 Enable rapid automated regression testing of code changes	CI4 Continuous feedback loop CI7 Test in a clone of the production environment	–
	Technology	CI11 Version control CI15 Automate the build	CI12 An automated build CI13 Use build servers	CI9 Use of cloud services	CI10 Tools interoperability for unifying force across diverse teams, skills, technology languages, and methodologies CI14 Maintain a single-source repository
	Culture	CI16 Collaboration between teams CI19 Agreement of the Team	CI17 Development and QA teams perform unit and integration testing	CI18 Operations participates in integration and load testing to assess operational readiness CI20 Make sure everyone can see what is happening	–

Table 21 Continuous monitoring MM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous monitoring	People	CM1 Analysis skills	–	–	–
	Process		CM4 Application monitoring CM5 System monitoring CM8 Delivery pipeline metrics CM11 Storing the data so it can easily be retrieved for analysis CM13 Setting up notifications so that people can find out about the events they care about	CM2 Define some useful measurement metrics CM6 Application user behaviour CM7 User sentiment CM9 Systems are monitored after deployment CM10 Instrumenting your applications and your infrastructure so you can collect the data you need CM12 Creating dashboards which aggregate the data and present it in a format suitable for operations and for the business	CM3 Ensure continuous feedback provided through the monitoring process and the users
	Technology		CM16 Basic services such as dashboards CM17 Use a realtime user monitoring tool CM18 APIs or services	CM19 The application should use to notify the operations team of its state	CM14 Analytics can be used to integrate the system and infrastructure performance data with customer usage behaviour CM15 Not just gather this data but also run analytics on it
	Culture		CM20 Collaboration between developers and operations so that the systems are designed to expose relevant information		

Table 22 Continuous testing MM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous monitoring	People	–	CT1 Understand test automation functions CT2 Automate tests CT3 Understand functionalities for test management	–	–
	Process	–	CT4 Script-based testing early and throughout the software delivery process CT6 Ensure continuous feedback on quality	CT5 Shorten later testing cycles CT7 Testing earlier and continuously across the life cycle CT8 High test coverage of high-risk areas CT9 integrate testing activities as closely as possible with coding	–
	Technology	–	CT10 Virtualisation to simulate the production environments CT11 test case generation	–	–
	Culture	CT15 driving development with tests CT16 TDD is a development practice that starts with writing tests before you write any code CT19 Testing/quality team is connected with Development team early in the development cycle to create the required test cases	CT13 Each developer should take personal responsibility for their code and write the test cases CT17 BDD encourages working with the business stakeholder to describe the desired business functionality of the application CT18 ATDD builds on TDD and BDD, and it is involved in finding scenarios from the end user perspective	CT12 Both IT development and IT Operations should carry out quality assurance and be responsible for test automation	CT14 Testing on real users at scale

Table 23 Infrastructure as code MM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous monitoring	People	–	–	–	–
	Process	–	IAC1 Versioning environments	–	–
	Technology	–	IAC2 Entire infrastructure in a common language IAC3 Automate server IAC4 Generic tools IAC5 Application or middleware-centric tools IAC6 Environment and deployment tools	–	–
	Culture	–	–	–	IAC7 Everyone knows how the execution environment of an application is provided and managed

Table 24 Feedback loops MM

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous monitoring	People	–	–	FL1 Feedback ability, in both directions – so, to give feedback but also to accept it	–
	Process	–	FL2 Shorten later testing cycles to ensure continuous feedback FL4 The frequency of integration is also important in that it should be regular enough to ensure quick feedback to developers FL7 Short feedback loops FL11 The delivery team must receive feedback and then act on it	FL3 Ensure continuous feedback provided through the monitoring process and the users FL5 Mechanisms to involve users in the development process and collect user feedback from deliveries as early as possible FL8 Feedback loops strategy the measurement results should be provided to not only the operation people, but also the development people FL10 The feedback must be delivered as soon as possible	FL6 Techniques need to be nonintrusive so that users are not stressed with continuous feedback requests FL9 Any change, of whatever kind, needs to trigger the feedback process

Table 24 Feedback loops MM (continued)

		Level 2	*Level 3*	*Level 4*	*Level 5*
Continuous monitoring	Technology	–	–	–	–
	Culture	–	–	FL13 High focus on requirements	FL12 Share feedback freely without blame
					FL14 Management through close relationship with the users to determine their needs and quickly react on their feedback
					FL15 Keeping a constant feedback about the current state of the system