ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 10, No. 3, June 2022, pp.221-235

DevOps model in practice: Applying a novel reference model to support and encourage the adoption of DevOps in a software development company as case study

César Pardo[1], Jonathan Guerrero[1] and Elizabeth Suescún[2]
[1] University of Cauca, GTI Research Group. Calle 5 No. 4-70, Popayán, Colombia
[2] EAFIT University, GIDITIC Research Group. Carrera 49 N° 7 Sur-50, Medellín, Colombia

ABSTRACT

DevOps has emerged as an approach to help organizations automate, cost optimization, increase profitability, improve the stability of the software development process and the responsiveness of organizations, and create a more agile development and release pipeline. However, its adoption, maintenance and evaluation continue to be a challenge for software organizations, due to the absence of solutions that formalize process elements in a detailed way, such as: practices, roles, artifacts, objectives, among others. This paper presents a DevOps Model, this model to support the adoption of DevOps, which provides a set of fundamental and complementary values, principles, dimensions, and practices. The practices suggest a set of items such as purpose, specific objectives and expected artifacts. The elements defined in proposed DevOps Model arise from the elements found in the studies analyzed through a systematic mapping study. Model evaluation was carried out through a software development company as case study. The results obtained have allowed the case study company to evaluate, diagnose and identify improvement opportunities to be carried out in the processes and projects where a DevOps-based approach is used, the above in a practical, useful, and adequate way that allows this type of companies and with a low use of resources, both economic investment and time. This is how the DevOps Model could guide professionals and organizations towards a better understanding of DevOps, in addition to minimizing the subjectivity and error of its interpretation, adoption and evaluation.

Keywords: Development and Operations (DevOps), Software Process Improvement (SPI), Agile Development, Automation, Pipeline

Corresponding Author:

César Pardo
PhD. MSc. Researcher
University of Cauca
Carrera 2º Calle 15n. Office 444, Colombia
E-mail: cpardo@unicauca.edu.co

1. Introduction

The process of building a software product involves the intervention of two different areas: (i) the *development* area and (ii) the *infrastructure* area, also called *Information Technology* (IT) or *Operations*. The development area is responsible for: analysis, design, and construction of the source code, which is tested and that allows to subsequently generate a version of a software product. On the other hand, the operations area is responsible for the configuration of the hardware that is required with the necessary services, in order to carry out the deployment of the versions of the software product, versions that are generated by the development area in productive environments for use by the end customer, and additionally; monitor the behavior of the software product during the time it is in use and ensure the stability of the software product in productive environments [1]. In this sense, with DevOps the teams are more empowered and the steps to production are more reliable due to automation and trust in the process. Currently, there is a wide portfolio of frameworks and approaches that guide the development process of a software product, these solutions can be traditional or agile approaches. Traditional approaches seek to impose discipline on the software development process and thereby make it predictable and efficient. To achieve this, they are supported by a detailed process with an emphasis on planning. On the other hand, agile approaches are adaptive - not predictive - and oriented to people and not processes and generation of value from early stages, which makes a great difference compared to traditional frameworks [2].

According to [3], the agile approaches that stand out the most in the software industry today are: Scrum [4], Extreme Programming (XP) [5], Crystal Clear [6], Lean Software Development (LSD) [7], Adaptive Software Development (ASD) [8], Dynamic Systems Development Method (DSDM) [9], Feature Driven Development (FDD) [10], Agile Unified Process (Agile UP) [11], Kanban [12], among others. On the other hand, among the most popular traditional approaches are: Rational Unified Process (RUP) [13], Microsoft Solutions Framework (MSF) [14], Capability Maturity Model Integration (CMMI) [15], Model-Based Architecture and Software Engineering (MBASE) [16], among others. From the perspective of the infrastructure area, we also find *de facto* frameworks, as well as: ITIL [17] and COBIT [18], and international standards such as ISO / IEC 2000 [19], where are proposed elements, practices and / or activities related to the management of IT services and the management of continuous improvement processes in IT services.

However, the aforementioned frameworks define their process elements from one of the two areas involved in the software development process, i.e.: from the development area or the infrastructure area, which has created a kind of dividing wall between these two areas significantly affecting the development process of a software product from its conception to its final delivery [20], because the proposed frameworks define workflows, roles and artifacts only from a particular perspective or area, which is often complementary to the objectives and / or needs of the other area involved, e.g.: development areas implement practices to be more agile and be open to changes in requirements at any time, while the infrastructure area implements practices to control the stability of the requirements, since these directly affect the processes related to the enlistment of the technological infrastructure, this mean the main priority of the infrastructure area is to take care of the integrity of the deployed applications and based on this, the policies and processes are defined [1]. For that reason, the approach known as DevOps is currently gaining strength, which, as its name indicates, focuses on improving the relationships between the development area (Dev) and the operations area (Ops) [1]. Some of the benefits of implementing DevOps in a software company are: (i) eliminate the existing barriers between development area and infrastructure area [21], (ii) significantly improve in reaction times (time to market) [22], (iii) favor the improvement and automation of processes [23], [24], product quality improvement [3], faster software delivery [3], among others. In [3] the importance and benefits of adopting DevOps are highlighted, however, it also highlights the importance and increasing need to understand and implement DevOps through the clarification and formalization of practices related to development, integration, monitoring and continuous delivery of software products.

Taking into account the above, and after carrying out a systematic review of the literature [1], the lack of clarification and formalization of DevOps in the development of software products is evident, e.g.: the need to standardize a formal definition for DevOps, in addition to providing a detailed description of the fundamental and complementary practices that could be taken into account when using this approach. On the other hand, no clear roles or responsibilities are defined for proposed new activities, likewise, it is not clear how conflicts or problems related to organizational culture, processes, people, and tools will be addressed. In this sense, this paper presents a reference model to facilitate and support the adoption of DevOps from a set of fundamental and complementary values, principles, dimensions, and practices. Likewise, the evaluation of the proposed solution is presented through a focus group made up of expert participants in DevOps.

The rest of the paper is structured as follows: Section 2 presents the background and analysis of related works. Section 3 describe DevOps Model in terms of its values, principles, dimensions, and practices. Section 4 presents in detail the evaluation of the DevOps Model through a detailed case study. Finally, the conclusions and future work is outlined.

2. Related work

From a systematic mapping of the literature presented in [1], it has been possible to identify some efforts related to the design of solutions related to DevOps. First, it was possible to observe different definitions about DevOps, with which it can be inferred that there is still no common unified conceptualization and what prevails in the literature are different specific perspectives about it. A generic definition proposed from the definitions found can be consulted in [1]. In total, 26 related papers were finally selected, which were the result from the application of different inclusion and quality criteria. The type of proposals found are related to the following categories: (i) the adoption of DevOps [22], [23], [25]–[32] and (ii) systematic reviews of the literature [21], [24], [33]–[38], where 90% of the studies in the first category registered the application of solutions in case studies, and only 70% of these studies, recorded the results achieved and the opinion of the participants through surveys or interviews to determine the impact achieved with DevOps. In the second category, only 38% propose at a conceptual level some solutions for the implementation of DevOps through some agile solution such as Scrum.

Although it is possible to find solutions to facilitate the understanding of DevOps in software companies, only 20% of them [25], [28], [32], [33], [39] propose solutions on what should be considered in a more concrete and detailed way. The remaining 80% [21], [22], [35]–[38], [40]–[45], [23], [46], [24], [26], [27], [29]–[31], [34] of the analyzed studies, propose non-specific or high-level solutions for the adoption of DevOps in development companies of Even software, most describe the experience of adopting DevOps without proposing new practices. Some of them (56%) [21], [22], [40]–[43], [24]–[26], [31]–[35] highlight the benefits and challenges. Likewise, 60% [23], [25], [37], [40], [43], [45], [46], [26], [27], [29], [30], [32]–[34], [36] of the studies suggest some process elements that could be considered for the adoption of DevOps. The remaining 40% [24], [27]–[33], [35], [46] mention some technological tools that could support DevOps practices. Both the benefits, challenges, tools, elements of processes identified, among other aspects identified in the analyzed literature can be consulted in greater detail in [1].

From the results obtained with the analysis of the literature, it is possible to observe that the proposals found are focused on supporting specific practices such as continuous integration, continuous deployment, and their automation. However, the proposed solutions are not very detailed and do not include different perspectives such as: architecture, culture, people, processes, tools, or other processes such as: continuous monitoring, quality, among others. Regarding the process elements suggested in the literature and related to DevOps, it could be observed that they are not clear, are not very detailed and there is ambiguity, e.g.: some of them are mentioned but not documented or described in detail, even the definitions in most cases are incomplete and confusing compared to other proposals. On the other hand, it was possible to note that most of the studies do not suggest the essential or complementary roles and responsibilities to support the performance of the proposed activities, tasks, or practices, nor is there clarity about the artifacts or objectives to be followed. Likewise, the solutions proposed mostly respond to the experience of the authors, however, a detailed analysis of the existing literature is not evidenced in them. In addition, there was also no evidence in the analyzed literature, a detailed analysis about the improvement in productivity achieved through the application of DevOps in the case study companies.

3. DevOps Model

The proposed DevOps Model is composed of four elements, the first one describes four values related to: automation, collaboration, measurement and communication. The second suggests 12 principles adapted from the Agile Manifesto. The third describes 18 practices, 12 fundamental and 6 complementary. Finally, the fourth element proposes 4 dimensions that allow categorizing the proposed practices, the suggested dimensions are: People, Culture, Processes and Tools. The four proposed elements are described below. These elements were proposed following the process: (i) characterize the process elements as well as activities, artifacts, roles, tools, among others, proposed in the literature (see the revision systematic mapping presented in [1]); (ii) design a set of fundamental and complementary practices from the literature found, (iii) analyze and adapt agile values and principles; (iv) establish a set of dimensions to group the practices; (v) establish relationships between: values and principles, principles and practices, and practices and dimensions; and (vi) evaluate the proposed elements and relationships established from a focus group made up of DevOps experts. It is important to mention that the proposal was constantly evaluated through the guidance of the DevOps experts who participated in this project (first and last author).

3.1. Values

Currently, software development companies seek in their processes to accelerate the delivery of value without diminishing the quality of the products / services they develop (value can be defined as the benefit in monetary / financial terms that an organization can receive for the expenses incurred and manage investment), which is why they have begun to implement agile frameworks [3], e.g.: Scrum [47] and Scrumban [48], among others.

Frameworks in which we find agile values as a key element, which represent the main attributes that a software development process must have to be considered agile according to what is defined in the Agile Manifesto [49], following this structure; and with the objective of promoting and informing the importance of DevOps, four (4) values are proposed for DevOps; which are based on the agile values proposed in [49] and can be considered as the natural evolution of these according to the current market needs 20 years after their definition.

The first value is related to *automation* (A); (i) it is essential to prioritize the automation of processes over the realization of manual processes whenever this results in improvements to the work team and task optimization, in short; everything that can be automated must be automated. The second value is related to *collaboration* (C); (ii) teamwork is key, although there are different roles or work areas, any action taken by one individual will benefit or affect another. The third value is related to *measurement* (M); (iii) continuous improvement is only achievable when continuous data is generated that allow better decisions to be made. The fourth and last value is related to *communication* (Co); (iv) communication and transparency is required at all levels of the organization and with the client.

3.2. Principles

The principles are the second element defined in the DevOps Model, they establish the general rules or standards that should guide the actions of software organizations within the framework of a software development process that includes the use of DevOps. The definition of the proposed principles for DevOps are based on the principles defined in the Agile Manifesto [49]. Table 1 shows the principles proposed for the DevOps Model which adapts and adopts the agile principles defined in [49]. Although most of the principles were rewritten, it is possible to observe the exception in principles 1 and 7; since for DevOps the construction of functional software that satisfies the needs of the client with increasingly early and continuous deliveries; it remains the main measure of progress.

Table 1. Principles proposed for the DevOps Model

#	DevOps Model’s principle proposed
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2	Any change in the requirements at any stage of development must be known and agreed upon jointly with representatives from all areas of work.
3	Any artifact of code that passes the defined tests must be available in production in the shortest possible time.
4	Those responsible for business, developers, and infrastructure; they work together on a daily basis throughout the project.
5	Projects are developed around motivated individuals. It is important to guarantee the environment and support they need, provide them with tools that allow them to automate their processes and entrust them with the execution of the work [49].
6	The most efficient and effective means of communication should always be sought among all team members.
7	Software that works is the main measure of progress [49].
8	The higher the speed of deliveries in production, the faster the reception of feedback and the possibility of improvement.
9	The team must have the knowledge and skills necessary to adapt to changes quickly.
10	Try to automate as many processes as possible.
11	The skills and vision for each individual in the work team must be harnessed for the construction of better architectures, requirements and designs.
12	All learning must be documented and communicated. This allows you to adjust and refine your team's actions.

3.3. Dimensions

Dimensions are the elements in the DevOps Model that characterize each of the activities, roles, products, practices, and tools required for the effective implementation of the principles and values proposed for DevOps. The definition of the dimensions takes into account the classifications proposed in other works as well as [42], [27] and [33]. Next, Table 2 presents a dimension identifier, the name of the dimension, its description, and the associated symbology for its association with other elements of the proposed model.

Table 2. Proposed DevOps dimensions

#	Dimension	Description
1	Tools (T)	This dimension relates the technological tools that support the processes, practices, tasks, or products developed within the framework of software development in all its phases under a DevOps approach.
2	Processes (Pr)	This dimension relates the entire set of policies, organizational structures, procedures, purposes, objectives, and work products required throughout the life cycle of a software product that must be considered to the adoption of DevOps.
3	Culture (C)	This dimension takes account aspects related to knowledge, ideas, traditions, and customs that characterize an organization, specifically an organization dedicated to the development of products in the software industry.
4	People (Pe)	This dimension refers to the entire set of characteristics at the level of the individuals that make up a work team.

3.4. Practices

Table 3 presents the 18 practices suggests by DevOps Model, of which 12 are proposed as essential, i.e.: their implementation is required to comply with most of the principles defined in the model. It is possible to define a practice as the performance of an activity continuously to increase engineering skill according to some rules. In addition, it suggests 6 additional practices that are presented as complementary (can be recognized by having an “*” at the end of their name), they require a greater investment on the part of the organizations. As can be noted, each of these practices is grouped into one or more of the proposed DevOps dimensions, in which many of them are related to *people* (Pe) dimension (16 related practices), which suggests that a development process based on DevOps must be people centric. Second are the dimensions of *tools* (T) and *processes* (Pr), with a total of 15 each, reflecting the importance of processes and tools. Finally, the *culture* (C) dimension is linked to 7 practices. Due to space limitations, information such as: purpose, specific objectives and expected artifacts can be consulted in greater detail at: https://bit.ly/36bWjzH.

Additional to that, using the GQM approach [50], it was possible to establish objectives and from these define and establish questions and metrics to support the evaluation of fundamental practices. Questions and metrics associated with the 12 fundamental practices proposed in the DevOps Model can be consulted in https://bit.ly/3x7nlqL and https://bit.ly/3JhsQpc, respectively.

Table 3. Relationship between practices and dimensions

Id.	Name	Description	Dimensions			
			T	Pr	C	Pe
1	Continuous Integration (CI)	Software development practice that requires developers to embed code in a shared repository many times a day [24]. Each new version of the project is then verified using an automated build, allowing teams to detect bugs more easily and quickly [24].	+	+		+
2	Continuous Delivery (CD)	Practice that allows developed source code that has passed all validation tests to be ready for a pre-production environment as soon as possible.	+	+		+
3	Continuous Testing (CT)	This practice incorporates continuous, pre-programmed, and automated code testing as the application code is written or updated. These tests can speed the delivery of code to production.	+	+		+
4	Requirements Management (RM)	Practice related to requirements management in tools to make it easier to track, test, analyse, visualize, and communicate project requirements to stakeholders.		+	+	+
5	Data Management (DM)	Practice related to selecting, obtaining, maintaining, and using data safely and efficiently.	+	+		+
6	Security Monitoring (SM)	Practice related to integrating testing and security controls into daily QA, operations, and development work.	+	+		+
7	Strategic Direction (SD)	Practice in charge of mobilizing people and resources towards the use of DevOps environments, making things happen without overvaluing the benefits that the tools can offer to work teams, generating action within the organization.		+		+
8	Configuration Management (CM)	The practice of controlling and managing changes to software using version control tool in a standard and repeatable way.	+	+		+
9	Continuous Monitoring and Observability (CMO)	The practice of proactively monitoring, alerting, and acting in key areas to provide teams with visibility into application health throughout the lifecycle of a software product.	+	+		+

Id.	Name	Description	Dimensions			
			T	Pr	C	Pe
10	Education around DevOps (EaD)	Practice that drives better software delivery and organizational performance.		+	+	
11	Continuous feedback and innovation (CFI)	Practice in charge of permanently maintaining the socialization and / or documentation of the work team's learning.			+	+
12	Measurement of Culture (MC)	Practice that allows to measure the capabilities that drive the delivery of software and organizational performance to determine improvement actions and continuous improvement.		+	+	
13	Continuous Deployment (CD)*	Practice that strives to automate the deployment of software to production environments with minimal human intervention.	+	+		+
14	Infrastructure as Code (IaC)*	Practice in charge of automating the provisioning of the infrastructure necessary for the construction of a software product at any stage, using descriptive or high-level languages to code more versatile and adaptable implementation and provisioning processes.	+			+
15	Privileged Access Management (PAM)*	A practice that integrates policies, processes, and tools to protect, manage, and monitor access, users, and privileged credentials.	+	+		+
16	Continuous Learning (CL)*	Practice that facilitates learning new skills, cross-team collaboration, systems thinking and knowledge continuously.		+	+	+
17	Continuous Experimentation (CE)*	Practice in charge of motivating the participation and generation of new ideas independently, writing and changing specifications during development, being willing to experiment.		+	+	+
18	Work Satisfaction (WS)*	Positive emotional state of the staff regarding the company and the work carried out in it.			+	+

Acronyms used: T (Tools), Pr (Processes), C (Culture), Pe (People), more information about the proposed dimensions can be consulted in section 3.1.

3.5. Relationship between values, principles, and practices

The relationship between the proposed DevOps fundamentals practices, principles and values is presented table 4, which shows the relationship of each of the suggested values and their relationship with each of the twelve proposed principles. Likewise, Table 4 also shows the relationship of DevOps principles and the fundamental practices that embody them.

Table 4. Relationships between values-principles and principles-practices

Values				DP	Practices																	
1	2	3	4		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	C	M	Co		CI	CD	CT	RM	DM	SM	SD	CM	CMO	EaD	CFI	MC	CD	IaC	PAM	CL	CE	WS
^			^	**1**	+	+	+		+	+		+	+				+	+	+			
		^	^	**2**	+	+	+	+	+	+	+	+	+	+		+	+		+	+	+	
^				**3**				+						+	+	+				+	+	+
	^		^	**4**	+	+	+	+	+	+	+	+	+		+		+	+	+	+	+	+
^	^			**5**										+	+	+				+	+	+
			^	**6**				+	+			+	+									
		^		**7**	+	+	+								+		+	+				
^	^		^	**8**	+	+	+						+		+		+	+				
	^			**9**				+	+			+	+	+	+					+	+	+
^				**10**	+	+	+	+	+			+					+	+				
	^			**11**				+	+		+				+					+	+	+
		^	^	**12**							+				+					+		

Acronyms used: A (Automation), C (Collaboration), M (Measurement), Co (Communication), DP (DevOps Principles), the acronyms for the practices can be consulted in Table 3.

4. Case Study Evaluation through

The evaluation of the DevOps Model was carried out through its application in a software development organization as a case study, for this, the guidelines established for carrying out case studies in [51] were followed. The following subsections describe the case study in terms of its: design, study subject, execution of the evaluation and data collection, analysis of results, validity analysis and study limitations.

4.1. Design

In order to conduct the case study, the following research question was defined: *Is the DevOps model suitable (useful and practical) to support the evaluation of software enterprises processes based on the elements it suggests?* From this research question, the following research sub-questions were defined: (i) Does the DevOps Model contain the necessary elements to evaluate the DevOps approach? (ii) Is the effort invested by a software company for the application of the DevOps Model suitable? and (iii) Does the proposed solution allow software companies to obtain reliable information that serves as a basis to improve the implementation of DevOps in their processes? With these questions, we sought to discover if the DevOps Model adequately fulfills its main function, if it is of practical use and if it is adequate according to the reality of companies that use a DevOps approach. In addition, taking into account the classification presented in [51], the design of this case study was holistic with a unit of analysis, which corresponded to the application of the proposed model in an organization dedicated to software development. The measures used in the study were: the opportunities for improvement identified through the DevOps Model and the metrics defined from the elements defined in it, the effort required to generate a diagnosis using the DevOps Model and the defined metrics, and the benefits perceived by the organization in relation to the DevOps Model.

4.2. Subjects and analysis units

For the case study, a software development company was selected, whose name will not be revealed for reasons of confidentiality, but it can be indicated that it is international in scope (LATAM), it has 1.300 employees and 32 years of experience in the industry. To date, its key focus areas are on the production and maintenance of services in the accounting area and as a provider of electronic invoicing services. Regarding the models and/or standards implemented in the organization, the use of agile practices of the Scrum framework, the ShapeUp methodology and the adaptation of the Spotify tribes model stand out. The criteria that were taken into account on their selection were: (i) a software development organization, (ii) the organization has practices associated with DevOps previously implemented and (iii) the organization is interested in evaluating and finding opportunities for improvement in its processes and aspects related to DevOps.

The case study was made up of the following participants: a research team made up of three people and the company's Vice President of Engineering (VPE). During the execution of the case study a total of seven people, selected from the case company because from his role in the company he is related to activities related to the management, configuration and use of automated processes. Regarding the research team, it was organized as follows: an advisor, who supervised the preparation of: (i) the evaluation instruments and (ii) how to analyze and summarize the information produced by them, and two researchers, who oversaw reviewing the quality of the work, as well as reviewing the technical reports and the scientific method applied in the case study. The advisor will also carry out the evaluation in the case study company, as well as provide the necessary advice.

4.3. Field procedure and data collection

The field procedure used was PvalCOMPETISOFT presented in [52], which allowed the collection of information during the execution of the case study. The activities carried out focused mainly on: (i) evaluation planning, (ii) evaluation execution and data collection and (iii) generation and reporting of results. A detailed description of the activities carried out is presented in the following sections.

4.3.1. Evaluation planning

In order to plan realization of the case study, a first telephone contact was established with the VPE of the organization in order to expose the objective of the study, the activities to be carried out, clarify doubts, characterize the organization and set a date to hold the meeting to start the study. After this first contact, a date and time was defined to hold a meeting remotely and synchronously using the Google Meet tool, which lasted one hour and was developed with the following agenda: presentation of the project and the research team, general presentation of the DevOps Model reference model, general presentation of the DevOps Assessment Model, assignment of roles, evaluation schedule and delivery of the evaluation instrument and supporting documentation.

4.3.2. Evaluation execution and data collection

The advisor selected 7 collaborators representing different key areas in the process of adopting DevOps practices in the company, the roles selected by the evaluator according to the positions of the company were the following:

Vice President of Engineering (VP of Engineering), Technical Lead (Tech Lead), Site Reliability Engineering (SRE), Site Reliability Engineering Tech Lead (SRE Tech Lead), Technical Lead Architect (Tech Lead Arq), Quality Assurance Automation (QA Automation) and DevOps Engineer. The selection of several participants in the assessment process was carried out with the objective of making assessments from different points of view, as well as identifying individual and general perceptions about the processes of adopting DevOps practices and for which the companies are responsible. selected people. Then, the questions of the assessment instrument provided in Google form format were answered, which allowed each question to be answered Yes or No depending on the individual perception of compliance or not with the activity, likewise, there was a space to add comments. The approximate time required for the evaluation was variable for each of the collaborators involved. Table 6 shows a summary of the times invested by each role.

Table 5 shows the distribution and percentage of the degree of individual and general compliance with each of the twelve fundamental practices proposed in the DevOps Model, as well as the value of the corresponding degree of compliance according to the scale defined based on the standard ISO/IEC 15504 since this supports the process according to the qualification of the process elements. Therefore, values are expressed on a discrete scale as follows: NA for Not Achieved (range 0% to 15%), PA for Partially Achieved (range 16% to 50%), LA for Largely Achieved (range 51% to 85%) and FA for Fully Achieved (range 86% to 100%). As can be seen in the evaluation carried out, the twelve fundamental practices evaluated obtained a high level of compliance in the organization, with the practices: Continuous Testing (CT) and the practice related to Education around DevOps (EaD) (see practice 10 in Table 3) being the lowest. compliance, with 67% each. The practice continuous delivery (CE) was the one with the highest degree of compliance, with a completely achieved (100%). In Table 5 it is also possible to observe the degree of general compliance with each of the principles previously defined in Table 1 and that are part of the DevOps Model, it can be observed that each of them is above 80%, which defines a degree of compliance Largely Achieved (LA) according to the defined scale. Principles 7, 8 and 12 reached a degree of compliance Fully Achieved (FA) by exceeding the threshold of 86%.

Table 5. Degree of compliance with fundamental practices and principles obtained in the case study

#	Fundamental practice	Val C1	Val C2	Val C3	Val C4	Val C5	Val C6	TPa	DoC	Principles	
										#	TPr
1	Continuous Integration (CI)	100	100	86	86	71	86	88	FA	1	82.1
2	Continuous Delivery (CD)	100	100	100	100	100	100	100	FA	2	81.4
3	Continuous Testing (CT)	88	88	100	63	13	50	67	LA	3	82.3
4	Requirements Management (RM)	100	100	100	100	50	75	88	FA	4	83.7
5	Data Management (DM)	88	75	75	75	75	75	77	LA	5	80.6
6	Security Monitoring (SM)	83	100	83	100	100	67	89	FA	6	82.1
7	Strategic Direction (SD)	100	75	100	100	50	75	83	LA	7	86.6
8	Configuration Management (CM)	100	100	50	75	50	50	71	LA	8	86.0
9	Continuous Monitoring and Observability (CMO)	67	100	100	83	83	67	83	LA	9	79.5
10	Education around DevOps (EaD)	100	100	100	0	50	50	67	LA	10	81.7
11	Continuous feedback and innovation (CFI)	100	100	100	100	100	50	92	FA	11	84.9
12	Measurement of Culture (MC)	100	100	100	67	100	33	83	LA	12	87.5

Acronyms used: **ValC#**: Valuation registered by collaborator; **TPa**: Average valuation of the company by practice; **DoC**: Degree of Compliance; **TPr:** Average valuation of the company by principle. The values are given in percentage.

Figure 1 shows the degree of general compliance with each of the dimensions defined in the DevOps Model. It can be seen as the four dimensions (tools, process, culture, and people) are at a LA level, with the People dimension being the one with the highest percentage of compliance with 83.7% and the dimension of processes with the lowest percentage of compliance with 81.4%. Figure 1 also shows each of the values defined in the DevOps Model and the degree of compliance achieved by the company according to the data recorded. It can be seen as all the values have a LA degree of compliance, where the Measurement in DevOps value has the highest compliance percentage with 85% and the Automation and Collaboration values have the lowest compliance percentages with an 83%.

Based on the data obtained and applying the respective metrics (questions and metrics associated with the 12 fundamental practices proposed in the DevOps Model can be consulted in https://bit.ly/3x7nlqL and https://bit.ly/3JhsQpc, respectively), a percentage of 84% was obtained in the degree of implementation of DevOps according to the elements defined in the DevOps Model, which categorizes the case study company to a widely reached degree. In addition, as part of the work product called the evaluation report, 8 opportunities for improvement were identified regarding the fundamental practices evaluated based on the DevOps Model, these can be consulted through the following link: https://bit.ly/3uRTTn2. The results obtained allow the company to carry out the definition of a general improvement plan that allows improving its set of DevOps implementation practices.

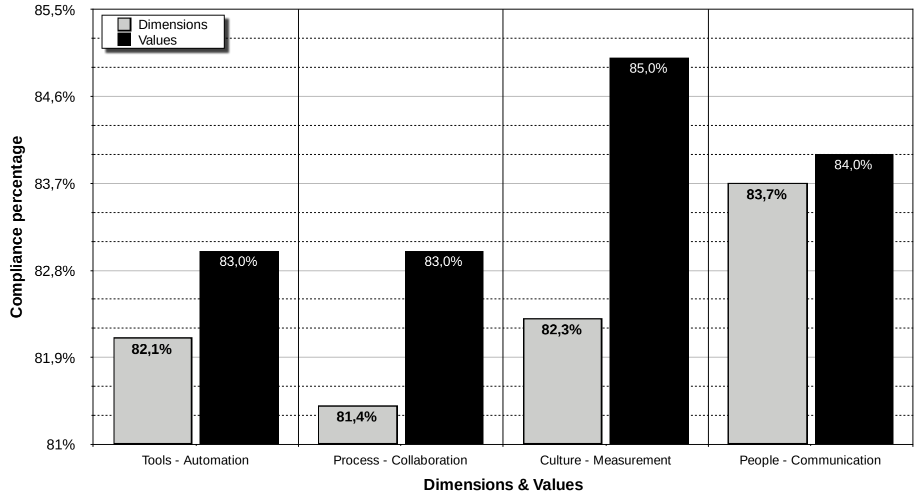

Figure 1. Degree of compliance with DevOps dimensions and values

4.3.3. Analysis of results

4.3.3.1. Effort and cost

Table 6 presents the effort and cost related to the execution of the case study, considering the roles involved in the process and the time dedicated to each of the activities and roles performed during the DevOps Model assessment process carried out in the company. The activities considered to calculate the effort were the following: (1) planning of the assessment, (2) execution of the assessment, (3) activity of clarifying doubts and (4) activity of generation and socialization of results. The roles were Vice President of Engineering (VP of Engineering), Technical Lead (Tech Lead), Site Reliability Engineering (SRE), Site Reliability Engineering Tech Lead (SRE Tech Lead), Technical Lead Architect (Tech Lead Arq), Quality Assurance Automation (QA Automation) and DevOps Engineer. Regarding the costs, the average values per hour in US dollars of the roles involved were used as a reference according to the rates defined in [53]. Likewise, the number of minutes used to carry out the activities was considered. Considering in this case, from Table 6 it can be thought that the assessment process carried out is not expensive and is applicable to both small and large companies that want to start building DevOps adoption processes and/or want to assess the current state of its processes to subsequently start the construction of improvement processes with greater complexity. The calculated cost is related to the evaluation only of the fundamental practices, dimensions, values and principles defined in the DevOps Model, according to this, the effort and cost of applying the DevOps Model is relatively low given the value that this generates for the organization, this is partly due to factors such as: (i) the valuation instrument can be filled out quickly if you have the required knowledge and information and (ii) any negative effect or considerable overload in the daily activities of the people involved. Additionally, metric results are quickly calculated from the generated spreadsheet.

Table 6. Effort and cost of the assessment process in case studies

Roles	**Time spent on scheduled activities measured in minutes**					Average hourly cost per role (US dollars) according to [53]	Total cost per role (US dollars)
	Assessment planning	Assessment execution	Clarification of doubts	Generation and socialization of results	Total effort by role		
VP of Engineering	55	10	2	20	82	72.13	98.57
Tech Lead	0	7	0	0	7	19.74	2.30
SRE	0	7	0	0	7	40.15	4.68
SRE Tech Lead	0	20	0	0	20	40.15	13.38
Tech Lead Arq	0	7	0	0	7	72.36	8.44
QA Automation	0	90	0	0	90	25.94	38.91
DevOps Engineer	55	120	0	0	175	47.69	139.1
Total	110	261	2	20	388 minutes		305.38

Once the process of generating and socializing results was completed, a survey was conducted with the participant who assumed the role of evaluator, to know how their general perception of the process carried out regarding the understandability, usefulness and practicality of the DevOps Model and DevOps Assessment. Model. For the information capture process, there was a rapporteur who was the person in charge of taking note of the observations and comments made by the focus group participants. In addition, as a complement to the observations made by the participants, they were asked to answer a questionnaire presented in Table 7 at the end of the session to discuss the proposal. As can be seen in Table 6, the survey was structured in two parts as follows: the first part of the questionnaire allowed collecting information from each participant through the questionnaire, which was focused on knowing the opinion of the participants about the proposal. The survey consisted of fifteen questions organized as follows: first 13 questions were designed to be answered using a 5-point Likert scale: Very Unsatisfied (VEU): Value (1); Little Satisfied (LIS): Value (2); Something Satisfied (SOS): Value (3); Satisfied (SAT): value (4); Very Satisfied (VES): value (5). The questions asked made it possible to measure the comprehensibility (4 questions), applicability (1 question), suitability (3 questions) and completeness (2 questions) of the proposed model. Questions were also defined that measured two variables at the same time, which were: suitability and completeness (1 question) and suitability and applicability (2 questions). Finally, 2 open questions were included that allowed participants to propose adjustments to the model and make additional comments.

Table 7. Evaluation questionnaire

Aspect	Id.	Question
Understandability	Q1	Do you find the proposed DevOps values easy to understand?
	Q2	Do you find the proposed DevOps principles easy to understand?
	Q3	Do you consider that the proposed dimensions are easy to understand?
	Q4	Do you consider that the proposed practices are easy to understand?
Applicability	Q5	Based on your experience: Do you consider that the elements defined in the reference model are appropriate and can be applied successfully in a software development project?
Suitability	Q6	Do you think the proposed values capture the core essence of DevOps?
	Q7	Based on your experience: Do you consider that the principles defined for DevOps are appropriate and do they correctly interpret the current dynamics of the Software industry?
Completeness	Q8	Based on your experience: Do you consider that the principles defined for DevOps are sufficient?
Suitability/ Completeness	Q9	Do you think that the dimensions defined for DevOps correctly group the defined principles and practices?
Suitability/ Applicability	Q10	Do you consider that the proposed fundamental practices are appropriate and can be successfully applied in a software development process?
Suitability/ Applicability	Q11	Do you consider that the proposed complementary practices are appropriate and can be successfully applied in a software development process?
Completeness	Q12	Do you consider that the proposed practices are sufficient for a DevOps adoption process?
Suitability	Q13	Do you consider that the proposed model is useful as a reference for improvement projects that aim to define processes related to DevOps?
		Open questions
Q14		Do you consider that elements (values, principles, dimensions, or practices) of the proposal should be added, eliminated, or modified?
Q15		Do you have any additional comments about the proposed reference model?

Once the focus group was held, the analysis of the contributions made by the participants in the debate session of the proposal and the results of the evaluation questionnaire filled out at the end of the session was carried out. Figure 2 shows the consolidated results for questions 1-13. According to the previous results, it can be observed that the participants had a favorable opinion about the understandability, applicability, suitability, and completeness of DevOps Model. However, with questions 4, 5, 7, 8, 10, 11 and 13, favorable responses were not evidenced by some participants, and they were considered to carry out improvement actions on the proposed model. Questions P14 and P15 allowed participants to propose adjustments to the model and make additional changes.

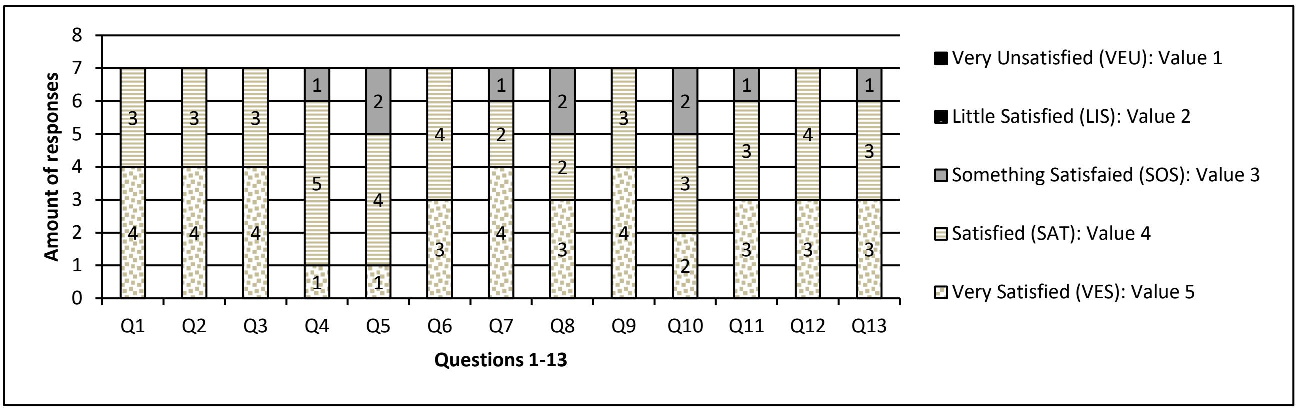

Figure 2. Consolidated answers to questions p1-p13

The opinions and suggestions were considered to make *improvements on the proposal*, these were related to: (i) it was suggested that some complementary practices such as *continuous deployment* become fundamental, however, this change was not made because this practice requires adopting automation tools and activities, which require a higher level of training so that minimal human intervention occurs for the deployment process. For this reason, only the practice of delivery and continuous integration is specified as a fundamental practice, which allows obtaining good results in the deployment in a manual and non-automated manner; (ii) add a practice of continuous tests, this practice was added indicating when to apply this but leaving the freedom of its definition and detail to the companies; (iii) It is suggested that the "security supervision" practice be complementary, however, this suggestion was not adopted due to the fact that DevOps favors the implementation of highly productive environments and due to the increase in the speed of putting a product into production or service cannot be skimped on security aspects; and (iv) add a practice related to shift left, this change only required a little more detail on the practice of continuous tests, it is the company that must decide how to implement its operational processes incorporating practices such as shift left or shift right.

4.4. Validity analysis

Different elements have been considered to better manage the threats and allow the validity plan to be carried out, for example, both the design and the data collection plan were carried out through checklists proposed especially for case studies in Software Engineering, see [54]. Construct validity, internal and external validity, and reliability are presented below:

- *Construct validity*: In order to maintain the validity of the construct and its traceability, different resources were used to manage the evidence that was generated, e.g.: surveys, interviews and documentation. The evidence was obtained from the meetings held, which were carried out in the activities described in the field procedure. Likewise, to preserve the validity of the construct so that the measurements obtained in the study represent the aspects that are really to be measured and are consistent with the research questions, an initial meeting was held where the project was contextualized, explaining the model of reference and the valuation method, the work plan for the case study and the existing concerns were resolved. Additionally, detailed supporting documentation was provided for the evaluator so that he could delve into the proposal and answer any questions that might arise along the way.
- *External validity*: It is not possible to generalize the results of the case study, since it was only carried out in a company with characteristics that may be particular to the business model and position in the Latin American market with respect to other companies. However, it is possible to replicate this study in software organizations, as long as the suggested selection criteria are followed: software company that uses processes related to DevOps, that uses the established field procedure and data collection process.
- *Reliability*: To achieve independence between the results obtained and the researchers who carried out the case study, the following activities were defined: (i) definition of the criteria for the selection of the

study subject, (ii) definition of the field procedure and data collection, (iii) elaboration of detailed documentation of the reference model and its assessment model to be used by the organization in the case study, and (iv) permanent support to the evaluator and case study participants during the assessment. execution phase.

4.5. Study limitations

The limitations found in carrying out this case study are related to: (i) due to the application of the case study in a single software development company, the ability to generalize the results obtained is impossible, for which it is done its application in more software development companies with diverse characteristics is necessary, and (ii) there is a possible bias in carrying out the case study regarding the subjectivity of the evaluator in the application of the assessment instrument and the possible subjectivity in the interpretation of the results by the research team.

5. Conclusions and future work

This paper presented a reference model to clarify, support, and encourage the implementation of DevOps-related values, principles, dimensions, and practices in the software development industry. The presented model aims to provide a clearer, more documented, and detailed view of the elements that must be considered to adopt DevOps. The elements defined in DevOps Model emerged from a systematic review of the literature presented in [1]. Likewise, the reference model is flexible, so that it can be adapted and enables the creation of processes and indicators according to the characteristics and needs of organizations related to the software industry that wish to advance DevOps adoption processes.

DevOps model has been proposed with the aim of reducing the ambiguity and misunderstood that could arise during the implementation of DevOps in software organizations, and therefore, facilitate its understanding, adoption, evaluation thanks to the detail that has been incorporated in each of the proposed elements and formalizes the appropriation of concepts. Likewise, a close relationship between agility and the proposed elements has been incorporated through the definition of a set of values and principles for DevOps. Likewise, it clarifies which practices contribute to a correct and clearer implementation of DevOps, describing in detail for each of them attributes such as: name, description, purpose, specific objectives and expected artifacts.

Has been carried out the application of DevOps Model in a software development company as case study, where from the results obtained it has been possible to observe that its application was satisfactory. The application was carried out by means of PvalCOMPETISOFT, which allows to clearly establish the field procedure regarding the activities and steps to be considered during the evaluation and identification of improvement opportunities to be considered when applying our model in the case study company. During the case study, a perception instrument focused on knowing the opinion of DevOps experts regarding the understandability, applicability, suitability, and completeness of the DevOps Model was applied. Likewise, the instrument allowed collecting opinions that were taken as opportunities for improvement. Based on the suggestions and comments of the participants, the adjustment, improvement and generation of a new version of DevOps Model was carried out, which is presented in this paper. On the other hand, it was possible to show that in a generalized way there was a very good acceptance of DevOps Model by the participants, who coincide in recognizing the value and importance of the proposed model and recommend using it as a reference in the software industry. They also suggest that these types of solutions in the DevOps field should be maintained to better guide efforts related to the adoption of DevOps in the agile / traditional development industry.

Part of the future work will focus on extending the reference model for DevOps through the incorporation of a set of questions based on the specific objectives of the proposed practices and the definition of possible metrics that allow measuring the level of implementation / adoption of DevOps in a software company. This will expand the capabilities of the DevOps Model, as well as enable continuous process improvement in the industry regarding DevOps. As part of the extension, it is expected to incorporate companies as case studies that allow to continue adjusting, refining and validating the proposal.

Declaration of competing interest

The authors declare that they have no any known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

Acknowledgements

We thank the anonymous reviewers. Likewise, César Pardo acknowledges the contribution of the Universidad del Cauca, where he works as an associate professor.

References

[1] J. Guerrero, K. Zuñiga, C. Certuche, and C. Pardo, "A systematic mapping study about DevOps," *Journal de Ciencia e Ingeniería*, vol. 12, no. 1, pp. 48–62, 2020, doi: https://doi.org/10.46571/JCI.2020.1.5.

[2] V. P. Munteanu and P. Dragos, "The Case for Agile Methodologies against Traditional Ones in Financial Software Projects," *European Journal of Business and Management Research*, vol. 6, no. 1 SE-Articles, Feb. 2021, doi: 10.24018/ejbmr.2021.6.1.741.

[3] VersionOne Inc. & Collabnet, "CollabNet & VersionOne," *15th Annual State Of Agile Report*, 2021. https://bit.ly/3x9cTLO (accessed Jul. 30, 2021).

[4] H. Takeuchi and I. Nonaka, "The New New Product Development Game," *Harvard Business Review*, vol. 64, no. 1, pp. 1–16, 1986.

[5] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed. Boston: Addison-Wesley Professional, 2004.

[6] A. Cockburn, *Crystal Clear: A Human-Powered Methodology for Small Teams*, 1st ed. Boston: Pearson Education, 2004.

[7] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*. USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[8] J. A. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, vol. 12. New York, NY: Dorset House Publishing editors, 2000.

[9] J. Stapleton, *DSDM: Dynamic Systems Development Method: The Method in Practice*, 1st ed. Addison-Wesley Professional, 1997.

[10] S. R. Palmer and J. Mac Felsing, *A Practical Guide to Feature Driven Development*, 1st ed. Prentice Hall, 2002.

[11] Scott W. Ambler, "The Agile Unified Process (AUP)," 2021. https://bit.ly/3hsgSx3 (accessed Jul. 02, 2021).

[12] D. J. Anderson and A. Carmichael, *Essential Kanban Condensed*, 1st ed. Seattle, Washington: Blue Hole Press, 2017.

[13] Davor Gornik, "IBM Rational Unified Process: Best Practices for Software Development Teams. TP026B, Rev 11/01," Somers, New York, 2020. [Online]. Available: https://ibm.co/3haHe80.

[14] G. Lory, D. Campbell, A. Robin, G. Simmons, and P. Rytkonen, "Microsoft Solutions Framework v3 Overview," 2003. [Online]. Available: http://www.microsoft.com/msf.

[15] CMMI Institute, "CMMI V2.0 model," 2020. [Online]. Available: https://cmmiinstitute.com/cmmi.

[16] B. Boehm, D. Port, A. Egyed, Abi-Antoun, and M. T., "The MBASE Life Cycle Architecture Milestone Package," in *Software Architecture: TC2 First Working IFIP Conference on Software Architecture (WICSA1)*, 1999, pp. 511–528.

[17] A. Hochstein, R. Zarnekow, and W. Brenner, "ITIL as common practice reference model for IT service management: formal assessment and implications for practice," in *IEEE International Conference on e-Technology, e-Commerce and e-Service*, 2005, pp. 704-710.

[18] G. Ridley, J. Young, and P. Carroll, "COBIT and its utilization: a framework from the literature," in *37th Annual Hawaii International Conference on System Sciences*, 2014, pp. 1–8.

[19] ISO/IEC, "ISO/IEC 20000:2018 Information Technology," London, United Kingdom, 2021. [Online]. Available: https://bit.ly/3qJqwzR.

[20] M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere, "Composable DevOps: Automated Ontology Based DevOps Maturity Analysis," in *IEEE International Conference on Services Computing*, Jun. 2015, pp. 600–607, doi: 10.1109/SCC.2015.87.

[21] G. de França, B., Jeronimo, H., & Travassos, "Characterizing DevOps by Hearing Multiple Voices," in *Proceedings of the 30th Brazilian Symposium on Software Engineering - SBES '16*, 2016, pp. 53–62,

doi: 10.1145/2973839.2973845.

[22] H. Senapathi, M., Buchan, J., & Osman, "DevOps Capabilities, Practices, and Challenges," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 - EASE'18*, 2018, pp. 57–67, doi: 10.1145/3210459.3210465.

[23] T. Riungu, L; Mäkinen, S; Lwakatare, L; Tiihonen, J & Männistö, "DevOps Adoption Benefits and Challenges in Practice: A Case Study," in *Product-Focused Software Process Improvement*, 2016, pp. 590–597, doi: 10.1007/978-3-319-49094-6_44.

[24] A. Ghantous, G.B & Gill, "DevOps: Concepts, Practices, Tools, Benefits and Challenges," in *Pacific Asia Conference on Information Systems (PACIS)*, 2017, pp. 1–12.

[25] S. S. Samarawickrama and I. Perera, "Continuous scrum: A framework to enhance scrum with DevOps," in *7th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Sep. 2017, pp. 19–25, doi: 10.1109/ICTER.2017.8257808.

[26] J. Kamuto, M., & Langerman, "Factors inhibiting the adoption of DevOps in large organisations: South African context," in *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, May 2017, pp. 48–51, doi: 10.1109/RTEICT.2017.8256556.

[27] I. Elberzhager, F., Arif, T., Naab, M., & Süß, "From Agile Development to DevOps: Going Towards Faster Releases at High Quality – Experiences from an Industrial Context," in *Software Quality. Complexity and Challenges of Software Engineering in Emerging Technologies - SWQD*, 2017, pp. 33–44, doi: 10.1007/978-3-319-49421-0_3.

[28] M. Krusche, S., Alperowitz, L., Bruegge, B., & Wagner, "Rugby: an agile process model based on continuous delivery," in *1st International Workshop on Rapid Continuous Software Engineering - RCoSE*, 2014, pp. 42–50, doi: 10.1145/2593812.2593818.

[29] L. Klepper, S., Krusche, S., Peters, S., Bruegge, B., & Alperowitz, "Introducing Continuous Delivery of Mobile Apps in a Corporate Environment: A Case Study," in *2nd International Workshop on Rapid Continuous Software Engineering*, May 2015, pp. 5–11, doi: 10.1109/RCoSE.2015.9.

[30] M. Benguria, G., Alonso, J., Etxaniz, I., Orue-Echevarria, L., & Escalante, "Agile Development and Operation of Complex Systems in Multi-technology and Multi-company Environments: Following a DevOps Approach," in *Systems, Software and Services Process Improvement - EuroSPI*, 2018, pp. 15–27, doi: 10.1007/978-3-319-97925-0_2.

[31] L. E. Lwakatare *et al.*, "DevOps in practice: A multiple case study of five companies," *Information and Software Technology*, vol. 114, no. 1, pp. 217–230, Oct. 2019, doi: 10.1016/j.infsof.2019.06.010.

[32] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A Survey of DevOps Concepts and Challenges," *ACM Comput. Surv.*, vol. 52, no. 6, Nov. 2019, doi: 10.1145/3359981.

[33] I. Bucena and M. Kirikova, "Simplifying the DevOps Adoption Process," in *16th International Conference on Perspectives in Business Informatics Research*, 2017, pp. 1–15.

[34] M. Erich, F; Amrit, C & Daneva, "A qualitative study of DevOps usage in practice," *Journal of Software: Evolution and Process*, vol. 29, no. 6, p. e1885, Jun. 2017, doi: 10.1002/smr.1885.

[35] Floris Erich, C. Amrit, and M. Daneva, "Report: DevOps Literature Review," Enschede, Netherlands, 2014. doi: 10.13140/2.1.5125.1201.

[36] A. D. Nagarajan and S. J. Overbeek, "A DevOps Implementation Framework for Large Agile-Based Financial Organizations," in *On the Move to Meaningful Internet Systems - OTM*, 2018, pp. 172–188, doi: 10.1007/978-3-030-02610-3_10.

[37] D. Teixeira, R. Pereira, T. A. Henriques, M. Silva, and J. Faustino, "A Systematic Literature Review on DevOps Capabilities and Areas," *International Journal of Human Capital and Information Technology Professionals (IJHCITP)*, vol. 11, no. 2, pp. 1–22, 2020, doi: 10.4018/IJHCITP.2020040101.

[38] M. Sánchez-Gordón and R. Colomo-Palacios, "Characterizing DevOps Culture: A Systematic Literature Review BT - Software Process Improvement and Capability Determination," in *18th Software Process Improvement and Capability Determination - SPICE*, 2018, pp. 3–15.

[39] J. Wettinger, U. Breitenbücher, and F. Leymann, "DevOpSlang – Bridging the Gap between Development and Operations," in *Service-Oriented and Cloud Computing - ESOCC*, 2014, pp. 108–122, doi: 10.1007/978-3-662-44879-3_8.

[40] F. Jones, S., Noppen, J & Lettice, "Management challenges for DevOps adoption within UK SMEs," in *Proceedings of the 2nd International Workshop on Quality-Aware DevOps - QUDOS 2016*, 2016, pp. 7–11, doi: 10.1145/2945408.2945410.

[41] L. Chen, "Continuous Delivery: Overcoming adoption challenges," *Journal of Systems and Software*,

vol. 128, pp. 72–86, Jun. 2017, doi: 10.1016/j.jss.2017.02.013.

[42] L. E. Lwakatare, P. Kuvaja, and M. Oivo, “Dimensions of DevOps,” in *Agile Processes in Software Engineering and Extreme Programming - XP*, 2015, pp. 212–217, doi: 10.1007/978-3-319-18612-2_19.

[43] V. Gupta, P. K. Kapur, and D. Kumar, “Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling,” *Information and Software Technology*, vol. 92, no. 1, pp. 75–91, Dec. 2017, doi: 10.1016/j.infsof.2017.07.010.

[44] B. Jabbari, R; Bin, N; Petersen, K & Tanveer, “What is DevOps?,” in *Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP ’16 Workshops*, 2016, pp. 1–11, doi: 10.1145/2962695.2962707.

[45] J. Cusick, “A Survey of Maturity Models from Nolon to DevOps and Their Applications in Process Improvement,” New York, NY, 2019. [Online]. Available: https://bit.ly/3qIAL7f.

[46] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, “From Agile to DevOps: Smart Skills and Collaborations,” *Information Systems Frontiers*, vol. 22, no. 4, pp. 927–945, 2020, doi: 10.1007/s10796-019-09905-1.

[47] K. Schwaber and J. Sutherland, “The scrum guide the definitive guide to scrum: The rules of the game,” USA, 2020. [Online]. Available: https://bit.ly/3iRzw2d.

[48] C. Ladas, *Scrumban-essays on kanban systems for lean software development*, 1st ed. Seattle: Modus Cooperandi Press, 2009.

[49] K. Beck *et al.*, “Manifesto for Agile Software Development,” *Agile Manifiest*, 2001. https://agilemanifesto.org (accessed Jul. 02, 2021).

[50] V. R. Basili, G. Caldiera, and H. D. Rombach, “The Goal, metric, and question Approach,” Kaiserslautern, Germany, 1994. doi: 10.1002/0471028959.sof142.

[51] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009, doi: 10.1007/s10664-008-9102-8.

[52] F. J. Pino, C. Pardo, F. García, and M. Piattini, “Assessment methodology for software process improvement in small organizations,” *Information Software and Technology*, vol. 52, no. 10, pp. 1044–1061, 2010, doi: http://dx.doi.org/10.1016/j.infsof.2010.04.004.

[53] I. PayScale, “PayScale - Salary Comparison, Salary Survey, Search Wages,” 2021. .

[54] M. Höst and P. Runeson, “Checklists for Software Engineering Case Study Research,” in *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement*, 2007, pp. 479–481, doi: http://dx.doi.org/10.1109/ESEM.2007.29.