ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 13, No. 1, April 2025, pp.159-194

Towards a metrics model for DevOps adoption: Case study of an enterprise application

Carlos Orozco[1*], César Pardo[1], 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

*Corresponding author E-mail: carlosorozco@unicauca.edu.co

ABSTRACT

The widespread use of DevOps in the industry has brought about the need to define mechanisms that allow for evaluation of its adoption and usage in organizations. With the purpose of providing a solution to assess the degree of DevOps adoption in the software industry, this article presents a metrics model to determine the extent to which the proposed practices and values in DevOps are being fulfilled. The metrics model is the result of a systematic literature review and a harmonization process of initiatives proposed by other authors. This led to the definition of 11 metrics following the formalism proposed by the Goal-Question-Metric approach. To extend the validity of the model, this article presents the results of two case studies conducted in two software development companies.

Keywords: Development and Operations, DevOps, Assessment, Appraisal, Evaluation, GQM.

1. Introduction

Currently, companies seek continuous improvement of their processes through the implementation of approaches, frameworks, and/or solutions that guarantee the deployment or production of functional software in increasingly shorter intervals; this ensures high quality standards [1] and alignment with customer needs [2]. Accordingly, companies have made efforts to define proposals that facilitate the standardization and optimization of the software development life cycle (hereafter referred to as Dev) through traditional and agile approaches. However, the software life cycle is complemented by activities related to the management of the operation in Information Technology - IT (hereafter referred to as Ops). The aim is to establish the necessary mechanisms to achieve solution integrity in productive environments, and thus manage business continuity through deployment, monitoring, and traceability of artifacts put into production. However, despite pursuing the same goal, development and operations have long been considered areas with contrasting purposes, interests, and intentions. For example, development focuses on the activities required for the production release of new functionalities through activities such as analysis, design, implementation, testing, among others. Operations focus on defining the set of tasks necessary to facilitate the continuous deployment, integrity, monitoring, and traceability of artifacts when they are in productive environments [3]. With the purpose of integrating the set of best practices proposed by development and operations into an approach with a common goal, in 2009, Debois [4] first suggested the term DevOps, after identifying that software projects faced value loss caused by delays in deployment cycles, lack of shared responsibility, difficulty in change management, and little to no visibility and feedback during the software development life cycle. To mitigate the aforementioned aspects, DevOps seeks to bridge the gap between development and operations [5], improving communication between both areas through the definition of a set of principles, values, and practices, and thus establishing a roadmap that allows organizations to automate the software development life cycle through the management of operations by enhancing tasks related to continuous integration [6], continuous deployment [7], continuous delivery and maintenance [8], change management [9], test automation [10], monitoring/observability [10], among others.

In general, the widespread use of DevOps in the industry has improved aspects related to the quality, productivity, efficiency, and competitiveness of software companies [10], [11]. According to the results report presented by the global survey on the state of agility conducted in 2022 [12], approximately 75% of the participants stated that supporting the development of internal projects in a company by applying DevOps brings benefits in terms of costs, effort, and time [11]. However, the State of Site Reliability and DevOps report conducted in 2022 [13] mentions that companies applying DevOps face challenges related to: (i) lack of hard skills - technical and/or operational knowledge - required for day-to-day work, (ii) little to no understanding of "what" DevOps is and "how" it should be applied, (iii) limited experience in using tools for task automation, and (iv) lack of mechanisms that allow a company to know if they are adopting the culture proposed by DevOps appropriately or if they need to take improvement actions. In this regard, it is observed that implementing DevOps is not a simple task [14]. Although it is true that DevOps seeks to solve specific problems related to defining activities that facilitate automation during the life cycle of solutions, these can only be refined when measured and characterized to determine their degree of compliance. Therefore, companies need practices, processes, knowledge, and tools that clearly quantify the degree and/or level of DevOps practice adoption in their projects. This ensures a starting point to identify improvement opportunities that refine their practices and enhance their internal processes [15], [16].

To establish a clear state of knowledge regarding the definition of proposals for measuring DevOps in the software industry, a systematic literature mapping (SLM) [17] was conducted. As a result of executing the SLM, it was observed that initiatives have been carried out through methodological solutions - capacity models, certification, collaboration, and maturity, among others - and technological tools developed by companies seeking to evaluate the degree or level of implementation of the set of practices, activities, and tasks proposed by DevOps. However, the analysis of the results showed a high degree of heterogeneity and ambiguity in the identified solutions. This is because each author and/or study proposes their solution based on the set of principles, values, activities, and practices considered most relevant according to their criteria. Although each solution pursues the same common goal of assessing the degree of conformity, adoption, capability, and/or maturity of DevOps, the reported results are heterogeneous, with different scopes and perceptions.

Based on the above, this proposal seeks to provide clarity and facilitate the evaluation of DevOps in software companies. Therefore, a metrics model was developed, defined following the Goal, Question, Metric (GQM) approach [18]. The model aims to support the evaluation of process elements related to DevOps. The metrics model is the result of identifying, comparing, and integrating the DevOps process elements identified in the SLM through a harmonization process of multiple models. This harmonization was applied to address the heterogeneity of existing solutions through a systematic approach that determined common elements and relationships among all the solutions identified in the literature. The metrics model organizes its elements around four dimensions: people, culture, technology, and processes, and four values: automation, collaboration, communication, and measurement. As a result, the metrics model clarifies what and how to evaluate compliance with DevOps in the software industry. Based on this, and with the purpose of extending the validity of the metrics model, this article presents the results obtained after applying the proposal at two software development companies as case studies.

Given this lack of standardization in DevOps measurement approaches, this study proposes a structured model to assess the degree of DevOps adoption in software projects. By analyzing existing methodologies and tools, a set of key dimensions that should be considered for a comprehensive evaluation were designed and evaluated. The proposed framework provides organizations with a systematic way to measure their DevOps maturity, identify areas for improvement, and optimize their internal processes.

The structure of the article is as follows: Section 2 presents the analysis of the state of the art regarding the definition of solutions for evaluating DevOps in software companies; Section 3 presents the harmonization process used to obtain the process elements that make up the model. It also presents the protocol carried out for the definition of 11 metrics that assess the degree of compliance with DevOps practices, dimensions, and values; Section 4 describes the application of the proposed metrics model in two case studies. Finally, Section 5 presents the conclusions and future work.

2. Analysis of the state of the art

Based on the results obtained from conducting a systematic literature mapping [17], efforts have been identified in studying and defining solutions to assess DevOps in software companies through tools, models, processes, and techniques. In general, the results can be classified into three types: (i) exploratory studies aimed at determining the elements that should be considered when evaluating DevOps, (ii) methodological solutions (models, processes, techniques, metrics), and (iii) technological tools developed by software companies that provide services focused on determining the degree and/or level of DevOps adoption for specialized consulting purposes.

Regarding exploratory studies, these focus on aspects such as comparative analyses of different tools for evaluating DevOps in small and medium-sized software enterprises [19]; systematic literature mappings to identify the essential process elements that should be considered when defining a DevOps certification model [3], [17], [20], [21]; and exploratory studies seeking to understand the state of the art in defining DevOps maturity models [22], [23], [24], [25]. On the other hand, methodological solutions have been observed, including metrics for evaluating practices related to continuous build, integration, and deployment [26], [27], [28], [29]; competency models [30], [31]; maturity models [22], [23], [24], [25], [32], [33], [34], [35], [36], [37], [38]; collaboration models [30]; certification models [39]; and models based on the principles proposed by agile approaches adapted to DevOps practices [37]. Finally, the results reported in the systematic mapping have made it possible to identify the implementation of technological tools that assess DevOps through surveys [40], [41], [42], [43], [44], [45], methodological guidelines [46], frameworks [47], and tools supported by specialized consulting services [48], [49], [50], [51], [52].

According to the results reported in the systematic literature mapping (SLM), progress has been made in defining exploratory studies, methodological solutions, and tools aimed at assessing DevOps in the industry. However, a high degree of heterogeneity remains in the proposals found in the literature, as there is no consensus regarding the state of knowledge associated with DevOps. In this regard, each company defines its own evaluation criteria based on the set of DevOps practices or elements that it considers most relevant during the project lifecycle. Consequently, various solutions have been proposed to evaluate DevOps through values, principles, activities, practices, roles, and tasks. As a result, in the absence of a generic reference model or standard, each company establishes its own evaluation criteria. While all companies share the same objective—determining DevOps compliance, capability, and/or maturity—there is no general solution that provides organizations with a comprehensive overview of the degree or level of adoption of their DevOps practices.

Given the above, the metric model presented in this study seeks to reduce the heterogeneity among the proposals identified during the state-of-the-art review by providing a structured solution that unifies the terminology used in the literature and industry, based on the studies reported in the SLM. Considering this, this article presents a metric model to support the evaluation of DevOps in the software industry, defining a set of 11 metrics to address the needs identified in the literature. The model's application is demonstrated through two case studies.

3. Harmonization of the fundamental elements proposed by DevOps in software companies

This article presents a metric model that provides a conceptual and technical solution to facilitate the evaluation of the degree, percentage, or level of compliance with DevOps practices. This section introduces a metric model that assesses a total of 12 fundamental practices, 6 complementary practices, 4 dimensions, and 4 values. The practices, dimensions, and values result from executing the following activities: (i) identifying the process elements suggested in the literature for defining a metric model [17], (ii) conducting a harmonization process of multiple models to establish a homogeneous state of knowledge regarding DevOps [53], and (iii) defining 11 metrics to evaluate the adoption level of the previously described practices, dimensions, and values following the protocol proposed by the GQM approach [18]. The following subsection presents the protocol carried out for the harmonization of the practices, dimensions, and values identified during the systematic literature mapping analysis. However, before introducing the proposed solution in this article, a brief introduction is provided to inform the reader about the importance of defining a metric model and its impact on the operational processes of an organization that decides to implement it.

3.1. What is a metric model and why is it important?

A metric model in the context of software engineering is a structured set of measures and criteria used to evaluate and improve the quality and efficiency of software development processes [54]. These metrics enable software engineers to quantify key aspects of development, such as code quality, team efficiency, customer satisfaction, and project management. For example, metrics may include the number of defects per line of code, system response time, test coverage, and the speed of delivering new functionalities. The importance of a metric model lies in its ability to provide an objective and quantifiable view of software development performance. Conducting periodic evaluations of processes within an organization allows teams to identify areas for improvement, make informed decisions, and justify changes in the process. Additionally, metrics help align team objectives with customer expectations and industry standards, ensuring that the final product is of high quality and meets established requirements. In this sense, a metric model facilitates the management and control of software development while driving continuous improvement and excellence in the delivery of technological products.

3.2. Protocol for the harmonization of DevOps process elements

The proposals identified during the analysis of the SLM results classify three main process elements when applying DevOps: (i) practices, (ii) dimensions, and (iii) values. However, the nature of the findings indicated that each study and/or technological tool defines its own process elements according to the specific needs of each company or author. In this regard, a harmonization process was necessary to standardize the process elements present in both the literature and industry within a common structure. In general, the models found in the literature and the technological tools implemented by different companies define their own structures, concepts, and characteristics that differentiate them from the rest. To homogenize and unify knowledge on this topic, a harmonization process was conducted across multiple models, applying the formalism defined in [55], which proposes the following methods: (i) identification, carried out during the state-of-the-art study; (ii) homogenization; (iii) comparison; and (iv) integration. The activities performed in each of these methods are presented in the following subsections.

3.2.1. Homogenization of DevOps elements

The homogenization of solutions enables the unification of general information found in the literature and technological tools within a common structure that groups shared concepts of interest. To carry out the homogenization of the proposals identified in the SLM [17], a process element structure [56] was applied, based on the concepts presented in the ontology proposed in [57]. As a result of applying this process element structure, the practices, dimensions, and values identified in the literature and tools were grouped. Table 1 presents a detailed characterization of the elements associated with the methodological solutions identified in the SLM. The columns labeled “Not Evidenced” represent studies that did not explicitly report practices, dimensions, and/or values. According to the information presented in Table 1, all studies propose practices related to process automation through activities such as integration, deployment, testing, delivery, and continuous monitoring. However, the studies also include aspects related to version control strategies, monitoring and incident management, configuration management, metric definition, communication, empowerment, maintainability, security, and improvements in engineering capabilities. In this regard, the results indicate a strong trend towards making efforts to enhance and/or facilitate project development through automation techniques that streamline the implementation, testing, and delivery of new functional code.

On the other hand, the dimensions presented in Table 1 focus on aspects such as culture, people, processes, technology, and monitoring. However, they also mention activities related to quality assurance, visibility, strategic alignment, and fostering an organizational culture within companies. Finally, the reported values include fundamental concepts such as automation, collaboration, measurement, and communication. Based on the information presented in Table 1, it is evident that all studies align with many of the concepts present in the culture proposed by DevOps. However, each study describes more or fewer concepts, resulting in a body of knowledge that needs to be standardized to ensure alignment across all studies within a common structure.

Table 1. Process elements identified in methodological solutions

No	**Ref**	**Process Elements**		
		Dimensions	**Practices**	**Values**
1	[57]	Culture, monitoring.	Continuous integration, continuous deployment, continuous monitoring, continuous testing, continuous delivery.	Automation, collaboration.
2	[31]	Culture, people, processes, technology.	Not evident.	Automation and tools, agility in value delivery.
3	[23]	Culture, lean practices, continuous delivery, continuous improvement, customer satisfaction, processes, quality, governance, products, foundation.	Sharing knowledge, trust and respect, team organization, delivery alignment, efficient releases, branch and merge, construction automation, development quality improvement, test automation, deployment automation, production releases, incident handling, configuration management, architecture alignments, infrastructure.	Collaboration, automation, communication.
4	[33]	Culture, monitoring.	Not evident.	Collaboration, communication, automation, measurement.
5	[58]	Culture, read, quality assurance.	Continuous integration, continuous delivery, continuous deployment.	Automation, collaboration, measurement.
6	[24]	Processes, quality, people, tools, technology, culture, quality assurance, visibility and reporting, products.	Continuous construction, continuous integration, continuous release.	Collaboration, automation.
7	[21]	Culture, quick feedback.	Continuous delivery, continuous deployment.	Collaboration, automation.
8	[25]	Quality, governance, culture and organization, processes, technology, visibility, strategic alignment.	Not evident.	Collaboration, automation.
9	[35]	Efficient management, culture and work environment, transformation and	Test information management, architecture, version control, code maintainability, monitoring, metrics, automation, continuous testing, security, team	Not evident.

No	Ref	Process Elements		
		Dimensions	**Practices**	**Values**
		leadership, continuous delivery.	empowerment, platforms, continuous integration, development, database change management.	
10	[36]	Culture, processes, products, quality.	Team organization, communication, trust and respect, knowledge transfer, release alignment, change management, test automation, deployment automation, delivery automation, incident handling, configuration management, architecture alignment, infrastructure.	Collaboration.
11	[32]	Culture.	Continuous integration, continuous deployment, continuous monitoring, continuous testing, feedback loops between Dev and Ops, infrastructure as code, change management, continuous planning, application prototyping, process	Not evident.
12	[37]	Culture, foundation, tools.	Not evident.	Automation, measurement, exchange.
13	[27]	Not evident.	Continuous integration, continuous deployment.	Not evident.
14	[38]	Processes, technology, organization, people.	Not evident.	Not evident.
15	[30]	Not evident.	Continuous delivery, continuous deployment, continuous construction, change control.	Not evident.
16	[59]	Culture, read, quality assurance.	Continuous integration, continuous delivery, continuous deployment.	Automation, collaboration, measurement.

Acronyms: No. Number; Ref. Reference

Similarly, Table 2 presents the detailed characterization of elements associated with the technological tools identified during the MSL.

Table 2. Process elements identified in the tools

No	Ref	Name	Process Elements		
			Dimensions	**Dimensions**	**Values**
1	[40]	ATOS DevOps Maturity Assessment	Agility, architecture and design, organizational structure, culture and incentives, standardization.	Source code management, continuous integration, continuous construction, continuous deployment, resiliency, code refactoring, technical debt management	Collaboration, automation.
2	[41]	Microsoft DevOps Self-Assessmet	Processes, technology, culture, outputs	Not evident.	Automation, measurement.

No	Ref	Name	Process Elements		
			Dimensions	Dimensions	Values
3	[44]	Xmatters	Design and architecture, Culture and organizational alignment, testing, and verification	Not evident.	Not evident.
4	[49]	Boxboat	Not evident.	Continuous integration, continuous delivery, infrastructure as code.	Not evident.
5	[60]	DORA DevOps Quick Chcek	Not evident.	Delivery time, frequency of deployments, restore times, failure rate.	Not evident.

Acronyms: No. Number; Ref. Reference

3.2.2. Comparison of DevOps elements

The comparison of methodological solutions and technological tools was carried out following the protocol proposed in [21] which was adapted to compare the dimensions, values, and practices resulting from the homogenization process. The comparison was conducted in three phases: (i) analyzing the solutions, (ii) designing the comparison, and (iii) conducting the comparison. To achieve the comparison, a base model was identified and crossed with all the solutions obtained during the homogenization process through a matrix that relates the practices, dimensions, and values described in each solution. The base model was selected considering the following selection criteria: (i) C1: the solution is generic, (ii) C2: the solution has a clearly defined set of dimensions, values, and practices, and (iii) C3: the solution was evaluated by expert peers in DevOps. After analyzing the solutions, it was identified that the reference model proposed in [16] meets all the criteria. The base model was compared with 23 methodological solutions and 3 tools that propose practices, dimensions, and values. The results obtained after the comparison process allowed for a clear identification of the degree or level of correspondence between the practices, dimensions, and values of each proposal through a unified structure. Table 3 presents an example of the comparison that cross-references the elements presented in the base model [16] and the study [57].

Table 3. Comparison between the base model and the study [1]

Mapping direction: From [16] to [57] **Mapped process elements:** Dimensions and Values **Mapping question:** ¿Which dimensions and values defined by [16] support specific activities in [57]? **Mapping goal:** Determinate which dimensions and values of [16] have a close relation with the activities in [57].		**Related Study** [57]			
		Culture	Automation	Monitoring	Collaboration
Reference model [16]	**Dimensions**			**X**	
		X			
	Values		**X**		
					X

The results presented in each comparison matrix were used to conduct a correspondence analysis, allowing us to identify the degree of alignment between the practices, dimensions, and values of each study in relation to the baseline case. As a result of this correspondence evaluation, it was possible to quantify the level of relationship among all elements, which were subsequently used for integration, ultimately leading to the final solution presented in this study.

3.2.3. Integration of DevOps elements

To carry out the integration, the method proposed in [23] was applied, which identified the integrated process elements from all the solutions resulting from the comparison process. The integration method describes five activities: (i) designing the integration, (ii) defining/establishing integration criteria, (iii) conducting the integration, (iv) analyzing the results, and (v) presenting the integrated model. As a result of the integration process, a total of 12 practices considered fundamental, 6 practices considered complementary, 4 dimensions, and 4 values were obtained, representing the state of knowledge related to all the solutions identified in the MSL. The following is a detailed description of each of the values, dimensions, and practices obtained because of the integration process.

Values: According to the results presented in the previous sections, there is a clear consensus on some of the values considered most important when applying DevOps in a team or project. Values are important because they enable development and operations teams to work together efficiently, effectively, and proactively, fostering high-quality deliveries, increased delivery speed, and improved end-user satisfaction. The values proposed by DevOps serve as a guide to promote continuous improvement and innovation in development and operations processes, allowing companies to quickly adapt to market changes and stay competitive. Additionally, values help establish a strong organizational culture that ensures a coordinated, communicative, and collaborative work environment. Table 4 presents the details and purpose of each value proposed in the metrics model that resulted from the harmonization process.

Table 4. Values proposed in the metrics model

Process element	Name	Purpose
Values	Automation	Prioritize the automation of processes over the realization of manual processes whenever this has an impact on improvements to the work team. Everything that can be automated must be automated.
	Collaboration	Establish a framework in which teamwork is key, although there are different roles or areas of work, any action taken by one individual will benefit or affect another.
	Measurement	Establish the mechanisms that allow obtaining feedback and continuous improvement to ensure better decision-making in the future.
	Communication	Ensure communication and promote communication that motivates collaboration and transparency at all levels of the organization.

Dimensions: As a result of the harmonization process, a total of four (4) dimensions were obtained, which represent the set of activities, roles, products, practices, and tools necessary to ensure that the values described in Table 4 are correctly adopted. The dimensions arise from the need to establish a set of policies, guidelines, and organizational structures that facilitate the execution of each individual practice performed by a team within an organization when implementing DevOps. Table 5 presents a summary of each dimension comprising the metrics model that resulted from the harmonization process.

Table 5. Dimensions proposed in the metrics model

Process element	Name	Purpose
Dimensions	Tools	Establish 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 according to the capabilities and objectives of the organization.
	Processes	Define the set of policies, organizational structures, procedures, purposes, objectives, and work products required throughout the lifecycle of a software product that must be considered for DevOps adoption.
	Culture	Characterize all aspects related to knowledge, ideas, traditions, and customs that characterize an organization dedicated to product development in the software industry.
	People	Define the set of characteristics at the level of the individuals that make up a work team with the aim of synchronizing people towards processes that facilitate collaboration and teamwork.

Practices: The harmonization process allowed for the identification, standardization, and comparison of practices proposed in all the studies and tools analyzed during the systematic literature mapping. As a result of consolidating, prioritizing, and characterizing all the practices identified in the SLM and tool analysis, a total of 18 practices were obtained, of which 12 are considered fundamental practices (FP) and 6 are considered complementary practices (CP). Fundamental practices define activities that enable the enhancement and/or facilitation of aspects such as: (i) the automation of processes throughout the software project lifecycle through continuous integration, delivery, and testing activities, (ii) facilitating strategic management and requirements management in a project, (iii) ensuring the protection of data and access to information through data management techniques and security monitoring, (iv) establishing version control policies through configuration management techniques, (v) defining tasks and activities that support artifact monitoring in a project, and (vi) fostering continuous learning of DevOps values and principles through education, feedback, and continuous measurement of the proposed DevOps culture. In addition, complementary practices provide strategies for implementing activities related to continuous deployment, infrastructure as code, and enhancing security monitoring activities through privilege access management mechanisms. Finally, complementary practices aim to continuously improve existing DevOps processes through continuous learning and experimentation activities, which align with efforts to create an environment that enhances team satisfaction Table 6 presents a summary of each practice resulting from the harmonization process.

Table 6. Proposed practices in the metrics model

Process element	Identifier	Name	Purpose
Fundamental practices (PF)	IC	Continuous Integration	Establish the necessary activities that guarantee the integrity of the changes made during development through version control strategies and software composition.
	EC	Continuous delivery	Produce software in short cycles, ensuring that software can be reliably released at any time.
	PC	Continuous testing	Assess software quality at every step of the continuous delivery process through early testing and frequent testing.
	GR	Requirements management	Operate with clear, realistic, and agreed requirements.

Process element	Identifier	Name	Purpose
	GD	Data management	Analyze the speed of changes in databases, constantly monitor the impact of changes on the DevOps process and replicate them to the place where they are required.
	SS	Security supervision	Improve software delivery and organizational performance through the identification, monitoring and resolution of vulnerabilities and/or threats sensitive to attacks of any kind.
	DE	Strategic direction	Align all the efforts of the organization around a goal.
	GC	Configuration management	Maintain the integrity and validity of developed products during all stages of the product life cycle.
	MC	Continuous monitoring and observability	Facilitate monitoring of all types of resources, personnel, and processes to find errors as early as possible and determine why they happen. Likewise, propose strategies that allow pre- and post-analysis of solutions to identify opportunities for improvement all the time.
	ED	Education around DevOps	Training and knowledge transfer that allows the work team to know, understand and internalize "what" DevOps is and "how" to apply it correctly in their projects.
	RC	Continuous feedback	Communication and lifelong learning that allows to reintroduce the knowledge acquired in the past.
	MCu	Culture measurement	Know the impact of culture on results in the process of adopting DevOps in the company.
Complementary practices (PC)	DC	Continuous deployment	Promote strategies necessary to deploy the artifacts present during the development process to the different environments arranged in the project and thus promote the frequent delivery of functional software.
	ICo	Infrastructure as code	Empower developers or operations teams to manage, monitor, and provision resources automatically.
	GA	Privilege Access management	Protect the infrastructure and applications used, manage the business efficiently and maintain the confidentiality of sensitive data and critical infrastructure.
	AC	Continuous learning	Promote a culture of development of new knowledge that favors competitiveness and innovation.
	EC	Continuous experimentation	Promote innovation and learning by encouraging experimentation that allows the definition of new solutions aligned with the needs of the organization.
	SL	Job satisfaction	Positively impact the performance and commitment of the work team with the aim of establishing environments of trust.

To establish a clear degree of relationship between the practices, dimensions, and values resulting from the integration process, a correspondence analysis was carried out to determine how the practices can be associated with the dimensions (see Table 7) and the dimensions with the values (see Table 8), respectively.

Table 7. Relation between practices and dimensions

Practices	Acronym	Name	Dimensions			
			Tools	Processes	Culture	People
Fundamental practices (PF)	IC	Continuous Integration	X	X		X
	EC	Continuous delivery	X	X		X
	PC	Continuous testing	X	X		X
	GR	Requirements management		X	X	X
	GD	Data management	X	X		X
	SS	Security supervision	X	X		X
	DE	Strategic direction		X		X
	GC	Configuration management	X	X		X
	MC	Continuous monitoring and observability	X	X		X
	ED	Education around DevOps		X	X	
	RC	Continuous feedback			X	X
	MCu	Culture measurement		X	X	
Complementary practices (PC)	DC	Continuous deployment	X	X		X
	ICo	Infrastructure as code	X			X
	GA	Privilege access management	X	X		X
	AC	Continuous learning		X	X	X
	ExC	Continuous experimentation		X	X	X
	SL	Job satisfaction			X	X

Table 8. Relation between dimensions and values

Dimensions	Values			
	Automation	Collaboration	Measurement	Communication
Tools	X		X	X
Processes	X		X	
Culture	X	X	X	X
People		X		X

To provide a macro view of the metrics model, Figure 1 presents a high-level diagram that describes each of the activities presented in the previous sections and an example of how the evaluation process would be carried out in a company using the Structured Analysis and Design Technique (SADT). The SADT technique provides a representation of activities, relationships, dependencies, and outputs of a process in an organized structure.

According to the flow presented in Figure 1: (i) activities A1.1, A1.2, and A1.3 represent a summarized version of the harmonization process explained in the previous subsections. As a result of performing these activities, each of the practices, dimensions, and values that compose the metrics model were obtained; (ii) activity A2 takes as input each of the fundamental and complementary practices obtained in activity A1.1 and relies on two artifacts: an evaluation template and a web tool. The result of this activity is the individual, weighted, and overall implementation percentage of practices; (iii) activity A3 takes as input the results obtained in activity A2.1, the dimensions resulting from activity A1.2, and relies on the web tool. The result of this activity is the individual and overall implementation percentage of dimensions; (iv) similarly, activity A4 takes as input the results obtained in activity A3, the values resulting from activity A1.3, and relies on the web tool. The result of this activity is the individual implementation percentage of each value and the implementation percentage of DevOps. Finally, the results obtained after calculating the implementation percentage of practices (A2), dimensions (A3), and values (A4) are reported to the end user in the web tool (A5). The details of the evaluation template and the web tool are presented in next sections. Furthermore, the details of the activities carried out to calculate each of the reported percentages for the case study are presented in the following subsections.

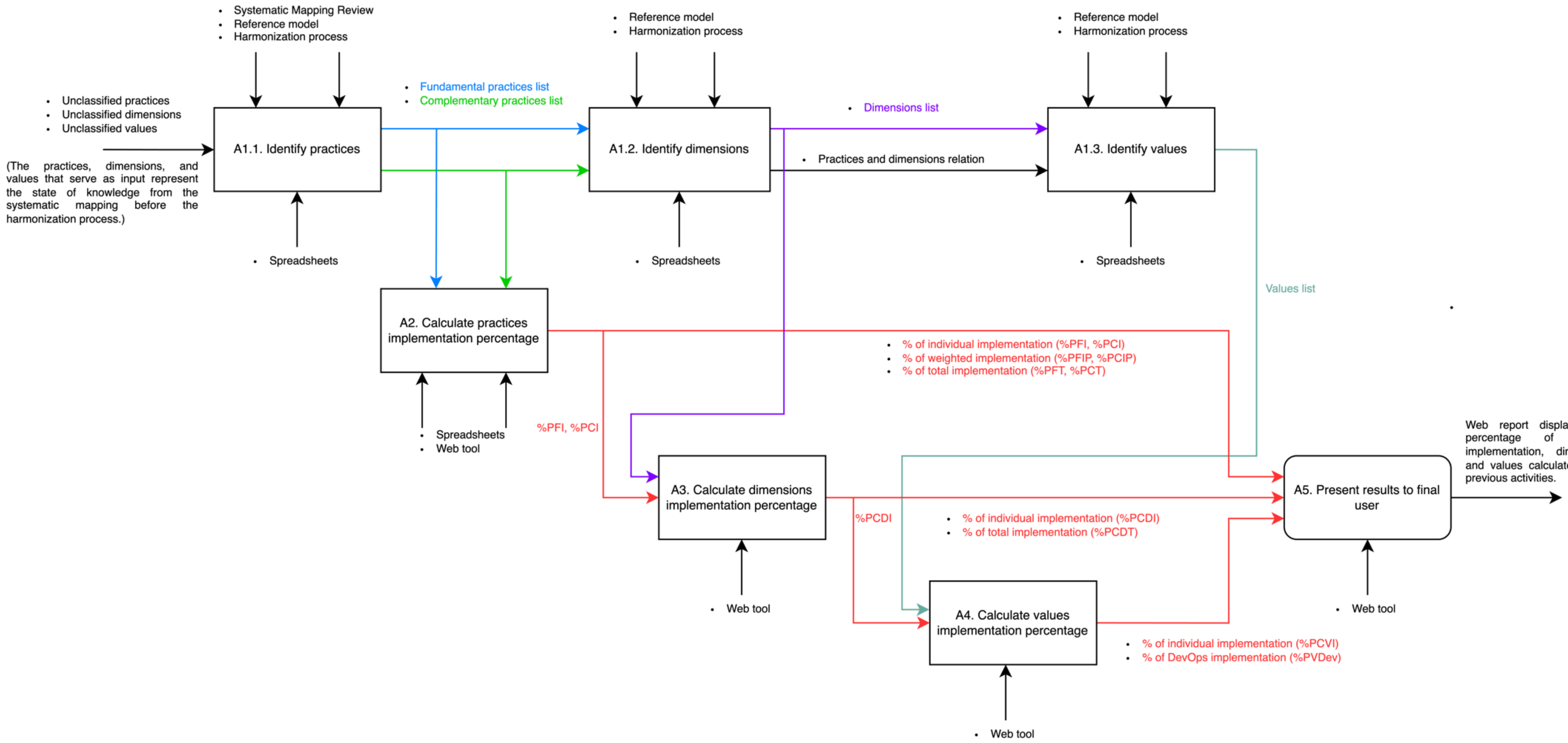

Figure 1. Overview of the practice evaluation process

3.3. Metrics model

The metrics model was defined following the guidelines described within the Goal, Question, Metric - GQM approach [24], which proposes: (i) a conceptual level (Goal), (ii) an operational level (Question), and (iii) a quantitative level (Metric). At the conceptual level, the dimensions, practices, and values proposed by DevOps were identified through the harmonization process. At the operational level, questions associated with each DevOps practice were defined according to a set of objectives associated with each practice. Finally, at the quantitative level, a set of metrics was defined to measure the implementation percentage of DevOps practices, dimensions, and values. By applying GQM, a total of 42 objectives and 63 questions related to fundamental practices were obtained. These numbers were determined based on the systematic literature mapping and the harmonization process, which identified the most relevant DevOps practices, dimensions, and values. Each objective was defined to cover a key aspect of DevOps implementation, ensuring a comprehensive evaluation. The corresponding questions were derived to assess compliance with these objectives, aligning with best practices and industry standards. This approach guarantees that the evaluation model captures both the breadth and depth of DevOps adoption in a structured and measurable manner.

Additionally, a total of 19 objectives and 29 questions related to complementary practices were obtained. To answer each of the questions, a questionnaire-type evaluation instrument was designed with two possible responses ("YES," "NO"). The response to each question is related to the following criteria: "YES" if there is (i) gathering of opinions from each of the roles involved in the practice or (ii) consistent historical records that demonstrate compliance with the practice. The response "NO" is applied if (i) the company does not provide evidence of compliance with the practice or (ii) the supporting documentation presented is not sufficient to determine compliance with the practice. After defining each of the objectives and questions described at the conceptual and operational levels, a total of 11 metrics were defined, of which: 7 metrics evaluate the compliance percentage of fundamental and complementary practices, 2 metrics evaluate the compliance percentage of dimensions, and 2 metrics evaluate the compliance percentage of values. Table 9 presents a summary of each of the metrics.

All the formulas used in the metrics model were originally developed as part of this study and were validated by experts throughout the project's development. These experts assessed the formulas and determined that they are appropriate for the specific context in which they were applied, ensuring their relevance and accuracy in evaluating the implementation of DevOps practices, dimensions, and values.

On the other hand, values related to $\%PPCF$ (Weighted percentage of impact of fundamental practices) and $\%PPCC$ (Weighted percentage of impact of complementary practices) are taken as 70% and 30%, respectively, as a basis calculated through expert judgment and empirical analysis conducted during the study. These values were determined considering the relative importance of fundamental and complementary practices in the DevOps context, ensuring that the weighting reflects their actual impact on the overall evaluation. However, these values can be adjusted if a company considers that its specific context indicates that this level of importance may vary.

Table 9. Summary of metrics

Element	Number	Metric name	Description	Identifier	Metric
Practices	1	Percentage of individual compliance with a core practice	Metric that determines the percentage of implementation of a fundamental practice taking as a unit of measurement	$\%PFI$	**Scale:** $[0\%, 100\%]$ **Equation:** $$\%PFI = \frac{1}{n}\sum_{i}^{n} \%P_i$$ **Variables:** $\boldsymbol{n}$: Number of questions related to the

Element	Number	Metric name	Description	Identifier	Metric
			the questions that it has associated.		practice; $\%\boldsymbol{P_i}$: Percentage obtained for a specific question (YES: 100%, NO: 0%).
	2	Weighted individual compliance percentage of a key practice	A metric that calculates the weighted percentage of a fundamental practice. Each fundamental practice has an associated weighted percentage (%PPA).	$\%PFIP$	**Scale:** $[0\%, \%PPA]$ **Equation:** $\%PFIP = \%PFI * \%PPA$ **Variables:** $\%\boldsymbol{PFI}$: Percentage of individual compliance for a fundamental practice; **%PPA:** Weighted associated percentage.
	3	Percentage of total compliance with key practices	Metric that calculates the percentage of compliance of all practices taking as a unit of measure the weighted individual percentage of compliance of each fundamental practice	$\%PFT$	**Scale:** $[0\%, 100\%]$ **Equation:** $$\%PFT = \sum_{i}^{n} \%PFIP_i$$ **Variables:** $\boldsymbol{n}$: Number of fundamental practices; $\%\boldsymbol{PFIP_i}$: Weighted percentage for a fundamental practice.
	4	Percentage of individual compliance with a complementary practice	Metric that determines the percentage of implementation of a complementary practice taking as a unit of measurement the questions that it has associated.	$\%PCI$	**Scale:** $[0\%, 100\%]$ **Equation:** $$\%PCI = \frac{1}{n}\sum_{i}^{n} \%P_i$$ **Variables:** $\boldsymbol{n}$: Number of questions related to the practice; $\%\boldsymbol{P_i}$: Percentage obtained for a specific question (YES: 100%, NO: 0%).
	5	Weighted individual compliance percentage of a complementary practice	Metric that calculates the weighted percentage of a complementary practice. Each complementary mental practice has an associated weighted percentage (%PPA).	$\%PCIP$	**Scale:** $[0\%, \%PPA]$ **Equation:** $\%PCIP = \%PCI * \%PPA$ **Variables:** $\%\boldsymbol{PCI}$: Percentage of individual compliance for a complementary practice; **%PPA:** Weighted associated percentage.
	6	Percentage of total compliance with	Metric that calculates the percentage of compliance of all	$\%PCT$	**Scale:** $[0\%, 100\%]$ **Equation:**

Element	Number	Metric name	Description	Identifier	Metric
		complementary practices	practices taking as a unit of measure the weighted individual percentage of compliance of each complementary practice		$\%PCT = \sum_{i}^{n} \%PCIP_i$ **Variables:** $\boldsymbol{n}$: Number of complementary practices; $\%\boldsymbol{PCIP_i}$: Weighted percentage for a complementary practice.
	7	Percentage of combined compliance with core and complementary practices	Metric that represents the percentage of compliance of all fundamental practices, and complementary together. This represents the total percentage of practices compliance.	$\%PTP$	**Scale:** $[0\%, 100\%]$ **Equation:** $\%PTP = (\%PPCF * \%PFT) + (\%PPCC * \%PCT)$ **Variables:** $\%\boldsymbol{PPCF}$: 70% $\%\boldsymbol{PPCC}$: 30%; $\%\boldsymbol{PFT}$: Total percentage of compliance of fundamental practices; $\%\boldsymbol{PCT}$: Total percentage of compliance of complementary practices.
Dimensions	8	Individual compliance percentage of a dimension	Metric that establishes the percentage of compliance of a dimension. Each dimension has a set of fundamental and complementary practices associated with it. Each summatory cross the related elements identified in Table 7 for the fundamental and complementary practices, respectively.	$\%PCDI$	**Scale:** $[0\%, 100\%]$ **Equation:** $\%PCDI = \frac{\%PPCF}{n}\sum_{i}^{n} \%PFI_i + \frac{\%PPCC}{m}\sum_{j}^{m} \%PCI_j$ **Variables:** $\boldsymbol{n}$: Number of fundamental practices related to a dimension; $\boldsymbol{m}$: Number of complementary practices related to a dimension; $\%\boldsymbol{PPCF}$: 70%; $\%\boldsymbol{PPCC}$: 30%; $\%\boldsymbol{PFI_i}$: Percentage of individual compliance for an specific fundamental practice related to a dimension; $\%\boldsymbol{PCI_i}$: Percentage of individual compliance for an specific complementary practice related to a dimension.
	9	Total dimension compliance percentage	Metric that establishes the percentage of combined compliance of all dimensions. Each	$\%PCDT$	**Scale:** $[0\%, 100\%]$ **Equation:**

Element	Number	Metric name	Description	Identifier	Metric
			dimension has an associated weighted percentage (%PCDI). Given the scope of this study, it is considered that all dimensions have the same importance.		$$\%PCDT = \sum_{i}^{n} \%PPAD_i * \%PCDI_i$$ **Variables:** $\boldsymbol{n}$: Number of dimensions; $\%\boldsymbol{PPAD_i}$: Weighed percentage related to a specific dimension; $\%\boldsymbol{PCDI_i}$: Percentage of individual compliance for a dimension.
Values	10	Percentage of individual compliance of a value	A metric that establishes the percentage of compliance for a value. Each value has a set of dimensions associated with it. The summatory cross the related elements identified in Table 8 for each value.	$\%PCVI$	**Scale:** $[0\%, 100\%]$ **Equation:** $$\%PCVI = \frac{1}{n}\sum_{i}^{n} \%PCDI_i$$ **Variables:** **n**: Number of dimensions related to a value; $\%\mathbf{PCDI_i}$: Percentage of compliance of a dimension associated with a value.
	11	DevOps compliance percentage	Metric that establishes the percentage of DevOps compliance. The DevOps compliance percentage is calculated by reference to the compliance percentage of each value. Each value has an associated weighted percentage (%PPAV). Given the scope of this study, it is considered that all values have the same importance.	$\%PCDev$	**Scale:** $[0\%, 100\%]$ **Equation:** $$\%PCDev = \sum_{i}^{n} \%PPAV_i * \%PCVI_i$$ **Variables:** **n**: Number of values; $\%\mathbf{PPAV_i}$: Weighted percentage related to a value; $\%\mathbf{PCVI_i}$: Individual value percentage.

3.3.1. Evaluation template

To answer the questions proposed in the metrics model, an evaluation tool in the form of a questionnaire was designed using an electronic spreadsheet that allows two possible responses ("YES," "NO"). The template contains a set of questions associated with each of the fundamental and complementary practices proposed in the model. The response to each question follows the criteria described in Table 10.

Table 10. Possible answers in the evaluation instrument

Answer	Description
YES	Sufficient and necessary evidence is presented to ensure that the set of activities associated with the question is carried out correctly. Evidence can be taken from different mechanisms, as well as: (i) direct evidence taken through observation of a practice, (ii) collection of opinions for each of the roles involved in the practice or (iii) consistent historical records that allow evidence of compliance with the practice.
NO	This response occurs in two scenarios: (i) the company does not have any type of evidence or (ii) partial compliance with the practice is observed; that is, not all the necessary aspects are met to ensure that the question is answered completely.

Additionally, the evaluation template includes a subsection providing basic company information, which is uploaded to a web tool implemented to support this process. This way, a company can perform new evaluation processes in the future. **Figure 2** presents an excerpt from the subsection of the template that describes the questions used to assess the compliance level of the continuous integration practice.

Question	Description	Answer		Observations
		Yes	No	
P1_IC	Have explicit policies, procedures, or strategies been defined to carry out version control in the project?	X		There are documented policies for version control management.
P2_IC	Have repositories been defined to automate version control for artifacts involved in the development process?	X		Confluence is used for documentation. GitLab and SVN for source code.
P3_IC	Does the project use tools that ensure the integrity of the source code before performing any updates?	X		Yes, Checkstyle and Sonar are used.
P4_IC	Are there clear procedures for implementing and documenting unit tests?		X	
P5_IC	Is the compilation, review, and deployment of the source code automated?		X	
P6_IC	Are there mechanisms to clearly identify any errors during the integration of new code in the development process?	X		
P7_IC	Are there mechanisms, policies, or procedures that allow the recovery of a stable system state in case of a failure during the integration of new code?	X		This is done through the change control policy and version control systems.
P8_IC	Do development and operations teams have access to the version control system set up for the project?	X		There are individual accounts for version control systems.

Figure 2. Evaluation template real example

The detailed evaluation template, which includes the questions associated with each practice, can be downloaded from the following link: https://bit.ly/3Zkshqt. Each of the questions presented in the template was derived from the process described in the previous section.

3.4. Proposed process for applying the model

The evaluation process is carried out following the set of activities proposed in [61], which are described below: (A1) plan the evaluation; the evaluator describes the characteristics of the company where the evaluation will take place, the evaluation date, activities to be carried out, and the effort and time required to perform the evaluation, (A2) delivering the documentation to the process responsible; the evaluator provides the template to the participants, (A3) execute the evaluation; the evaluator gathers the necessary evidence with the required information to apply the metrics, (A4) generate and analyze the results; the evaluator analyzes the results obtained after executing the metrics in the tool, and (A5) delivering the results report to the process responsible; the evaluator presents a detailed report of the results obtained to the process owner designated in the company.

The activities to conduct the evaluation were performed following the protocol described in [62], it is suggested that the activities be carried out based on the company's capacity in terms of human resources, effort, and available time [63].

Table 11, provides a recommendation on the criteria that should be considered for applying the model according to the classification of the type of company, based on the number of employees [64], [65] and the effort that the process owner should exert.

Table 11. Proposed roles

Type of company	Number of employees (n)	Recommended time for the evaluation [Week]	Effort [Hour]	Recommended time between evaluations [month]
Big	$n \geq 250$	1	3	9
Medium	$50 \leq n < 250$	1	3	9
Small	$10 \leq n < 50$	1.5	2	6
Very Small	$n < 10$	2	1	6

In Table 12, the details of each activity that make up the evaluation process and the resulting output artifacts (AS) are presented.

Table 12. Evaluation process activities

Activity	Description	Output
A1. Plan the evaluation	The planning is carried out in different stages: (i) establish a first contact with the company to raise the objective of the case study, activities to be carried out, clarification of doubts, characterization of the company and definition of a date to start the case study; and (ii) a remote and/or synchronous meeting lasting approximately two hours to explain the model to stakeholders.	**AS1.** Evaluation implementation plan. **AS2.** Description of the elements to be evaluated. **AS3.** Evaluation template.
A2. Deliver the documentation to the process responsible	The evaluator delivers the evaluation template to all participants designated by the company. Each participant has the task of answering each of the questions presented in the template and providing the necessary supports to guarantee their compliance.	
A3. Execute the evaluation	The execution is carried out following the following activities: (i) identify, characterize and document the role of each of the participants of the evaluation; (ii) deliver the evaluation template to each of the participants to answer the questions designated in an evaluation template provided by the evaluation team; and (iii) deliver the completed template to the evaluator, who has the responsibility of uploading the file to a web tool to obtain the results of the evaluation.	**AS1.** Supports to assure the responses filled out in the template. **AS2.** Filled evaluation template.
A4. Generate and analyze the results	From the results delivered by the tool, the evaluator must report a detailed analysis according to the results obtained by the company.	**AS1.** Evaluation results report.
A5. Deliver the results report to the process responsible	The evaluator delivers the analysis of the results to the company in an estimated time of five business days after carrying out the evaluation.	**AS1.** Evaluation results detail report.

3.5. Tool to automate the process of applying the metrics model

To facilitate the work of an evaluator during the process of applying the metrics model, a comparative analysis was carried out to identify the level of detail in the evaluation performed by each of the technological tools identified in the SLM. The following criteria were evaluated: (i) C1; the tool evaluates the degree of adoption of DevOps practices, dimensions, and values, (ii) C2; the tool is free, (iii) C3; the tool is supported by a clearly defined set of metrics, (iv) C4; the results provided by the tool are complemented with an analysis conducted by an external consultant, and (v) C5; the tool is supported by a reference model. Table 13 presents the result after analyzing each of the tools. It is necessary to clarify that the purpose of the evaluation is not to determine whether the solution proposed in this work is better than the tools identified in the literature; this strategy is used to establish a set of criteria that facilitate the identification of possible improvements or additions that add value to this proposal and that may not have been considered in previous solutions.

The selection of these five criteria was based on the need to assess both the comprehensiveness and practical applicability of the tools in real-world DevOps evaluation processes. Each criterion addresses a key aspect that impacts the usability, accessibility, and methodological rigor of the evaluation: C1 ensures that the tool aligns with the core elements of DevOps assessment, C2 considers accessibility and cost-related barriers, C3 verifies the presence of a structured and transparent evaluation framework, C4 assesses whether expert insights complement automated analysis, and C5 examines the tool's alignment with established reference models. These criteria were defined through a review of existing literature and expert validation to ensure their relevance to the context of DevOps evaluation. While other criteria could have been considered, these were prioritized as they encapsulate essential characteristics that directly impact the practical utility of the tools, ensuring a balanced comparison that highlights both technical and contextual factors relevant to DevOps adoption.

Table 13. Comparison of tools

Ref	**Name**	**C1**			**C2**		**C3**		**C4**		**C5**	
		Practice	**Dimension**	**Value**	**YES**	**NO**	**YES**	**NO**	**YES**	**NO**	**YES**	**NO**
[47]	IVI's DevOps Assessment	X				X		X		X		X
[40]	DevOps Maturity Assessment	X			X			X		X		X
[41]	Microsoft DevOps Self-Assessment	X			X			X	X			X
[52]	Eficode	X				X		X		X		X
[42]	Infostretch	X			X			X		X		X
[43]	InCycle Evaluacion de devops	X			X			X		X		X
[48]	Veritis	X				X		X		X		X
[49]	Boxboat	X				X		X		X		X
[46]	IBM DevOps Self Assesment	X			X		X			X		X

Ref	Name	C1			C2		C3		C4		C5	
		Practice	**Dimension**	**Value**	**YES**	**NO**	**YES**	**NO**	**YES**	**NO**	**YES**	**NO**
[44]	DevOps Maturity Survey Report	X			X		X		X			X
[50]	Humanitec DevOps Assessment	X				X		X		X		X
[51]	Attlasian DevOps Assessment	X				X		X		X		X
[45]	DevOps Maturity model	X			X			X		X		X
[66]	DevOps Heath Radar Assessment	X	X		X		X					X
[60]	DORA DevOps Quick Check	X			X			X		X		X

According to the results obtained from the analysis of the tools, the following observations were made:

- **Regarding criterion C1:** 15 tools (100%) evaluate DevOps practices, 1 tool (6.6%) evaluates dimensions, and no tool (0%) evaluates values.
- **Regarding criterion C2:** 9 tools (60%) are free, and 6 tools (40%) are paid.
- **Regarding criterion C3:** 3 tools (20%) propose a structured evaluation mechanism, while 12 tools (80%) only present the metrics without providing a detailed report that explains how they were defined.
- **Regarding criterion C4:** 2 tools (13.3%) offer consulting services as support to the tool, and 13 tools (86.7%) provide a report that only shows the evaluation results.
- **Regarding criterion C5:** no tools (0%) follow a reference model. Based on the above, the following observations can be made: several tools have been proposed to evaluate DevOps using different approaches; however, none of the proposals provide a comprehensive evaluation that encompasses practices, dimensions, and values.

Furthermore, 40% of the proposed tools limit the user by offering partial results that require a subscription payment to access the complete report. Additionally, 80% of the tools do not have a clear mechanism to understand the scope of their evaluation, and only 13.3% of the tools offer specialized consulting services to support the evaluation. Finally, no tool follows a reference model, resulting in heterogeneous solutions that are defined based on each company's individual considerations. To support the application process of metrics through a tool that meets the criteria described in the previous section, it was decided to implement a web application that automates the calculation of the results obtained when applying each of the metrics and weights proposed in the evaluation model. The following subsections provide detailed information about the tool. Based on the above, it can be observed that the proposed model consistently aligns with the five comparison criteria. However, it is important to clarify that the reported results are not intended to determine whether this solution is superior to others but rather to indicate that it seeks to meet a set of criteria that facilitate its application in different contexts.

4. Case studies

The evaluation of the metrics model was carried out through case studies applied to two software development companies following the guidelines and protocols established in [67]. The following sections describe the structure of the case studies according to their design, object of study, execution and data extraction, analysis of results, validity analysis, and limitations. Figure 3 presents the procedure carried out to execute each of the activities described in the case studies.

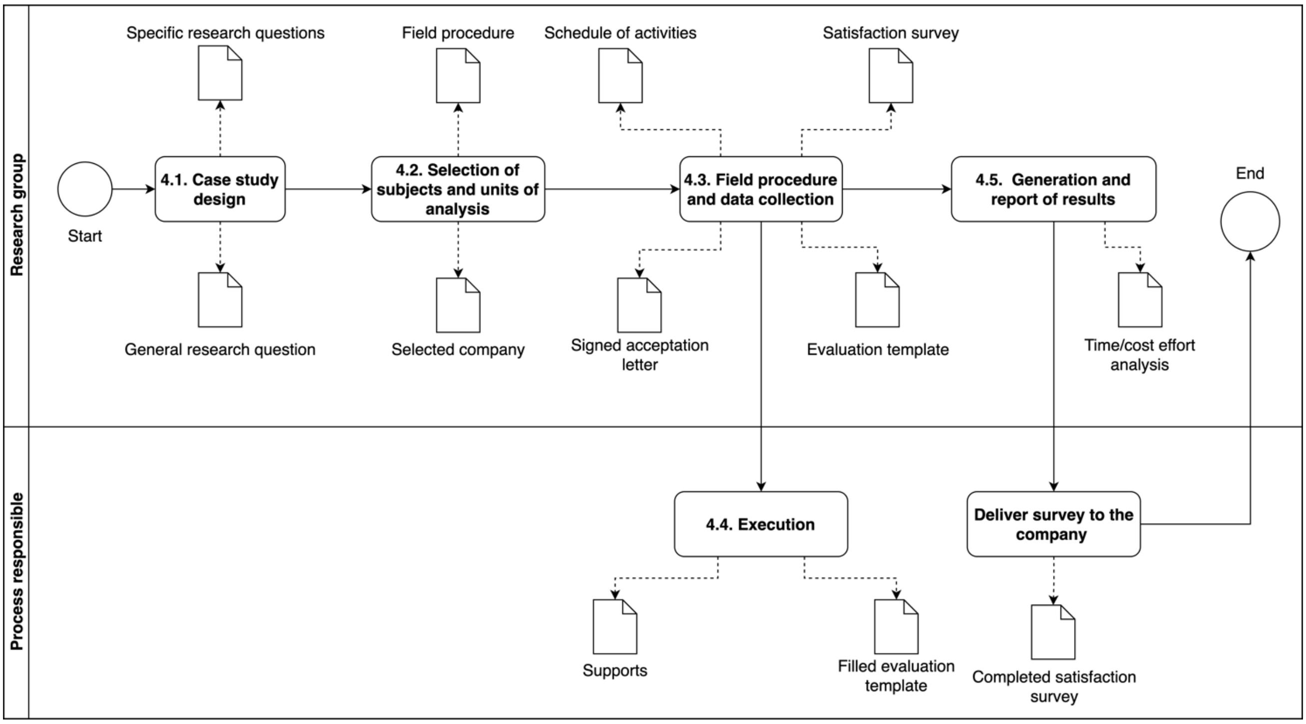

Figure 3. Procedure for conducting the case studies

4.1. Case studies design

Based on the knowledge obtained during the literature review, the harmonization of solutions for evaluating DevOps, and the definition of the metrics model, it was possible to define a general research question (GQ) to conduct the case study, which is addressed by defining three specific research questions (SQ). The research questions were formulated to determine if the metrics model is useful, applicable in the industry, and meets the needs of software development companies that implement DevOps in their projects. Table 14 presents the details of each research question.

Table 14. Research questions to carry out the case studies

Research question	Description
General question (GQ)	Does the model allow to evaluate in an understandable and useful way the set of practices, dimensions and values proposed by DevOps in a software development company?
Specific questions (SQ)	PI1. Are the variables used for the evaluation of the metrics clear and easy to understand?
	PI2. Do the results provide valuable information to companies to improve their DevOps adoption process?
	PI3. Can a company apply the model with an acceptable degree of effort?

The design of each case study was holistic with a unit of analysis [40], which consisted of applying the model in each company. The measures used in each case study were:

1. The effort applied by each company to implement the proposed solution: Characterized using time (hours) and the number of people involved in each case study.
2. The improvement opportunities identified after applying the model: Through the suggestions and/or comments made by the evaluator after analyzing the results obtained in each company.
3. The benefits obtained in each company after applying the metrics: Determined by the comments and perceptions of the participants after conducting the evaluation.

4.2. Selection of subjects and units of analysis

The two companies in which the case study was conducted were selected based on the following criteria: (i) the company provides services related to software development, (ii) the company implements DevOps in one or more projects, (iii) the company is interested in evaluating the degree or percentage of DevOps implementation in its projects in order to identify improvement opportunities in its software development process, and (iv) the company has extensive experience in using agile frameworks such as Scrum, Kanban, and Lean. As a result, a total of two (2) software development companies were selected, whose names are not disclosed for confidentiality reasons and will be referred to as C1 (Company 1) and C2 (Company 2) hereafter. Table 15 presents the information for each company, including their identifier, number of employees, sector or business area, time in the industry in the industry, company description, and the objective set for the evaluation.

Table 15. Information of each company.

Id	Number of employees	Sector	Time in the industry[years]	Company description	Evaluation goal
E1	**54**	**Health**	**18**	Company focused on providing software solutions in the health sector. The company has been using SCRUM as a development methodology for 7 years and began applying DevOps as an automation culture in 2019.	Know your current state of DevOps adoption and identify possible improvement actions.
E2	**14**	**Financial**	**5**	Fintech focused on providing customized solutions for clients in the financial sector. The company applies agile approaches such as SCRUM and XP to support its development process. The company began applying DevOps in 2021 to facilitate the process of continuous delivery of new functionality.	Identify which DevOps practices have not been adopted in the enterprise and take corrective actions to improve their current practices.

Acronyms: Id. Identifier

4.3. Field procedure and data collection

The field procedure is divided into three main activities: (i) planning, (ii) execution, and (iii) report generation and results. The details of each activity carried out for the execution of each case study are presented below.

Planning: The planning was carried out as follows: (i) Initially, a contact was made with each company to present the objective of the case study, the activities that would be carried out if accepted, clarification of any doubts that may arise regarding the scope, purpose, or activities of the case study, and finally, the definition of a date to start the case study; (ii) After establishing the initial contact, a remote and synchronous meeting was held, lasting two hours, in which the project was presented to the stakeholders. Subsequently, the metrics model was presented, roles were assigned to each participant in the case study, and a schedule was proposed to establish a start and end date for each study.

Execution of each case: During the execution process, the role of each participant was identified, characterized, and documented. The purpose of this characterization is determining each company profiles and roles based on their own business target and organizational goals. This study allows the research group understand how the model needs to be applied in specific contexts.

The details of each role per company are presented in Table 16.

Table 16. Roles per company

Company	Role identifier	Role name	Role description
C1	**C1_HR1**	**Project leader**	Person with knowledge of the entire life cycle of the project.
	C1_HR2	**Developer**	Senior Developer in charge of tasks focused on infrastructure management and deployment of environments.
C2	**C2_HR1**	**Operations leader**	Person in charge of defining the guidelines, policies, and strategies necessary to adopt DevOps in the company.
	C2_HR2	**Developer/DevOps**	Senior developer who performs tasks related to the study and implementation of tools, technologies and mechanisms that can support the use of DevOps practices in the company.

Acronyms: HR. Human resource.

Subsequently, each participant was asked to answer the questions presented in the evaluation template. Additionally, each company was requested to provide the necessary evidence to determine that the response to each question had a clear and consistent justification to certify its level of compliance. Table 17 describes the effort in hours applied during the execution stage of the evaluation. The detailed report of the total effort applied by each company is addressed in the following subsection.

Table 17. Effort in hours applied by each participant

Company	Involved Roles	Individual effort	Total effort
C1	C1_HR1. Project Leader	4	7
	C1_HR2. Developer	3	
C2	C2_HR1. Operations leader	5	13
	C2_HR2. Developer/DevOps	8	

Results generation and report: The results obtained after conducting the evaluation in each company were analyzed to establish a comprehensive view from multiple perspectives that allowed the identification of relationships, differences, and commonalities regarding the degree of implementation of DevOps practices, dimensions, and values in each company. The scheduled time to conduct and deliver the analysis of the results was five (5) business days from the day each company submitted the evaluation report and uploaded the completed template to the web tool. The time effort was recorded by documenting the hours spent by the research team and the study participants during the stages of (i) proposal presentation, (ii) evaluation planning, (iii) clarification of doubts, (iv) execution, (v) analysis, and (vi) reporting of the results. Table 18 presents the detailed time effort in each of the studies.

Table 18. Time effort applied in the case studies

Stage	Effort [Hour]	
	E1	E2
Socialization of the proposal	2	2
Planification	1	2
Clarification of doubts	0.5	1
Execution	7	13
Analysis of results	12	12
Results report	1	1
Total	23.5	31

The total number of hours presented in the execution stage corresponds to the hours applied during the business days of the evaluation in each company, respectively. Since the participants had other assignments related to the projects they were assigned to, they dedicated a different time effort each day to carry out the evaluation, ensuring that it did not delay their previous assignments. Considering this, each participant was requested to document the number of hours they dedicated each day to determine the exact number of hours spent by each participant during the execution of the case study.

The breakdown of the hourly effort applied by each company is presented in Table 19. Spaces marked with the symbol "-" represent days in which a participant did not report effective hours related to the evaluation. This behavior was consistent with the fact that every participant had different activities related with their own role responsabilities.

Table 19. Hourly effort applied per day by each company

Day	Effort [Hour]			
	Company 1		Company 2	
	C1_HR1	C1_HR2	C2_HR1	C2_HR2
1	1	-	1	1
2	-	1	1	-
3	1	-	1	2
4	1	0.5	1	1
5	-	-	1	0.5
6	1	0.5	-	0.5
7	-	1	-	1
8	-	-	-	1
9	-	-	-	1
Individual effort [Hour]	4	3	5	8
Total effort [Hour]	7		13	

Based on the information presented in Table 20 the following observations were made: (i) Company C1 took seven (7) business days to complete the evaluation. It was observed that participant HR1 dedicated one hour of effort for four (4) days, while participant HR2 dedicated one hour for two days and half an hour (0.5 hours) for two days. (ii) Company C2 took nine (9) business days. Participant RH1 dedicated one hour of effort for five (5) days, and participant HR2 dedicated one hour for five (5) days, half an hour (0.5 hours) for two (2) days, and two (2) hours for one day. Regarding the total effort in hours dedicated by each company, although the effort in days for each company was seven (7) and nine (9) business days, respectively, the effective hours were seven (7) and thirteen (13), which corresponds to 0.875 and 1.625 business days. In this regard, a participant without parallel assignments or tasks could complete the evaluation within a timeframe of one to two business days.

Based on the analysis and discussion of the results in each company, the following findings were identified: (i) Each participant had to allocate a variable time slot of one to two hours exclusively for carrying out the evaluation without interrupting their ongoing assignments. (ii) The chosen hourly intensity by each participant was directly related to their daily workload and assignments, which is why there were no progress reports on some days. (iii) The hourly intensity dedicated by each participant was determined by their level of experience and knowledge within the organization. As a result, some participants completed the evaluation much faster compared to their peers.

Results report to each company: After receiving and uploading the template to the tool, the system generated the corresponding results for each company. Table 20 presents a summary of the obtained results. According to the tool's report, the total implementation percentage of practices for Company C1 was 56.06%, and the total implementation percentage of practices for Company C2 was 37.12%.

Table 20. Percentage of practice implementation in each company

Practice	Name	Results					
		Company 1 (C1)			Company 1 (C2)		
		Individual result	Combined result	Total result	Individual result	Combined result	Total result
PF	Continuous Integration	62.50%	65.07%	56.06%	75%	30.09%	37.12%
	Continuous delivery	100%			100%		
	Continuous testing	87.50%			62.50%		
	Requirements management	100%			25%		
	Data management	100%			28.57%		
	Security supervision	71.43%			0%		
	Strategic direction	83.33%			33.33%		
	Configuration management	57.14%			14.29%		
	Continuous monitoring and observability	20%			0%		
	Education around DevOps	33.33%			0%		
	Continuous feedback	40%			20%		
	Culture measurement	25%			0%		
PC	Continuous deployment	75%	35.06%		25%	35.53%	

Practice	Name	Results					
		Company 1 (C1)			Company 1 (C2)		
		Individual result	Combined result	Total result	Individual result	Combined result	Total result
	Infrastructure as code	40%			33.33%		
	Privilege Access management	33.33%			0%		
	Continuous learning	33.33%			40%		
	Continuous experimentation	25%			33.33%		
	Job satisfaction	0%			75%		

Table 21 presents a summary of the results obtained after applying the metrics to obtain the percentage of dimension implementation in each of the companies. The full description of how each metric is applied was presenter in Table 9.

Table 21. Percentage of dimension implementation in each company

Dimension	Results			
	Company 1 (C1)		Company 1 (C2)	
	Individual percentage	Total percentage	Individual percentage	Total percentage
Tools	55.69%	55.15%	38.34%	31.14%
Processes	54.48%		33.05%	
Culture	49.54%		18.21%	
People	60.87%		35.27%	

Table 22 shows the summary of results obtained after calculating the percentage of value implementation. The percentage obtained for each value is used to calculate the percentage of DevOps implementation.

Table 22. Percentage of value implementation in each company

Value	Results			
	Company 1 (C1)		Company 1 (C2)	
	Individual percentage	DevOps implementation percentage	Individual percentage	DevOps implementation percentage
Automation	55.24%	54.26%	29.77%	29.20%
Collaboration	55.20%		26.74%	
Measurement	53.24%		29.77%	
Communication	55.37%		30.51%	

The results presented in Table 21 and Table 22 were analyzed individually and collectively based on the response to each completed question and the supporting documents provided by each company. Based on the results, potential improvement actions related to core and complementary practices were identified and delivered to the responsible person in each company's process. Table 23 presents an excerpt of the improvement opportunities identified for continuous integration, continuous testing, and DevOps education practices. The recommendations were based on the set of best practices proposed by DevOps, as detailed in [68].

Table 23. Identified improvement actions (excerpt)

Id	Practice	Improvement actions	
		Company 1 (C1)	**Company 1 (C2)**
IC	**Continuous integration**	It is recommended to implement tools that allow automating the publication of new source code that is uploaded to integration, stage, and production environments by executing automated pipelines that are responsible for verifying the integrity of the new code implemented during the development stage.	It is suggested to define a clearly defined action plan that describes the next step for the implementation of version control in the project. In addition, it is suggested the implementation of tools that allow identifying common errors of style, smells and good practices when new changes are uploaded to the integration environment.
PC	**Continuous testing**	It is suggested to establish test sets focused on verifying the integration of the APIs.	Action plans must be implemented that allow testing APIs in projects. In addition, it is suggested to create a clearly defined plan that describes how functional tests should be carried out. Finally, it is suggested to define policies for the management and solution of bugs identified during the automated testing process.
ED	**Education around DevOps**	suggests defining spaces that allow team members to be trained in DevOps and follow up to verify that DevOps practices are carried out properly.	It is suggested to define spaces that allow team members to be trained in DevOps and to follow up to verify that DevOps practices are carried out properly.

The purpose of the improvement actions is to provide an additional perspective that goes beyond the quantitative results of the study, allowing each company to define strategies to improve their practices and achieve a higher score in the future. This fosters a culture of continuous improvement, focusing on enhancing existing practices and adopting those that have not been considered prior to the evaluation.

The total cost per company was calculated based on the effort in hours, converting the hours per human resource into an effective cost using the average hourly rates in U.S. dollars (USD) described in [69]. Table 24 presents the total cost obtained for each company. The results show that applying the metrics model cost 240.84 USD and 391.80 USD, respectively, which is relatively low compared to common business expenses such as training, certifications, and internal audits. Additionally, the cost includes a detailed report on the results obtained at the level of practices, dimensions, and values, feedback from the research group with recommendations on improving DevOps processes based on the findings, and suggestions for possible improvements in internal processes. In this regard, the proposed improvements during each evaluation process could contribute to future operational cost savings.

Table 24. Total cost after applying the model

Company	Role	Annual average salary [USD]	Hour average salary [USD]	Dedicated hours	Total cost [USD]
C1	RH1. Project leader	92536	32.13	4	240.84
	RH2. Developer	107831	37.44	3	
C2	RH1. Operations leader	54489	18.92	5	391.80
	RH2. Developer/DevOps	107000	37.15	8	

4.3.1.1.1 Qualitative analysis

To complement the validation of the proposal, a survey was conducted with each participant to gather their overall opinion of the model by evaluating aspects related to its applicability, comprehensibility, completeness, and suitability. The analysis of the obtained results aimed to assess the participants' perception based on their experience, considering their suggestions, comments, or recommendations regarding the artifacts presented in the metrics model. For this purpose, a form was created consisting of two sections: (i) general company information (name, number of employees, years in the industry, market description, and current models/standards used by the company), and (ii) a questionnaire composed of a total of 2 open-ended questions and 17 questions with a discrete Likert scale [70], which includes the following scale: very poor, very dissatisfied (1), poor, somewhat dissatisfied (2), fair, adequate, somewhat satisfied (3), good, quite satisfied (4), and excellent, very satisfied (5).

Table 25 presents the count of responses to the questions with the discrete scale according to the level of agreement in each of the companies.

Table 25. Count of responses to questions with the Likert scale

Question	**Description**	**Conformity level (Likert scale)**	
		C1	**C2**
1	Do you consider that the practices proposed in the model are clear and easy to understand?	5	5
2	Do you consider that the dimensions proposed in the model are clear and easy to understand?	5	5
3	Do you consider that the values proposed in the model are clear and easy to understand?	5	5
4	According to your experience: Do you consider that the evaluation of practices, dimensions and values is adequate and allows to identify aspects of value for software development companies?	5	5
5	Do you consider that the weighting defined for core and complementary practices is appropriate?	5	5
6	Do you consider that the weighting defined for the dimensions is adequate?	5	5
7	Do you consider that the weighting defined for the values is adequate?		
8	According to your experience, do you consider that metrics have sufficient mathematical rigor?	5	5
9	Do you consider that the metrics defined to assess the degree of adoption of fundamental practices are adequate?	5	5
10	Do you consider that the metrics defined to assess the degree of adoption of complementary practices are adequate?	5	5
11	Do you consider that the metric to calculate the degree of total adoption of practices is adequate?	5	5
12	Do you consider that the metrics to calculate the degree of adoption of dimensions are adequate?	5	5
13	Do you think the metric for calculating the degree of adoption of values is adequate?	5	5
14	Do you think the metric to calculate the degree of DevOps adoption is adequate?	5	5
15	Do you consider that the metrics proposed in the model are sufficient to ensure a complete evaluation of DevOps?	5	5
16	Do you think that the results obtained after applying the metrics model will allow a company to identify aspects of improvement in its processes?	5	5
17	According to your experience, do you consider that the proposed metrics model can be applied in companies with infrastructure, capital, personnel or size limitations?	5	5

To establish a broader evaluation that included the comments, perceptions, and suggestions of the participants, two open-ended questions were posed to identify possible improvement actions identified by each of the roles involved in the evaluation. The responses to the open-ended questions are presented in Table 26. According to the results, participants from both companies had a favorable perception regarding the definition of practices, dimensions, and values. In addition, participants noted the following aspects: (i) the proposal is applicable according to the characteristics of each company; (ii) the results obtained after applying the evaluation allowed them to identify areas for improvement related to continuous learning, continuous experimentation, organizational culture, and the need to establish spaces for educating team members in DevOps; finally, (iii) the results obtained after conducting the evaluation are consistent with the expected outcomes by the evaluation participants. After applying the metrics model, participants from both companies expressed that the evaluation model is useful and provides valuable information that will enable the establishment of continuous improvement actions within each organization.

Table 26. Response to the open-ended questions

Question	Answer	
	Company 1	**Company 2**
18	None	None
19	The model allowed us to see shortcomings that we had not considered to date.	Metrics helped us understand things we didn't know we could improve

4.4. Validity analysis

Next, the analysis of construct validity, external validity, and reliability regarding the results obtained in the case study is described.

- **Construct Validity Analysis**: To maintain construct validity and ensure that the obtained results objectively represent what is intended to be evaluated in accordance with the research questions, an initial meeting was held. During this meeting, the project context, proposed solution, work plan for each case study, and any concerns or ambiguities expressed by the stakeholders were addressed. Detailed supporting documentation was also provided to ensure that the designated role in the evaluation had the necessary elements to address any doubts that may arise during the execution of the case study in each company.
- **External Validity**: It was emphasized during the case studies that the obtained results cannot be generalized, as each study was conducted in companies with specific characteristics. Therefore, they cannot be considered a generic evaluation. However, the study can be replicated in companies with different characteristics by following the protocol and set of activities described in this document.
- **Reliability**: To reduce bias and ensure independence between the obtained results and the researchers conducting the case study, the following activities were carried out: (i) definition of criteria for the selection of study subjects, (ii) definition of the field procedure and data collection, (iii) development of detailed documentation of the metrics model to be used in each case study, and (iv) continuous support to the evaluator and participants of the case study during the execution phase.

4.5. Limitations

The following limitations were identified during the execution of the case studies:

- The case study was conducted at two software development companies, so the results cannot be generalized. It is necessary to replicate the study in companies with different characteristics to extend the nature of the results.

- There is a degree of bias associated with the subjectivity of the evaluator when applying the assessment tool. Additionally, the interpretation of the results is subject to a degree of subjectivity by the research team.
- During the execution of the two case studies, it was observed that participants had different levels of theoretical and/or practical knowledge about DevOps concepts, activities, and process elements. As a result, the effort and time dedicated to conducting the DevOps evaluation in each company varied.

5. Limitations of the proposed solution

The proposed model demonstrated its effectiveness in two independent case studies, providing valuable insights into the evaluation of DevOps practices. However, some limitations must be acknowledged to ensure a comprehensive understanding of its applicability and potential areas for improvement:

- While the model successfully assessed DevOps maturity in the selected companies, its applicability to organizations with significantly different structures, development methodologies, or operational constraints has not been extensively validated. Additional case studies in companies of varying sizes, industries, and technological landscapes would be necessary to confirm its generalizability.
- The model relies on predefined metrics derived from a harmonization process of multiple frameworks. Although this ensures a solid theoretical foundation, it may not fully capture the unique characteristics or evolving nature of DevOps practices in certain organizations.
- The evaluation process assumes that the available documentation and responses provided by stakeholders accurately reflect the actual implementation of DevOps practices. However, discrepancies between documented policies and real-world execution may introduce bias in the assessment. Future iterations of the model could integrate automated data collection mechanisms, such as monitoring tools or software analytics, to enhance objectivity.
- While the correspondence analysis used to quantify relationships among DevOps elements provided a structured approach for model integration, it does not inherently address the dynamic and iterative nature of DevOps transformations. Organizations often undergo continuous process refinements, and static assessments may not fully capture the incremental improvements over time. Establishing periodic evaluations and feedback loops could strengthen the model’s ability to track DevOps maturity evolution.

Despite these limitations, the proposed model offers a structured approach to evaluating DevOps practices and provides a solid foundation for further refinement. Future research should focus on addressing these challenges to enhance the model’s robustness and applicability across a broader range of organizations.

6. Conclusions and future work

DevOps has become a widely adopted culture in the software industry as it enables the continuous delivery of new software functionalities through the automation of all activities throughout the application development lifecycle. In this regard, DevOps has facilitated the enhancement of monitoring activities, early diagnosis, and incident reporting in production environments. However, as mentioned in previous sections, companies face difficulties in assessing the adoption status of DevOps practices in their projects. With the aim of supporting the DevOps evaluation process in the software industry, this article presented a metrics model that seeks to facilitate the task of an expert evaluator through a total of 11 metrics defined following the formalism for metrics definition in the context of software engineering defined by the Goal-Metric-Question (GMQ) model. The elements proposed in the metrics model resulted from the following activities: (i) the execution of a systematic literature mapping to identify the necessary elements for the definition of a metrics model [41], (ii) a harmonization process of multiple DevOps evaluation models carried out to standardize, unify, and clarify DevOps-related practices based on a reference model, (iii) the definition of 11 metrics to assess the proposed DevOps practices, dimensions, and values and (iv) the evaluation of the model through a focus group composed of DevOps experts. This article presented the results obtained after evaluating the metrics model at two software development companies, thereby validating its applicability in the software industry. The application of the two case studies allowed us to gather the opinions, suggestions, and comments of each participant regarding the applicability, completeness, comprehensibility, and suitability of the metrics model, thus establishing improvement actions for the model.

Subsequently, each participant was asked to answer the questions presented in the evaluation template. Additionally, each company was requested to provide the necessary evidence to determine that the response to each question had a clear and consistent justification to certify its level of compliance. The effort applied during the execution stage of the evaluation was documented, detailing the hours dedicated by each participant and company. The results obtained after conducting the evaluation in each company were analyzed to establish a comprehensive view from multiple perspectives that allowed the identification of relationships, differences, and commonalities regarding the degree of implementation of DevOps practices, dimensions, and values in each company. The total number of hours presented in the execution stage corresponds to the hours applied during the business days of the evaluation in each company. Since the participants had other assignments related to the projects they were assigned to, they dedicated different amounts of time each day to carry out the evaluation, ensuring that it did not delay their previous assignments. Each participant documented the number of hours they dedicated each day to determine the exact number of hours spent during the execution of the case study. Based on this information, it was observed that Company C1 completed the evaluation over seven business days, with varying levels of individual effort among its participants. Similarly, Company C2 required nine business days, with a higher overall effort recorded due to a more extensive involvement of its participants. The effective effort in hours demonstrated that, despite differences in duration, the intensity of participation varied, influencing the time required for completion.

Although these results are limited to two companies, the observed differences between them indicate that the nature of the evaluation process can vary depending on the organizational structure, resource availability, and internal workflows of each company. This suggests that while the evaluation model provides a standardized framework, its application may need to be adjusted to accommodate the specific characteristics of different organizations.

According to the comments made by the participants, the metrics model had a high level of acceptance, recognizing its importance and relevance. Additionally, the participants stated that the results obtained allowed each company to establish improvement actions in their internal practices. Furthermore, the participants determined that the total cost of applying the model is relatively low and provides insight into the adoption status of DevOps practices within each company. The evaluation allowed for a starting point for improvement actions or decisions that will refine the DevOps process in each company according to their characteristics.

Regarding possible future initiatives and work, the following activities were identified: (i) while the study was conducted in two companies, it is necessary to extend the evaluation to reflect the diversity of how DevOps is applied in different contexts. In this sense, conducting additional case studies in companies with different characteristics will help determine the applicability of the metrics model in various contexts, (ii) applying artificial intelligence techniques to refine the evaluation process and reduce human intervention, (iii) combining the proposal with existing certification models to extend the scope of the metrics model to other process areas present throughout the software project lifecycle, (iv) supporting the evaluation process with generative artificial intelligence techniques to enhance the evaluation process through predictive analysis techniques that refine recommendations made to companies in the future, (v) conducting future studies to determine the degree of correlation between the metrics model and the engineering capabilities present in software companies, including architecture, quality, management, on-site reliability, and security, among others, and (vi) Conducting a comprehensive review and update of the referenced literature to ensure alignment with the latest advancements in DevOps. While the current study incorporates foundational and widely recognized sources, future work will prioritize integrating more recent publications to reflect emerging trends, evolving best practices, and the latest research findings in the field. This will enhance the relevance and timeliness of the proposed model, ensuring it remains adaptable to the continuous evolution of DevOps methodologies.

Conflict of interest statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

[1] H. Conradi and A. Fuggetta, "Improving software process improvement," *IEEE Software*, vol. 19, no. 4, pp. 92–99, Jul. 2002, doi: 10.1109/MS.2002.1020295.

[2] K. Beck et al., "Manifesto for Agile Software Development."

[3] J. Guerrero, K. Zúniga, C. Certuche, and C. Pardo, "A systematic mapping study about DevOps," *Journal de Ciencia e Ingeniería*, vol. 12, no. 1, pp. 48–62, 2020.

[4] P. Debois, "Devopsdays - Organizing Guide," 2009.

[5] D. Edwards, "What is devops," *Retrieved*, vol. 3, no. 2014, p. 5, 2010.

[6] M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017.

[7] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "Relationship of DevOps to agile, lean and continuous deployment," in *International Conference on Product-Focused Software Process Improvement*, Springer, 2016, pp. 399–415.

[8] J. Michelsen, "Dysfunction Junction: A Pragmatic Guide to Getting Started with DevOps," *CA Technologies*, p. 26, 2014.

[9] C. Orozco, C. Pardo, S. Vásquez, H. Ordoñez, and E. Suescún, "An agile process to support software configuration management," *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, vol. 2020, no. E32, 2020.

[10] M. Virmani, "Understanding DevOps & Bridging the Gap from Continuous Integration to Continuous Delivery," in *International Conference on Innovative Computing Technology*, 2015, pp. 78–82.

[11] S. S. Samarawickrama and I. Perera, "Continuous Scrum: A Framework to Enhance Scrum with DevOps," in *Int. Conf. Adv. ICT for Emer. Reg.*, IEEE, Sep. 2017, pp. 1–7.

[12] Digital.ai, "15th Annual State of Agile Report." Accessed: May 16, 2022. [Online]. Available: https://bit.ly/3Lgw4KE

[13] Business Research Company, "State of SRE Report," Sep. 2022.

[14] J. Wettinger, V. Andrikopoulos, and F. Leymann, "Automated Capturing and Systematic Usage of DevOps Knowledge for Cloud Applications," in *Int. Conf. on Cloud Eng.*, 2015, pp. 60–65.

[15] M. E. F. A. C. & Daneva, “Report: DevOps Literature Review,” 2014. doi: 10.13140/2.1.5125.1201.

[16] J. Guerrero, “DevOps Model - Modelo de referencia para la adopción de DevOps en empresas de desarrollo de software,” Universidad del Cauca, 2021.

[17] C.-E. Orozco-Garcés, C.-J. Pardo-Calvache, and Y.-H. Salazar-Mondragón, “What is There About DevOps Assessment? A Systematic Mapping,” *Revista Facultad de Ingeniería*, vol. 31, no. 59, pp. e13896–e13896, 2022.

[18] V. R. B. G. Caldiera and H. D. Rombach, “The Goal Question Metric Approach,” *Encyclopedia of Software Engineering*, pp. 528–532, 1994.

[19] M. Muñoz et al., “Analysis of Tools for Assessing the Implementation and Use of Agile Methodologies in SMEs,” in *Int. Conf. on Soft. Proc. Impr. Cap. Det.*, Springer, 2016, pp. 123–134.

[20] A. Mishra and Z. Otaiwi, “DevOps and software quality: A systematic mapping,” *Comput. Sci. Rev.*, vol. 38, p. 100308, 2020.

[21] G. Rong, H. Zhang, and D. Shao, “CMMI guided process improvement for DevOps projects: an exploratory case study,” in *Proc. of Int. Conf. on Soft. Syst.*, 2016, pp. 76–85.

[22] M. Gasparaite and S. Ragaisis, “Comparison of DevOps maturity models,” in *IVUS*, 2019, pp. 65–69.

[23] M. Zarour et al., “A research on DevOps maturity models,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 4854–4862, 2019.

[24] C. Marnewick and J. Langerman, “DevOps and Organisational Performance: The Fallacy of Chasing Maturity,” *IEEE Software*, 2020.

[25] R. Feijter et al., “Towards the Adoption of DevOps in Software Product Organizations: A Maturity Model Approach,” 2017, UU BETA ICS Department Informatica.

[26] L. König and A. Steffens, “Towards a Quality Model for DevOps,” *Continuous Software Engineering & Full-scale Software Engineering*, vol. 37, 2018.

[27] S. Kruis, “Designing a Metrics Model for DevOps at Philips IT,” Master’s thesis, Eindhoven University of Technology, 2014.

[28] L. Prates et al., “DevSecOps Metrics,” in *Eur. Symp. Syst. Ana. and Des.*, Springer, 2019, pp. 77–90.

[29] P. Batra and A. Jatain, “Measurement Based Performance Evaluation of DevOps,” in *2020 Int. Conf. Comp. Perf. Eval.*, IEEE, 2020, pp. 757–760.

[30] P. Rittgen, S. Cronholm, and H. Göbel, “Towards a Model for Assessing Collaboration Capability Between Development and Operations,” in *Eur. Conf. Soft. Proc. Impr.*, Springer, 2019, pp. 111–122.

[31] T. Masombuka and E. Mnkandla, “A DevOps Collaboration Culture Acceptance Model,” in *Proc. of Ann. Conf. of South African Inst. Comp. Sci. Inf. Tech.*, 2018, pp. 279–285.

[32] J. M. Radstaak, “Developing a DevOps Maturity Model: A Validated Model to Evaluate the Maturity of DevOps in Organizations,” 2019, University of Twente.

[33] D. Teixeira et al., “A Maturity Model for DevOps,” *International Journal of Agile Systems and Management*, vol. 13, no. 4, pp. 464–511, 2020.

[34] T. Neubrand and T. Haendler, “Development of a GQM-based Technique for Assessing DevOps Maturity,” 2020.

[35] R. de Feijter et al., “DevOps Competences and Maturity for Software Producing Organizations,” in *Ent. Proc. and Inf. Syst. Mod.*, Springer, 2018, pp. 244–259.

[36] T. Seppä-Lassila, A. Järvi, and S. Hyrynsalmi, “An Assessment of DevOps Maturity in a Software Project,” *Computer Science*, 2017.

[37] R. Costa, R. Rodrigues, and A. C. S. Dutra, “Application of Scrum Maturity Model in SoftDesign Company,” in *Brazilian Workshop on Agile Methods*, Springer, 2016, pp. 39–49.

[38] A. Kumar, M. Nadeem, and M. Shameem, “Assessing the Maturity of DevOps Practices in Software Industry: An Empirical Study of HELENA2 Dataset,” in *Proc. of the Int. Conf. on Eval. and Assess. in Soft. Eng.*, 2022, pp. 428–432.

[39] M. Anisetti et al., “A Continuous Certification Methodology for DevOps,” in *11th Int. Conf. Man. Dig EcoSys.*, 2019, pp. 205–212.

[40] ATOS, “DevOps Maturity Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3uTbPve

[41] Microsoft, “Microsoft DevOps Self-Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/2RZCHLz

[42] Infostretch, “Infostretch DevOps Self-Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3fh4kr

[43] InCycle, “InCycle Evaluación de DevOps.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/2RqYQ

[44] Xmatters, “DevOps Maturity Survey Report.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/33N8iC

[45] Atlassian, “DevOps Maturity Model.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/2Rq1o3N

[46] IBM, “IBM DevOps Practice Self Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://ibm.co/3w2bWE

[47] IVI, “IVI’s DevOps Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3w9LGZ

[48] Veritis, “Veritis.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3yhZhQ

[49] Boxboat, “Boxboat.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3yqsDM

[50] Humanitec, “DevOps Assessment.” [Online]. Available: https://humanitec.com/devops-assessmen

[51] Atlassian, “DevOps Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3fChUpB

[52] Eficode, “Eficode DevOps Assessment.” Accessed: Mar. 29, 2021. [Online]. Available: https://bit.ly/3omPkfD

[53] C.-E. Orozco-Garcés, C.-J. Pardo-Calvache, and E. Suescún-Monsalve, “Metrics Model to Complement the Evaluation of DevOps in Software Companies,” *Revista Facultad de Ingeniería*, vol. 31, no. 62, pp. e14766–e14766, 2022.

[54] S. H. Kan, *Metrics and Models in Software Quality Engineering*. Addison-Wesley Professional, 2003

[55] C. Pardo et al., “A Process for Driving the Harmonization of Models,” in *Proc. of the 11th Int. Conf. on Product Focused Software*, 2010, pp. 51–54.

[56] C. Pardo, F. J. Pino, and F. Garcia, “Towards an Integrated Management System (IMS), Harmonizing the ISO/IEC 27001 and ISO/IEC 20000-2 Standards,” *International Journal of Software Engineering and Its Applications*, vol. 10, no. 9, pp. 217–230, 2016.

[57] C. Pardo et al., “PrMO: An Ontology of Process-reference Models,” 2012.

[58] N. Tomas, J. Li, and H. Huang, “An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps,” in *Int. Conf. Cyb. Sec. Prot. Dig. Serv.*, IEEE, 2019, pp. 1–8.

[59] A. Caprarelli, E. di Nitto, and D. Tamburri, “Fallacies and Pitfalls on the Road to DevOps: A Longitudinal Industrial Study,” in *Int. Work. Soft. Eng. Asp. Cont. Dev. Parad.*, Springer, 2019, pp. 200–210.

[60] Google Cloud, “DORA DevOps Quick Check.” Accessed: Apr. 21, 2022. [Online]. Available: https://www.devops-research.com/quickcheck.html#questions

[61] F. J. Pino et al., “Harmonizing Maturity Levels from CMMI-DEV and ISO/IEC 15504,” *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 22, no. 4, pp. 279–296, 2010.

[62] C. Pardo et al., “Integrating Multiple Models for Definition of IT Governance Model for Banking ITGSM,” *International Business Management*, vol. 10, no. 19, pp. 4644–4653, 2016.

[63] F. J. Pino et al., “Assessment Methodology for Software Process Improvement in Small Organizations,” *Information and Software Technology*, vol. 52, no. 10, pp. 1044–1061, 2010.

[64] Unión Europea, “Reglamento 651/2014 de la Comisión del 17 de junio de 2014,” 2014.

[65] Unión Europea, “Recomendación de la Comisión del 6 de mayo de 2003 sobre la definición de microempresas, pequeñas y medianas empresas,” 2003.

[66] Agility Health, “DevOps Health Radar Assessment.” Accessed: Apr. 21, 2022. [Online]. Available: https://agilityhealthradar.com/devops-health-radar-assessment

[67] R. K. Yin, *Case Study Research: Design and Methods*, vol. 5. Sage, 2009.

[68] M. Shahin, M. A. Babar, and L. Zhu, “Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices,” *IEEE Access*, vol. 5, pp. 3909–3943, 2017

[69] Payscale, “Payscale - Salary Comparison, Salary Survey, Search Wages.” Accessed: Apr. 10, 2022. [Online]. Available: https://www.payscale.co

[70] R. Likert, “A Technique for the Measurement of Attitudes,” *Archives of Psychology*, 1932.