Received April 11, 2022, accepted April 27, 2022, date of publication May 2, 2022, date of current version June 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3171803

Applying and Researching DevOps: A Tertiary Study

ELVIRA MARIA ARVANITOU[1], APOSTOLOS AMPATZOGLOU[1], STAMATIA BIBI[2], ALEXANDER CHATZIGEORGIOU[1], AND IGNATIOS DELIGIANNIS[3]

[1]Department of Applied Informatics, University of Macedonia, 546 36 Thessaloniki, Greece
[2]Department of Electrical and Computer Engineering, University of Western Macedonia, 501 00 Kozani, Greece
[3]Department of Information and Electronic Engineering, International Hellenic University, 574 00 Thessaloniki, Greece

Corresponding author: Elvira Maria Arvanitou (e.arvanitou@uom.edu.gr)

This work was supported by the European Union's H2020 Research and Innovation Program (SmartCLIDE) under Grant 871177. The work of Elvira Maria Arvanitou was supported in part by the Action ''Strengthening Human Resources Research Potential via Doctorate Research'' of the Operational Program ''Human Resources Development Program, Education and Lifelong Learning, in 2014–2020,'' implemented from the State Scholarship Foundation (IKY); and in part by the European Social Fund and the Greek Public (National Strategic Reference Framework (NSRF), in 2014–2020).

ABSTRACT DevOps is an emerging software development methodology, that differs from more traditional approaches due to the closer involvement of the customer and the adoption of ''*continuous-∗*'' (e.g., integration, deployment, delivery, etc.) practices. The vast research on DevOps (including numerous secondary studies) published in a short timeframe, and the diversity of the authors' research backgrounds (e.g., from a *Dev* or an *Ops* perspective), has inevitably produced a long list of investigated topics, which use inconsistent terminology. The goal of this study is to analyze literature reviews on DevOps with respect to: (a) the research topics in DevOps; (b) the terms that are mapped to each topic; and (c) the consistency of terminology. To achieve this goal, we have performed a tertiary study, i.e., a systematic mapping study that uses as primary studies ''*Systematic Literature Reviews*'' and ''*Mapping Studies*''. For Data Extraction, Analysis, and Synthesis (DEAS) we propose a novel approach relying on *thematic analysis*, *statistical analysis*, and *meta-analysis*. The results unveiled 7 core topics on DevOps research, out of which *DevOps features* and *DevOps practices* are dominant ones. Additionally, as expected various terminology ambiguities have been identified, most between features as practices, as well as, between challenges faced before adopting DevOps and while applying DevOps. The main contribution of this study is the disambiguation of the mapping of terms to topics. Along this process we highlight both inconsistencies—attempting to resolves ambiguities, as well as topics and terms with high levels of consistency; aiding researchers and practitioners.

INDEX TERMS DevOps, software development process, software engineering, tertiary study.

I. INTRODUCTION

In recent years, DevOps has emerged as one of the most ''modern'' and widely discussed terms in the field of software engineering [1], [2]. DevOps aims to bridge the gap between development and operations, in the sense that it promotes the collaboration between the development teams (i.e., designers, developers, testers, etc.) with businesses, creating additional teams that are responsible for developing, managing and supporting the performance of customer-side systems [3]. By definition the DevOps methodology builds a living bridge between development teams (DEVelopment) and system users (OPeration), enabling them to collaborate efficiently and seamlessly. However, due to the emerging and dual nature of DevOps, several challenges appear both in industry and academia.

First, the literature contains a vast number of proposed approaches for DevOps (see list of primary studies), attempting to cover the complete range of the software lifecycle from requirements to development, deployment, and maintenance. Such approaches are introduced by researchers from different backgrounds, rendering the comprehensive knowledge and understanding of the literature unrealistic [3]. Thus, there is a need for a detailed research panorama of the DevOps research area, in the form of usually studied topics (***need-1***). Second, there are tentative misunderstandings between the ''Dev'' and the ''Ops'' teams, originating mostly from their business goals (e.g., understand requirements, develop software,

The associate editor coordinating the review of this manuscript and approving it for publication was Weizhi Meng.

sell software, maintain software, compile bug reports), the artifacts they use (e.g., source code, designs, working software, customer complaints), as well as the different background of their members (e.g., architects, testers, customers, customer support). As a consequence, different stakeholders might use the same terminology targeting at a different concept, and vice versa [1], [4]. For instance, the term "complexity" for a "Dev" team would probably refer to the complexity of the code; whereas for the "Ops" team it would refer to the complexity of using the software. For this reason, there is a need to propose a consolidated terminology for DevOps, which would be understandable from all stakeholders (***need-2***). To alleviate the aforementioned problems, we have performed a tertiary study on DevOps that aims at: (a) *providing a **unified research landscape for DevOps*** (related to *need-1)*. The unified research landscape will include a list of topics and terms that have emerged from the literature. This list will act as a dictionary of terms that can be used in future research and in practice; *and* (*b*) *exploring the **consistency of terminology** within the specific topics* (related to *need-2)*. To achieve this goal, we will contrast the use of terms within and between topics, so as to identify terms that are used in a different manner across studies. Identifying and consolidating confusing terms can lead to more accurate presentation of future research ideas, as well as reduce the misunderstandings in an industrial setting.

The rest of the paper is organized as follows: In Section II, we discuss related work that is relevant to DevOps area, whereas, in Section III, we present the adopted systematic mapping protocol. Next, in Section IV we present the results of our study, and in Section V we present the discussion and the threats to validity, and in Section VI we conclude the paper.

II. BACKGROUND INFORMATION & RELATED WORK

In this section we present background information and related work for this paper. In particular, in Section II.A, we present background information on DevOps, so that the non-expert reader can be acquainted to the basics of DevOps. In the software engineering literature, there are no tertiary studies on DevOps; thus, in Section II.B we expand our related work discussion, on tertiary studies for software processes. Nevertheless, we need to note that the findings of papers presented in Section II.B are not directly comparable to ours, since their scope and context are different.

A. BACKGROUND INFORMATION

The term DevOps was introduced in the 2007-2009 period and it represents the combination of Development (Dev) and Operations (Ops) [5]. More specifically, DevOps methodology [6] is the extension of agile processes (such as: Scrum, XP, etc.), having as a primary goal the promotion of good collaboration between development and operation teams, throughout the software development lifecycle, from design until the delivery of the product to the customer. The success of the DevOps methodology in an organization, is monitored by considering three key areas of evaluation: the culture of collaboration, processes and tools [7]. People are at the heart of the methodology, and in many cases the main obstacle to cultivating the culture of the DevOps methodology in an organization. Adopting the methodology can lead to a complete transformation of product development processes. In addition, monitoring the effectiveness of DevOps interconnected processes throughout the pipeline of product manufacturing and delivery is considered particularly relevant. The selection of tools can also play an important role in the correct and effective implementation of the DevOps methodology. Typical factors to be monitored are operating time and capacity among others. Furthermore, based on the literature, Culture, Automation, Measurement, Sharing, and Services have been identified as the main dimensions of DevOps [8]. DevOps encourages Continuous Integration (CI) and Continuous Delivery (CD). It aims at shortening the product delivery cycle, enabling enterprises to timely launch software products and services without compromising their quality.

B. RELATED WORK

Hoda *et al.* [9] performed a tertiary study in agile software development (ASD). More specifically, the study focused on identifying: (a) the number of SLRs that have been published; (b) the research areas and topics of interest; (c) the venues that are most active in publishing SLR in the domain of ASD; (d) the quality of the SLRs in ASD; and (e) the progress that have been achieved in ASD research. The search process covers the time frame from 1990 to December 2015 in five digital libraries (IEEE, ACM, Springer, ScienceDirect, and ISI Web of Science). After applying the selection criteria, 28 studies were identified. As a result, the study revealed ten different ASD research areas: adoption, methods, practices, human and social aspects, CMMI, usability, global software engineering, organizational agility, embedded systems, and software product line engineering.

Hanssen *et al.* [10] conducted a tertiary study on global software engineering. More specifically, the aim of this study is to identify the current trends and the role of agile topics in the global software engineering literature. The search process is applied in ISI Web of Science and Google Scholar and returned 21 studies. The results of the study suggest that agility is one of the topics that are attracting attention in the global software engineering area. Additionally, Curcio *et al.* [11] performed a tertiary study on the usability of agile software development. In particular, the goal of this study is to categorize secondary studies related to the co-existence of usability and agile software development, and discuss the quality of the selected studies. The search process is conducted between 2001 and February 2018 in four digital libraries (IEEE, ACM, Springer, and ScienceDirect). After applying the selection criteria, 14 studies were identified. The results identify six main categories for representing ways of integrating usability into agile development: processes, techniques, practices, recommendations, principles and different approaches. Additionally, regarding the challenges for the integration, the authors identified seven main categories: issues related to tests, time, work balance, modularization, feedback, prioritization, and documentation.

Nurdiani *et al.* [12] conducted a tertiary study of Agile and Lean practices. More specifically, the goal of this study is to identify the impact of Agile and Lean practices on project constraints. The search process is performed in five digital libraries (Compendex & Scopus, Inspec, IEEE, ACM, and ISI Web of Knowledge) between 1990 and 2014. 41 secondary studies were retrieved and analyzed. The results of this study highlighted 13 Agile and Lean practices as the most prominent ones, whereas Test-Driven Development (TDD) is studied in ten secondary studies, concluding that it has a positive impact on external quality. Finally, Khan *et al.* [13] performed a tertiary study on software process improvement. In particular, the aim of this study is to identify the software process improvement topics that have been discussed in the secondary studies; and the quality of these articles. The search strategy identified papers between 2004 and October 2015, and was conducted on five databases (IEEE, Scopus, Google Scholar, ScienceDirect and ISI Web of Science). At the end of the selection process, 24 secondary studies were retained. The results suggest that the secondary studies focus on five topics, i.e., factors, small and medium-sized enterprises (SMEs), process models, software quality, and testing. Factors and process models were the most common topics in software process improvement.

III. TERTIARY STUDY DESIGN

The first tertiary study on software engineering has been published by Kitchenham *et al.* [14] in 2010. In that work, Kitchenham *et al.* [14] adopted the guidelines for performing systematic literature review. Nevertheless, given the goals of this work, we preferred to perform our tertiary study by applying the Systematic Mapping process. In this section, we present the protocol of the systematic mapping study, based on the guidelines described by Petersen *et al.* [15]. A protocol constitutes a plan that describes the investigated research questions and how the mapping study has been conducted. More specifically, the protocol involves three activities, namely: (a) defining research objectives and questions—see Section III.A, (b) defining search and article selection process—see Section III.B, and (c) defining data extraction, analysis and synthesis strategy—see Section III.C.

A. RESEARCH OBJECTIVES AND QUESTIONS

The goal of this study, expressed in the Goal-Question-Metrics (GQM) format [16], is to *analyze* existing literature reviews on DevOps for *the purpose of* characterization and evaluation *with respect to*: (a) the research topics in DevOps area; (b) the terms that are mapped to each topic; and (c) the consistency among terms and within and between topics, *from the point of view of* researchers and practitioners. Based on this goal, we define the following research questions.

RQ$_1$: *What are the most common research topics in the DevOps domain?*

RQ$_2$: *What terms can be mapped to each research topic?*

RQ$_3$: *Is the terminology used in the DevOps research consistent?*

To answer RQ$_1$, we have identified the topics that have been most frequently studied. The topics have been retrieved from the research questions of each secondary study (e.g., benefits for adopting DevOps, used practices, definition of DevOps). Next, for each topic, (to answer RQ$_2$) we recorded all the answers (terms) for each research question, for each secondary study. Finally, to answer RQ$_3$, we compared all the terms of each topic, to identify the consistency of the terminology within each topic. Inconsistency in terminology exists in two forms: (a) a term defining two meanings; or (b) a meaning represented by more than one term. By answering these research questions, the industrial and academic stakeholders could easily identify the most interesting and active topics in the area of DevOps, as well as a consolidated terminology that can contribute towards the avoidance of possible misunderstandings.

B. SEARCH AND ARTICLE SELECTION PROCESS

defined by considering the goal and research questions of the study. In Figure 1 we provide an overview of the process along with the number of studies retrieved at each phase.

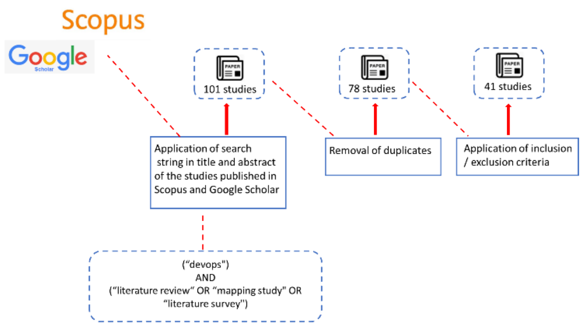

Scopus
Google Scholar
101 studies
78 studies
41 studies
Application of search string in title and abstract of the studies published in Scopus and Google Scholar
Removal of duplicates
Application of inclusion / exclusion criteria
("devops") AND ("literature review" OR "mapping study" OR "literature survey")

FIGURE 1. Overview of search and article selection process.

Search Process: More specifically, we have selected to perform an automated search of the complete content of two well-known indexing engines i.e., ***Google Scholar*** and ***Scopus*** and not in specific venues, so as to not exclude studies that are relevant to our work. First, we developed a search string (see box below) to identify studies relevant to DevOps and applied this search on the title and abstract of the papers. We note that for the first part of the search string we have not employed any synonyms, since the term is very distinct and researchers referring to DevOps do not use alternatives. This search has returned 101 candidate studies.

("devops") AND ("literature review" OR "mapping study" OR "literature survey")

Article Selection Process: Next, we removed duplicate papers (78 articles remained). As a last step of this process, we identified all the primary studies satisfying the Inclusion Criteria (IC). First, it was mandatory to assess the study as

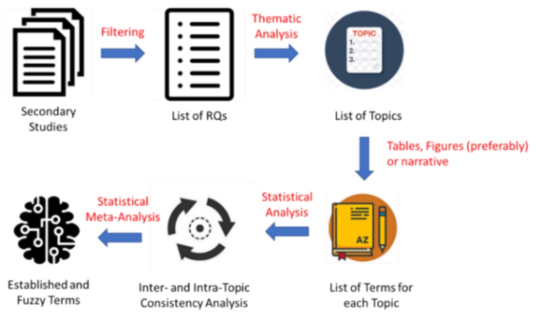

Secondary Studies
Filtering
List of RQs
Thematic Analysis
TOPIC
List of Topics
Tables, Figures (preferably) or narrative
Statistical Analysis
List of Terms for each Topic
Inter- and Intra-Topic Consistency Analysis
Statistical Meta-Analysis
Established and Fuzzy Terms

FIGURE 2. **Data extraction, analysis, and synthesis process.**

relevant to DevOps domain and then as secondary study. The Inclusion Criteria of our tertiary study are:

IC1—The study deals with the DevOps domain. AND

IC2—The study is a secondary study (i.e., literature review, mapping study, or literature survey).

The Exclusion Criteria of our tertiary study are:

EC1—The study is written in a language other than English;

EC2—The study is an editorial, keynote, biography, opinion, tutorial, workshop summary report, progress report, poster, or panel.

In the final dataset, we kept studies that satisfied both IC1 and IC2, and did not satisfy any Exclusion Criteria (EC). The article selection process has been handled by the first three authors of this study, using a simplified version of the voting method, as described by Farhoodi *et al.* [17]. In particular, the first three authors inspected the publication's full text and assigned a binary vote (include / exclude). Studies with 3 include votes have been included in the study, whereas studies with 3 exclude votes have been automatically excluded. The inclusion of the rest primary studies has been discussed in plenary. In total, since the level of clarity for the inclusion/exclusion was high, only 4 articles have been discussed. At the end of this process, the final set of primary studies was comprised of 41 secondary studies.

C. DATA EXTRACTION, ANALYSIS, AND SYNTHESIS

In this section we present the data extraction, analysis, and synthesis process that we have used for answering the RQs. The proposed process relies heavily on synthesis and meta-analysis methods that are applied in the software engineering domain, as presented by Cruzes and Dybå [18], dos Santos and Travassos, [19], and Kitchenham *et al.* [20]. In Figure 2, we provide an overview of the proposed Data Extraction, Analysis, and Synthesis (DEAS) process that can be applied to any tertiary study that aims at building a dictionary of a field of research, comprised of topics and associated terms, exploring their consistent usage among secondary studies.

As a 1st step, we ***extract all the research questions*** that are answered in secondary studies, compiling a list of research questions that are of interest in the field (e.g., "*What are the main concepts related to DevOps?*", "*What are the main expected benefits and challenges of adopting DevOps?*")—***the research questions are noted exactly as they appear in the study***, without any interference of researchers. In case (a rare one) that a study does not have explicitly stated research questions, we use the goal. If no goals exist, we use the goal based on the organization of the results section. We note that since in this work we rely on the systematic mapping study process, no quality appraisal has been performed. For instance, using DARE would have excluded studies without RQs [21].

As a 2nd step, we perform thematic analysis[1] so as to consolidate a ***list of topics of research interest***. To achieve this, we first extract a topic for each RQ (e.g., feature and benefits and challenges, according to the previous examples); and subsequently we merge similar topics together (e.g., we merge areas and features under the same topic, named features). To extract topics, we used ***open coding*** [22]. In particular, we examined the text of the RQs, subdivided them into words, and labeled the important ones with codes. When possible, codes are generated as words, "*as-are*" in the RQ. Otherwise, "*synthetic*" codes representing the semantic meaning of the research topic were created by the researchers. Next, topics were clustered into fundamental categories, which guided the future data collection.

In the 3rd step, we build a collection of 2D arrays, in which for every study, we note ***a tuple of terms and topics***. The terms are recorded as presented by the authors of the secondary study in tables or figures that answer the corresponding research question (e.g., Jira, Jenkins, Chef for the topic tool)—without any interference from the researchers. In case that an original (in the secondary study) RQ is not answered in a compact form (quite infrequent: only 5 out of the 41 examined studies), then terms are extracted from the corresponding text. The thematic analysis has been performed by using the Open Card Sorting method, introduced by Spencer [23]. In particular, we: (a) identified "*Consolidated Terms*" (i.e., super-categories)—e.g., we developed a term "Deployment"; (b) we mapped "Original Terms" to the consolidated ones—e.g., we mapped "Continuous Deployment" and "Automated Deployment" to "Deployment"; and (c) defined the final names of the Consolidated Terms, after we mapped all Original Terms. The first and the second author performed the process of identifying the terms, and the third and fourth authors validated the results. During the process of consolidating terms, along with their naming, there were some disagreements (approximately 2%), which have been resolved by a discussion among the authors. The low rate of disagreement was reached, by applying a process to guarantee the common understanding of researchers. In particular, first a thorough discussion among authors was performed. Next, we piloted the first 10 papers, which have been assessed in pairs by the three authors, so as to have an open discussion on the recording of variables' scores. All authors explained their scores, until a consensus was reached.

[1]According to Cruzes and Dyba [4] thematic analysis is defined as a method for identifying, analyzing, and reporting themes within data.

Upon the completion of these steps, the following data are extracted for every secondary study:

[V1] Title: title of the paper.

[V2] Author: list of authors of the paper.

[V3] Year: publication year of the paper.

[V4] Type of Paper: conference or journal.

[V5] Publication Venue: name of the journal or conference.

[V6] Used Approach: the type of study (i.e., SLR/SMS).

[V7] Topics: the topics studied in each secondary study.

[V8] Terms: consolidated / original terms.

[V1] to [V6] are collected for documentation purposes. [V7] is an array of the topics that are studied in the secondary study (outcome of step 2 of DEAS), and [V8] is a 2D array, for which rows are the topics of [V7] and columns the identified terms (outcome of step 3 of DEAS). To answer RQ_1 and RQ_2, we have produced basic descriptive statistics (i.e., frequencies) and visualization methods (i.e., word clouds) on [V7] and [V8]. The complete dataset is available online.[2]

As a 4th step of DEAS, we perform statistical analysis to answer RQ_3. The statistical analysis explores the usage of the terms for pairs of topics, defined in different studies, calculating a Consistency Factor (CF). Given the set of terms mapped to two topics into two studies, CF is calculated as a fraction of same terms, divided by the size of the largest set. The calculations have been automated with an open-source tool developed by the authors for this purpose. The tool has been substantially tested before performing the data analysis process, following software engineering testing principles. The tool receives as input text files containing the terms and topics explored in each study, and calculates IntraCF and InterCF metrics. In a domain with consistent and well-established terminology, it is expected that same topics (referred in different studies) are having a high CF (i.e., use the same terms); whereas different topics are having a limited overlap in terms (i.e., low CF) is expected.

$$\begin{aligned} terms(topic_i) &= \{term | term \in topic_i\} \\ terms_{Sx}(topic_i) &= \{term | term \rightarrow S_x \cap terms(topic_i)\} \\ CF_{Sx,Sy}(topic_i) &= \frac{|\boldsymbol{termsSx}(\boldsymbol{topici}) \cap \boldsymbol{termsSy}(\boldsymbol{topici})}{|\boldsymbol{max}(|\boldsymbol{termsSx}|, |\boldsymbol{termsSy}|)} \end{aligned}$$

where:

- *terms(topic$_i$)* denote the terms that are used for *topic $_i$*
- S_x, S_y are two studies for which the consistency of terms on *topic $_i$* is calculated.
- $terms_{Sx}(topic_i)$ denote the terms used by study S_x for *topic $_i$*
- $|terms_{Sx}(topic_i)|$ number of terms used by study S_x for *topic$_i$*

For example, suppose that study S13 notes as *SE* practices the following set: {patterns, quality, planning, coding, testing, release, deployment, monitoring, operation}, whereas S14 the {continuous integration, continuous delivery, deployment, release} set. For this case, $CF_{S13,S14}$(SE practices) equals 2 / 9, i.e., 22%.

[2]https://users.uom.gr/~a.ampatzoglou/aux_material/TS_DevOps.xlsx

As part of the 5th step, we have performed meta-analysis for interpretation purposes. For this step, we synthesize CF values per topic, calculating IntraCF and InterCF. InterCF is calculated as an average of the CF of one topic, against all others. IntraCF for $topic_i$ is the consistency of terms on $topic_i$ between all tuples of studies that are related to $topic_i$. An example of the calculations is presented below:

Suppose that studies **S1-S3** explore the following topics

[S1] topics: {practice}

[S2] topics: {practice, features}

[S3] topics: {practice, features}.

For the sake of illustration consider that **CF** across studies for the aforementioned practices, are as follows:

topics: {practice, practice} study: {S1, S2} CF: 80%

topics: {practice, practice} study: {S1, S3} CF: 60%

topics: {practice, features} study: {S1, S2} CF: 25%

topics: {practice, features} study: {S1, S3} CF: 10%

topics: {features, features} study: {S2, S3} CF: 90%.

IntraCF, use rows with the **same topic**. Thus, rows 1-2 for practice and row 5 for features

$$\begin{aligned} \text{IntraCF}_{\text{practice}} &= (80\% + 60\%)/2 = 70\% \\ \text{IntraCF}_{\text{features}} &= 90\% \end{aligned}$$

InterCF, use rows with ***different topics***, i.e., rows 3 and 4

$$\text{InterCF}_{\text{features}} = \text{InterCF}_{\text{SE Practice}} = (25\% + 10\%)/2 = 17\%$$

To present the outcomes of RQ_3, we have: (a) used frequencies of studies in which a term is used in conflicting topics; and (b) Venn diagrams for visualizing the "grey zones" among topics, i.e., terms used inconsistently.

IV. RESULTS

In this section we present the results of the performed tertiary study, organized by research question. More specifically, in Section IV.A we present the results on the most commonly researched topics in the DevOps area (RQ_1). In Section IV.B, we focus on the terms that are mapped to each topic (RQ_2). Finally, in Section IV.C we present our findings regarding the consistency of the DevOps terminology (RQ_3). We note that in this section we mostly provide raw results, as well as their interpretations, since implications to researchers and practitioners are discussed in Section V.

A. DevOps RESEARCH TOPICS ($RQ_{1.1}$)

To identify the most commonly studied DevOps topics in the secondary literature, we have analyzed the research questions answered by each secondary study, following the DEAS approach (see Section III.C). Upon synthesizing the topics of focus for each research question, we have identified 8 DevOps research topics as the most prominent ones—see Table 1. We note that from the table, we have excluded research questions targeting demographics analysis, e.g., load of research per year, publication venues, top authors, application domains, etc.

The most commonly studied topic is the identification of the ***Practices*** used while applying DevOps, which is

TABLE 1. DevOps topics of research.

Topic	Description	Secondary Studies [a]
Definition	RQs exploring the DevOps literature so as to synthesize a global definition of DevOps, based on the individual definitions of primary studies.	S3, S4, S8, S10, S14, S21, S24, S34, S41
Benefits	RQs that deal with the benefits that can be identified after applying DevOps. The benefits can be related to all viewpoints of DevOps: software product, customer relationship, process improvement, etc.	S8, S10, S20, S21, S22, S24, S25, S32, S38
Challenges	RQs that focus on required activities, problems, etc. identified before the adoption of DevOps. Apart from the term "Challenges", in this topic we also classified RQs mentioning "Readiness Models", "Problems in Adoption", and "Adoption Success Factors".	S4, S10, S15, S17, S19, S20, S23, S26, S30, S31, S33, S36, S40
Features	RQs that describe the features that differentiate DevOps from other development methodologies. In this topic we have classified also RQs mentioning: "Area", since after exploring the answers to such questions, we conclude it that the terms are used for the same purpose.	S1, S3, S5, S10, S11, S15, S18, S21, S22, S24, S29, S35, S37, S38
Practices	RQs that deal with practices (i.e., specific processes or methods) that can be used while applying DevOps. Practices can be related to either software development (e.g., patterns), processes (monitoring), or any other viewpoint of the DevOps methodology. In this topic we have classified also RQs mentioning: "Methods", and "Capabilities".	S4, S5, S6, S7, S8, S9, S10, S12, S13, S14, S16, S17, S18, S19, S20, S21, S22, S24, S25, S30, S31, S32, S34, S38, S40, S41
Problems	RQs that focus on problems, challenges etc. identified while applying the DevOps development methodology (i.e., after adoption).	S7, S8, S10, S20, S21, S22, S24, S28, S29, S30, S31, S38
Quality Characteristics	RQs identifying the quality characteristics that improve upon the application of DevOps, or are of interest while monitoring DevOps. In this topic we have classified also RQs mentioning: "Measurement", and "Metrics".	S7, S9, S13, S17, S18, S25, S32
Tools	RQs cataloguing and discussing the available tools that can be used when applying the DevOps development methodology.	S4, S6, S7, S8, S12, S19, S22, S34, S39

researched in 63% of the secondary studies focusing on DevOps. Such practices can be software engineering ones (e.g., patterns, traceability, etc.) or operations-related (e.g., improved customer satisfaction). The popularity of Practices as a research topic is considered intuitive in the sense that understanding the practices that need to be considered for applying DevOps can lead to the development of practical guides of needed skills, tools, and potential training targets for DevOps industries. Second ranks DevOps ***Features***, which are discussed in 34% of the secondary studies. We need to note that the term DevOps features appears to correspond to a list of foundations of applying DevOps (e.g., Culture, Communication, Sharing, etc.). The popularity of DevOps features can be explained, since various studies targeted at understanding the essentials of DevOps, as well as the differences from other development methodologies, especially, since the DevOps methodology is still not widely established.

The next group of topics relates to problems or benefits while applying DevOps. In particular, the ***Challenges*** topic (studied in 31% of the secondary studies) corresponds to problems that might be faced while mitigating the development methodology from a traditional one to DevOps, as well as the readiness to perform such a migration. Within the topic ***Problems***, we have classified studies (29%) that pose research questions on the problems that industries encounter after adopting DevOps; whereas questions on the obtained ***Benefits*** are addressed by 22% of the studies. This group of topics targets to answer basic questions of DevOps adopters in terms of what to expect from DevOps, and if DevOps adoption is fitting to their organization. Additionally, 22% of the secondary studies attempt to catalogue the ***Tools*** that are used while applying DevOps in industry; as well as, ***Definitions*** of DevOps. Finally, 17% of secondary studies have dealt with specialized gains in terms of the ***Qualities*** that DevOps application can improve.

B. DevOps TERMINOLOGY (RQ_2)

In this section, for every topic identified in RQ_1 we present the terms it comprises. We note that from this analysis, we have excluded "*Definitions*", since only one secondary study [S10] presented a definition of DevOps synthesized from the primary studies; the rest of the studies presented definitions from their primary studies, but did not synthesize. Therefore, no further synthesis from our side was possible. Regarding "*Features*", "*Practices*", "*Tools*", and "*Quality Characteristics*" we present the results in Tables 2-5; whereas regarding "*Benefits*", "*Challenges*" and "*Adoption*" in Figures 3-5.

For tabular data, in the first column we list the (consolidated) terms identified for the topic, in the second column the percentage of secondary studies in which the term is reported as part of this topic, and in the third column the (original) terms as identified in the secondary study. As part of interpretation, we note that consolidated terms with high percentage refer to terms that many studies acknowledge as important. Within those, there are terms which are mentioned with many possible synonyms, a fact that indicates that this term is not uniformly used in the literature, and attention from the community is required. In Table 2, we present the terminology under the topic ***DevOps Features***. Based on our findings, when researchers refer to DevOps features, they (with some certainty—Freq. >50% and low number of synonyms) refer to the need for ***Automation***, ***Sharing***, ***Measurement***, ***DevOps Culture***, and ***Collaboration***. Quality Assessment is referred as a Feature in 50% of the secondary studies, but both as a process (QA) as well as specific quality characteristics, e.g., Resilience, Complexity, Cost,

TABLE 2. Terms of features.

Consolidated Term	Pct,	Original Term
Automation	78%	Automation, Auto generation of Test cases
Sharing	78%	Sharing
Measurement	71%	Measurement, Alternative Metrics
Culture	64%	Culture
Collaboration	64%	Collaboration, Teamwork
Quality Assessment	50%	Quality Assurance, Reduce Complexity and Cost, Resilience, Scale
Communication	36%	Communication, Feedback, Commitment and agreement
Project Management	36%	Governance, Hiring personnel, People, Process, Management
Services	36%	Services, Micro services, Orchestration Framework
Deployment	28%	Continuous deployment, Virtualization
Skills	28%	Skills, Leadership, Social Aspects
Standard	21%	Standard
Structures	21%	Structures
Transparency	21%	Transparency
Agility	21%	Agility, Leanness
Design / Architecture	21%	Open Architecture, Shift left, TDD, Tested Design & Development
Cont. Improvement	14%	Continuous improvement, Improvement cycle
Blameless	14%	Blameless
Delivery	14%	Continuous Delivery
Experimentation	14%	Experimentation
Responsibility	14%	Responsibility
Trust	14%	Trust
Testing	7%	Continuous Testing, Simulation Testing
BDD	7%	BDD
Infrastructure	7%	Cloud Management
Planning	7%	Continuous Planning
Reuse	7%	Reuse Methods
Version Control	7%	Roll back code
Technology	7%	Technology

Scalability. We remind the reader that despite the existence in the list of the quality characteristics, we needed to mention them as features, since they are classified as such in the secondary study. Lower in the list, we can identify terms that are related to more specific software development activities, such as ***Project Management***, ***Service-Based Development***, ***Deployment***, ***Design/Architecture***, etc., which however are not comprehensively classified as DevOps features (Freq. < 40%) and some of them being identified with various naming alternatives.

Regarding ***DevOps Practices*** (see Table 3), a long list of non-consolidated practices has been identified. However, with limited level of synthesis (e.g., "*Continuous Deployment*", "*Automated Deployment*", etc. are consolidated under the term "*Deployment*") 4 practices have occurred in more than 50% of the studies, namely: ***Deployment***, ***Testing***, ***Monitoring***, and ***Quality Assessment***. Among these practices, the first 2 are solely "*Dev*"-oriented, whereas Monitoring and Quality Assessment can be applied to either the "*Dev*" or the "*Ops*" branch of DevOps. The application of these practices seems to be non-negotiable for applying the DevOps process. Next, we can identify practices that appear in more than 30% of the studies, namely: ***Delivery***, ***Continuous Integration***, ***Design / Architecture***, ***Continuous Improvement***, ***Planning***, ***Version Control***, and ***Infrastructure***. Additionally, an interesting observation is that after these DevOps practices, we have spotted ***Project Management***, ***Automation***, and ***Collaboration***, which are defined as DevOps Features, as well—denoting a possible confusion or overlap between the two topics. This also leads us to the conclusion that a DevOps feature (i.e., Collaboration) is very important and needs support by the associated practices (i.e., face-to-face communication, shared responsibility, etc.). Finally, by inspecting Table 3, we can conclude that the terminology used under the topic DevOps Practices is much more diverse, compared to the terminology under DevOps Features, which is considered expected, since Practices pre-existed the DevOps initiative, and therefore the terminology follows the more general terminology of Software Engineering. The last observation denotes that there is common understanding among stakeholders in terms like Deployment, Testing, Planning that are commonly used by Software Engineers unlike more general terms like Collaboration.

With respect to ***tool support***, while applying DevOps, in Table 4 we have mapped the identified tools to the DevOps practice (as mentioned in the secondary study) that they can support. The table is ordered alphabetically, by DevOps practice. By inspecting the results, we can observe that the majority of the tools are related to ***Monitoring***, ***Security***, ***Project Management***, ***Infrastructure Management***, and ***Continuous Integration and Continuous deployment*** (***CI/CD***). By contrasting the results of Tables 3 and 4, we need to note that despite the prevalence of ***Testing*** as a DevOps practice, the tool support for it is limited compared to other practices. This finding can be interpreted either as: (a) lack of tool support; or (b) the existence of so well-established tools for this DevOps practice that can cover the current needs, leading to no requirement for introducing additional ones. We believe that the same arguments apply also to ***Build*** and ***Version Control*** tools.

Finally, regarding ***Quality Characteristics*** (see Table 5) of interest when applying the DevOps, ***Testability*** stands out as the most important one, followed by ***Maintainability***, ***Performance***, and ***Security***. These results comply with the ones on DevOps practices, as well as tool support. The additional evidence on the importance of Testability probably signifies that regarding tool-support there is probably no lack of tools, but that due their significance some well-established solutions monopolize the market and hinder future research. On the contrary, the importance of security, combined with the number of existing tools, suggests that probably this is an open research field and that practitioners have not concluded with the tool support in this direction.

Regarding the rest of the topics (Benefits, DevOps Adoption, and Challenges), we have visualized the main terms in the form of word clouds. In the word clouds, the larger the fonts of a term, the higher the number of papers in which

TABLE 3. Terms of practices.

Consolidated Term	Pct.	Original Term
Deployment	61%	Continuous deployment, automated deployment, Orchestration, Small independent deployment units vertical layering, Safe deployment parameters, Containerization, Containers
Testing	58%	Testing, Automated Testing, Continuous Testing, Stakeholder Participation, A/B testing, Canary releasing for quality testing, Continuous Penetration Testing, Multi-stage Testing, Non-functional Testing, Prioritize defects corrections, Dynamic and systematic analysis before executing unit test
Monitoring	54%	Monitoring, Continuous Monitoring, Automated Monitoring, Automated Dashboard, Continuous Infrastructure Monitoring And Optimization, Continuous Production Monitoring, Real-Time User Monitoring, Automated Performance Monitoring, Post Deployment QA, Continuous App Performance Modeling, Daily Check-In Of System, Developers must be able to access the IT operations incident reports, Update system iterations based on monitors, Display Metrics and Events, DevOps team must provide overall visibility into project scope and release timing to stakeholders, Modeling & Simulation, Notify Unexpected Behaviors, Production Data Analysis, Staging Application virtualization of development and testing environment, Create development sandboxes for minimum code deployment, Development and Deployment, Multi-stage deployment
Quality Assessment	54%	Quality, Analytics, Use of data to guide QA, Code maintainability, Architecture modifiability, Automated code review to support feature experimentation, Display metrics and events, Application severity matrix, Maintainability and scalability of infra, Ops processes, Source code peer review, Standard depth to support granularity levels to check quality of software, Standard quality profiles, Tracking compliance breaches through automated reporting of violations
Delivery	46%	Continuous Delivery, Build Process, Continuous Release, DevOps team must be able to increase release frequency to satisfy business demand, End-to-end application delivery processes to check value streams, Short cycles & incremental change, Release
Continuous Integration	38%	Continuous Integration, Continuous integration infrastructure, Standardized pipelines
Design / Architecture	38%	Designing Architecture, Microservices, Shift Left, Architectural Specification, Architecture Profiling to Dev / Ops Sides of Teams, IaaS / PaaS, Process-Oriented Data Pipelines, Service Models, Automated Documentation, Design
Continuous Improvement	31%	Continuous Improvement, Performance Metrics, Continuous Process Improvement, Data-Driven Improvement, DevOps team optimize SD project based on Behavior-Driven Development and Retro-QA results of a process, Direct Observation for Continuous Improvement, Inject Faults to Measure Quality, Involve Quality Assurance on Initiatives, Statistical Process Control
Planning	31%	Continuous Planning, Short Planning Cycles, Development incident response plans collaboratively, Release Management
Version Control	31%	Change Management, Version Control, DevOps team use central repository for versioning, synchronization of application source code, Source Code Management, Source Control Management, Use of Artifact Management Tools
Infrastructure	31%	Infrastructure as code, Maintainability and scalability of infra, Ops processes, Sharing of resources, Infrastructure standardization within organization, DevOps team must provide self-service and resources management of platform to stakeholders
Collaboration	27%	Continuous collaboration, Promoting team mindset and expertise, Bridging the gap between development and operation teams, Collaborative Culture, Common goals for development and operations, Common metrics for development and operations, Share common tools across teams, Create open channels for communication, DevOps team synchronizes critical services such as transactions, performance, uptime, deployment schedule, run-time costs, version control, and project scope, Collaboration & trust, Incentive to teams, Involve everyone when making adoption decisions, Shared code responsibility, Promoting team, mindset and expertise, Face-to-face communication, Share responsibility, Constructive communication environment
Project Management	27%	Process Standardization, Risk Assessment, Training, Track Development Progress, Role Rotation, Decision-Making in Product Development, Follow Accelerated time to market technology track, Keep variance between development and production to minimum, Radical intermix of Dev- and -Ops professionals in the same team, Product Management, Stable Teams Composition, Self-Organized Teams, Shared code of conduct, a formal roles assignment, and clear and simple processes to help in understanding responsibilities
Security	27%	Security as Code, Security Configuration, Automatic Code Security Testing, Maintenance of a service catalog with tested and certified services, Improved customer satisfaction level by controlling vulnerability and showing future vision, Record / Security, Red-Team and security drills
Automated SE	27%	Automation, Automated Artifact for On-Demand Troubleshooting, Automated Infrastructure Provisioning, Automated Tools Chain, Integrate Development, Test And Operation Tools, Metrics automation support the team's capability of continuous measurement of appropriate metrics, Practice Appropriate tools and technology, Share Common Tools Across Teams, Manage the environment through automation, Use Automated tools chain, Use DevOps tools to automate deployment, build, testing, update, synchronize, continuous deployment of sandbox code
Configuration	23%	Configuration Management, Configuration Automation
Requirements	19%	Stakeholder Participation, Defining Requirements, Development Supporting Operational Requirements, Feature Toggle to Support Distributed Nature Infrastructure, Incremental Operation Features, Lean Requirements at Scale, Feature Flags, Operational Requirements Based on Several Resources, Structured Requirements Management

TABLE 3. ***(Continued.)*** **Terms of practices.**

Communication	19%	Continuous Feedback, Automated Feedback, Proper communication channel for team to give feedback, Adjust objectives streamline to measure communication across silos
Patterns	19%	Architectural Patterns, Branching Pattern to counter check resources and code, Design Patterns
Agility	11%	Agile Methodologies
Prototyping	11%	Prototyping application
Knowledge	11%	Knowledge Exchange / Management, Knowledge Sharing
Building trust	8%	Building trust and share values and goals
Fault Tolerance	8%	Failure recovery without delay, Back-end as a service for failure
Reuse	8%	Reusable artifacts for scalability, Reuse of artifacts
Maturity Modeling	8%	DevOps maturity evaluation model, DevOps maturity ontology
Production Support	8%	Production Support
Elasticity	8%	Elasticity
Behavior-Driven SE	4%	Behavior-Driven Operations
Team Building	4%	Cohesive team work to fill gap during Isolation changes
Integration Env.	4%	Integration build environment

TABLE 4. **Identified tools.**

Practice	#Tools	Tools
Build	6	Glu, Apache Ant, Gradle, Maven, NixOS, Atlassian Bamboo
CI/CD	15	CodeShip, DBMaestro, GoCD, Jenkins, LiquiBase, RedGate, Squid Proxy, TeamCity, Travis, Loggly, Chef, Drone CI, Circle CI, Hudson, Atlassian Bamboo
Code Execution	4	Sell, Ansible, Apache Kafka, AWS CodePipeline
CRM	1	Crowdbase
Deployment	3	Charon, Docker, NixOS
Design / Architecture	1	ABS Modeling Language
Infrastructure	13	AWS, Cloud Router, Foreman, Heroku, IBM Smart Cloud, MongoDB, Open Stack, Open VPN, PagerDuty, Terraform, Vagrant, Apache Zookeeper, SONATA
Issue Tracking	3	Bugzilla, Jira, MantisBT
Monitoring	25	Track and TesTtrack, AppDynamics, AWS Cloud Formation, AWS Cloud-Watch, Cacti, CloudWave, Filebeat, Ganglia, Grafana, Graphite, Graylog, iPerf, AWS Kinesis, Logstash, Apache Mesos, Nagios, Papertrail, PingDom, Prometheus, Sensu, Splunk, SumoLogic, Zabbix, New Relic, Loggly
Project Management	14	Nuclion, Asana, Clarizen, ClearCase, Confluence, DevOpsLang, HipChat, Rake, Redmine, Rocker, SaltStack, Slack, Trello
Quality Assessment	3	JSLint, Omnia, SonarQube
Security	15	ENSURE, Arachni Scanner, Argon, Boundry, Azure DevOps Security Scanner, Etcd, Gauntlt, J-PAKE, OWASP, Snorby threat stack, Snort, ThreadFix, Tripwire, Chef, SONATA
SOA	4	Juju, Nexus, Rancher, Spring Actuator
Testing	8	Cucumber, Apache JMeter, Junit, Mockito, Selenium, TestComplete, Drone CI, New Relic
Version Control	9	Apache Subversion, Assembla, Bitbucket, CVS, Flyway, GIT, Mercurial, Puppet, SVN

they appear. In terms of "*Benefits*" from adopting DevOps most of the studies focus on the "*Ops*" branch (see Figure 3). In particular, as the main benefit they highlight the ***improved customer satisfaction***, which is an intuitive outcome in the sense that the customer is actively involved in the "*Dev*" branch, mostly by ***validating requirements*** in almost real-time. Additional benefits are related to improved ***security control***, since the system is operational from early stages, enabling the run-time security assessment for a longer period. Finally, the agile principles that govern DevOps development enable a ***continuous planning***, which can be updated based on customer requests, as well as, the development resources, leading to a ***better application***.

TABLE 5. **Quality characteristics.**

Consolidated Term	Pct,
Testability	71%
Maintainability	57%
Performance	57%
Security	57%
Deployability	43%
Efficiency	43%
Flexibility	43%
Scalability	43%
Usability	43%
Availability	28%
Modifiability	28%
Portability	28%
Monitorability	28%
Fault Tolerance	28%
Reliability	28%
Reusability	28%

On the other hand, with respect to problems / challenges, in Figure 4 we present the most common challenges in "*Adopting*" DevOps; whereas in Figure 5 the "*Challenges*" faced when applying DevOps. The main challenge before adopting DevOps is the ***selection of and familiarization with the tools*** that will be used while applying DevOps. As can be observed from Table 4, despite the fact that some of the required tools are generic (not necessarily DevOps-specific), others (e.g., related to Monitoring) might be unfamiliar to non-DevOps organizations. Additionally, another aspect that concerns companies is the fitness of the software

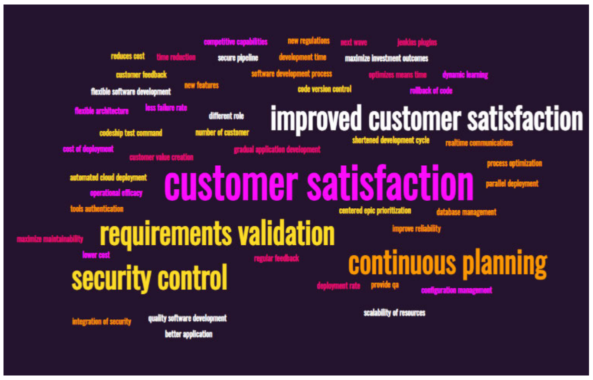


FIGURE 3. Perceived benefits from applying DevOps.

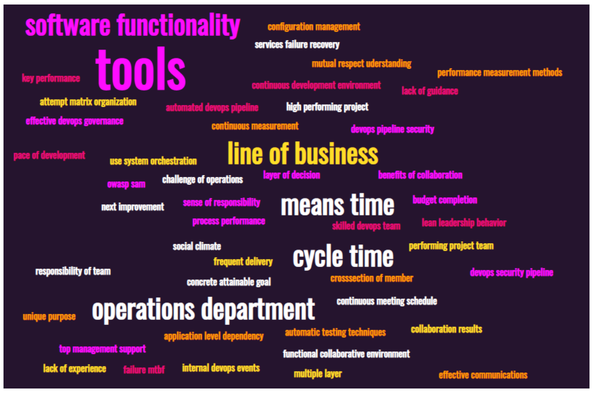


FIGURE 4. Perceived challenges before adopting DevOps.

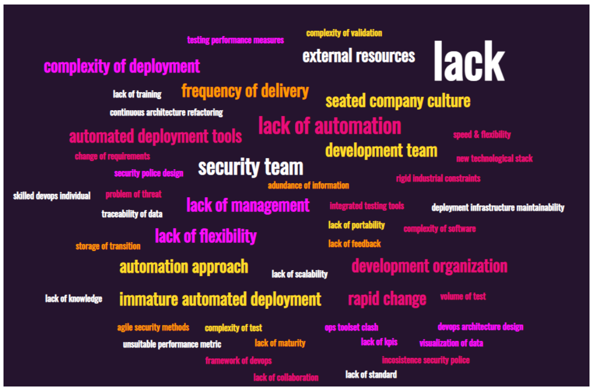


FIGURE 5. Perceived problems while applying DevOps.

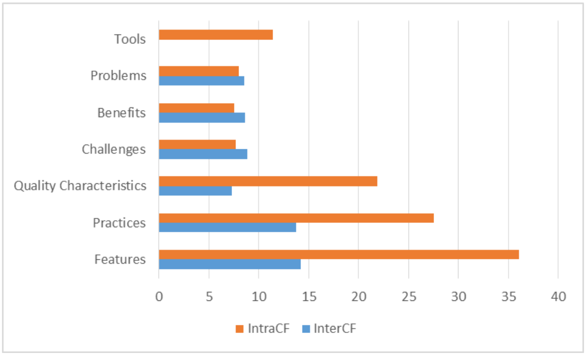


FIGURE 6. Intra- and inter-topic consistency.

(e.g., ***software functionality***) and the customer (e.g., ***line of business***) for applying DevOps. From this point of view, we can deduce that not all application domains are fitting for DevOps, e.g., ***continuous development / deployment / delivery*** might not be applicable, or the customer might not have the ability to be as active and ***knowledgeable***. Furthermore, the application of DevOps, dictates changes in the "*Ops*" branch of the company, since the ***operations department*** needs to adopt its processes to more central role of the customer, providing him/her the ability for providing continuous ***feedback***. Finally, it seems that setting ***mean cycle times*** (and more generally time management) is a challenge for companies that are not experienced in DevOps, considering it as a main challenge before migrating their process to DevOps.

Finally, with respect to the "*Problems*" that are faced after the adoption of DevOps, from Figure 5, we can observe that problems are related to various aspects of DevOps that might be ***lacking*** from the company. On the one hand, in terms of processes, problem might be related to ***lack of automation***, ***lack of management***, ***lack of flexibility***, ***frequency of delivery***, etc. On the other hand, from a more technical perspective, the company might face challenges from the ***complexity of deployment***, ***immature automated deployment***, etc. Additionally, as a problem that might arise along development is the need for ***security*** in modern applications, which is considered as problem, while applying DevOps.

C. DevOps TERMINOLOGY CONSISTENCY (RQ$_3$)

In this section we present the results of the consistency analysis, following the DEAS approach. For each identified topic, we explore the consistency in the used terminology, i.e., that the same terms are consistently and orthogonally mapped into research topics.

In Figure 6, we present the IntraCF (orange bar) and InterCF (blue bar) scores for each topic. The results suggest that the topic with the maximum consistency is "*DevOps Features*" followed by "*DevOps Practices*". As expected more generic topics, such as "*Challenges*", "*Benefits*", and "*Problems*" are less consistent. With respect to "*Tools*" we can observe that although no confusion is tools' names can be performed, the number of tools mentioned in all secondary studies is rather limited. This finding can be explained by the different focus of secondary studies that can potentially lead to the mention of different tools (used for different purposes). Another interesting observation is that "*Quality Characteristics*" is a topic that has limited overlap with others (low InterCF).

Finally, very similar values of InterCF between "*Practices*" and "*Features*", as well as "*Challenges*" and "*Problems*", provide a hint of a possible confusion between the two pairs of topics by the DevOps researchers, and needs further consideration. First, to explore the confusion between "*Practices*" and "*Features*" in Figure 7, we present the 16 terms (37% of all terms used) that are used interchangeably in the secondary studies (intersection of the Venn diagram). As a

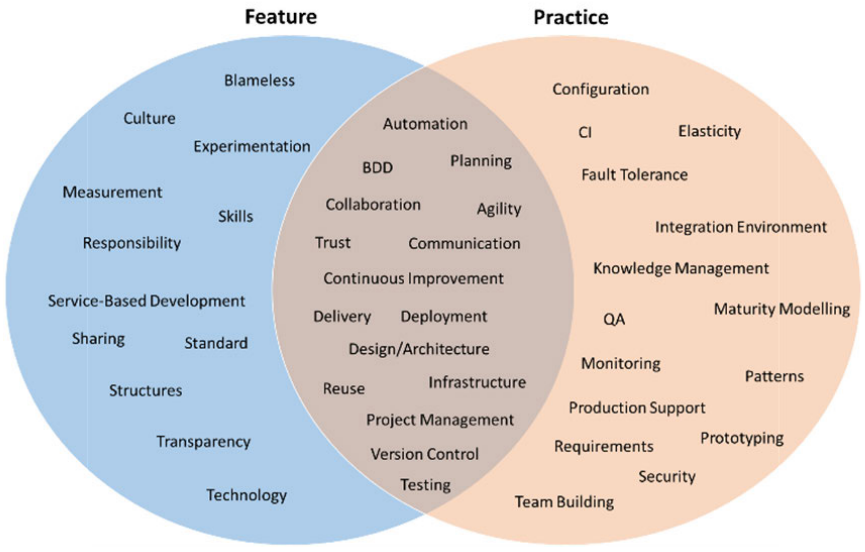


FIGURE 7. Confusion among practices and features.

second step in this analysis, in Figure 8, we present the number of studies, in which each one of these "grey zone" terms are classified as either "*DevOps Practice*", or as "*DevOps Feature*". Based on this analysis, we can classify each term to a single topic for cases with a clear difference (e.g., ***Version Control*** can be more safely classified as a term for ***DevOps Practices***, or ***Automation*** mapped to ***DevOps Features***). Nevertheless, even after this analysis, some terms are safer to be considered as both Features and Practices (e.g., ***Communication***, and ***Trust***). The last approach may help the community towards associating the desired feature (Communication) to the associated practices (Open channel Communication with continuous feedback).

Regarding the second most usual confusion (i.e., "*Challenges*" and "*Problems*"), we can observe a logical continuation: challenges before adopting, not being resolved before starting the project; therefore, being identified as problems along the application of the DevOps methodology. For example, when the challenges: (a) ***adopting automatic testing techniques***; and (b) ***setting up an automated DevOps pipeline***; are not satisfied before the start of the projects, leads to the: (a) ***lack*** of ***automation***; and (b) ***lack*** or ***immaturity of automated deployment*** problems.

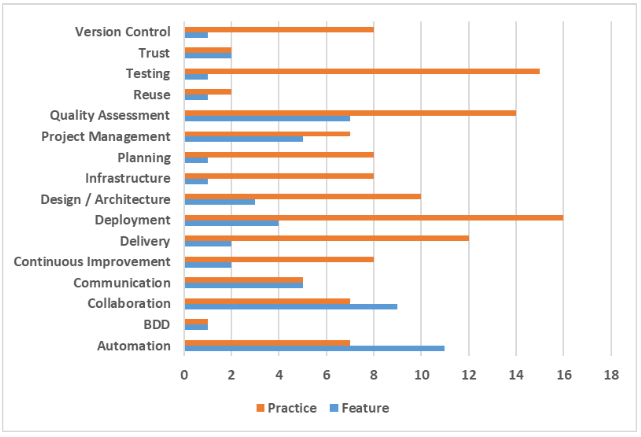


FIGURE 8. "Grey zone" features and practices.

V. DISCUSSION

A. RECAP AND SYNTHESIS OF RESEARCH QUESTIONS

In this section we synthesize the findings presented in Section IV, based on the answers to research questions. The main findings are summarized below:

F1 Secondary studies on DevOps appear to investigate two main lines of research: (a) understand the main DevOps features, as well as the practices and tools that are used for DevOps application; and (b) catalogue the problems that are faced when applying and before adopting DevOps, as well as the benefits from applying DevOps.

F2 Automation, Sharing, Measurement, DevOps Culture, and Collaboration are indisputable features of DevOps that need to be considered for the successful application of DevOps.

F3 Continuous Deployment, Testing, Continuous Monitoring, and Quality Assessment are practices that need to be employed for a successful application of DevOps.

F4 A variety of tools exist for most of the practices.

F5 The majority of DevOps benefits are related to customer involvement, whereas the majority of the problems are more related to Dev branch of the methodology.

F6 The terminology of DevOps is ambiguous, especially for practices, since various terms are used for referring to the same practice.

F7 The terminology in terms of practices and features is mixed: many practices are also listed as features, and vice versa.

F8 A significant number of problems before adopting DevOps continue to be problems during the application of the DevOps methodology.

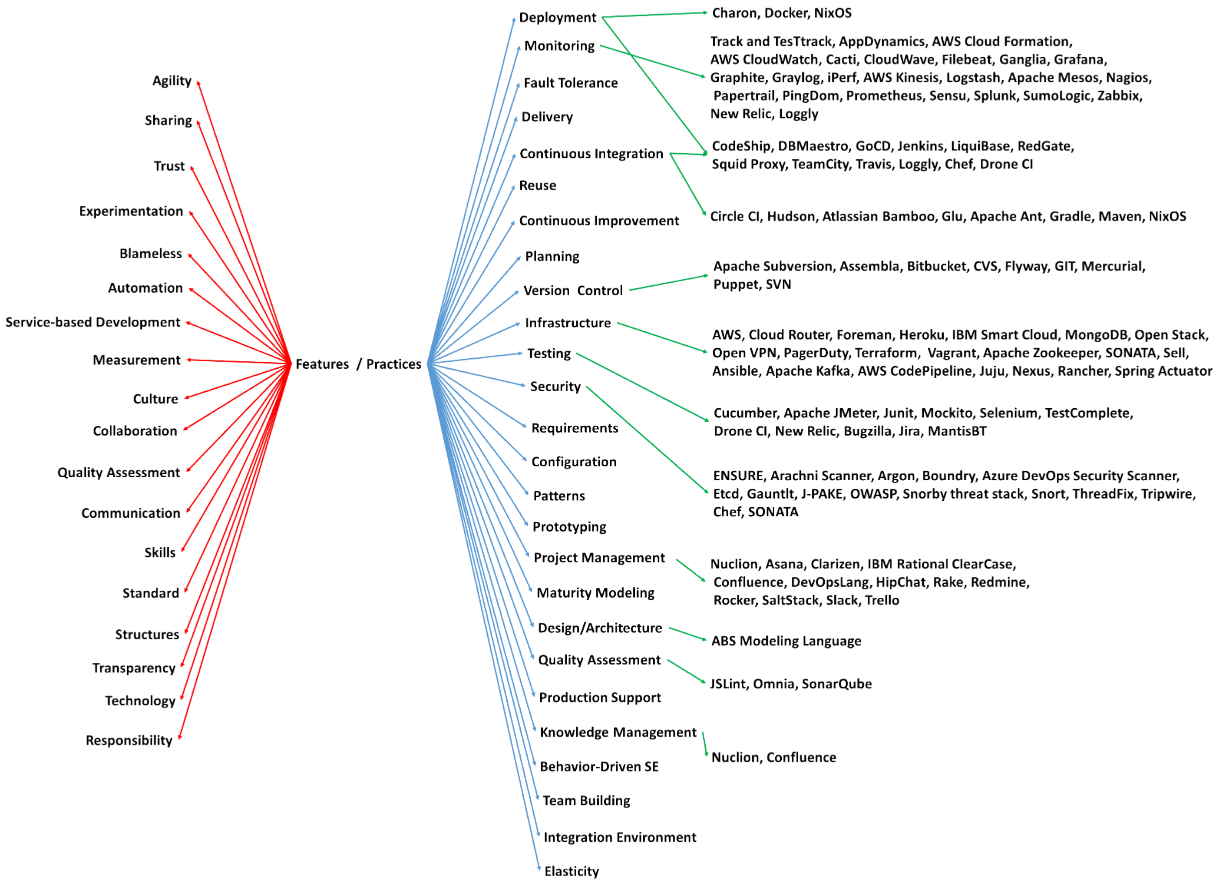

Agility
Sharing
Trust
Experimentation
Blameless
Automation
Service-based Development
Measurement
Culture
Collaboration
Quality Assessment
Communication
Skills
Standard
Structures
Transparency
Technology
Responsibility
Features / Practices
Deployment
Monitoring
Fault Tolerance
Delivery
Continuous Integration
Reuse
Continuous Improvement
Planning
Version Control
Infrastructure
Testing
Security
Requirements
Configuration
Patterns
Prototyping
Project Management
Maturity Modeling
Design/Architecture
Quality Assessment
Production Support
Knowledge Management
Behavior-Driven SE
Team Building
Integration Environment
Elasticity
Charon, Docker, NixOS
Track and TesTtrack, AppDynamics, AWS Cloud Formation, AWS CloudWatch, Cacti, CloudWave, Filebeat, Ganglia, Grafana, Graphite, Graylog, iPerf, AWS Kinesis, Logstash, Apache Mesos, Nagios, Papertrail, PingDom, Prometheus, Sensu, Splunk, SumoLogic, Zabbix, New Relic, Loggly
CodeShip, DBMaestro, GoCD, Jenkins, LiquiBase, RedGate, Squid Proxy, TeamCity, Travis, Loggly, Chef, Drone CI
Circle CI, Hudson, Atlassian Bamboo, Glu, Apache Ant, Gradle, Maven, NixOS
Apache Subversion, Assembla, Bitbucket, CVS, Flyway, GIT, Mercurial, Puppet, SVN
AWS, Cloud Router, Foreman, Heroku, IBM Smart Cloud, MongoDB, Open Stack, Open VPN, PagerDuty, Terraform, Vagrant, Apache Zookeeper, SONATA, Sell, Ansible, Apache Kafka, AWS CodePipeline, Juju, Nexus, Rancher, Spring Actuator
Cucumber, Apache JMeter, Junit, Mockito, Selenium, TestComplete, Drone CI, New Relic, Bugzilla, Jira, MantisBT
ENSURE, Arachni Scanner, Argon, Boundry, Azure DevOps Security Scanner, Etcd, Gauntlt, J-PAKE, OWASP, Snorby threat stack, Snort, ThreadFix, Tripwire, Chef, SONATA
Nuclion, Asana, Clarizen, IBM Rational ClearCase, Confluence, DevOpsLang, HipChat, Rake, Redmine, Rocker, SaltStack, Slack, Trello
ABS Modeling Language
JSLint, Omnia, SonarQube
Nuclion, Confluence

FIGURE 9. Terminology classification schema.

Driven by ***F1***, we perform two synthesis actions: one for the first line of research (features, practices, tools) and another for the second (problems and benefits). Considering ***F4***, ***F6***, and ***F7*** in Figure 10 we present a synthesized classification of terms to features, practices, and tools. The main contribution of this classification is the disambiguation of the "grey-zone" features and practices presented in Figure 8. To classify the "grey-zone" features and practices, even after the analysis of Figure 8, we performed the classification, based on the definitions of Table 1. Therefore, ***Trust*** and ***Communication*** have been classified as "*Feature*", since they are more conceptual term; whereas ***Behavior-Driven SE*** as a "Practice" since it is an activity that can be performed along DevOps application. Additionally, we have mapped the tools presented in Table 4 to the final set of DevOps Practices, so as to provide a comprehensive panorama on how certain DevOps activities can be automated, or at least be tool-supported.

Next, driven by ***F8***, in Figure 10, we present a mapping of challenges faced before the adoption of DevOps to problems that have been identified along the application of the DevOps methodology. The rationale for this synthesis process was the identification of a common practice or feature in the raw data of Figure 4 and Figure 5. The column on the left side of the figure corresponds to challenges that a team can face before DevOps adoption, whereas the right side of Figure 10 corresponds to problems that the secondary studies have identified as important along the application of DevOps. The majority of the "*unresolved*" problems from adoption to application are mostly related to practices (rather than features) and are mostly linked to the "*Dev*" branch of DevOps (e.g., security, testing, development productivity, etc.)

B. IMPLICATIONS TO RESEARCHERS AND PRACTITIONERS

Based on the findings of this study, several implications to researchers and practitioners can be highlighted. On the one hand, regarding ***researchers***, we propose the use of the synthesized terminology presented in Figure 9, as a unified vocabulary so as to reduce ambiguity in the DevOps terminology, and enhance communication among stakeholders and researchers. Also, we could stress the need for identifying specific security tools as state-of-the-art ones, and propose their consistent use in practice. Finally, it seems important to consider the quality characteristics that are important to DevOps, such as testability and maintainability and focus their future research endeavors into proposing methods for safeguarding them. For instance (given the problem highlighted in Figure 10), with respect to testability it seems important to propose methods that enhance automated deployment and testing through a pipeline that will enable continuous delivery. Whereas for maintainability, it might be interesting to explore the technical debt metaphor to speed-up development, achieving a higher pace.

On the other hand, regarding ***practitioners***, the following implications can be highlighted: (a) attempt to *seek for automation solution* (or at least tool support) for their DevOps activities. Guidance is provided by Figure 9. Emphasis shall be placed on automating the most prolific practices: Continuous Deployment, Testing, Continuous Monitoring, and Quality Assessment; (b) attempt to *promote a DevOps Culture* to the employees of the company, so as to secure the adoption of the main DevOps features: Sharing, Measurement, and Collaboration; (c) make full benefit of the main advances that DevOps bring, such as involvement of customer, leading to a more relevant product, and at the same time consider try to prevent the problems that are faced along DevOps application. To this end, during DevOps planning, special emphasis shall be given to the adoption challenges that lead to the most common problems (see Figure 10).

VI. THREATS TO VALIDITY

In this section we present the threats to validity of the current study based on guidelines for identifying, reporting, and mitigating threats to validity, specialized for secondary research studies in software engineering, as they are suggested by Ampatzoglou *et al.* [24].

A. STUDY SELECTION PROCESS

Study selection validity concerns the early phases of the research, i.e., the search process and the filtering of studies. To guarantee that our search process adequately identified all relevant studies (from the studied top-quality venues) we used a well-defined process, based on strict guidelines [15]. The search string was systematically constructed (see Section III.B), in the sense that we have used the term DevOps combined with well-established terminology for secondary studies. However, it could be possible to exclude studies that have used different terminology from the more established ones—i.e., not referring to a secondary study as "*Mapping Study*" or "*Literature Review*". Regarding the first part of the search string, we have preferred to use only the broader term (i.e., *DevOps*), instead of DevOps alternatives, such as: *DevSecOps*, *BizDevOps*, *DataOps*, etc. This decision was made to protect our dataset from being biased due to the specifics of alternatives, e.g., *DevSecOps* boosting the *Security Quality Characteristic*. Furthermore, in the inclusion / exclusion phase, it could be possible to exclude relevant articles. To mitigate this threat, we used three authors in the selection process, discussing any potential conflicts and a systematic voting procedure. Also, the inclusion / exclusion criteria have been extensively discussed by the authors to ensure their clarity and to avoid misinterpretations (see Section III.C). Moreover, from our process we have excluded grey literature, since the goal of the study focuses on systematic secondary studies, almost never published in grey literature. Our study suffers from missing non-English papers; however, most top venues (journals and conferences) in software engineering are only publishing in the English language. Finally, we were able to access all publications because our institutions provide access to DLs.

B. DATA VALIDITY

The main threat for the data validity is related to data extraction bias and the selection of publications. First, all relevant data were extracted and recorded manually by the first and the second author. Due to the potential for subjectivity in this process (e.g., regarding the classification of each term), two other authors reviewed and further refined the collected data, re-validating them. After this process, the results were discussed among all authors and they resolved any conflicts (see Section III.D). Additionally, there is no publication bias in the selected studies, in the sense that the secondary studies have been retrieved by various venues. Thus, the aforementioned studies are not affected by a closed and small circle of researchers. Our tertiary study is not affected from the following threats: (a) small sample size, as we retrieved all possible secondary studies that focus on DevOps; (b) lack of relationships, the study did not aim to identify relationships between data, but only to classify; and (c) the selection of variables to be extracted, as the research questions of this study did not create disagreements in the discussions between authors based on the variables to be extracted.

Moreover, we did not identify issues with the use of statistical analysis, in the sense that the nature of our research questions did not require hypothesis testing, but only basic statistical analysis (descriptive statistics). Finally, to mitigate the researchers' bias in data interpretation and analysis, the authors discussed the data clustering based on the topics that the research questions of each secondary study focuses and the terminology that they have been used. Nevertheless, we need to note that some explanations express the viewpoints and personal opinion of the authors, based on the understanding of the results.

C. RESEARCH VALIDITY

In terms of research validity, threats are related with research method bias and repeatability. Regarding the first one, the majority of the authors are very familiar with the process of conducting secondary studies, as they have participated in a large number of secondary studies as co-authors and reviewers. On the other hand, it could be argued that the following evaluation process ensures the reliability and replication of this study. Therefore, all important decisions for the review process have been thoroughly documented in this manuscript and can be easily reproduced by other researchers. Second, the fact that the export of data is based on the opinion of four authors can to some extent guarantee the reduction of potential bias. Finally, all extracted data have been made public so that the results can be compared and validated. Additionally, through discussion among the authors, we have defined three main research questions in which they accurately map to

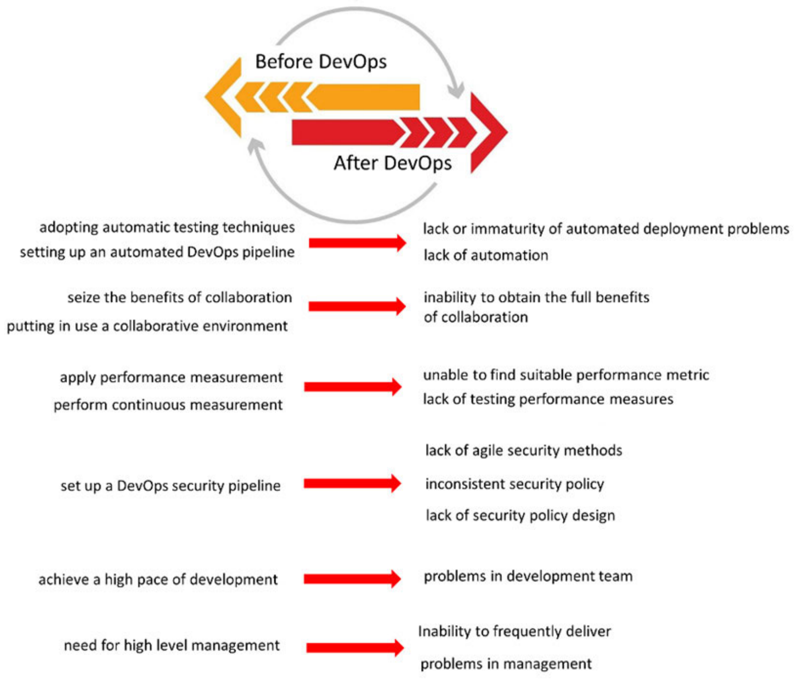

Before DevOps
After DevOps
adopting automatic testing techniques
setting up an automated DevOps pipeline
lack or immaturity of automated deployment problems
lack of automation
seize the benefits of collaboration
putting in use a collaborative environment
inability to obtain the full benefits of collaboration
apply performance measurement
perform continuous measurement
unable to find suitable performance metric
lack of testing performance measures
set up a DevOps security pipeline
lack of agile security methods
inconsistent security policy
lack of security policy design
achieve a high pace of development
problems in development team
need for high level management
Inability to frequently deliver
problems in management

FIGURE 10. **Transformation of challenges to problems.**

the study goal. This is clearly illustrated by the mapping of each research question to the research objectives / goals. Furthermore, in the literature we have been able to identify a substantial amount of related works that can be used for comparison to our results. Finally, the selection of the research method is adequate for the goal of this study and no deviations from the guidelines have been performed.

VII. CONCLUSION

This tertiary study provides a structured understanding of the state-of-research on the DevOps development methodology. For this purpose, 41 secondary studies focusing on DevOps have been identified and analyzed with respect to: (a) the topics that the authors address, (b) the mapping of terms to the different topics and (c) the consistency in the use of terms across different studies. Based on the findings, it seems that at the moment there is ambiguity regarding the terminology used among DevOps stakeholders. This fact makes the retrieval of information relevant to DevOps practices, features and supporting tools a difficult process. We believe that despite the fact that there are research potentials on the particular topic, the impact of the studies and their theoretical development will be limited unless: (a) the DevOps community adopts a common terminology; and (b) the researchers and practitioners focus on cumulative building of knowledge.

Future research can focus on ways that the various features of DevOps methodology should be integrated into practices and automation tools enabling the smooth collaboration between the Dev and the Ops teams supporting the whole process of deploying software, collecting and communicating real-time data for achieving measurable goals. Finally, we believe that an additional study that would explore possible confusing aspects of DevOps terminology with practitioners, and validate the proposed consolidated terms would be highly valuable for both researchers and industrial stakeholders.

REFERENCES

[1] R. Jabbari, N. B. Ali, K. Petersen, and B. Tanveer, "What is DevOps? A systematic mapping study on definitions and practices," in *Proc. Sci. Workshop XP*, Edinburgh, Scotland, May 2016, pp. 1–11.

[2] J. F. Perez, W. Wang, and G. Casale, "Towards a DevOps approach for software quality engineering," in *Proc. Workshop Challenges Performance Methods Softw. Development (WOSP)*, Austin, TX, USA, Jan. 2015, pp. 5–10.

[3] G. G. Claps, R. B. Svensson, and A. Aurum, "On the journey to continuous deployment: Technical and social challenges along the way," *Inf. Softw. Technol.*, vol. 57, pp. 21–31, Jan. 2015.

[4] E. Engstrom and K. Petersen, "Mapping software testing practice with software testing research—SERP-test taxonomy," in *Proc. IEEE 8th Int. Conf. Softw. Test., Verification Validation Workshops (ICSTW)*, Apr. 2015, pp. 13–17.

[5] P. Debois, "Devops: A software revolution in the making?" *Cutter Bus. Technol. J.*, vol. 24, no. 8, 2011.

[6] M. Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery," in *Proc. 5th Int. Conf. Innov. Comput. Technol. (INTECH)*, Pontevedra, Spain, May 2015, pp. 78–82.

[7] M. Krief, *Learning DevOps: The Complete Guide to Accelerate Collaboration With Jenkins, Kubernetes, Terraform, and Azure DevOps*. Birmingham, U.K.: Packt Publishing Ltd, 2019.

[8] F. Erich, C. Amrit, and M. Daneva, "Report: Devops literature review," Univ. Twente, Enschede, The Netherlands, Tech. Rep, 2014.

[9] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Inf. Softw. Technol.*, vol. 85, pp. 60–70, May 2017.

[10] G. K. Hanssen, D. Šmite, and N. B. Moe, "Signs of agile trends in global software engineering research: A tertiary study," in *Proc. IEEE 6th Int. Conf. Global Softw. Eng. Workshop*, Helsinki, Finland, Aug. 2011, pp. 15–18.

[11] K. Curcio, R. Santana, S. Reinehr, and A. Malucelli, "Usability in agile software development: A tertiary study," *Comput. Standards Interfaces*, vol. 64, pp. 61–77, May 2019.

[12] I. Nurdiani, J. Börstler, and S. A. Fricker, "The impacts of agile and lean practices on project constraints: A tertiary study," *J. Syst. Softw.*, vol. 119, pp. 162–183, Sep. 2016.

[13] A. Khan, J. Keung, M. Niazi, S. Hussain, and H. Zhang, "Systematic literature reviews of software process improvement: A tertiary study," in *Systems, Software and Services Process Improvement*. Ostrava, Czech Republic: Springer, Sep. 2017, pp. 6–8.

[14] B. Kitchenham, R. E. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering—A tertiary study," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, 2010.

[15] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. 12th Int. Conf. Eval. Assessment Softw. Eng.*, Bari, Italy, Jun. 2008, pp. 68–77.

[16] V. Basili, "Software modelling and measurement: The goal/question/metric paradigm," Univ. Maryland, College Park, MD, USA, Tech. Rep., 1992.

[17] R. Farhoodi, V. Garousi, D. Pfahl, and J. Sillito, "Development of scientific software: A systematic mapping, a bibliometrics study, and a paper repository," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 23, no. 4, pp. 463–506, May 2013.

[18] D. S. Cruzes and T. Dybå, "Research synthesis in software engineering: A tertiary study," *Inf. Softw. Technol.*, vol. 53, no. 5, pp. 440–455, May 2011.

[19] P. Santos and G. H. Travassos, "Research synthesis in software engineering," *Contemp. Empirical Methods Softw. Eng.*, vol. 53, no. 5, pp. 443–474, 2020.

[20] B. Kitchenham, L. Madeyski, and P. Brereton, "Meta-analysis for families of experiments in software engineering: A systematic review and reproducibility and validity assessment," *Empirical Softw. Eng.*, vol. 25, no. 1, pp. 353–401, Jan. 2020.

[21] B. A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in *Proc. 26th Int. Conf. Softw. Eng.*, Edinburgh, U.K., May 2004, pp. 273–281.

[22] B. Glaser, *Theoretical Sensitivity*. USA: Sociology Press, 1978.

[23] D. Spencer, *Card Sorting: Designing Usable Categories*. USA: Rosenfeld Media, Apr. 2009.

[24] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Verbeek, and A. Chatzigeorgiou, "Identifying, categorizing and mitigating threats to validity in software engineering secondary studies," *Inf. Softw. Technol.*, vol. 106, pp. 201–230, Feb. 2019.

INCLUDED SECONDARY STUDIES

[S1] F. Erich, C. Amrit, and M. Daneva, "A Mapping Study on Cooperation between Information System Development and Operations," International Conference on Product-Focused Software Process Improvement (PROFES'14), Helsinki, Finland, 10-12 December 2014.

[S2] M. Sánchez-Gordón and R. Colomo-Palacios, "A Multivocal Literature Review on the use of DevOps for e-Learning systems," 6th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'18), Salamanca, Spain, 24 - 26 October 2018.

[S3] F. M. A. Erich, C. Amrit, and M. Daneva, "A qualitative study of DevOps usage in practice," Journal of Software: Evolution and Process, 29 (6), June 2017.

[S4] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A Survey of DevOps Concepts and Challenges," ACM Computing Surveys, 52 (6), January 2020.

[S5] D. Teixeira, R. Pereira, T. Henriques, J. Faustino, and M. Silva, "A systematic literature review on DevOps capabilities and areas," International Journal of Human Capital and Information Technology Professionals, 11 (2), 2020.

[S6] A. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A systematic mapping study of infrastructure as code research," Information and Software Technology, 108, pp. 65-77, 2019.

[S7] M. Waseem, P. Liang, and M. Shahin, "A Systematic Mapping Study on Microservices Architecture in DevOps," Journal of Systems and Software, 170, 2020.

[S8] J. Guerrero, K. Zúñiga, K. Certuche, and C. Pardp, "A systematic mapping study about DevOps," Journal de Ciencia e Ingeniería, 2020.

[S9] G. Márquez, F. Osses, and H. Astudillo, "An Exploratory Study of Academic Architectural Tactics and Patterns in Microservices: A systematic literature review," CIbSE 2018, Bogota, Colombia, 23-27 April 2018.

[S10] B. B. N. de França, H. Jeronimo, and G. H. Travassos, "Characterizing DevOps by Hearing Multiple Voices," 30th Brazilian Symposium on Software Engineering (SBES '16), Maringá, Brazil, 19 - 23 September 2016.

[S11] M. Sánchez-Gordón, and R. Colomo-Palacios, "Characterizing DevOps Culture: A Systematic Literature Review," International Conference on Software Process Improvement and Capability Determination (SPICE'18), 2018.

[S12] M. Koskinen T. Mikkonen and P. Abrahamsson, "Containers in Software Development: A Systematic Mapping Study," International Conference on Product-Focused Software Process Improvement (PROFES'19), 2019.

[S13] T. Davide, V. Lenarduzzi, and C. Pahl, "Continuous architecting with microservices and DevOps: A systematic mapping study," Cloud Computing and Services Science, August 2019.

[S14] D. Stahl, T. Martensson and J. Bosch, "Continuous practices and devops: beyond the buzz, what does it all mean?" 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA'17), Vienna, Austria, 30 August - 1 September 2017.

[S15] F. Erich, C. Amrit, and M. Daneva, "Cooperation between information system development and operations: a literature review," 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'14), Torino, Italy, 18 - 19 September 2014.

[S16] J. Ereth, "DataOps - Towards a definition," LWDA 2018, 2018.

[S17] R. Bolscher and M. Daneva, "Designing software architecture to support continuous delivery and DevOps: A systematic literature review," 14th International Conference on Software Technologies (ICSOFT 2019), Prague, Czech Republic, May 2019.

[S18] A. Mishra, and Z. Otaiwi, "DevOps and software quality: A systematic mapping," Computer Science Review, 38, 2020.

[S19] A. Q. Gill, A. Loumish, I. Riyat, and S. Han, "DevOps for information management systems," Journal of Information and Knowledge Management Systems, 48(5), December 2017.

[S20] M. F. Lie, M. Sánchez-Gordón, and R. Colomo-Palacios, "DevOps in an ISO 13485 Regulated Environment: A Multivocal Literature Review," 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '20), Bari, Italy, 5-7 October 2020.

[S21] M. Rütz, "DevOps: A systematic literature review," 27th European Conference on Information Systems (ECIS2019), Stockholm-Uppsala, Sweden, 2019.

[S22] G. B. Ghantous and A. Gill, "DevOps: Concepts, practices, tools, benefits and challenges," 21st Pacific-Asia Conference on Information Systems (PACIS 2017), Langkawi Island, Malaysia, 16-20 July 2017.

[S23] M. Hüttermann and C. Rosenkranz, "DevOps: Walking the shadowy bridge from development success to information systems success," International Conference on Information Systems, Munich, Germany, December 2019.

[S24] H. Myrbakken and R. Colomo-Palacios, "DevSecOps: A multivocal literature review," International Conference on Software Process Improvement and Capability Determination (SPICE'17), Palma de Mallorca, Spain, 4-5 October 2017.

[S25] P. Haindl and R. Plösch,"Focus Areas, Themes, and Objectives of Non-Functional Requirements in DevOps: A Systematic Mapping Study," 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'20), Portoroz, Slovenia, 26-28 Aug. 2020.

[S26] M. A. Akbar, S. Mahmood, M. Shafiq, A. Alsanad, A. A.-A. Alsanad, and A. Gumaei, "Identification and prioritization of DevOps success factors using fuzzy-AHP approach," Soft Computing, 2020.

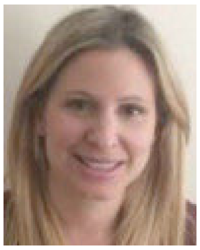
[S27] S. Lavirotte, G. Rocher, J.-Y. Tigli, and T. Gonnin, "IoT-based systems actuation conflicts management towards devops: A systematic mapping study," 5th International Conference on Internet of Things, Big Data and Security, May 7, 2020 - May 9, 2020.

[S28] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad and A. Gumaei, "Multicriteria Based Decision Making of DevOps Data Quality Assessment Challenges Using Fuzzy TOPSI," IEEE Access, 8, 2020.

[S29] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad and A. Gumaei,"Prioritization Based Taxonomy of DevOps Security Challenges Using PROMETHEE," IEEE Access, 8, 2020.
[S30] M. Munoz, M. Negrete, and J. Mejia, "Proposal to Avoid Issues in the DevOps Implementation: A Systematic Literature Review," New Knowledge in Information Systems and Technologies, April 2019.
[S31] S. Rafi, W. Yu, M. A. Akbar, S. Mahmood, A. Alsanad, and A. Gumaei, "Readiness model for DevOps implementation in software organizations," Journal of Software: Evolution and Process, 33 (4), October 2020.
[S32] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "Relationship of devops to agile, lean and continuous deployment: A multivocal literature review study," International Conference on Product-Focused Software Process Improvement, November 2016.
[S33] S. Rafi, W. Yu, and M. A. Akbar, "RMDevOps: A Road Map for Improvement in DevOps Activities in Context of Software Organizations," Evaluation and Assessment in Software Engineering (EASE '20), Trondheim, Norway, 15-17 April 2020.
[S34] V. Mohan and L. B. Othmane, "SecDevOps: Is It a Marketing Buzzword? - Mapping Research on Security in DevOps," 11th International Conference on Availability, Reliability and Security (ARES'16), Salzburg, Austria, 31 August - 2 September 2016.
[S35] M. Sánchez-Gordón and R. Colomo-Palacios, "Security as Culture: A Systematic Literature Review of DevSecOps," 42nd International Conference on Software Engineering Workshops (ICSEW'20), Seoul, Republic of Korea, 27 June 2020 - 19 July 2020.
[S36] M. Van Belzen, D. de Kruijff, and J. J. M. Trienekens, "Success Factors of Collaboration in the Context of DevOps," 12th IADIS International Conference Information Systems, Utrecht, The Netherlands, 11 - 13 April, 2019.
[S37] J. Angara, S. Prasad, and S. Gutta, "The factors driving testing in devops setting-a systematic literature survey," Indian Journal of Science and Technology, 9 (48), January 2017.
[S38] R. Jabbary, N. bin Ali, K. Petersen, and B. Tanveer, "Towards a benefits dependency network for DevOps based on a systematic literature review," Journal of Software: Evolution and Process, 30 (11), July 2018.
[S39] M. A. Akbar, Z. Huang, Z. Yu, F. Mehmood, Y. Hussain, and M. Hamza, "Towards continues code recommendation and implementation system: An Initial Framework," Evaluation and Assessment in Software Engineering (EASE '20), Trondheim. Norway, 15 - 17 April 2020.
[S40] S. Badshah, A. A. Khan, and B. Khan, "Towards Process Improvement in DevOps: A Systematic Literature Review," Evaluation and Assessment in Software Engineering (EASE '20), Trondheim. Norway, 15 - 17 April 2020.
[S41] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is DevOps? A Systematic Mapping Study on Definitions and Practices," Scientific Workshop Proceedings of XP2016 (XP'16), Edinburgh, Scotland, 24 May 2016.

ELVIRA MARIA ARVANITOU received the B.Sc. degree in information technology from the Technological Institute of Thessaloniki, Greece, in 2011, the M.Sc. degree in information systems from the Aristotle University of Thessaloniki, Greece, in 2013, and the Ph.D. degree in software engineering from the University of Groningen, The Netherlands, in 2018. She is currently a Postdoctoral Researcher at the Department of Applied Informatics, University of Macedonia, Greece. Her research interests include technical debt management, software quality metrics, and software maintainability. Her Ph.D. thesis has been awarded as being part of the top-3 ICT-related in The Netherlands, in 2018.

APOSTOLOS AMPATZOGLOU received the B.Sc. degree in information systems, the M.Sc. degree in computer systems, and the Ph.D. degree in software engineering from the Aristotle University of Thessaloniki, in 2003, 2005, and 2012, respectively. Before joining the University of Macedonia, he was an Assistant Professor with the University of Groningen, The Netherlands. He is currently an Assistant Professor in software engineering with the Department of Applied Informatics, University of Macedonia, Greece. He has been nominated as the 3rd most active Early-Stage Researcher in software engineering, from 2010 to 2017. He has published more than 100 articles in international journals and conferences. He is/was involved over 15 research and development ICT projects, with funding from national and international organizations. His current research interests include technical debt management, software maintainability, game engineering, software quality management, open-source software, and software design.


STAMATIA BIBI received the B.Sc. degree in informatics and the Ph.D. degree in software engineering from the Aristotle University of Thessaloniki, Greece, in 2002 and 2008, respectively. She is currently an Assistant Professor of software engineering with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece. She has 40 publications among journals, conference papers, and book chapters. Her research interests include process models, cost estimation, quality assessment, and cloud computing.

ALEXANDER CHATZIGEORGIOU received the Diploma degree in electrical engineering and the Ph.D. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 1996 and 2000, respectively. From 1997 to 1999, he was with Intracom S.A., Greece, as a Software Designer. He is currently a Professor of software engineering with the Department of Applied Informatics and the Dean of the School of Information Sciences, University of Macedonia, Thessaloniki, Greece. He has published more than 150 articles in international journals and conferences and participated in a number of European and national research programs. His research interests include object-oriented design, software maintenance, technical debt, and evolution analysis. He is a Senior Associate Editor of the *Journal of Systems and Software*.

IGNATIOS DELIGIANNIS received the B.Sc. degree in computer science from the University of Lund, Sweden. He was a member of the Empirical Software Engineering Research Group (ESERG), Bournemouth University, U.K. He is currently a Professor at International Hellenic University. He worked for several years in software development at Siemens Telecommunications industry. His main research interests include object-oriented software assessment, and in particular design heuristics and measurement.