What is DevOps? A Systematic Mapping Study on Definitions and Practices

Ramtin Jabbari, Nauman bin Ali, Kai Petersen
Blekinge Institute of Technology
Karlskrona, Sweden
{rjb, nal, kps}@bth.se

Binish Tanveer
Fraunhofer Institute for Experimental Software Engineering, IESE
Kaiserslautern, Germany
binish.tanveer@iese.fraunhofer.de

ABSTRACT

Context: DevOps, the combination of Development and Operations, is a new way of thinking in the software engineering domain that recently received much attention. Given that DevOps is a new term and novel concept recently introduced, no common understanding of what it entails has been achieved yet. Consequently, definitions of DevOps often only represent a part that is relevant to the concept.

Objective: This study aims to characterize DevOps by exploring central components of DevOps definitions reported in the literature, specifying practices explicitly proposed for DevOps and investigating the similarities and differences between DevOps and other existing methods in software engineering.

Method: A systematic mapping study was conducted that used six electronic databases: IEEE, ACM, Inspec, Scopus, Wiley Online Library and Web of Science.

Result: 44 studies have been selected that report a definition of DevOps, 15 studies explicitly stating DevOps practices, and 15 studies stating how DevOps is related to other existing methods. Papers in some cases stated a combination of a definition, practices, and relations to other methods, the total number of primary studies was 49.

Conclusion: We proposed a definition for DevOps which may overcome inconsistencies over the various existing definitions of individual research studies. In addition, the practices explicitly proposed for DevOps have been presented as well as the relation to other software development methods.

CCS Concepts

•**Software creation and its engineering** → **Software creation and management;**

Keywords

DevOps definition; DevOps practice; Software development method

XP '16 Workshops, May 24 2016, Edinburgh, Scotland Uk

DOI: http://dx.doi.org/10.1145/2962695.2962707

1. INTRODUCTION

There is considerable interest in "DevOps" from practitioners. Numerous consultancies offer their services to help organizations to adopt and leverage on the benefits of DevOps. Several recent conferences and systematic reviews on the topic suggest that software engineering researchers also have a strong interest in the topic.

Software engineering research is sometimes criticized for repackaging existing terms and concepts, often resulting in inconsistent definitions for the same concept [45]. Therefore, inconsistent use of terms and hence the concepts leads to difficulties in finding relevant research as authors refer to the same conceptual contribution by totally different terms [12, 52].

Often the distinctions have hampered instead of fostering progress and collaboration in the fields. For example, in several areas of software engineering after the decades of research, we have had a paper suggesting that our research is not all that different [3, 21].

As DevOps is a recent topic there is no standard definition for DevOps [48]. Thus, in this paper we attempt to conceptualize DevOps through the following contributions:

- Analyze and compare definitions of DevOps from the research literature.
- Identify and classify practices associated with DevOps.
- Compare DevOps with other development methods .

We follow a systematic mapping study approach [30, 47] to achieve the contributions. The focus is on scientific and peer reviewed literature.

The remainder of the paper is structured as follows: Section 2 describes the related work. Section 3 presents the research method. Thereafter, in Section 4 we present the results, followed by discussions and conclusions in Sections 5 and 6, respectively.

2. RELATED WORK

In the related work we provide an overview of secondary studies, which review the primary studies on the topic of DevOps, and elaborate on how these are related to our contributions to synthesize/integrate the evidence related to the topic.

Erich *et al.* [13, 15] conducted a systematic literature review on how the relation between development and operations can influence the development of information systems (IS). In particular, they attempted to identify the main concepts of DevOps, and those challenges occurring while using

them in IS development, which were specifically related to development and operations. They also investigated how DevOps can mitigate the identified challenges. In addition, they mentioned culture, automation, measurement, sharing, services, quality assurance, etc. as the concepts for DevOps, but the problems related to IS development have not been reported. In their research work, they found no strong evidence in the literature of how to address the challenges of using DevOps identified in their study. The potential for further research on the relation between DevOps and IS development has been highlighted. Erich *et al.* did not consider the definition of DevOps, identification of DevOps practices and its relation to the other methods that have been targeted in our study.

Smeds *et al.* [51] aimed to specify the concept of DevOps and what practitioners perceive as impediments of adopting DevOps. They defined DevOps by proposing three main attributes, namely capabilities, culture, and technology. The study did not provide a synthesized definition through a systematic analysis of existing definitions, which is one of the contributions of our study.

Furthermore, we complement the existing work by assessing the consistency of the definitions based on 49 primary studies. A systematic approach has been used to derive the definitions. To investigate and understand the impediments of adopting DevOps, Smeds *et al.* explicitly concluded that the definition of DevOps still needs much attention from researchers.

As additional contributions complementing the existing work, we also identified the practices associated with DevOps, as well as how authors of the studies included in our mapping study compare DevOps to other development approaches (such as agile software development).

3. RESEARCH METHOD

We conducted a systematic mapping study [30, 47] to get an understanding of the following question: *How has "DevOps" been characterized in the research literature?* We formulated three research questions (RQs):

- *RQ1: How is "DevOps" defined in peer-reviewed literature on the topic?* The goal of this research question is to achieve an aligned understanding of the concept of DevOps, aiding communication among researchers and practitioners when presenting results related to DevOps.
- *RQ2. Which practices were associated with DevOps in the literature?* Specific agile methodologies prescribe practices to be followed, such as Extreme Programming proposing a set of practices, e.g., continuous integration, coding standards, simple design, etc. As DevOps is a relatively new concept, we are investigating which practices authors associated with DevOps.
- *RQ3. What are the similarities and differences reported by the authors of primary studies between "DevOps" and the other development methods?* In order for DevOps to benefit from earlier lessons learned and evidences obtained, it is of interest to compare and relate DevOps with other development methods.

3.1 Search strings

Table 1 shows the search strings, databases and number of papers found per database. As shown in Table 1, the search conducted aims to find all papers, in which DevOps has been mentioned. We did not limit the search string to e.g., DevOps definition, practice, etc. in order to avoid missing any potential paper that might be related to DevOps. In order to increase the confidence in the search process, the process was done twice with a gap of one week, which is also referred to as the test-retest procedure [35].

Table 1: Search strings, databases used and results from search conducted on September 4, 2015

Database	Search string	No. of papers
IEEE	("DevoOps")	46
ACM	("DevOps") and (PublishedAs:journal OR PublishedAs:proceeding)	85
Inspec	1969-2016: (("DevOps") WN All fields)	41
Scopus	TITLE-ABS-KEY ("DevOps") AND (LIMIT-TO (SRCTYPE, "p") OR LIMIT-TO (SRCTYPE, "j"))	51
Wiley Online Library	"DevOps" in All Fields	7
Web of Science	TOPIC:("DevOps") Timespan: All years.	8
Total		238
Duplicates (Auto Detection)		−79
Duplicates (Manual Detection)		−1
Remove proceedings		−10
Non peer-reviewed		−23
Irrelevant		−2
Not available in Full-text		−6
Remaining		117

3.2 Selection criteria and process

To improve the reliability of the study, several measures [2] were taken starting with the describing the study selection criteria and process. Very simple criteria were used to assess the relevance of the articles:

- Exclude any article not published in English.
- Exclude any article not peer-reviewed.
- Exclude proceedings of conferences (e.g., messages from chair of editorial boards, etc.).
- Exclude any article not available in full-text.

To have a reliable understanding of the studies and realize the role of DevOps in particular, we only consider those studies which are available in full text. To this end it should be mentioned that we only include the articles explicitly discussing "DevOps" to identify how the term itself is defined and which practices are associated with the DevOps concept, studies not using the term "DevOps" have been excluded. For articles not available in full-text we contacted the university librarian and author(s). We could obtain two more studies through contacting the librarian or author(s) that were not available in full-text. There is always a risk that selection criteria are interpreted differently. Hence, it is important to involve multiple persons in the selection process. In order to avoid individual biases, the first two authors were selecting the papers jointly allowing for an immediate discussion of uncertainties.

3.3 Data Extraction

To extract data, we designed the data extraction form as illustrated in Table 2 based upon the objectives of this study [30], which were respectively; to define DevOps, to identify those practices which have been explicitly proposed in the context of DevOps, and to explore any comparison and/or reflection of the relation to other methods.

The data extraction form has been evaluated through conducting a pilot study. Five articles, among the 117 finally selected ones, were randomly selected to avoid bias. The evaluation was done in two rounds. In the first round five studies were extracted by the first two authors to evaluate the data extraction form. The data extraction form was improved based on the pilot. A second pilot was conducted with all four authors. After consensus building, the studies have been distributed between the authors based upon the contributions.

After data extraction, it became evident that many studies were not relevant as they did not focus on DevOps definitions, practices, and relationships to other methods. This was not possible to deduce from the title and abstract only, which was the reason to read the full-text on all 117 studies. Of these only 49 studies were included. Table 3 depicts the number of studies used in our research work in accordance with each specific RQ.

3.4 Analysis

For definition, we investigated the studies which have explicitly used the term DevOps, and then proposed a definition for that (see section 4.1). Based upon the frequency of repetition to define DevOps in these selected studies, we identified the central components for the DevOps definition. (see Figure 1 and Table 4).

To better display the identified central components of DevOps definition, we excluded 'Development' and 'Operations' in Figure 1, as the most common definition for DevOps. The figure shows the identified terms, which is a good starting point for the analysis. Though, it does not put the terms in a context. Consequently, we utilized open coding in order to identify the components of the definitions.

For DevOps practices, we investigated the studies which explicitly proposed activities which are executed in the context of DevOps. After identifying these practices, we have categorized them according to the fundamental knowledge areas, and corresponding sub-knowledge areas, proposed in software engineering body of knowledge (Swebok), as an international standard for providing a comprehensive categorized collection of the bounds of software engineering [1] (see Table 5). The Swebok knowledge areas also provides an indication how many areas are covered by DevOps. The individual practices have been identified through open coding. Thereafter they have been associated and placed under the Swebok knowledge areas.

To investigate the relation between DevOps and other existing method, we investigated the papers, in which an explicit comparison, discussion or reflection has been done relating DevOps and other methods. Then, we categorized the data in accordance with the specified methods (see Table 6).

3.5 Validity Threats

Theoretical validity (confounding factors, inability to capture what we intend to capture:) One common confounding factor is publication bias (i.e. primarily positive studies getting published). Given that the focus of this study is on the definition of DevOps, and not on, for example, benefits and limitations, the publication bias may not play a significant role for this study.

Generalizability (ability to generalize to different contexts): This study is focused on a limited set of studies, and the definition of DevOps may not be static, but rather evolve over time. That is, new practices and new components may potentially added in future studies. This is a threat to validity, though not in the control of the researchers during the study design and execution. Hence, our study presents an aggregated definition and set of practices that may be further extended based on the evolution of the research field.

Objectivity (ability to objectively describe observations): Multiple biases are possible during the search, study selection, data extraction, and analysis. To reduce the biases, multiple researchers have been involved in all steps of the study. For example, we used two researchers during study selection, piloted the data extraction, and reviewed the data analysis. There is also a risk of missing relevant studies in the search. To avoid this, we followed the recommendation by Kitchenham and Brereton to include publisher databases and index databases in our search procedures [34]. Furthermore, a test-retest has been used to reduce the risk of mis-

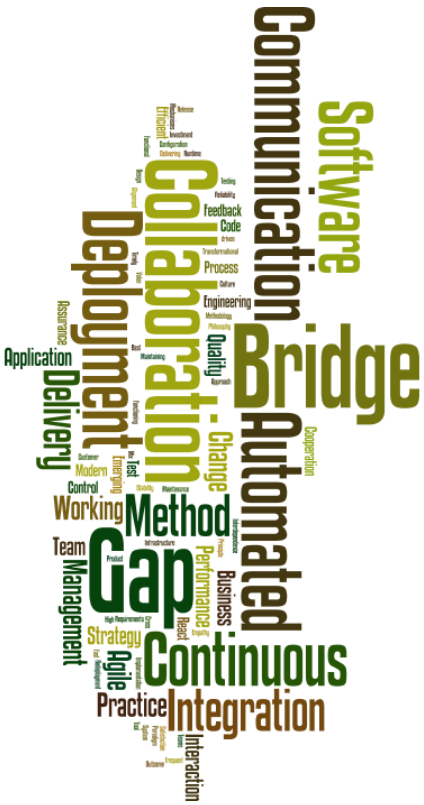

Figure 1: Wordcloud of common terms used in the DevOps definitions

Table 2: Data Extraction Form

Data Extraction Form	
Study ID Name of Reviewer	
Definition? (RQ1)	Author states; e.g., DevOps is OR defined as OR ...
Practices? (RQ2)	Activities that are proposed to be executed in the context of DevOps.
Comparison/Discussion/Reflection of the relation to other methodologies? (RQ3)	If paper has explicitly compared and contrasted DevOps to other methodologies like Agile, Lean, VBSE, etc.
Additional Note	
Do you exclude the paper based on your findings? Yes/ Uncertain/ No if yes/uncertain, why?	E.g., as the paper does not discuss DevOps explicitly.

Table 3: Number of studies

References	**#Ref**	**DevOps Definition (RQ1)**	**DevOps Practice (RQ2)**	**Relation to other methods (RQ3)**
[16, 25, 39, 48, 55, 62]	6	Yes	Yes	Yes
[18, 19, 23, 27, 41, 54]	6	Yes	Yes	No
[5, 9, 10, 31, 40, 59, 65]	7	Yes	No	Yes
[6, 7, 8, 11, 14, 17, 24, 26, 29, 32, 33, 37, 38] [42, 43, 44, 46, 50, 53, 57, 58, 60, 61, 63, 64]	25	Yes	No	No
[22, 28, 49]	3	No	Yes	No
[4, 20]	2	No	No	Yes
Total number of papers	**49**			

takes in the search.

Interpretive validity (drawing the right conclusions from the given data): One of the key outcomes of this study is the definition of DevOps, which should follow and be consistent with the extracted data. Thus, multiple researchers have been involved in the analysis, reflections, and conclusions related to the research questions.

4. RESULTS

4.1 DevOps Definitions (RQ1)

To explore the definitions for DevOps in literature, we attempted to identify the central components to define DevOps explicitly. Table 4 shows the identified components elicited from the corresponding studies.

4.2 DevOps Practices (RQ2)

To explore the practices proposed as DevOps practices in the literature, we attempted to identify those practices, which have been explicitly presented as DevOps practices. Then, we categorized the identified practices in accordance with the software engineering knowledge area categorization and corresponding sub knowledge areas [1]. Table 5 shows the identified practices elicited from the corresponding studies. The table shows that the knowledge areas are covered with respect to the practices suggested in the primary studies.

4.3 Relation to other methods (RQ3)

The relation of DevOps to the other existing software development methods have been elicited from the primary studies. The relations were explicitly discussed in the papers, allowing to identify the similarities and differences between DevOps and the other methods. The relations to other methods are shown in Table 6. Explicitly mentioned relations were highlighted to agile, cloud computing,

Table 4: Central components of DevOps definition

Component	Definition of Component	References
C1: Development and Operations	It has been clarified that the term "DevOps" has been coined by a combination of Development and Operations (or Developers and Operators).	[5, 6, 7, 8, 9, 10, 11, 16, 17, 19, 23, 24, 26, 27, 29, 31, 32, 33, 37, 38, 39, 40, 41, 43, 44, 46, 48, 53, 54, 55, 56, 57, 59, 65]
C2: Communication, Collaboration, Team working	DevOps is defined as a paradigm or method or set of principles and/or practices that enables communication and collaboration resulting in efficient team working between developers and operators.	[6, 7, 8, 11, 16, 24, 25, 27, 33, 41, 44, 57, 58, 64, 65]
C3: Bridge the gap	DevOps is defined as a paradigm or method or set of principles and/or practices that bridges the gap (as the main goal of DevOps) between development and operations. This component is in a close correlation with C2 in the literature.	[7, 8, 9, 16, 25, 39, 46, 54, 55, 56, 57, 60, 61, 63]
C4: Development method	DevOps is defined as a modern software development method to respond to the inter-dependencies between development and operations by unifying modern methods and tools resulting in a real convergence between developers and operators.	[10, 24, 26, 44, 48]
C5: Software delivery	DevOps is defined as a paradigm or set of principles focuses on software delivery through enabling continuous feedback, quick response to changes and using automated delivery pipelines resulting in reduced cycle time. Bayser *et al.* [10] have explicitly mentioned that DevOps was born for fast delivery of web-based systems.	[10, 11, 18, 27, 55, 65]
C6: Automated deployment	DevOps is defined as a paradigm or set of principles that enables automating deployment process from the source code in version control to the production environment.	[7, 29, 38, 61]
C7: Continuous integration	Devops is defined as a practice that emphasizes the tasks enabling continuous integration between software development and its operational deployment needs.	[18, 19, 26, 65]
C8: Quality assurance	DevOps is defined as a method that combines the concerns of quality assurance with operations and development practices to improve performance	[10, 43]

cloud management, development processes (waterfall development), ITIL, and quality assurance activities.

4.3.1 Agile

DevOps extends Agile: DevOps extends agile in terms of the principles as DevOps can provide a pragmatic extension for the current agile activities. For example, as DevOps stresses more on the communication and collaboration between developers and operators rather than tools and processes, it can achieve agile goals to reduce team working latency and extend agile principles to entire software delivery pipeline [16, 55].

Agile web application development and delivery: Bayser *et al.* [10] have mentioned very briefly that DevOps has a particular relation to agile web application development and delivery.

Agile as an enabler for DevOps: Hosono [25] has mentioned that agile methods can be considered as enablers to adopt DevOps thinking.

Agile supports DevOps: Agile can support DevOps by encouraging collaboration between team members, automation of build, deployment and test, measurement and metrics of cost, value and processes, knowledge sharing and tools [5].

DevOps versus Agile: Terhi *et al.* [31] have mentioned that Agile methods for continuous integration and deployment have shared properties with DevOps, while DevOps itself cannot meet all principles proposed in the agile manifesto. Császár *et al.* [9] say that *"Scaling DevOps to software-defined carrier networks is, in a sense, like scaling agile development to large projects in multi-national software companies"*, and finally Miglierina states *"DevOps eliminate the gap between developers and operators, while agile makes an alignment between business requirements and development"* [40].

4.3.2 Cloud computing

Cloud computing: Cloud computing can be considered as an enabler for DevOps, in particular for frequent releases, continuous delivery and integration, as competitive advantages [4, 62].

Model-driven cloud management: Wettinger *et al.* [59] have made a comparison between DevOps and cloud management that mentions, model-driven cloud management can provide a more comprehensive approach for service management, but it cannot replace DevOps, as it comes from completely different background.

4.3.3 Waterfall

In DevOps, *development and operations flow together*, which has not been provided in Waterfall, as an traditional plan-driven methodology [48]. In other word, Waterfall contradicts DevOps because of the standards and principles [14].

4.3.4 ITIL

To define ITIL (i.e., Information Technology Infrastructure Library, a set of practices for IT service management that focuses on aligning IT services with the needs of business), McCarthy *et al.* [39] has mentioned explicitly; like DevOps it is not a set of standards but a set of practices that can be integrated and organized according to the needs of an IT organization.

Table 5: DevOps Practices

Knowledge Area (KA)	Sub-Knowledge Area (Sub-KA)	Practice	References
Software engineering management	**Software Project Planning**	Continuous planning	[55]
		Feedback loop between developers and operators	[23]
	Software Project Enactment	Continuous monitoring	[55]
		Automated performance monitoring during test and continuous integration	[23]
		Automated feedback for performance models and performance predictions	[54]
		Application monitoring	[16]
		Automated dashboards	[16]
Software construction	**Practical Considerations**	Continuous integration	[16, 18, 19, 55]
	Software Construction Fundamentals	Prototyping application	[25]
Software configuration management	**Software Release Management and Delivery**	Integrated deployment planning	[16]
		Continuous deployment	[16, 55]
		Automated deployment	[22, 62]
		Continuous delivery	[22, 62]
		Cooperative application configurations	[41]
		Monitoring application and next development	[25]
	Management of the SCM Process	Staging application	[25]
		Integrated configuration management	[16]
	Software Configuration Control	Integrated change management	[16]
		Change management	[27]
Software testing	**Test Techniques**	Continuous testing	[54, 55]
		Automated testing	[16]
Software Process	**Process Definition**	Process standardization	[48]
		Production support	[16]
Software quality	**Practical Considerations**	Use of data to guide QA	[48]
Software engineering tools and methods	**Software Engineering Methods**	Infrastructure as code	[49]
		Modeling & Simulation	[41]
		Measure performance metrics [in CI, Test & Ops]	[23]
		Continuous application performance	[54]
	Software Tools	DevOps maturity evaluation model	[39]
		Elasticity practice	[28]
Software requirements	**Software Requirements Fundamentals**	Defining requirements	[25]
	Requirements Process	Stakeholder participation	[16]
Software design	**Software Structure and Architecture**	Designing architecture	[25]

4.3.5 *Quality assurance*

Roche [48] discusses a convergence of quality assurance and DevOps.

5. DISCUSSION

Consistency of the use of DevOps definitions: The research results of this study showed the need for a definition as individual studies do not consistently define DevOps. As an example, determining the number of papers all defining the most common components C1, C2, and C3 (Table 4), that is $\mid C1 \cap C2 \cap C3 \mid = 4$, the studies being [7, 8, 16, 57]. Interestingly, none of these references were defining any of the remaining components, namely C4 to C8. This shows the need to propose a definition incorporating the different views of what DevOps stands for.

Derived definition: Based on our synthesis, considering the components identified in Table 4, we define DevOps as follows: *"DevOps is a development methodology (C4) aimed at bridging the gap (C3) between Development (Dev) and Operations (C1), emphasizing communication and collaboration (C2), continuous integration (C7), quality assurance (C8) and delivery (C5) with automated deployment (C6) utilizing a set of development practices."*

The development practices associated with DevOps in the literature are presented in Table 5.

Practices - DevOps versus Agile: In Section 4.3 we presented how authors have explicitly compared DevOps to other methods. It was visible that cloud computing played an important role in DevOps to continuously deploy solutions to the customers, with regard to development processes DevOps has been related to agile software development. Thus, it is interesting to compare the practices in Table 5 with the practices of agile software development, as this facilitates the reuse of previous knowledge obtained in the field of agile software development. For this purpose we utilize the set of practices identified in Kurapati et al.

Table 7: Comparison of DevOps practices with agile practices (cf. Kurapati et al. [36])

Knowledge Area (KA)	Sub-Knowledge Area (Sub-KA)	DevOps Practices	Agile Practices [36]
Software engineering management	**Software Project Planning**	Continuous planning	Sprint planning meeting, sprints and iterations, sprint review meeting
		Feedback loop between developers and operators	Work in teams, communication
	Software Project Enactment	Continuous monitoring	Tracking progress, retrospective, sprint review meetings
		Automated performance monitoring during test and continuous integration	•
		Automated feedback for performance models and performance predictions	•
		Application monitoring	•
		Automated dashboards	•
Software construction	**Practical Considerations**	Continuous integration	Continuous integration, continuous testing
	Software Construction Fundamentals	Prototyping application	[25]
Software configuration management	**Software Release Management and Delivery**	Integrated deployment planning	•
		Continuous deployment	Short/small releases
		Automated deployment	•
		Continuous delivery	Configuration and change management, Continuous integration, short/small releases
		Cooperative application configurations	Work in teams
		Monitoring application and next development	Short/small releases
	Management of the SCM Process	Staging application	Configuration and change management
		Integrated configuration management	Configuration and change management
	Software Configuration Control	Integrated change management	Configuration and change management
		Change management	Configuration and change management
Software testing	**Test Techniques**	Continuous testing	Continuous testing
		Automated testing	Test-driven development (coding/automating execution on unit test level)
Software Process	**Process Definition**	Process standardization	Coding standards
		Production support	•
Software quality	**Practical Considerations**	Use of data to guide QA	•
Software engineering tools and methods	**Software Engineering Methods**	Infrastructure as code	•
		Modeling & Simulation	•
		Measure performance metrics [in CI, Test & Ops]	•
		Continuous application performance	•
	Software Tools	DevOps maturity evaluation model	•
		Elasticity practice	•
Software requirements	**Software Requirements Fundamentals**	Defining requirements	Stories and features
	Requirements Process	Stakeholder participation	On-site customer, communication, sprint review meeting
Software design	**Software Structure and Architecture**	Designing architecture	Simple design

[36], mapping them to the practices in Table 5. The mapping of DevOps and agile practices, shown in Table 7, has been done by expert judgment of the third author, who has particular experiences in the domain, thorough investigating

Table 6: In comparison to other methods

Method	Relations as specified by the authors of the primary studies	Reference
Agile	DevOps extends Agile	[16, 55]
	Agile web application development	[10]
	Agile web application delivery	[10]
	Agile as an enabler for DevOps	[25]
	Agile supports DevOps	[5]
	DevOps versus Agile	[9, 31, 40]
Cloud comp.	Cloud computing as an enabler	[4, 62]
Cloud management	Cloud management versus DevOps	[59]
Waterfall	Waterfall versus DevOps	[48, 65]
ITIL (Information Technology Infrastructure Library)	similarity between ITIL and DevOps	[39]
Quality Assurance	Quality Assurance and DevOps	[48]

the corresponding activities related to pair of the mapped practices, e.g., 'definition requirement' (DevOps practice) has been mapped to 'stories and features' (Agile practice), or 'cooperative application configuration' mapped to 'work in teams', etc. As is evident, the practices in relation to software project planning, software release management, configuration management, and testing are similar from a principle point of view. For example, both DevOps and agile have an emphasis on continuous integration, testing, and delivery of working software. In particular, agile emphasizes small and continuous releases. What DevOps adds to agile software development is the emphasis on automating the analysis of applications, i.e. continuously monitoring the performance of the system, presenting the results in automated dashboards, etc. Furthermore, DevOps literature provided specific approaches and tools, such as infrastructure as code, or the DevOps maturity evaluation model.

6. CONCLUSION

In this paper we present a systematic mapping study focusing on DevOps definitions, practices, and how these are positioned in relation to other development methods. Three research questions have been asked.

RQ1: How is "DevOps" defined in peer-reviewed literature on the topic? With regard to RQ1 we identified eight components of the definition characterizing DevOps. The most frequently highlighted component was that development and operations appear together to coin the term, in particular both components are essential by definition. Furthermore communication, collaboration and team-work as well as bridging the gap between Dev and Ops were highlighted. In addition, DevOps is a development method with an emphasis on software delivery, automated deployment, continuous integration and quality assurance. In synthesis, we proposed the following definition for DevOps: *"DevOps is a development methodology (C4) aimed at bridging the gap (C3) between Development (Dev) and Operations (C1), emphasizing communication and collaboration (C2), continuous integration (C7), quality assurance (C8) and delivery (C5) with automated deployment (C6) utilizing a set of development practices."* Although C7 and C8 have been only mentioned by few studies, we decided to use them in the proposed definition for DevOps, as they specify the core aims of adopting DevOps. The importance of a common definition was evident from the differences of definitions for individual studies. That is, only very few studies shared components of the definition.

RQ2. Which practices were associated with DevOps in the literature? We classified practices of DevOps using the Swebok knowledge areas, and sub-knowledge areas. This provided an idea of the coverage of practices associated with DevOps in relation to software engineering as a whole. When comparing the DevOps practices with those of agile software development many practices were shared, while several practices were specific to DevOps, such as automating the analysis of applications, i.e. continuously monitoring the performance of the system, presenting the results in automated dashboards,

RQ3. What are the similarities and differences reported by the authors of primary studies between "DevOps" and the other development methods? A sub-set of studies explicitly described how DevOps is related to other development methods. Relations were identified with agile software development, cloud computing and management, ITIL, and quality assurance. With regard to agile, different types of relations were defined, such as Devops extends agile, and agile supports DevOps. Looking at the practices of DevOps and comparing them with the common practices of agile software development, these findings were consistent with RQ2.

In this study we focused on the investigation of how research articles view DevOps and which definitions, practices, and relations to other methods were studied. In future work, in order to investigate the understandability and usability of the findings, e.g., the proposed definition for DevOps, the practitioner perspectives should also be considered. This can, for example, be done through interview studies and the investigation of Blogs authored by thought leaders. We also observed that practices were mentioned, but few details of how to use them were presented. Hence, the value of different practices (also agile practices) and their combinations for DevOps need to be understood and investigated in the future.

7. REFERENCES

[1] A. Abran, J. W. Moore, P. Bourque, R. Dupuis, and L. Tripp. Guide to the software engineering body of knowledge: 2004 version. *IEEE Computer Society*, 1, 2004.

[2] N. B. Ali and K. Petersen. Evaluating strategies for study selection in systematic literature studies. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '14, 2014.

[3] N. B. Ali, K. Petersen, and C. Wohlin. A systematic literature review on the industrial use of software process simulation. *Journal of Systems and Software*, 97:65–85, 2014.

[4] P. Austel, H. Chen, T. A. Mikalsen, I. Rouvellou, U. Sharma, I. Silva-Lepe, and R. Subramanian. Continuous delivery of composite solutions: A case for collaborative software defined paas environments. In *Proceedings of the 2nd International Workshop on Software-Defined Ecosystems, BigSystem 2015, Portland, Oregon, USA, June 16, 2015*, pages 3–6, 2015.

[5] S. K. Bang, S. Chung, Y. Choh, and M. Dupuis. A grounded theory analysis of modern web applications: Knowledge, skills, and abilities for devops. In *Proceedings of the 2Nd Annual Conference on Research in Information Technology*, RIIT '13, 2013.

[6] L. Bass, D. R. Jeffery, H. Wada, I. Weber, and L. Zhu. Eliciting operations requirements for applications. In *Proceedings of the 1st International Workshop on Release Engineering, RELENG 2013, San Francisco, California, USA, May 20, 2013*, pages 5–8, 2013.

[7] D. Bruneo, T. Fritz, S. Keidar-Barner, P. Leitner, F. Longo, C. C. Marquezan, A. Metzger, K. Pohl, A. Puliafito, D. Raz, A. Roth, E. Salant, I. Segall, M. Villari, Y. Wolfsthal, and C. Woods. Cloudwave: Where adaptive cloud management meets devops. In *IEEE Symposium on Computers and Communications, ISCC 2014, Funchal, Madeira, Portugal, June 23-26, 2014*, pages 1–6, 2014.

[8] C. A. Cois, J. Yankel, and A. Connell. Modern devops: Optimizing software development through effective system interactions. In *2014 IEEE International Professional Communication Conference, IPCC 2014, Pittsburgh, PA, USA, October 13-15, 2014*, pages 1–7, 2014.

[9] A. Császár, W. John, M. Kind, C. Meirosu, G. Pongrácz, D. Staessens, A. Takács, and F. Westphal. Unifying cloud and carrier network: EU FP7 project UNIFY. In *IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC 2013, Dresden, Germany, December 9-12, 2013*, pages 452–457, 2013.

[10] M. de Bayser, L. G. Azevedo, and R. F. G. Cerqueira. Researchops: The case for devops in scientific applications. In *IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11-15 May, 2015*, pages 1398–1404, 2015.

[11] A. Dyck, R. Penners, and H. Lichter. Towards definitions for release engineering and devops. In *3rd IEEE/ACM International Workshop on Release Engineering, RELENG 2015, Florence, Italy, May 19, 2015*, page 3, 2015.

[12] E. Engström and K. Petersen. Mapping software testing practice with software testing research - serp-test taxonomy. In *Eighth IEEE International Conference on Software Testing, Verification and Validation, ICST 2015 Workshops, Graz, Austria, April 13-17, 2015*, pages 1–4, 2015.

[13] F. Erich, C. Amrit, and M. Daneva. Cooperation between information system development and operations: a literature review. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 69, 2014.

[14] F. Erich, C. Amrit, and M. Daneva. Cooperation between information system development and operations: a literature review. In *2014 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, Torino, Italy, September 18-19, 2014*, page 69:1, 2014.

[15] F. Erich, C. Amrit, and M. Daneva. A mapping study on cooperation between information system development and operations. In *Product-Focused Software Process Improvement*, pages 277–280. 2014.

[16] B. Farroha and D. Farroha. A framework for managing mission needs, compliance, and trust in the devops environment. In *Military Communications Conference (MILCOM), 2014 IEEE*, 2014.

[17] D. G. Feitelson, E. Frachtenberg, and K. L. Beck. Development and deployment at facebook. *IEEE Internet Computing*, 17(4):8–17, 2013.

[18] B. Fitzgerald and K. Stol. Continuous software engineering and beyond: trends and challenges. In *1st International Workshop on Rapid Continuous Software Engineering, RCoSE 2014, Hyderabad, India, June 3, 2014*, pages 1–9, 2014.

[19] B. Fitzgerald and K.-J. Stol. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 2015.

[20] G. Fox, J. Qiu, S. Kamburugamuve, S. Jha, and A. Luckow. HPC-ABDS high performance computing enhanced apache big data stack. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2015, Shenzhen, China, May 4-7, 2015*, pages 1057–1066, 2015.

[21] A. Fuggetta and E. Di Nitto. Software process. In *Proceedings of the on Future of Software Engineering*, pages 1–12, 2014.

[22] K. Gohil, N. Alapati, and S. Joglekar. Towards behavior driven operations (bdops). In *Advances in Recent Technologies in Communication and Computing (ARTCom 2011), 3rd International Conference on*, 2011.

[23] W. Gottesheim. Challenges, benefits and best practices of performance focused devops. In *Proceedings of the 4th International Workshop on Large-Scale Testing, LT'15, Austin, TX, USA, February 1, 2015*, page 3, 2015.

[24] M. Guerriero, M. Ciavotta, G. P. Gibilisco, and D. Ardagna. Space4cloud: a devops environment for multi-cloud applications. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, Bergamo, Italy, September 1, 2015*,

pages 29–30, 2015.

[25] S. Hosono. A devops framework to shorten delivery time for cloud applications. *IJCSE*, 7(4):329–344, 2012.

[26] S. Hosono and Y. Shimomura. Application lifecycle kit for mass customization on paas platforms. In *Eighth IEEE World Congress on Services, SERVICES 2012, Honolulu, HI, USA, June 24-29, 2012*, pages 397–398, 2012.

[27] S. Hussaini. Strengthening harmonization of development (dev) and operations (ops) silos in it environment through systems approach. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014.

[28] K. R. Jayaram. Towards explicitly elastic programming frameworks. In *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16-24, 2015, Volume 2*, pages 619–622, 2015.

[29] X. Ju, L. Soares, K. G. Shin, K. D. Ryu, and D. D. Silva. On fault resilience of openstack. In *ACM Symposium on Cloud Computing, SOCC '13, Santa Clara, CA, USA, October 1-3, 2013*, pages 2:1–2:16, 2013.

[30] S. Keele. Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. 2007.

[31] T. Kilamo, M. Leppänen, and T. Mikkonen. The social developer: Now, then, and tomorrow. In *Proceedings of the 7th International Workshop on Social Software Engineering*, SSE 2015, 2015.

[32] J. Kim. Preparing the end-to-end virtualized networking over software-defined infrastrure. In *Optical Internet 2014 (COIN), 2014 12th International Conference on*, 2014.

[33] J. Kim, C. Meirosu, I. Papafili, R. Steinert, S. Sharma, F. Westphal, M. Kind, A. Shukla, F. Németh, and A. Manzalini. Service provider devops for large scale modern network services. In *IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11-15 May, 2015*, pages 1391–1397, 2015.

[34] B. Kitchenham and P. Brereton. A systematic review of systematic review process research in software engineering. *Information & Software Technology*, 55(12):2049–2075, 2013.

[35] B. A. Kitchenham. What's up with software metrics? - A preliminary mapping study. *Journal of Systems and Software*, 83(1):37–51, 2010.

[36] N. Kurapati, V. S. C. Manyam, and K. Petersen. Agile software development practice adoption survey. In *Agile Processes in Software Engineering and Extreme Programming - 13th International Conference, XP 2012, Malmö, Sweden, May 21-25, 2012. Proceedings*, pages 16–30, 2012.

[37] T. Leesatapornwongsa, M. Hao, P. Joshi, J. F. Lukman, and H. S. Gunawi. SAMC: semantic-aware model checking for fast discovery of deep bugs in cloud systems. In *11th USENIX Symposium on Operating Systems Design and Implementation, OSDI '14, Broomfield, CO, USA, October 6-8, 2014.*, pages 399–414, 2014.

[38] L. Lemus ZÃžÃśiga, N. Pintos, E. Pardo, A. Garza, and J. MontaÃśana Aliaga. Assessment and intervention with wii fit in the elderly. In G. Jezic, R. J. Howlett, and L. C. Jain, editors, *Agent and Multi-Agent Systems: Technologies and Applications*, volume 38 of *Smart Innovation, Systems and Technologies*. Springer International Publishing, 2015.

[39] M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere. Composable devops: Automated ontology based devops maturity analysis. In *2015 IEEE International Conference on Services Computing, SCC 2015, New York City, NY, USA, June 27 - July 2, 2015*, pages 600–607, 2015.

[40] M. Miglierina. Application deployment and management in the cloud. In *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, Timisoara, Romania, September 22-25, 2014*, pages 422–428, 2014.

[41] S. Murphy, S. Gallant, C. Gaughan, and M. Diego. U.S. army modeling and simulation executable architecture deployment cloud virtualization strategy. In *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012, Ottawa, Canada, May 13-16, 2012*, pages 880–885, 2012.

[42] J. Obstfeld, S. Knight, E. Kern, Q. S. Wang, T. Bryan, and D. Bourque. VIRL: the virtual internet routing lab. In *ACM SIGCOMM 2014 Conference, SIGCOMM'14, Chicago, IL, USA, August 17-22, 2014*, pages 577–578, 2014.

[43] M. Olszewska and M. A. Waldén. Devops meets formal modelling in high-criticality complex systems. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, Bergamo, Italy, September 1, 2015*, pages 7–12, 2015.

[44] S. Park, B. Cha, and J. Kim. Preparing and inter-connecting hyper-converged smartx boxes for iot-cloud testbed. In *29th IEEE International Conference on Advanced Information Networking and Applications, AINA 2015, Gwangju, South Korea, March 24-27, 2015*, pages 695–697, 2015.

[45] D. L. Parnas. Stop the numbers game. *Commun. ACM*, 50(11):19–21, 2007.

[46] J. F. Pérez, W. Wang, and G. Casale. Towards a devops approach for software quality engineering. In *Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development, WOSP-C'15, Austin, TX, USA, January 31, 2015*, pages 5–10, 2015.

[47] K. Petersen, S. Vakkalanka, and L. Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.

[48] J. Roche. Adopting devops practices in quality assurance. *Commun. ACM*, 56(11):38–43, 2013.

[49] J. Scheuner, J. Cito, P. Leitner, and H. C. Gall. Cloud workbench: Benchmarking iaas providers based on infrastructure-as-code. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 239–242, 2015.

[50] A. Sill. Factors in development and adoption of new cloud software and standards. *IEEE Cloud Computing*, 1(4):10–13, 2014.

[51] J. Smeds, K. Nybom, and I. Porres. Devops: a definition and perceived adoption impediments. In *Agile Processes, in Software Engineering, and Extreme Programming*, pages 166–177. 2015.

[52] D. Smite, C. Wohlin, Z. Galvina, and R. Prikladnicki. An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19(1):105–153, 2014.

[53] D. Spinellis. Package management systems. *IEEE Software*, 29(2):84–86, 2012.

[54] T. Ustinova and P. Jamshidi. Modelling multi-tier enterprise applications behaviour with design of experiments technique. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, Bergamo, Italy, September 1, 2015*, pages 13–18, 2015.

[55] M. Virmani. Understanding devops bridging the gap from continuous integration to continuous delivery. In *Innovative Computing Technology (INTECH), 2015 Fifth International Conference on*, 2015.

[56] W. Wang, J. F. Pérez, and G. Casale. Filling the gap: a tool to automate parameter estimation for software performance models. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, Bergamo, Italy, September 1, 2015*, pages 31–32, 2015.

[57] J. Wettinger. Streamlining devops automation for cloud applications using {TOSCA} as standardized metamodel. *Future Generation Computer Systems*, 2015.

[58] J. Wettinger, V. Andrikopoulos, and F. Leymann. Automated capturing and systematic usage of devops knowledge for cloud applications. In *2015 IEEE International Conference on Cloud Engineering, IC2E 2015, Tempe, AZ, USA, March 9-13, 2015*, pages 60–65, 2015.

[59] J. Wettinger, M. Behrendt, T. Binz, U. Breitenbücher, G. Breiter, F. Leymann, S. Moser, I. Schwertle, and T. Spatzier. Integrating configuration management with model-driven cloud management based on TOSCA. In *CLOSER 2013 - Proceedings of the 3rd International Conference on Cloud Computing and Services Science, Aachen, Germany, 8-10 May, 2013*, pages 437–446, 2013.

[60] J. Wettinger, T. Binz, U. Breitenbücher, O. Kopp, F. Leymann, and M. Zimmermann. Unified invocation of scripts and services for provisioning, deployment, and management of cloud applications based on TOSCA. In *CLOSER 2014 - Proceedings of the 4th International Conference on Cloud Computing and Services Science, Barcelona, Spain, April 3-5, 2014.*, pages 559–568, 2014.

[61] J. Wettinger, U. Breitenbücher, and F. Leymann. Compensation-based vs. convergent deployment automation for services operated in the cloud. In *Service-Oriented Computing - 12th International Conference, ICSOC 2014, Paris, France, November 3-6, 2014. Proceedings*, pages 336–350, 2014.

[62] J. Wettinger, U. Breitenbücher, and F. Leymann. Devopslang - bridging the gap between development and operations. In *Service-Oriented and Cloud Computing - Third European Conference, ESOCC 2014, Manchester, UK, September 2-4, 2014. Proceedings*, pages 108–122, 2014.

[63] J. Wettinger, U. Breitenbücher, and F. Leymann. Compensation and convergence - comparing and combining deployment automation approaches. *Int. J. Cooperative Inf. Syst.*, 24(3), 2015.

[64] J. Wettinger, U. Breitenbucher, and F. Leymann. Dyn Tail - Dynamically Tailored Deployment Engines for Cloud Applications. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, June 2015.

[65] S. A. Wright and D. Druta. Open source and standards: The role of open source in the dialogue between research and standardization. In *2014 IEEE GLOBECOM Workshops, Austin, TX, USA, December 8-12, 2014*, pages 650–655, 2014.