Computer Science Review 38 (2020) 100308

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

DevOps and software quality: A systematic mapping

Alok Mishra [a,b,*], Ziadoon Otaiwi [b]

[a] Faculty of Logistics, Molde University College-Specialized University in Logistics, Molde, Norway
[b] Department of Software Engineering, Atilim University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 10 December 2019
Received in revised form 15 August 2020
Accepted 17 September 2020
Available online 3 October 2020

Keywords:
DevOps
Development
Operations
Software
Software quality
Automation
Measurement
Systematic mapping

ABSTRACT

Quality pressure is one of the factors affecting processes for software development in its various stages. DevOps is one of the proposed solutions to such pressure. The primary focus of DevOps is to increase the deployment speed, frequency and quality. DevOps is a mixture of different developments and operations to its multitudinous ramifications in software development industries, DevOps have attracted the interest of many researchers. There are considerable literature surveys on this critical innovation in software development, yet, little attention has been given to DevOps impact on software quality. This research is aimed at analyzing the implications of DevOps features on software quality. DevOps can also be referred to a change in organization cultures aimed at removal of gaps between the development and operations of an organization. The adoption of DevOps in an organization provides many benefits including quality but also brings challenges to an organization. This study presents systematic mapping of the impact of DevOps on software quality. The results of this study provide a better understanding of DevOps on software quality for both professionals and researchers working in this area. The study shows research was mainly focused in automation, culture, continuous delivery, fast feedback of DevOps. There is need of further research in many areas of DevOps (for instance: measurement, development of metrics of different stages to assess its performance, culture, practices toward ensuring quality assurance, and quality factors such as usability, efficiency, software maintainability and portability).

Contents

* Corresponding author at: Faculty of Logistics, Molde University College-Specialized University in Logistics, Molde, Norway and Department of Software Engineering, Atilim University, Ankara, Turkey.
E-mail addresses: alok.mishra@himolde.no, alok.mishra@atilim.edu.tr (A. Mishra), otaiwi.ziadoon@student.atilim.edu.tr (Z. Otaiwi).

https://doi.org/10.1016/j.cosrev.2020.100308

1. Introduction

We are in an era where almost every company uses software in running its business operations. Software has become prevalent in companies and daily human activities. This has shown the need to have software dependent products and services that are reliable, useful and secure every time during operations. In the present days, the capability of companies to recurrently and continuously provide new application features that are not only innovative and suitable to use but are also high quality products has become a critical factor in the software development industry. When provided with enough time, any software development organization can be able to provide great software applications and services. In this sense, speed for developing a product (from the stance of faster time a given product is launched to the market) and software quality are the key to success in the software industry. With the speed, developers will have an opportunity to provide quick response to consumers' needs and be able to get a quick feedback regarding the software that is released to the market. Customer feedback is very important as it provides the information that is significant in making an informed decision concerning the software development efforts to stakeholders. A recently developed model, called DevOps (development and operations), aims at producing fast delivery to customers by bringing the development and operation team to work together. The use of the term DevOps began in 2008 when Patrick Debois, at the Agile 2008 Conference, mentioned the need for an agile infrastructure and interaction between the development and operations teams [1]. In 2009, the term DevOps became popular with the beginning of events called DevOpsDays [2]. DevOps is now becoming an essential part of software industry over the last few years focusing on developers and operations to communicate well and deliver reliable and high-quality software services [3]. DevOps is a set of methods in which developers and operations communicate and collaborate to deliver software and services rapidly, reliably and with higher quality. DevOps is sharing of tasks and responsibilities within a team empowered with full accountability of their service and its underlying technology stack; from development, to deployment and support [3].

Softwares have constantly become bigger, more complex, and require high quality. DevOps is the new software process that extends the agility practices within the collaborative culture to enhance the process of software development and delivery. DevOps is concerned with improving the collaboration between the development and operation teams to achieve fast high-quality releases [4]. However, establishing DevOps culture (e.g., shared responsibility) and implementing its practices such as Continuous Delivery and Deployment (CD) require new organizational capabilities and innovative techniques and tools for some, *if not all*, Software Engineering (SE) activities [3,5,6]. Furthermore, in the DevOps transformation, the development side is more emphasized than the operations side [7]. This could be mainly justified by the fact that most of the business values come from the development side (e.g., adding more features).

DevOps increases both deployment frequency and the pace by which companies can serve their customers without compromising the quality of deliveries [8]. DevOps is not only culture aspects it is also a set of engineering practices influenced by cultural aspects and supported by technological enablers [9]. DevOps capabilities are Continuous planning, Continuous integration and testing, Continuous release and deployment, Continuous infrastructure monitoring and optimization, Collaborative and continuous development, Continuous user behavior monitoring and feedback [9,10]. Cheriyan et al. [11] proposed an approach for SQA professional that needs to have deeper understanding of the technical areas of continuous delivery, inspection/static quality assurance, and other areas related to DevOps. This will facilitate in finding the product quality-related issues in systematic manner using this model and may aid to automate the quality assurance parameters and bring predictability in the product quality. Agile team uses continuous delivery approach where as soon as the code is checked in, and the build runs in deployment pipeline (divided into stages) in order to give quick feedback about the quality of the check-in. The quality checkpoints are built into the deployment pipeline. DevOps is also a relevant practice where the development and operations team work together closely to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality [11].

Software quality can be defined as "the totality of characteristics of a software product that satisfy stated or implied needs" (ISO/IEC 9126 2001) [12]. According to (IEEE SA 610.2 1990), quality software assurance that ensures that the software development lifecycle process and its products according to the requirements, standards, and procedures of the industry through the planned and systematic set of operations. Quality has been divided into two categories as product quality and quality of the development process, i.e. the process quality [3,13].

2. Background

In order to have a correct definition of DevOps impacts in software quality, this systematic mapping is conducted with the following specific objectives; first is that we would like to know the qualities that define DevOps impacts. Second, we would

like to investigate the phenomenon behind DevOps quality and finally growing interest in the field. DevOps approach is one of the techniques that has been proven to not only increase the rate of production but also promise a quality product [14]. DevOps is the core enabler of the throughput and high production speed [15]. Due to its potential benefits to organizations, DevOps has attracted the attention of researchers in the recent times [8]. According to Perera et al. [3] there is a lot of literature regarding automation and speed, however, its impact on software quality have been given little attention. The traditional solutions are becoming constraining factor thus leading to inadequacy which can put reputation of an organization at risk [16]. According to World Quality Report, the image of an organization is the first executive concern when it comes to matters concerning quality [8].

It is further reported that attaining the required software quality within a short duration of time is a challenge to many organizations. Research shows that almost a half of organizations have difficulty in determining the right quality coverage in the DevOps process [17]. When an organization fails to overcome this challenge, it translates that the company either accept high risk of their products failure which would lead to considerable impacts to not only the organization's image, but also to its business operations or the organization has accepted the lower speed in implementing the current technology, this shows that the company risk to loose in the current competitive industry. In this sense, study by Ebert et al. [18] seeks to provide a systematic analysis on the impacts of DevOps in software development that such kinds of organizations can benefit from. Though, little is known about the impact of this approach on software product quality. Céspedes et al. [19] study aims to analyze the influence of the application of DevOps on software product quality; therefore, a systematic literature review was conducted. However, their systematic review is aligned to model proposed in the ISO/IEC 25010 standard, and with three research questions only. Although DevOps is in use now for several years, but it is still in its infancy period as well as empirical studies that document the experience of its implementation worldwide. Accordingly, more research and empirical work is vitally needed to practice and validate the proposed DevOps maturity models as observed by Zarour et al. [4]. Jabbari et al. [20] also supported that few systematic studies have been conducted so far and there is need of further studies. Therefore, this study further advances in this direction which is based on DevOps features contribution towards quality. The rest of the paper is structured as follows: Section 3 describes research methodology along with research questions. Section 4 illustrate results of systematic mapping. A discussion is provided in Section 5. Section 6 outline threats to validity. Paper concludes with future research directions in Section 7.

3. Research methodology

3.1. Research questions

Before performing a systematic mapping on the impacts of DevOps on software quality, it was conducted to determine existing secondary research related to our topic. In order to get adequate information, we found the two mostly used online search academic article search engines: the Scopus and the Google scholar. This was because the two major search engines covers all major publisher venues for databases such as the IEEE, ACM, Science Direct, Web of science and the Inspect among others [15]. The research was conducted during November 2019 using the search threads (''DevOps'' and ''quality'') and the (''systematic mapping'' or ''mapping study'' or ''a systematic literature review''). After the searches were performed 20 results were found on Scopus while 234 results were found from the Google Scholar [21]. Regardless of how we searched, most of the results were not actually a secondary study. After a systematic review of the literature in the secondary studies with the same research objectives, it can be observed that there is just one published research on impacts of DevOps on software quality so far but with different perspective. Therefore, this study explores further research in this direction.

This study seeks to provide answers to the below stated questions to explore further insight in this area. In the process of answering the questions, the research seeks to answer the main research question: How does DevOps approach impact software quality?

The research question will be answered in Discussion section while motivation of each is as given in Table 1 and Fig. 1 shows the systematic mapping process.

3.2. Research question definition

The process of research question definition is very important for any type of research paper and hence, it is the first and the foremost step in conducting a research. The research questions highlight the current procedures which are required to be resolved and also the need for the research and the resolution of any particular type of problems. Hence, the research question would be helpful in guiding the researcher towards the review of the scope of research.

3.3. Outcome: Scope review

The scope review is used for reviewing the scope of the research. There scopes are then classified into in-scope for research and out of scope of research.

3.4. Research process

The research process is used for the development of the research and the research process is defined here. In addition to this, there are various type of methodologies which can be selected for a particular research. The primary form of research is used for the researches where the researcher conducts the research all by himself/herself and this type of research yields qualitative data and is not dependent on the other type of researches. The other form of research is the secondary method in which a large number of papers and journals are selected for the research and the data is gathered from the papers. This also helps in the collection of the quantitative data which provides the researcher with a huge data set and also helps in the directing the research in the right direction.

3.5. Outcome: Paper selection

The papers are selected for the research to be conducted and data gathering process.

3.6. Research paper screening

After it has been decided that the secondary method of research is being used for the research it is important that the research papers of the other researchers are looked into and also the relevant papers are reviewed. After a good quantity of research paper has been obtained the research paper are further classified into relevant categories. The filtered papers are screened thoroughly and the best papers are selected for the further research into the subject. The papers for research are finalized in this process.

Table 1
Research questions.

Research questions	Main motivation
RQ1: What are DevOps objectives (features) which helps towards ensuring software quality?	Determine the features of DevOps that impact on software quality attributes.
RQ2: Will the inclusion of automation in DevOps contribute to software quality?	Assess the extent of automation in DevOps process and analyze the effects of automation on software quality.
RQ3: Can measurement in the DevOps lead to increased software quality?	Identify the methods of measurement applied in DevOps aimed to increased software quality. Assessment towards improved software quality as a result of measurements in DevOps.
RQ4: Can sharing in DevOps impact the software quality?	Examine how sharing impacts on software quality.
RQ5: Does DevOps culture has an impact on software quality?	Analyze the DevOps cultures impact on different attributes of software quality.
RQ6: Does DevOps enable fast feedback helps in software quality?	Identify how often DevOps gives feedback and whether it helps in software quality.
RQ7: Does DevOps practice bridge development of software and software quality assurance?	Describe how DevOps practice bridge between software development and software quality assurance.
RQ8: How Software architecture contributes in DevOps success and quality	Identify the software architecture role towards software quality in DevOps practice
RQ9: Does continuous delivery in DevOps helps in ensuring completion on time along with quality?	Discuss how the continuous delivery in DevOps helps in ensuring software quality.
RQ10: How does DevOps impacts Usability, Efficiency, Maintainability and Portability in software?	Describe the ways in which DevOps impacts usability, efficiency and many other impacts on software development.

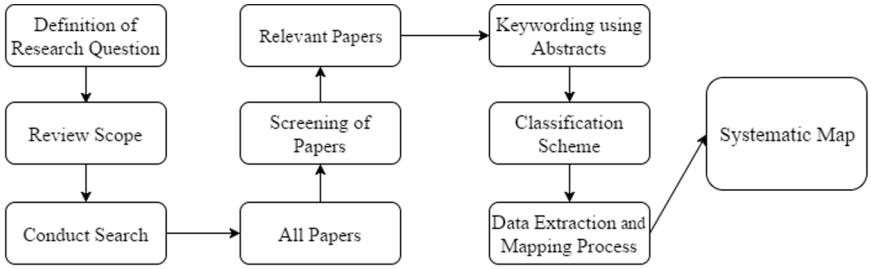

Fig. 1. Systematic mapping process.

3.7. Outcomes: Selection of the relevant papers

The relevant papers to the research are further selected for finalizing the research.

3.8. Key wording the abstracts

The abstract for the completions of the research is selected and the keywords that are being used in the research are searched for. The keywords are obtained by the researcher and this process also helps the researcher in construction of the paper for their research. This also helps identifying the category in which the research has been conducted and the classification of the research in a particular field can be done.

3.9. Outcomes: Scheme classification

The scheme under which the research is classified is known as scheme classification.

3.10. Mapping process and data extraction

After the research procedures have been finalized and the development of a system has been designed, the procedures required to complete the research are mapped. The data analysis on the collected data is performed and the resultant data is obtained which can be used in the development of the system.

3.11. Outcomes: System mapping

The data to be stored in the data storage facilities are also identified and the main business process to be recorded in the system are mapped and designed with the help of the process mapping and system mapping techniques obtained from this process.

3.12. Research method

The answers to the above stated question were achieved through methods that are yet to be discussed in this section. Since this study was primarily based on the literature works, the best method that was proposed is based on systematic review of the literature. By this, a number of academic journals that are relevant to the subject of study were reviewed in order to build a theoretical framework of the study. In this rationale, two key terms were used in the research, the terms include "DevOps", and "Software quality" [3]. We used various reference management facilities for the success of this research, the reference management applications used include Google Scholar, Web

Table 2
Distribution of articles in database.

Database	Total outcome	First results	Final selection
Google Scholar	72	19	10
Web of Science	42	16	3
IEEE Xplore	57	9	4
Scopus	65	55	16
ACM	24	11	2

of Science, IEEE Xplore, Scopus, and ACM library. We managed to access voluminous articles, however, we only selected peer reviewed articles including journals, books and periodicals that were coherent to the subject of the study. The sources which were considered for the study were the documents which were published within 2009, when DevOps was born to late 2019 (see Table 2).

3.13. A synthesis of findings

The results retrieved from literature review were transferred into a publication for further analysis. All of the articles, journals and books accessed contributed to answering the research question stated above. As such the results were synthesized to come up with an overall research results based on the research questions [22]. After the research was completed, the following quality assessment was used to evaluate the validity of the findings obtained. The questions were used as a measure of success of the research process.

3.14. Quality assessment

After primary papers were selected based on the study selection criteria, we performed a quality assessment for the selected studies. We used ten quality criteria derived from Kitchenham et al. [23] and other SLR studies. Each quality assessment question was answered based on the following options: yes = 1, no = 0, somewhat = 0.5. If the total score of the paper gets less than or equal to 4, it is excluded from the list because the quality threshold is set to the value 4. All authors involved in the quality assessment. When there is a conflict about the scoring per paper, an additional meeting was held.

The quality criteria are listed as follows:

- Q1: Are the aims of the study clearly stated?
- Q2: Are the scope and experimental design of the study defined clearly?
- Q3: Are the variables in the study likely to be valid and reliable?
- Q4: Is the research process documented adequately?
- Q5: Are all the study questions answered?
- Q6: To what extent has the research explained the wider view of DevOps?
- Q7: How well was the analysis approach and formulated?
- Q8: Are the data sources and contexts described appropriately for future reference?
- Q9: What additional knowledge has the research added?
- Q10: Do the conclusions relate to the aim of the purpose of the study?

After the report was evaluated, the obtained data was synthesized basing on the source of the data. The sources were categorized into groups; one is those that are relevant in development of DevOps (inclusion) and those that are not relevant (exclusion). Content analysis was used in categorizing these sources and the categories are tabulated below in Table 3 while stages and selection criteria is illustrated in Table 4.

Table 3
Inclusion and exclusion criteria.

Items	Criteria	Category
Source 1	Studies that have been registered in indexed databases	Relevant
Source 2	Studies that are relevant in Software Engineering	Relevant
Source 3	Studies that illustrate how DevOps and quality in software products relate to one another.	Relevant
Source 4	Studies that describe various context and case studies relating to DevOps.	Relevant
Source 5	Studies that do not focus on the quality of software products	Irrelevant
Source 6	Studies not related with DevOps	Irrelevant
Source 7	Studies that are not firsthand i.e. secondary sources	Irrelevant
Source 8	Studies in languages other than English	Irrelevant

Table 4
Stages and selection criteria.

Stages	Scope	Selection criteria
First stage	Title	1, 2, 3, 5, 6 and 7, 8
Second stage	Abstract and conclusions	1, 2 and 3, 4, 5, 6
Third stage	Complete content	3, 4, 5, 6 and 7

3.15. Discussion on database

The current study researched for five different electronic data bases namely; Google Scholar, IEEE explore, the Springer, Scopus, Web of Science and the ACM digital library in order to get the research papers related with impacts of DevOps on software quality. Due to the search, an approximate number of 195 research papers on DevOps were found, even though only 35 of them were related to the development of quality software.

A spreadsheet was developed for the data extraction according to the criteria of Table 3 and Table 4. There were exactly 72 journal, conference papers and other newsletters. It is simpler to note that most papers are in journals and conferences. The final selection used in the paper amounted to thirty five. Therefore, they appear to be the most significant references for impacts of DevOps on software quality. There is no individual article with greater publications in this field.

The ACM digital library contributed the lowest number of databases about impacts of DevOps on software quality. This shows that there was a minimal number of authors who concentrated on DevOps on software quality. Among these results only the final 35 results had the objectives of the study. By this, a number of academic journals that are relevant to the subject of study were reviewed in order to build a theoretical framework of the study. In this rationale, two key terms were used in the research, the terms include “DevOps”, and “Software quality”. Google Scholar, Web of Science, IEEE Xplore, and ACM library were used for this process. Only peer reviewed articles including journals, conferences, and periodicals were included that were coherent to the subject of study. Fig. 2 shows relationships between DevOps features and software quality attributes.

4. Results of systematic mapping

4.1. DevOps definition

Over the past few years, DevOps has gained a considerable popularity and this has been attributed to enabling organizations

Table 5
DevOps definitions in literature.

Authors	Definition	Scope of the definition
Jabbari, et al. [24]	DevOps is a development methodology aimed at bridging the gap) between Development and Operations emphasizing communication and collaboration, continuous integration, quality assurance, and delivery with automated deployment utilizing a set of development practices.	Focus on multifunctional, quality, automated deployment and collaboration.
Perera et al. [3]	DevOps is described as different ways whereby developers and operations discuss and share so as to produce software and give services faster, timely and with increased quality.	Fast deployment and quality

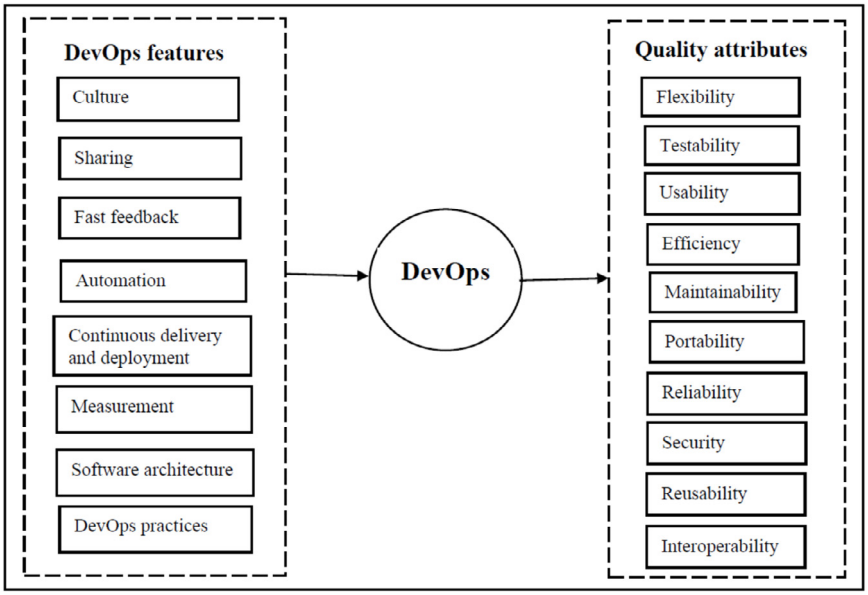

Fig. 2. Relationship between DevOps features and software quality.

to create and improve products at a more increased pace compared to the traditional software development approaches. As such, it is important to explore this critical approach and find out DevOps features which are associated to improving software quality. DevOps is a recent concept and as such, it has no consensus definition. Various definitions of DevOps have been provided in the literature. A good number of the definitions suggest that the term emphasizes the collaboration between development and operations. Jabbari, et al. [24] after systematic literature review provided definition in Table 5. On the other hand, Perera et al. [3] describe DevOps as different ways through which developers and operations discuss and engage in collective discussion so as to produce software and deliver services faster, reliably and with increased quality. All of the definitions above describe DevOps, however, these are limited to software development and operations. As it can be seen, the practices of DevOps are so wide, they can be captured from the key dimensions of culture, collaboration, automation, measurements and monitoring. Table 6 shows studies which are related to DevOps features and its impact on software development process and quality.

4.2. DevOps objectives (features) towards quality

Objectives are what take every organization forward. They are the first step to software quality. And therefore, every process employed by any organization to access quality should support its objectives. To this end, the value of the question: What are the objectives of DevOps? This can be seen by software development organizations. Perera et al. [16] have examined some of the goals of DevOps including improving deployment frequency, lower rates of failure and faster mean time to recover on the off chance that a new release crush. It is further observed that the deployment frequency is one of the major metrics where the organizations implementing DevOps shine by deploying at a considerably higher frequency of more than 40 times compared to the non-performers of DevOps [3]. The researchers Mohan, & Othmane [8], further supported that it does not only provide such a considerable deployment frequency but also ensure software quality. Further, it contributes to software quality by reducing the rates of failure. As discussed in the previous sections, DevOps is all about automation, increasing the feedback loop; this tremendously reduces the amount of labor that would be involved in the release pipeline thus reducing the chances of failure. Moreover, it is shown by that mean time to recover is one of the interesting cases of the culture of DevOps. If building, measuring and feedback cycle are all that DevOps is concerned with then it should be undeniable truth that the development operation can ensure software quality [14].

Table 6
DevOps features related to software development process and quality.

DevOps features	Description	Studies
DevOps objectives (features) towards quality	These are the first steps towards quality	[78], [79], [87], [100], [102]
Automation in DevOps	Automation in development process increases deployment rate to quality deliveries with short cycle time	[79], [83], [84], [86], [87], [89], [96], [97], [98], [106], [107],
Measurement in DevOps	Checking performance metrics show consistent results thus helps in ensuring quality.	[91], [101], [108]
Sharing in DevOps	Embracing collaboration in process, tools, goals in operation and development improves communication which enhances the efficiency lead to quality	[74], [85], [90], [97], [104]
DevOps Culture	DevOps culture promotes considerable improvement in software quality	[79], [80], [83], [96], [97], [103], [106]
Fast feedback in DevOps	Capturing end-user feedback is improved by deployment of operations or DevOps which ensure software quality with less effort.	[84], [88], [91], [96], [107]
DevOps practice towards quality assurance	DevOps is linked to quality assurance and enables the production of a flawless software.	[75], [95], [97], [104]
Software architecture in DevOps	Software Architecture (SA) is foundation for moving towards the highest level of DevOps success	[77], [81], [82], [94], [99], [104], [105]
Continuous delivery in DevOps	Continuous delivery (CD) enables to deliver high quality in a more efficient way at a reduced time	[77], [78], [92], [93], [102], [105], [106]
DevOps impact on Usability, Efficiency, Maintainability and Portability	Continuous delivery and test automation helps in ensuring reliability, maintainability, safety, reliability(availability), maintainability (testability and modifiability)	[74], [76], [79], [91], [104]

4.3. DevOps automation towards quality

Automation of organizations is one of the key performance indicators of quality and reliability. There is substantial literature that reviewed this crucial aspect, however, little has been done as far as DevOps is concerned. This has precipitated the need to find out the bridge between software quality and DevOps concerning automation in organizations. An organization can instill automation in various ways. Automation by DevOps is supported by different designs. Some of which include utilizing cloud for big data storage, use of cloud based email and logging services, utilize a real-time monitoring tool [25]. Using SaaS and IaaS are also capable of supporting DevOps automation [26].

Perera et al. [16], evaluates the impacts of DevOps on software quality in an organization. They accomplish this through quantitative analysis where they used online questionnaires which were distributed to more than 300 software professionals in the organization of which 62% of the respondents had two to three years' experience, 24% had less than one year and 14% were software professionals who have used DevOps for more than three years. The analysis demonstrates that implementing DevOps in an organization has considerable positive impacts on the software quality or value. The practice of DevOps increases the value of the software. Along with that, the research also revealed that there is a considerable relationship between DevOps and automation. They suggested that DevOps encourages automation thus enhancing the quality of the application.

4.4. DevOps technical issues (measurement) towards quality

Measurement is another handy approach in software production. Measuring the reliability of an application before it is released for users is a contributing factor towards the software quality. As such, we are motivated to examine if this would contribute to software quality in DevOps. Measurements refer to examining the high level business metrics while doing a careful selection on the system performance as well as quality metrics in production. In their study, Bou Ghantous and Gill [27] provides clarification of a software which is reliable. They note that measurements concerning business metrics has to be transparent and must be able to be visualized through DevOps model. The measurements by DevOps tend to provide a consistent result through its range which is quite different from the traditional estimates which are based on experience as such measurements seem to be having low accuracy [25]. To the reason that DevOps increases the rate of change, possessing the right change, promise considerable benefits to an organization. DevOps provide its users with chances to do amazing things to their business, however, they are also associated with risks sometimes [8]. By utilizing the automated software analysis, companies can leverage these benefits without encountering any risk; this shows the potential benefits of DevOps [8].

Zarour et al. [4] observed that maturity models help to assess effectiveness of an organizational processes and assist in identifying capabilities required to improve their performance to move towards higher maturity levels. In their study Bou Ghantous and Gill [27], present clarification of a software which is reliable. Prates et al. [28] identified metrics which can use to measure the effectiveness of DevSecOps methodology implementation inside organizations. Perera [13] and Perera, et al. [16] have examined some of the goals of DevOps including improving deployment frequency, lower rates of failure and faster mean time to recover on the off chance that a new release crush. In another study it was noticed that the deployment frequency is one of the major

metrics where the organizations implementing DevOps shine by deploying at a considerably higher frequency of more than 40 times compared to the non-performers of DevOps [3].

4.5. Collaboration and sharing in DevOps towards quality

DevOps is characterized by culture of sharing information and the information sharing has a huge potential for positive impacts on software quality. It is therefore important to find out the relation between sharing in DevOps and software quality. Sharing involves a collaboration which is developed by sharing of information, shifting responsibilities, expanding skillsets, as well as putting some sense of responsibility between development and operation as noted by Gill et al. [29]. With a close collaboration with operations, organizations can be assured that tests can be executed in a production environment signaling production. This helps in increasing confidence while launching a new product versus software quality as observed by Lwakatare et al. [30]. DevOps embraces collaboration in various areas including processes and tools as well as goals and incentives. This is also due to the reason that as when a team from various background come together in a common environment (DevOps environment), it is important to share the common success, goals and problem-solving responsibilities, hence enables the success [3]. This shows that there is a link between collaboration and software value suggesting the positive impacts of the DevOps on the quality of software through collaboration [31].

4.6. DevOps culture towards quality

Culture is significant in every organization as it changes how employees work and share responsibility to ensure the quality of an end product [32]. As such, we sought to investigate the culture in DevOps and software quality. DevOps culture mainly focuses on sharing responsibilities, open communication and trust as well as mutual respect. The interplay of these factors is essential when it comes to quality assurance. In implementation of DevOps practice in quality assurance they illustrate that the DevOps culture is a fundamental aspect when it comes to software support role. The researches further add that embedding this DevOps nature throughout the life cycle of a product promise a considerable improvement in software quality [33]. This analysis concludes that DevOps culture offers a significant contribution to quality assurance in applications development phases. It is normally difficult to predict the quality assurance, but studies have shown that with DevOps, the quality assurance process can be much improved by harnessing the above mentioned natures of DevOps.

4.7. DevOps fast feedback and automation towards quality

The environment in which software systems run have become complex thus nagging feedback loop which may ultimately compromise software quality. It is essential to explore this crucial factor as far as DevOps is concerned. This is in the sense that feedback loop is one of the most critical considerations in software development. It is the fundamental approach used in validating and obtaining feedback about the process of software development. As such, getting feedback as fast as possible is essential for software quality in the overall development process as noted by Forsgren et al. [34].

DevOps is described here with the three principles along with their accompanied loops. According to the Erich et al. [35], the principle of flow enables a fast feedback loop throughout a software development process to the production stage. The process is by establishment of automated deployment strategies along with the tests that examines the production fitness by use of trunk-base development strategy. Combined with automation, this strategy provides the best means for achieving software quality throughout the software development process to the stage of production [36]. The trunk-based development drives down batches proportions and ensure that all changes are in the right order. It also enables transformation of testing workflows as work is done in a shared space, this helps in avoiding potential development conflicts (Jones, 2018). Obtaining fast feedback reduces the need for separate tests as well as stabilization phases hence agility. When automation is enabled in testing, the software development team can divert their attention to identify and improve other quality aspects earlier in the software development phases as noted by Kroll et al. [36].

4.8. DevOps practice enables quality assurance

Erich et al. [35], have shown that quality assurance is significant to ensure the development of high-quality software. However, a few literatures have explored the significance of quality assurance for software quality with regards to DevOps [3]. Ensuring quality assurance practices have significant contribution in software quality. However, achieving quality assurance has been a challenge to many organizations, more so to the companies that deploy agile software development [37]. As stated earlier quality assurance is one of the major areas in which DevOps have a significant impact. Further, confirm that DevOps is the key determinant for quality assurance in software development. Toh et al. [38] conclude in their review that DevOps can be seen as a gift for quality assurance. DevOps enables quality assurance through cooperation. Automation also facilitates gathering of more data for decision making by the quality assurance team. Additionally, the duty to ensure quality assurance can get assigned to organization's members, who also have software development and operation responsibilities, this enhances software quality in the development phase [38].

4.9. Software architecture aid towards quality

Software architecture provides infrastructure in DevOps to monitor and ensure quality assurance. Shahin and Babar [39] suggested that DevOps success is best associated with modular architectures and require to needs to prioritize various quality attributes. Daneva and Blosher [40] studied and found that (a) 17 software architecture characteristics are beneficial for CD and DevOps adoption. Chen [41] observed that a set of quality attributes such as deployability, security, modifiability, and monitorability, require more consideration when designing architectures in the context of CD in DevOps background. Therefore, it is significant to observe software architecture relationship with quality in DevOps.

4.10. DevOps continuous delivery towards quality

DevOps continuous delivery and software for on-time completion with quality is a factor of fundamental relevance for its performance thus worth investigating. The continuous delivery involves automating and streamlining processes for deployment. Continuous Delivery can be used to aid the process of product delivery within the DevOps environment. This practice can fulfill the requirement of better quality during the software development process [42]. Mohan, & Othmane [8], further noted that it consumes a great deal of efforts and time when done in traditional method which would not only be slow and costly but is also likely to compromise quality. According to Erich et al. [35], in continuous delivery process can ensure efficient completion when monitoring dashboards are availed for their production

environment in order to eliminate performance challenges and ensure quick response time. It is further observed that continuous delivery enables organizations to develop and deliver a high quality product in a more efficient manner and in a reduced amount of time. Studies by Hamzehloui et al. [43] show that implementing DevOps enables faster time to production and delivery times which consequently leads to improvement in software quality. This is exemplified by Lwakatare et al. [30] study that examined organizations that have employed DevOps having collaborative cross-functional teams working towards delivering high quality at a maximum speed.

4.11. DevOps impacts on usability, efficiency, maintainability and portability

Usability, efficiency, maintainability and portability are the key software quality attributes. These attributes are essential in gauging software quality thus the relevance of this question in examining the impacts of DevOps on software quality. The scholars Jabbari et al. [20], examine various quality attributes which the development operation promises to software development organizations. It is shown that DevOps enables effective usability test without slowing the continuous delivery pipeline. Its automation also promise efficiency. Regarding maintainability, DevOps team focuses on standardization of the deployment environment as well as automated delivery process that ensure software quality. Kroll et al. [36] have also expound that DevOps and cloud's flexibility and efficiency could be attributed to software quality.

5. Discussions

DevOps brings many benefits however there are challenges associated with it for instance: cultural and organizational aspects, adoption challenges, new concepts like automation in processes and tools, in general separation between developers and operations team, lack of explicit control on reusability poses challenges to practicing Continuous Delivery and Deployment (CD), automatic deployment activity, performance challenges in providing fast feedback, etc. [9,20,44,45].

RQ1: What are DevOps features which helps towards ensuring software quality?

The definitions show that DevOps is a software quality-oriented approach. It has also been found that culture, collaboration, automation, measurements and monitoring are the key enablers of DevOps for software quality. In DevOps the development and operations team work together closely to reduce the time between committing a change to a system and it is moved into normal production, while assuring high quality [11]. DevOps is intended to increase the frequency, quality and speed of deploying software from development into production by means of new organizational structures and processes with a high degree of automation [15].

According to Céspedes et al. [19] main practices associated with DevOps that influence the software product quality are: deployment automation, test automation, cloud computing and team cooperation. They further argued that using tests, automatic or not, combined with the deployment automation increases significantly the product quality although deployment automation and cloud computing should be taken a greater care with security. Céspedes et al. [19] further argued that there is an influence on product quality by the use of DevOps, mainly related on reliability and maintainability; and to a minor extent functional suitability, security and performance efficiency. With the DevOps team setup, we could align team efforts into one direction with the common objective of high-value delivery on-time with quality and in meeting the project milestones and version increments [46]. Perez-Palacin et al. [47] reported an experience on the usage of a software quality evaluation tool during a DevOps-oriented software development.

RQ2: Will the inclusion of automation in DevOps contributes to software quality?

Automation is an immensely helpful to organizations. With more work automated, high performers free their technical staff to do innovative work that adds real value to their organizations [34]. Forsgren et al. [34] further reported for instance: HP LaserJet, The firmware division based on critical path for hardware releases was able to increase time spent on developing new features by 700 percent. The DevOps paradigm focuses mainly on making cultural changes in the organization, relying on approaches such as Continuous Delivery, which is aimed at the automation and optimization of the delivery process [10]. Quality deliveries with short cycle time need a high degree of automation [18]. Automation, which is a great enabler for swift delivery, the practice of automate the software delivery lifecycle [48]. Muñoz et al. [48] further argued that adapting DevOps practices to the current process is a big challenge that involves cultural and organizational aspects, bringing new concepts like automation in processes and tools. Furthermore, presently the segregation between developers and operations team is commonly found in many organizations, causing one of the main obstacles for fast and frequent releases of software [49].

Before automation it is significant to set up quality environment to automate the scenarios. Tools are essential in automating DevOps and quality deliveries with short cycle time requires a high degree of automation. So, selecting the right tools for your environment or project is important when you move to DevOps [18]. Ebert et al. [18] further argued that the quality assurance team must ensure automation of all test cases and full code coverage. The test cases and ops activities automation make the DevOps setup useful in short release cycles. Ops automation in a cloud environment helps to control the ops cost [46]. There are number of tools to use when practice DevOps such as Jenkins for Continuous Integration, Cucumber for BDD, Junit for TDD, GIT for configuration management, Quality Centre, JIRA, ALM for Test lifecycle and defect management, Selenium, QTP and UFT for automation [3].

DevOps environment implies an automation in process and tools as shown in the results of an interview performed to 18 case organizations from various software domains, in which it is mentioned that commodity tools such as version control, continuous integration, UI testing, performance, are used in almost every organizations, which have implemented a DevOps environment [50]. The DevOps paradigm focuses mainly on making cultural changes in the organization, depending on ways such as Continuous Delivery, which is directed at the automation and optimization of the delivery process [19] and showed in their review study strong relationship exists between deployment automation with reliability and maintainability which are significant attributes of software quality. This is also supported in qualitative study by Jabbari et al. [20] that automation is the key contributor to business proliferation and automation improve accuracy, reliability and agility. Thus, it contributes to software quality.

RQ3: Can measurement in the DevOps leads to increased software quality?

Measurement tends to produce consistent results enhancing software quality. Measurement ensures reliability which is the key metric in a software quality. Zarour et al. [4] observed that maturity models help to assess effectiveness of an organizational processes and assist in identifying capabilities required to improve their performance to move towards higher maturity levels. They found only few such models are available to assess in DevOps adopted practices and same applies for the assessment methods of these DevOps maturity models that how to assess the DevOps adoption for organizations to improve their maturity incrementally. They further recommended more research and

empirical work required to practice and validate the proposed DevOps maturity models. They noted that there are large similarities in the measured dimensions in most of models expect Bucena's and Feijter's models. Both models assist the culture dimension and both of them have validated their model in an independent organization they do not work for. Moreover, they further noticed that Bucena's model is holistic and covers all of the DevOps dimensions while Feijter's model is dedicated for SPO organizations. So, they suggested that Bucena's and Eficode's models are comprehensive and promising models to assess DevOps maturity.

Myrbakken and Colomo-Palacios [51] observed that implementing security that can sustain with DevOps is a challenge, but it can attain immense benefits if done properly. Prates et al. [28] identified metrics which can use to measure the effectiveness of DevSecOps methodology implementation inside organizations. DevSecOps is an emerging paradigm which includes security practices to the Software Development Cycle (SDL). Security practices in SDL are significant to avert data breaches, guarantee compliance with the law and is an obligation to protect customers data.

RQ4: Can sharing in DevOps impact the software quality?

In DevOps sharing is an important feature where the collaboration and knowledge sharing between people helps to make the process better as observed by Kaiser [52]. Gill et al. [29] reported that sharing enhances software quality. Sharing improves communication which ensures a product which improves the efficiency thus quality. It has demonstrated culture, automation, measurement and sharing (CAMS) are significant factors to consider to improve quality of the software [3]. According to Agarwal et al. [53] DevOps is all CAMS and it is gaining acceptance because of its continuous approach — continuous integration(CI), continuous deployment (CD) and software delivery. Shahin and Babar [39] observed that establishing DevOps culture (e.g., shared responsibility) and implementing its practices such as Continuous Delivery and Deployment (CD) requires new organizational capabilities and innovative techniques along with tools.

RQ5: Does DevOps culture has an impact on software quality?

DevOps provides a cultural change in organizations that uses: (1) a set of practices to get a visible workflow, which can be performed by the whole team; (2) the automation of processes, which cause a continuous feedback during the project and; (3) the improvement of learning through that experimentation [10]. Culture is another significant factor because it changes the manner in which teams work together and share the responsibility for the end users of their application. It is crucial to initiate mutual exchange with Dev and Ops to break down the barrier between teams and both teams will learn from each other [3].

DevOps means a culture shift towards collaboration between development, quality assurance, and operations [18]. According to Céspedes et al. [19] DevOps is a change in the organizational culture that aims to reduce the difference between development and operation teams, expediting the software release process. DevOps is not only cultural attitude, it is also a set of engineering practices impacted by cultural aspects and supported by technological enablers [9]. Smeds et al. [9] further argued that practitioners perceive as impediments of adopting DevOps, namely capabilities, culture, and technology.

According to Muñoz, and Negrete [54] it is important to highlight that the generic DevOps process has 4 phases, 8 activities and 40 tasks, which have to be followed in order to perform a software development process with DevOps culture. Colomo-Palacios et al. [25] observed that DevOps culture has an impact on software quality. It determines the employee's operation which in turn improve interoperability of an application.

RQ6: Does DevOps enable fast feedback helps in software quality?

Today, DevOps is an understood set of practices and cultural values that has been proven to help organizations of all sizes, improve their software release cycles, software quality, security, and ability to get rapid feedback on product development [34]. DevOps practices can help drive quality assurance by improving communication and feedback loops as noticed by Ibrahim et al. [42]. Nowadays customers are expecting faster feedback and changes related to issues or feature requests but contributing a result to this need, therefore organizations such as IBM, Facebook and Firefox are implementing DevOps practices [54]. Muñoz, and Negrete [54] further argued that feedback demonstrates how to get continuous augment feedback to keep the quality from the source avert the rework.

Presently users and customers of applications expect fast feedback to issues and feature requests [49]. DevOps facilitates provision of fast feedback loops. It is also observed that the feedback loops enable the achievement of software quality. This improves software efficiency. According to Muñoz, and Negrete [54] DevOps process could be improved with ISO/IEC 29110 series as reinforced DevOps which includes 4 phases: Inception, Construction, Transition, and Feedback.

RQ7: Does DevOps practices bridge development of software and software quality assurance?

The findings by Bou Ghantous, and Gill [27], indicate that DevOps practices link software development with quality assurance. Quality assurance improves the reliability of an application. Perara et al. [3] using multiple regression analysis has demonstrated culture, automation, measurement and sharing (CAMS) are significant factors to consider to improve quality of the software and proposed following model:

$$\text{Software Quality} = 1.409 + 0.176(\text{C}) + 0.227(\text{A}) + 0.096(\text{M}) + 0.172(\text{S})$$

Equation is represented with following notation C — Culture, A — Automation, M — Measurement and S — Sharing. Therefore, software quality will be increased if A, M, C, and S factors get increased. Perara et al. [3] argued that Automation is the critical success factor to improve software quality in DevOps environment. Culture, Sharing and Measure factors also require to be considered to enhance the software quality. Shahin and Babar [39] noticed that teams can improve their performance in DevOps by (1) including operations specialists in the teams to perform the operations tasks that require advanced expertise and (2) investment in testing, in particular automating tests arise during the last stages of DevOps pipelines, to release software changes in swift manner.

RQ8: How Software architecture contributes in DevOps success and quality?

Software Architecture (SA) is slated to be the foundation for reaching the highest level of DevOps success [6,55]. Most of the noted research related to SA and DevOps carried on in the context of CD as a key practice of DevOps [7,56,57]. Shahin et al. [7] have conducted a mixed-methods study to explore how SA is being impacted by or is impacting CD. They present a conceptual framework to support (re-) architecting for CD. Shahin et al. [7] also observed that a lack of explicit control on reusability poses challenges to practicing CD. Shahin and Babar [39] suggested that DevOps success is best associated with modular architectures and needs to prioritize deployability, testability, supportability, and modifiability over other quality attributes. They further argued that the successful architectural decisions made by the teams to support DevOps will be valuable for other organizations.

Daneva and Blosher [40] studied and found that (a) 17 software architecture characteristics are beneficial for CD and DevOps adoption, (b) micro-services are a dominant architectural style in this context, and (c) large-scale organizational contexts are the most studied, and (d) qualitative approaches (case study based) are the most applied research method. Pérez et al. [58] introduced a tool to fill the gap between development and operations in DevOps. Their tool was designed to identify architecture and user requirements and capable of providing feedback to the developer on the performance, reliability, and in general quality characteristics of the application at runtime. Chen [41] argue that a set of quality attributes such as deployability, security, modifiability, and monitorability, require more attention when designing architectures in the context of CD. Di Nitto et al. [59] delineates architecturally significant stakeholders (e.g., infrastructure provider) and their concerns (e.g., monitoring) in DevOps scenarios and proposed a framework called SQUID towards supporting the documentation of DevOps-driven software architectures and their quality properties.

Microservices architecture which pursue to deliver small, self-contained, and rigidly enforced atoms of functionality [60]. The tool set and microservices architecture supports the frequent delivery of new commercial-grade software features that can seamlessly connect with preexisting operational systems [61]. Recently Microservices Architecture (MSA) in DevOps has received significant consideration. Waseem et al. [62] conducted systematic analysis on MSA in DevOps and found : Three themes on the research on MSA in DevOps are "microservices development and operations in DevOps", "approaches and tool support for MSA based systems in DevOps", and "MSA migration experiences in DevOps". They further observed that most of the quality attributes (QAs) are positively affected during the implementation of MSA in DevOps.

RQ9: Does Continuous Delivery and Deployment (CD) in DevOps helps in ensuring completion on time along with quality?

The ultimate difference between DevOps and Continuous Delivery is that the first one focuses on the synergy between the development and operations areas, seeking the optimization of the product, in addition the continuous improvement of it [10]. Continuous delivery (CD) enables companies to deliver high quality software in a more efficient manner at a reduced time. Continuous Integration (CI) and CD has emerged as an aid for software development and release management practices to bring the capability to release quality artifacts continuously to customers in an integrated feedback as observed by Soni [63]. Team culture is the foundation for effective continuous delivery. There should be strong cross-cutting collaboration between development, test, security, operations, and related roles. Knowledge can be cross-pollinated through knowledge management practices. Team members can be trained on continuous delivery mode of requirement development and analysis, system architecture → CI architecture→ build pipeline, coding, testing, build scripting and related practices [11]. According to Céspedes et al. [19] test automation is related to the continuous delivery process and a relationship with reliability, maintainability and safety was observed.

Successful adoption of continuous deployment practice in software intensive organizations is reported to provide many benefits, including improvements in the delivery speed of software changes, improved software quality, improved developer productivity and improved customer satisfaction [41,64,65]. Whilst DevOps, CD, and micro-services share common characteristics (e.g., automating repetitive tasks) and support each other [66,67], organizations may embrace only one of the practices to achieve their business goals, for instance, providing quality software in a shorter time more reliably [68]. It is established that DevOps aims at improving the deployment frequency, low rates of failure and faster meantime for recovery in case of crush. Chen [41] worked towards adopting continuous deployment which focusses on how software companies can benefit from applying several automated tools for automatic deployment and almost every aspect of system development life cycle. Shahin et al. [55] performed the empirical investigation on deployment challenges. It provided the survey results conducted on small and big organizations adopting DevOps regarding the several challenges faced in automatic deployment activity. Waseem et al. [62] in their study on microservices architecture services in DevOps noticed that the leading positively affected QAs are Deployability and Scalability and other leading positively affected QAs are Performance, Maintainability, Monitoring, and Testability. They argued these results indicate that MSA in DevOps brings significant benefits, including, independent scalability, flexibility to employ novel frameworks, improved product quality, and zero downtime deployment.

RQ10: How does DevOps impacts on Usability, Efficiency, Maintainability and Portability in software?

In State of DevOps 2017 report, Forsgren et al. [34] found there is strong evidence that DevOps practices lead to higher IT performance based on more than 27,000 DevOps survey responses in past six years. According to Céspedes et al. [19] test automation is related to the continuous delivery process and thus helps in ensuring reliability, maintainability, and safety attributes of software quality. They further argued that reliability (availability) and maintainability (testability and modifiability) are the most referenced characteristics related to DevOps. It is shown by Bezemer et al. [69] that DevOps enables usability. Additionally, automation in the development operation is an enabler of efficiency.

Shahin and Babar (2020) reported that achieving DevOps-driven software architecture requires loosely coupled architectures and prioritizing deployability, testability, supportability, and modifiability over other quality attributes (i.e., confirming and extending) [7,57]. In DevOps, Céspedes et al. [19] noticed performance efficiency with cloud computing as it improves resources utilization and its modifiability is enhanced due to scalable on demand. Maintainability, on the other hand provide a standardized environment that ensures software quality. It has been observed that the flexibility in DevOps is an enabler of portability thus ensuring software quality. Kim et al. [70] also studied the aspects to include functionality, security, maintainability and reliability of DevOps systems. Waseem et al. [62] observed that only a few problems were reported regarding QAs (i.e., performance, scalability, security) of MSA based systems and could not find problems and solutions related to other QAs (e.g., availability, reusability, reliability, maintainability, modularity, portability), for instance, problems and solutions related to improving reusability of existing microservices for new microservices in MSA based systems.

Microservices pose security challenges due to inter-service communication over the distributed network [62] and Waseem et al. noticed that security is the most negatively affected, suggesting that MSA may introduce more vulnerabilities than monolithic applications for instance, microservices run via HTTP and use vulnerable third-party components, which may expose them for hackers' attack [62]. It is claimed that there are not many mature solutions available to address security issues of MSA based systems (Korolov, 2020) [71].

6. Threats to validity

As much as this systematic analysis would be important, like every other study, the analysis is accompanied by a few threats to its validity which are discussed in this section. The first threat is generalization. It involves viewing contexts from a general

perspective. There is a limitation on the academic search engines represent the academic DevOps search. This study has focused on a limited number of peer-reviewed studies available in selected electronic databases. However, the research on DevOps impact on software quality is not static and continues to evolve over time. The use of English-only papers might mean that works in other languages is left out. It is important to note that the categorization was also put under review by another researcher in order to reduce the risks of doing the work in the wrong way [72]. This systematic analysis is based on a limited number of literature and the definitions and features of DevOps may be evolving with time, this emerges one of the threats to this study. The second threat is biases. There are high chances that biases occurred in the selection of the literature. However, to deal with the biases data were randomly selected during the study. To ensure the significance of the outcomes, the initial findings were made available for discussion to the other researchers before writing this paper. Few meetings and a workshop program with other researchers were also involved to get feedback from the researchers working in this area. However, there was limited time to allow for the collection of data due to the larger volume of sources of data [24]. Threats to conclusion validity are concerned with issues that affect to come to correct conclusions and these risks were mitigated by applying Kitchenham & Charters [22] guidelines and Petersen et al. [73].

The validity from the external environment and the generalization of the research results is threatened by the small number of interviews and our overreliance on Google scholar and the Scopus for our data sources. Another threat to the validity of DevOps in impacting to quality software is creation of a DevOps department in an organization. This is because it will involve tedious work that is time consuming and eventually increase the gap between the development and operations of quality software. This will in turn affect the organization negatively.

7. Conclusion

Presently DevOps adoption has been increased in software development around the globe. In summary, we have analyzed the impacts of DevOps on the quality of software. In this process, we have examined various literature on the subject by use of various reference applications including Google scholar, IEEE Xplore among others. The analysis has also revealed automation, sharing and measurement characteristics (features) of DevOps have strong relationship with the quality and success in software development. It has also been found that DevOps is a fast feedback loop enabler which is essential in achieving software quality. The analysis has also revealed a connection between DevOps software architecture and quality assurance. Therefore, it can be concluded that DevOps contributes positive impacts towards ensuring software quality. Based on our analysis this study shows research was mainly focused in automation, culture, continuous delivery, fast feedback of DevOps. Also, this study found that DevOps in software quality is not just in theory but can also be identified in practice towards ensuring quality software. There is still limited primary studies related to the topic, though it is growing and there is need for empirical research along with survey based qualitative research to compare different contexts in various organizations and countries.

There is need of further research in many areas of DevOps (for instance: measurement, development of metrics of different stages to assess its performance, culture, practices toward ensuring quality assurance, and quality factors such as usability, efficiency, software maintainability and portability). Further, different survey questionnaires categorized by role may be developed to be answered by DevOps professionals.

Declaration of competing interest

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

Acknowledgment

We would like to thanks Timo Peder Brøyn, Director Library and Documentation of Molde University College - Special University in Logistics, for facilitating support towards open access publication as part of pilot agreement of Elsevier with the Norwegian institutions (Unit consortium).

References

[1] P. Debois, Agile infrastructure and operations: how infragile are you? in: Agile 2008 Conference, IEEE, 2008, August, pp. 202–207.
[2] R. Rapaport, A short history of DevOps, 2014, , (Accessed 28 December 2017).
[3] P. Perera, R. Silva, I. Perera, Improve software quality through practicing devops, in: 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions, ICTer, IEEE, 2017, pp. 1–6.
[4] M. Zarour, N. Alhammad, M. Alenezi, K. Alsarayrah, A research on DevOps maturity models, Int. J. Recent Technol. Eng. 8 (3) (2019) 4854–4862.
[5] L. Bass, I. Weber, L. Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley Professional, 2015.
[6] L. Leite, C. Rocha, F. Kon, D. Milojicic, P. Meirelles, A survey of devops concepts and challenges, ACM Comput. Surv. 52 (6) (2019) 1–35.
[7] M. Shahin, M. Zahedi, M.A. Babar, L. Zhu, An empirical study of architecting for continuous delivery and deployment, Empir. Softw. Eng. 24 (3) (2019) 1061–1108.
[8] V. Mohan, L.B. Othmane, Secdevops: Is it a marketing buzzword?-mapping research on security in devops, in: 2016 11th International Conference on Availability, Reliability and Security, ARES, IEEE, 2016, August, pp. 542–547.
[9] J. Smeds, K. Nybom, I. Porres, DevOps: A definition and perceived adoption impediments, in: C. Lassenius, T. Dingsøyr, M. Paasivaara (Eds.), XP 2015, in: LNBIP, vol. 212, Springer, Cham, 2015, pp. 166–177, http://dx.doi.org/10.1007/978-3-319-18612-2_14.
[10] M. Virmani, Understanding DevOps & bridging the gap from continuous integration to continuous delivery, in: 5th International Conference on Innovative Computing Technology, INTECH 2015, 2015, pp. 78–82.
[11] A. Cheriyan, R.R. Gondkar, S.S. Babu, Quality assurance practices and techniques used by QA professional in continuous delivery, in: Information and Communication Technology for Sustainable Development, Springer, Singapore, 2020, pp. 83–92.
[12] ISO/IEC, Software Engineering—Software Product Quality. Part 1: Quality Model, ISO/IEC 9126-1, Geneva, Switzerland, International Organization for Standardization, 2001.
[13] N.A.P.O. Perera, An Analysis on Implementing DevOps in Software Companies in Sri Lanka (Doctoral dissertation), 2016, Available at http://dl.lib.mrt.ac.lk/handle/123/12372.
[14] L. Banica, M. Radulescu, D. Rosca, A. Hagiu, Is devops another project management methodology? Inform. Econ. 21 (3) (2017) 39.
[15] M. Sánchez-Gordón, R. Colomo-Palacios, Characterizing DevOps culture: a systematic literature review, in: International Conference on Software Process Improvement and Capability Determination, Springer, Cham, 2018, October.
[16] P. Perera, M. Bandara, I. Perera, Evaluating the impact of devops practice in Sri Lankan software development organizations, in: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions, ICTer, IEEE, 2016, September, pp. 281–287.
[17] G. Casale, C. Chesta, P. Deussen, E. Di Nitto, P. Gouvas, S. Koussouris, et al., Current and future challenges of software engineering for services and applications, Cloud Forward 97 (2016) 34–42.
[18] C. Ebert, G. Gallardo, J. Hernantes, N. Serrano, DevOps, IEEE Softw. 33 (3) (2016) 94–100.
[19] D. Céspedes, P. Angeleri, K. Melendez, A. Dávila, Software product quality in DevOps contexts: A systematic literature review, in: International Conference on Software Process Improvement, Springer, Cham, 2019, October.
[20] R. Jabbari, N. bin Ali, K. Petersen, B. Tanveer, Towards a benefits dependency network for DevOps based on a systematic literature review, J. Softw.: Evol. Process 30 (11) (2018) e1957.
[21] F. Erich, C. Amrit, M. Daneva, Report: Devops Literature Review, Technical Report, University of Twente, 2014.

[22] B.A. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering Technical Report. Software Engineering Group, EBSE Technical Report, Keele University and Department of Computer Science University of Durham, 2, 2007.
[23] B. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering–a systematic literature review, Inf. Softw. Technol. 51 (1) (2009) 7–15.
[24] R. Jabbari, N. bin Ali, K. Petersen, B. Tanveer, What is DevOps? A systematic mapping study on definitions and practices, in: Proceedings of the Scientific Workshop Proceedings of XP2016, 2016, May, pp. 1–11.
[25] R. Colomo-Palacios, E. Fernandes, P. Soto-Acosta, X. Larrucea, A case analysis of enabling continuous software deployment through knowledge management, Int. J. Inf. Manage. 40 (2018) 186–189.
[26] M. Sánchez-Gordón, R. Colomo-Palacios, A multivocal literature review on the use of devops for e-learning systems, in: Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM'18, ACM, New York, NY, USA, 2018, October, pp. 883–888, http://dx.doi.org/10.1145/3284179.3284328.
[27] G. Bou Ghantous, A. Gill, DevOps: Concepts, Practices, Tools, Benefits and Challenges. PACIS2017, 2017.
[28] L. Prates, J. Faustino, M. Silva, R. Pereira, DevSecOps metrics, in: Euro Symposium on Systems Analysis and Design, Springer, Cham, 2019, September, pp. 77–90.
[29] A.Q. Gill, A. Loumish, I. Riyat, S. Han, DevOps for information management systems, VINE J. Inf. Knowl. Manage. Syst. (2018).
[30] L.E. Lwakatare, P. Kuvaja, M. Oivo, Dimensions of devops, in: International Conference on Agile Software Development, Springer, Cham, 2015, May, pp. 212–217.
[31] C. Jones, A proposal for integrating devops into software engineering curricula, in: International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment, 2018, March, pp. 33-47.
[32] S.I. Mohamed, Software release management evolution-comparative analysis across agile and DevOps continuous delivery, Int. J. Adv. Eng. Res. Sci. 3 (6) (2016) 236745.
[33] S.I. Mohamed, Devops shifting software engineering strategy-value based perspective, Int. J. Comput. Eng. 17 (2) (2015) 51–57.
[34] N. Forsgren, G. Kim, J. Humble, A. Brown, N. Kersten, State of DevOps Report. Puppet and DevOps Research & Assessment, 2017, Retrieved June, 10, 2017.
[35] F. Erich, C. Amrit, M. Daneva, A mapping study on cooperation between information system development and operations, in: International Conference on Product-Focused Software Process Improvement, Springer, Cham, 2014, December, pp. 277–280.
[36] J. Kroll, I. Richardson, R. Prikladnicki, J.L. Audy, Empirical evidence in follow the sun software development: A systematic mapping study, Inf. Softw. Technol. 93 (2018) 30–44.
[37] R. Laigner, M. Kalinowski, S. Lifschitz, R.S. Monteiro, D. de Oliveira, A systematic mapping of software engineering approaches to develop big data systems, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2018, August, pp. 446–453.
[38] M.Z. Toh, S. Sahibuddin, M.N.R. Mahrin, Adoption issues in devops from the perspective of continuous delivery pipeline, in: Proceedings of the 2019 8th International Conference on Software and Computer Applications, 2019, February, pp. 173–177.
[39] M. Shahin, M.A. Babar, On the role of software architecture in DevOps transformation: An industrial case study, 2020, arXiv preprint arXiv:2003.06108.
[40] M. Daneva, R. Bolscher, What we know about software architecture styles in continuous delivery and DevOps? in: M. van Sinderen, L. Maciaszek (Eds.), Software Technologies. ICSOFT 2019, in: Communications in Computer and Information Science, vol. 1250, Springer, Cham, 2020, http://dx.doi.org/10.1007/978-3-030-52991-8_2.
[41] L. Chen, Continuous delivery: Overcoming adoption challenges, J. Syst. Softw. 128 (2017) 72–86.
[42] M.M.A. Ibrahim, S.M. Syed-Mohamad, M.H. Husin, Managing quality assurance challenges of devops through analytics, in: Proceedings of the 2019 8th International Conference on Software and Computer Applications, 2019, February, pp. 194–198.
[43] M.S. Hamzehloui, S. Sahibuddin, K. Salah, A systematic mapping study on microservices, in: International Conference of Reliable Information and Communication Technology, Springer, Cham, 2018, June, pp. 1079–1090.
[44] S. Jones, J. Noppen, F. Lettice, Management challenges for DevOps adoption within UK SMEs, in: Proceedings of the 2nd International Workshop on quality-aware devops, 2016, July, pp. 7–11.
[45] L.E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, et al., DevOps in practice: A multiple case study of five companies, Inf. Softw. Technol. 114 (2019) 217–230.
[46] R.K. Gupta, M. Venkatachalapathy, F.K. Jeberla, Challenges in adopting continuous delivery and DevOps in a globally distributed product team: a case study of a healthcare organization, in: 2019 ACM/IEEE 14th International Conference on Global Software Engineering, ICGSE, IEEE, 2019, May, pp. 30–34.
[47] D. Perez-Palacin, Y. Ridene, J. Merseguer, Quality assessment in DevOps: automated analysis of a tax fraud detection system, in: Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion, 2017, April, pp. 133–138.
[48] M. Muñoz, M. Negrete, J. Mejía, Proposal to avoid issues in the devops implementation: A systematic literature review, in: World Conference on Information Systems and Technologies, Springer, Cham, 2019, April, pp. 666–677, Proceedings of XP2016 (p. 12).
[49] J. Wettinger, U. Breitenbücher, O. Kopp, F. Leymann, Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel 56 (2016) 317–332.
[50] V. Gupta, P.J. Kapur, D. Kumar, Modeling and measuring attributes influencing devops implementation in an enterprise using structural equation modeling, Inf. Softw. Technol. 92 (2017) 75–91.
[51] H. Myrbakken, R. Colomo-Palacios, DevSecOps: a multivocal literature review, in: International Conference on Software Process Improvement and Capability Determination, Springer, Cham, 2017, October, pp. 17–29.
[52] A.K. Kaiser, Introduction to DevOps, in: Reinventing ITIL® in the Age of DevOps, Apress, Berkeley, CA, 2018, pp. 1–35.
[53] A. Agarwal, S. Gupta, T. Choudhury, Continuous and integrated software development using DevOps, in: 2018 International Conference on Advances in Computing and Communication Engineering, ICACCE, IEEE, 2018, June, pp. 290–293.
[54] M. Muñoz, M. Negrete, Reinforcing DevOps generic process with a guidance based on the basic profile of ISO/IEC 29110, in: International Conference on Software Process Improvement, Springer, Cham, 2019, October, pp. 65–79.
[55] M. Shahin, M.A. Babar, L. Zhu, Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices, IEEE Access 5 (2017) 3909–3943.
[56] S. Bellomo, N. Ernst, R. Nord, R. Kazman, Toward design decisions to enable deployability: Empirical study of three projects reaching for the continuous delivery holy grail, in: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE, 2014, June, pp. 702–707.
[57] L. Chen, Towards architecting for continuous delivery, in: 2015 12th Working IEEE/IFIP Conference on Software Architecture, IEEE, 2015, May, pp. 131–134.
[58] J.F. Pérez, W. Wang, G. Casale, Towards a DevOps approach for software quality engineering, in: Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development, 2015, January, pp. 5–10.
[59] E. Di Nitto, P. Jamshidi, M. Guerriero, I. Spais, D.A. Tamburri, A software architecture framework for quality-aware DevOps, in: Proceedings of the 2nd International Workshop on Quality-Aware DevOps, 2016, July, pp. 12–17.
[60] J. Thönes, Microservices, IEEE Softw. 32 (1) (2015) 116.
[61] R.V. O'Connor, P. Elger, P.M. Clarke, Continuous software engineering—A microservices architecture perspective, J. Softw.: Evol. Process 29 (11) (2017) e1866.
[62] M. Waseem, P. Liang, M. Shahin, A systematic mapping study on microservices architecture in devops, J. Syst. Softw. (2020) 110798.
[63] M. Soni, End to end automation on cloud with build pipeline: the case for devops in insurance industry, continuous integration, continuous testing, and continuous delivery, in: 2015 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM, IEEE, 2015, November, pp. 85–89.
[64] M. Leppänen, S. Mäkinen, M. Pagels, V.P. Eloranta, J. Itkonen, .M.V. Mäntylä, T. Männistö, The highways and country roads to continuous deployment, IEEE Softw. 32 (2) (2015) 64–72.
[65] C. Parnin, E. Helms, C. Atlee, H. Boughton, M. Ghattas, A. Glover, et al., The top 10 adages in continuous deployment, IEEE Softw. 34 (3) (2017) 86–95.
[66] XebiaLabs, Exploring microservices: 14 questions answered by experts, 2020, Available at https://blog.xebialabs.com/wp-content/uploads/2015/04/Microservices-Whitepaper.pdf.
[67] M. Schmidt, Devops and continuous delivery: Not the same, Mediaops, LLC 8 (04) (2016) https://bit.ly/2vEme4H.
[68] T. Laukkarinen, K. Kuusinen, T. Mikkonen, Devops in regulated software development: case medical devices, in: 2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track, ICSE-NIER, IEEE, 2017, May, pp. 15–18.
[69] C.P. Bezemer, S. Eismann, V. Ferme, J. Grohmann, R. Heinrich, P. Jamshidi, et al., How is performance addressed in DevOps? in: Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, 2019, April, pp. 45–50.
[70] G. Kim, J. Humble, P. Debois, J. Willis, The DevOps Handbook:: How To Create World-Class Agility, Reliability, and Security in Technology Organizations, IT Revolution, 2016.

[71] M. Korolov, Why securing containers and microservices is a challenge, 2020, https://www.csoonline.com/article/3268922/why-securing-containers-and-microservices-is-a-challenge.html.%20Accessed%20on%202020-08-10Accessedon2020-08-10.
[72] M. Rajkumar, A.K. Pole, V.S. Adige, P. Mahanta, DevOps culture and its impact on cloud delivery and software development, in: 2016 International Conference on Advances in Computing, Communication, & Automation, ICACCA, Spring, IEEE, 2016, April, pp. 1–6.
[73] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), vol. 12, 2008, June, pp. 1–10.

Appendix: Selected Studies: (35)

[74] Eismann Bezemer, Grohmann Ferme, Jamshidi Heinrich, Willnecker, How is Performance Addressed in DevOps? in: Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, ACM, 2019, April, pp. 45–50.
[75] G. Bou Ghantous, A. Gill, DevOps: Concepts, practices, tools, benefits and challenges, in: PACIS2017, 2017.
[76] L. Chen, Towards architecting for continuous delivery, in: 2015 12th Working IEEE/IFIP Conference on Software Architecture, IEEE, 2015, May, pp. 131–134.
[77] L. Chen, Continuous delivery: Overcoming adoption challenges, J. Syst. Softw. 128 (2017) 72–86.
[78] A. Cheriyan, R.R. Gondkar, S.S. Babu, Quality assurance practices and techniques used by QA professional in continuous delivery, in: Information and Communication Technology for Sustainable Development, Springer, Singapore, 2020, pp. 83–92.
[79] D. Céspedes, P. Angeleri, K. Melendez, A. Dávila, Software product quality in DevOps contexts: A systematic literature review, in: International Conference on Software Process Improvement, Springer, Cham, 2019, October, pp. 51–64.
[80] R. Colomo-Palacios, E. Fernandes, P. Soto-Acosta, X. Larrucea, A case analysis of enabling continuous software deployment through knowledge management, Int. J. Inf. Manage. 40 (2018) 186–189.
[81] M. Daneva, R. Bolscher, What we know about software architecture styles in continuous delivery and devops? in: M. van Sinderen, L. Maciaszek (Eds.), Software Technologies. ICSOFT 2019, in: Communications in Computer and Information Science, vol. 1250, Springer, Cham, 2020, http://dx.doi.org/10.1007/978-3-030-52991-8_2.
[82] E. Di Nitto, P. Jamshidi, M. Guerriero, I. Spais, D.A. Tamburri, A software architecture framework for quality-aware DevOps, in: Proceedings of the 2nd International Workshop on Quality-Aware DevOps, 2016, July, pp. 12–17.
[83] C. Ebert, G. Gallardo, J. Hernantes, N. Serrano, Devops, IEEE Softw. 33 (3) (2016) 94–100.
[84] N. Forsgren, G. Kim, J. Humble, A. Brown, N. Kersten, State of DevOps Report. Puppet and DevOps Research & Assessment, 2017, Retrieved June, 10, 2017.
[85] A.Q. Gill, A. Loumish, I. Riyat, S. Han, DevOps for information management systems, VINE J. Inf. Knowl. Manage. Syst. (2018).
[86] V. Gupta, P.J. Kapur, D. Kumar, Modeling and measuring attributes influencing devops implementation in an enterprise using structural equation modeling, Inf. Softw. Technol. 92 (2017) 75–91.
[87] R.K. Gupta, M. Venkatachalapathy, F.K. Jeberla, Challenges in adopting continuous delivery and DevOps in a globally distributed product team: a case study of a healthcare organization, in: 2019 ACM/IEEE 14th International Conference on Global Software Engineering, ICGSE, IEEE, 2019, May, pp. 30–34.
[88] M.M.A. Ibrahim, S.M. Syed-Mohamad, M.H. Husin, Managing quality assurance challenges of DevOps through analytics, in: Proceedings of the 2019 8th International Conference on Software and Computer Applications, 2019, February, pp. 194–198.
[89] R. Jabbari, N. bin Ali, K. Petersen, B. Tanveer, Towards a benefits dependency network for DevOps based on a systematic literature review, J. Softw.: Evol. Process 30 (11) (2018) e957.
[90] A.K. Kaiser, Introduction to devops, in: Reinventing ITIL® in the Age of DevOps, Apress, Berkeley, CA, 2018, pp. 1–35.
[91] G. Kim, J. Humble, P. Debois, J. Willis, The DevOps Handbook:: How To Create World-Class Agility, Reliability, and Security in Technology Organizations, IT Revolution, 2016.
[92] T. Laukkarinen, K. Kuusinen, T. Mikkonen, DevOps in regulated software development: case medical devices, in: 2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track, ICSE-NIER, IEEE, 2017, May, pp. 15–18.
[93] M. Leppänen, S. Mäkinen, M. Pagels, V.P. Eloranta, J. Itkonen, .M.V. Mäntylä, T. Männistö, The highways and country roads to continuous deployment, IEEE Softw. 32 (2) (2015) 64–72.
[94] L. Leite, C. Rocha, F. Kon, D. Milojicic, P. Meirelles, A survey of DevOps concepts and challenges, ACM Comput. Surv. 52 (6) (2019) 1–35.
[95] L.E. Lwakatare, P. Kuvaja, M. Oivo, Dimensions of DevOps, in: International Conference on Agile Software Development, Springer, Cham, 2015, May, pp. 212–217.
[96] M. Muñoz, M. Negrete, Reinforcing DevOps generic process with a guidance based on the basic profile of ISO/IEC 29110, in: International Conference on Software Process Improvement, Springer, Cham, 2019, October, pp. 65–79.
[97] P. Perera, R. Silva, I. Perera, Improve software quality through practicing DevOps, in: 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions, ICTer, IEEE, 2017, pp. 1–6.
[98] P. Perera, M. Bandara, I. Perera, Evaluating the impact of devops practice in Sri Lankan software development organizations, in: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions, ICTer, IEEE, 2016, September, pp. 281–287.
[99] J.F. Pérez, W. Wang, G. Casale, Towards a DevOps approach for software quality engineering, in: Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development, 2015, January, pp. 5–10.
[100] D. Perez-Palacin, Y. Ridene, J. Merseguer, Quality assessment in DevOps: automated analysis of a tax fraud detection system, in: Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion, 2017, April, pp. 133–138.
[101] L. Prates, J. Faustino, M. Silva, R. Pereira, DevSecOps metrics, in: Euro Symposium on Systems Analysis and Design, Springer, Cham, 2019, September, pp. 77–90.
[102] M. Sánchez-Gordón, R. Colomo-Palacios, Characterizing DevOps culture: a systematic literature review, in: International Conference on Software Process Improvement and Capability Determination, Springer, Cham, 2018, October, pp. 3–15.
[103] J. Smeds, K. Nybom, I. Porres, in: C. Lassenius, T. Dingsøyr, M. Paasivaara (Eds.), XP 2015, in: LNBIP, vol. 212, Springer, Cham, 2015, pp. 166–177, http://dx.doi.org/10.1007/978-3-319-18612-2_14.
[104] M. Shahin, M. Zahedi, M.A. Babar, L. Zhu, An empirical study of architecting for continuous delivery and deployment, Empir. Softw. Eng. 24 (3) (2019) 1061–1108.
[105] M. Shahin, M.A. Babar, L. Zhu, Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices, IEEE Access 5 (2017) 3909–3943.
[106] M. Virmani, Understanding DevOps & bridging the gap from continuous integration to continuous delivery, in: 5th International Conference on Innovative Computing Technology, INTECH 2015, 2015, pp. 78–82.
[107] J. Wettinger, U. Breitenbücher, O. Kopp, F. Leymann, Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel 56 (2016) 317–332.
[108] M. Zarour, N. Alhammad, M. Alenezi, K. Alsarayrah, A research on devops maturity models, Int. J. Recent Technol. Eng. 8 (3) (2019) 4854–4862.