SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943


# DevOps Adoption Guidelines, Challenges and Benefits: A Systematic Literature Review

Usman Hamza[1,*], Nasuha Lee Abdullah[1], Sharifah Mashita Syed-Mohamad[2]

[1] School of Computer Sciences, Universiti Sains Malaysia, 11700 Gelugor, Pulau Pinang, Malaysia
[2] Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, 21300 Kuala Terengganu, Terengganu, Malaysia

**ARTICLE INFO**

***Article history:***
Received
Received in revised form
Accepted
Available online

***Keywords:***

DevOps guidelines; Challenges; Benefits; Systematic literature review

**ABSTRACT**

The DevOps methodology in software engineering is currently trending and many software development organisations are adopting it to stay competitive. However, these organisations face difficulties in adopting DevOps due to a lack of common standard guidelines and methodology. We argued that guidelines across multiple platforms are needed to succeed in DevOps implementation. This paper aims to identify the benefits, challenges, and guidelines associated with DevOps adoption using a systematic literature review. The search spanned over ten digital libraries such as ACM Digital Library, ScienceDirect, EBSCOhost, SAGE Journals, Wiley Online Library, and so on. The review also included keyword searches in relevant studies i.e DevOps guidelines, DevOps and guidelines, DevOps benefits, DevOps and benefits, DevOps challenges, DevOps and challenges. Papers written in English and published from 2011 through July 2020 were considered. 33 studies passed the two-step quality assessment. Finally, the data were extracted and analysed based on the themes and research questions, revealing the benefits and challenges of DevOps adoption guidelines. The findings show that improved software delivery speed increases code quality and application quality with frequent releases. It also shows that there was no transparent scientific approach to determining the DevOps capabilities regarding the process, practice, tools, and implementations, cost implications, and insufficient knowledge of the automation process. Finally, the reviewed studies indicated that software quality increases whenever DevOps best practices are adopted. DevOps is about continuous software delivery. Previous studies have not considered the guidelines for scientifically adopting DevOps for organisations that want to migrate. The present review identified the DevOps adoption guideline model represented in a cyclic format that indicates an infinite process representing continuous integration, continuous delivery, and continuous deployment, with an understanding of benefits and challenges.

## 1. Introduction

DevOps originated in agile software development by combining development and operations (developers and operators) [1]. DevOps is considered a phenomenon or set of principles and practices

* *Corresponding author.*
*E-mail address: usmanhamza@student.usm.my*

https://doi.org/10.37934/araset.63.2.114136

that enable collaborations and communications to produce an efficient team of developers and operators [2,3]. DevOps is seen as a modern software development methodology and a response to the inter-dependencies between development and operations by taking a modern approach and producing tools through collaboration between developers and operators [4,5] . The DevOps methodology in software engineering is evolving rapidly due to customers' high expectations and technological advancements. The current trend has motivated software development organisations and teams to adopt DevOps principles to stay competitive. However, these organisations face difficulties in adopting these principles. DevOps implementation strategies lack common standard guidelines and methodology [6,7].

Researchers have been actively investigating DevOps' scientific and technical stand for organisations' adoption and practices and have developed many techniques and models [8-10] . However, none of such studies focuses on specific aspects related to DevOps adoption guidelines to our best knowledge. Therefore, this review explores some considerations and procedures for DevOps adoption and identifies challenges and obstacles that prevent its optimal adoption. Forward/backward snowballing iterations were used to ensure that all papers matched keywords. As a result, 810 primary studies were scrutinised, of which 33 preliminary studies eventually passed the selection criterion and were extensively studied. A classification framework was developed based on the main components of DevOps methodologies.

Based on the research of Al Smin *et al.,* [11] conducted a case study using a web survey on the DevOp adoption process to characterise relevant factors of successful DevOps adoption. The reported findings from the web survey analysis could serve as inputs for future framework development [12,13] . However, several studies with existing methods and software development tools could benefit from improvements in the deployment, delivery and maintenance phases [14] . DevOps improves these phases by collecting principles and practices and collaborating between development and operations teams [15]. DevOps is the collaboration between development and operations teams and represents a new way of thinking in the context of the software engineering field that has recently received much attention [16]. A similar study conducted systematic mapping and selected 44 articles that discuss DevOps' definition based on DevOps practices and how DevOps is related to other existing methods. Some studies mentioned a combination of descriptions, techniques, and relations to other forms, leading other researchers to propose a definition for DevOps that includes its benefits. Despite an increasing demand for software delivery, both in industry and academia, deployment obstacles often impede the quick delivery of the new and improved features to end users [17,18]. Similar studies have adopted an SLR approach to identify challenges to adopting DevOps in practice before disembarking on the journey effectively. There is a need to clearly understand DevOps concepts, training, tools, and benefits. The present study applied the SLR method, and the results indicate that the techniques could create shorter lead times and better communication within and between development teams [19] . There are also some challenges and drawbacks to change management, software quality assurance, and stakeholder acceptance [20,21]. Other studies suggest considering the comprehensive reporting of context and how to implement the practice, and the adoption of a metric is very relevant for achieving an efficient security approach instead of referring to the use of the procedure [21-24] . Some studies have classified DevOps based on principles, practices, benefits, and challenges, which were not sufficiently addressed in the peer-reviewed literature [24]. This review further conducts an SLR to identify the benefits and challenges of DevOps adoption and the guidelines for comprehending the current difficulties in adoption. None of the reviewed studies indicates interest or signs of developing policies for DevOps adoption. Our study identified some common guidelines that organisations need to consider before adopting DevOps that would help them succeed in implementing DevOps.

DevOps is a collection of processes and philosophies that include the four fundamental elements of culture, collaboration, tools, and operations. Due to a lack of common standard guidelines and methodology, as well as a transparent scientific approach to determining DevOps capabilities in terms of process, practise, tools, and implementations, cost implications and a lack of knowledge of the automation process, organisations are having difficulty adopting DevOps. Using a systematic literature review, this paper aims to identify the benefits, challenges, and proposed guidelines associated with DevOps adoption. The findings from the previous studies show an increased software delivery speed with improved code and application quality with frequent releases when best practices were implemented. The reviewed study also found that when DevOps best practices are implemented, software quality improves while continuous software delivery is the focus of DevOps.

## 2. Methodology

This study follows the guidelines for systematic reviews in software engineering research [21,25]. The methodology has three phases: planning the review, conducting the study, and reporting the extracted results. The literature search involved an automated search of ten popular digital libraries and several iterations of the snowballing approach to ensure the completeness of results [26,27].

### *2.1 Planning the Review*

The plan is to discover the current DevOps adoption guidelines and review the existing approach. In particular, the research questions addressed by this review are:

RQ1 What are the DevOps guidelines?
RQ2 What are the challenges in DevOps adoption?
RQ3 What are the benefits of DevOps adoption?

### *2.2 Conducting the Review*

The review started with the identification of relevant primary studies. The search spanned over the 10 digital libraries most commonly used in similar studies: SCOPUS, EBSCO-Discovery Service, ACM Digital Library, ScienceDirect, Wiley Online Library, SAGE Journal, Google Scholar, SpringerLink, EBSCO Academic Search Complete, and ProQuest. The process was carried out in the five steps described in Table 1.

**Table 1**
Inclusion and Exclusion Criteria

| Step | Inclusion Criteria | Exclusion Criteria |
|---|---|---|
| 1 | Studies presenting DevOps concepts/definitions and methodology | Studies not explicitly providing solutions to DevOps challenges and practice |
| 2 | Studies presenting DevOps strategies/practice | Studies presenting surveys |
| 3 | Studies presenting DevOps issues/goals, models, and their benefits | Editorials, abstract, thesis panels, monographs, books, book sections |
| 4 | Studies presenting DevOps principles | Articles that are not written in English |
| 5 | Studies presenting case studies related to DevOps strategies/guidelines | Studies published before 2011 |

A checklist was used as a reference for the quality assessment. Each statement is evaluated based on a three-point numeric scale: yes (two points), partial (one point), and no (0 points). Since there were five questions on the checklist, the maximum number of points any study could be assigned was 10 points; the minimum was 0 points. An article had to receive at least five points to be included in the final primary studies. All items with four or fewer points were excluded from the research for low quality. It is also essential to understand that the assessed article was chosen based on their high relevance to our study. A perfect item could have been excluded if it did not have the desired relevant grade points per the checklist, which was designed to locate highly relevant reviews only. The aim was not to rank the articles depending on their scores; the goal was to exclude articles that contributed little to our research.

A quality assessment using sensitivity analysis was performed in this review by checking whether the selected primary reviews from the previous phase reached the study's minimum quality requirements. Papers that did not meet the stated condition were excluded from the study. The studies that passed this quality assessment (QA) are the final papers considered in the primary research [28].

### *2.3 Automated Search of Digital Libraries*

The process involved conducting an automated search in the following electronic sources that are relevant to software engineering research:

i. Scopus
ii. ACM Digital Library
iii. ScienceDirect
iv. EBSCOhost
v. SAGE Journals
vi. Wiley Online Library
vii. ProQuest

Specifically, we searched by title, abstract, and keywords. Papers written in English and published from 2011 to July 2020 were considered. We tried to be as comprehensive as possible. Consequently, we used a general search string across all explored databases, such as "DevOps guidelines", DevOps and guidelines", "DevOps or guidelines", and "guideline of DevOps." We retrieved similar results from databases. We used a single relation in our search string because we wanted to focus on a specific data analysis direction to produce transparent and reliable results.

### *2.4 The Selection Approach Based on Inclusion and Exclusion Criteria*

We started the selection process by reading the title, abstract, and keywords of each study and selecting them according to the criteria in Table 1. Then, we excluded studies that are not primary studies and works that were not peer-reviewed or contained less than four (4) pages (e.g., theses or abstracts). We also excluded books, book titles, and monographs because they usually present mature research work and merge results that are previously presented or consumed by journals or conferences.

We also used a sensitivity analysis approach. After that, we performed a second selection by reading the full text of each included paper. We defined a sensitivity analysis on the item checklist composed of the five criteria in Table 2 and an acceptance or rejection of either "Yes" or "No", respectively, provided by the individual research assessment scores as shown in the last three (3) columns in Table 2.

The sensitivity analysis aimed to exclude studies of inferior quality per our research focus. This analysis included two stages. In the first stage, each study was read by a randomly selected author assigned to each criterion in Table 2. The acceptance of "Yes" and "No" was provided so that any study that satisfied the sensitivity criteria would be assigned a pass of "Yes" and would be included in the present review. Our "No" sensitivity criteria were excluded if a unique research paper failed. The second stage of the sensitivity analysis was carried out to ensure that high-quality documents were selected from those that passed the first stage. The author, different from the initial one, in the early stage, was assigned to read and assess the task using the same procedure used in the first stage and produce a second sensitivity analysis [28].

**Table 2**
Sensitivity Analysis Table

| Step | Items Checklist | Included (2 point) | Partial (1 point) | Excluded (0 point) |
|---|---|---|---|---|
| 1 | Was the problem statement of the study clearly stated? | Yes | 1 | No |
| 2 | Were the initial research gaps and contributions of the study clearly stated? | Yes | 1 | No |
| 3 | Were the results focused on a single relationship and validated? | Yes | 1 | No |
| 4 | Are the limitations and recommendations for future research clearly stated? | Yes | 1 | No |
| 5 | Are the focus of the DevOps approach and its principles clearly defined? | Yes | 1 | No |

### *2.5 Searching Based on Snowballing*

Snowballing is a search technique typically used to complement automated queries. We adopted backward/forward snowballing to identify additional papers that the initial computerised search might not have captured. In each selected preliminary study, we adapted backward snowballing, its list of references, and forward snowballed its citations in Google Scholar (http://scholar.google.com/). We first decided to consider all papers published from 2011–2019 that were not included in the initial search; then, we applied the previously used sensitivity analysis to the other retrieved primary studies [27].

### *2.6 Assessing the Research Methodology*

Lastly, we performed another literature search to verify all the relevant primary studies. We conducted an automated search over the following electronic databases:

i. Wiley Online Library (https://onlinelibrary.wiley.com)
ii. Ebsco Discovery Service (http://eds.b.ebscohost.com/eds/search/)
iii. Springer Link (https://link.springer.com)
iv. From this search, we found 18 relevant primary studies, confirming the snowballing approach's validity, reliability, and completeness.

The search results (Table 3) were filtered, and a duplicate was removed from the first assessment.

**Table 3**
Search Results with the First Assessment

| S/N | Databases Used | Search Results | Databases with First Assessment | |
|---|---|---|---|---|
| 1 | ACM Dig Lib | 27,058 | ACM Dig Lib | 117 |
| 2 | Ebsco Discov Service | 1,570 | Ebsco Dis S. | 83 |
| 3 | Ebsco Academic Search Complete | 18 | Ebsco Academic Search Complete | 12 |
| 4 | Google Scholar | 3,723 | Google Scholar | 17 |
| 5 | Scopus | 14 | Scopus | 3 |
| 6 | SAGE Journal | 01 | SAGE Journal | 2 |
| 7 | ScienceDirec | 109 | ScienceDirect | 23 |
| 8 | Wiley Oline Lib | 89 | Wiley OLt | 15 |
| 9 | SpringerLink | 233 | SpringerLink | 77 |
| 10 | ProQuest | 1544 | ProQuest | 09 |
| TOTAL | 32, 960 | TOTAL | 358 | |

### *2.7 Literature Classification Steps*

This review classified papers based on the classification criteria and then presented a framework model characterised by different DevOps research areas. The research review derived this framework gradually. Initially, draft schedule topics were generated based on the reading of titles, keywords, and abstracts during the paper-selection stages. This topic comprises six areas with possible branches. We later used these labelled topics to classify the documents while reading the full text and continued to feed new subtopics as needed within each area.

#### *2.7.1 DevOps concepts*

We describe six (6) areas of the DevOps loop flow framework model and their topics based on the DevOps concepts' linkage. All the papers present perspectives on all other relevant DevOps research areas. They address DevOps definitions and methodologies, practice, principles, strategies, benefits, challenges, and software domains concerning the adoption and future research for manageable DevOps implementation.

#### *2.7.2 DevOps adoption methodologies*

Studies belonging to this category present a methodological approach to DevOps implementation, such as DevOps models and automation.

#### *2.7.3 DevOps practices*

The papers belonging to this area describe DevOps best practices and solutions to some challenges of DevOps adaption. They analyse the definition of DevOps metrics and different strategies for DevOps implementations such as configuration management, continuous testing, continuous monitoring, continuous integration, and continuous deployment/delivery.

#### *2.7.4 DevOps principles*

These studies mention the different stages of DevOps, such as continuous integration (CI), continuous delivery (CD), and continuous deployment (CD). They also comply with the DevOps pipeline showing a specified performance with reliability, automation [29] , measurement, sharing feedback, and fast software release, among other matters [26,30].

*2.7.5 DevOps strategies*

These studies evaluated different DevOps teams' activities and various management approaches. Silos and targeted results, providing support and quick responses for each group, analysing the process, and quality assurance evaluation of different software delivery. This also involves defining a strategy [29], understanding requirements, and selecting and testing tools, as shown in Figure 1.

*2.7.6 DevOps benefits*

These papers present DevOps benefits using an automated toolset for both development and operation teams to break the silos arising from each specific application domain, including web transaction payments of web applications and mobile applications.

*2.7.7 DevOps challenges*

Based on the articles in this category, present DevOps challenges include different team locations or different countries' policies, a lack of a standard methodology, and the lack of a framework for adoption.

## 3. Results

Finally, the resulting framework serves as a valuable contribution to DevOps research trends during data analysis.

Figure 1 is a continuous DevOps implementation guide that requires best practices with a delivery cycle comprising planning, development, testing, deployment, release, and monitoring. And with active cooperation between different team members, consisting of a series of activities, such as continuous development, continuous automated testing, deployment, continuous monitoring, continuous integration, and continuous delivery [31] . And high-level principles with the acronym "CAMS" (culture, automation, measurement, and sharing). And that includes culture, constant collaboration, and communication.

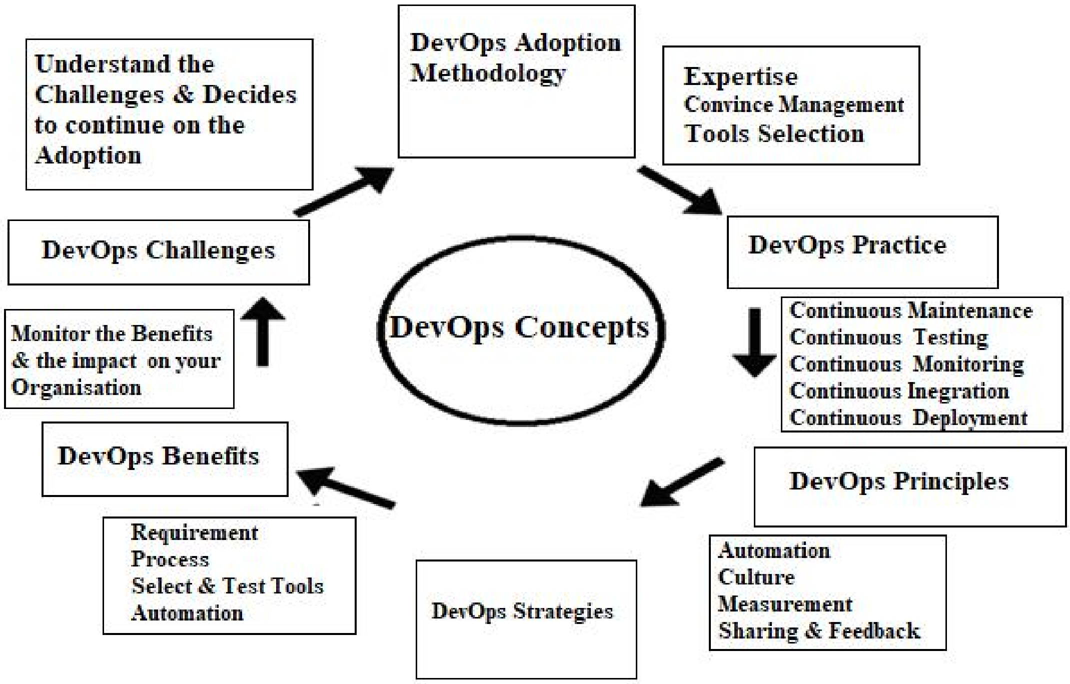


**Fig. 1.** DevOps Loop-Flow Framework Model

Figure 1 is a loop flow framework model that characterises the relevant guides to DevOps implementation and pinpoints the understanding of continuous delivery, which served as a starting point for identifying a common methodology for DevOps adoption in the existing laid-down guidelines.

This research shaped the framework in a looping system because DevOps is a continuous and repeated process along the pipeline.

A further brief explanation of an individual loop flow framework stage is provided presently [32,33].

This section also reports the number of primary studies selected in each of the previous steps and presents some quantitative analyses of the results, as shown in Table 4.

**Table 4**
Primary Studies Included, along with the Research Questions Where They Were Mapped

| Study Indicators (S) | References | RQs Mapped |
|---|---|---|
| S1 | [29] | RQ1 & RQ2 |
| S2 | [26] | RQ2 |
| S3 | [7] | RQ1, RQ2 |
| S4 | [19] | RQ2 |
| S5 | [13] | RQ1, RQ2 |
| S6 | [29] | RQ1, RQ2 |
| S7 | [25] | RQ1, RQ2 |
| S8 | [34] | RQ1 & RQ2 |
| S9 | [35] | RQ1 & RQ2 |
| S10 | [23], | RQ1 |
| S11 | [15] | RQ1, RQ2 |
| S12 | [34] | RQ1, RQ2 |
| S13 | [32,36] | RQ1 |
| S14 | [14] | RQ2 |
| S15 | [25] | RQ2 |

| | | |
|---|---|---|
| S16 | [6] | RQ1 & RQ2 |
| S17 | [7] | RQ1 & RQ2 |
| S18 | [5] | RQ2 |
| S19 | [8] | RQ2 |
| S20 | [35] | RQ1 |
| S21 | [37] | RQ2 & RQ1 |
| S22 | [1,38] | RQ1 |
| S23 | [39] | RQ2 |
| S24 | [40] | RQ1 |
| S25 | [41] | RQ1 |
| S26 | [42] | RQ1 |
| S27 | [43] | RQ1 |
| S28 | [44] | RQ1, RQ2 |
| S29 | [45] | RQ1 |
| S30 | [43] | RQ2 |
| S31 | [46] | RQ2 |
| S32 | [47] | RQ2 |
| S33 | [48] | RQ2 |

The data was extracted based on the themes and research questions [31] . Except for DevOps adoption, six (6) themes were identified, with three (3) pieces considered for each research question. Articles were selected based on their impact and relevance to answering the research questions and their strong support from the primary literature. The selected articles are listed in Table 5.

**Table 5**
Identified Themes and the Primary Studies Supporting Them

| Themes | Study Indicators |
|---|---|
| DevOps Methodology | |
| CI/CDs Practice & Collaborations | [8,13-17,19,23,25,28] |
| Strategies and<br>Shared Responsibility | [11,12,14,16,17,24,25] |
| Principles, Domains<br>And guideline | [10,13,16,20,25,28,30,33] |
| <u>DevOps Adoption</u> | |
| CI/CD Pipelines | [12,19,20,23] |
| Monitoring | [12,14,15,18,19,23,24] |
| Automation | [9,S11,14,17,21,23-26] |
| <u>DevOps Advantages</u> | |
| Faster release | |
| Cycles and | [11,-13,20,21,23] |
| Low Cost | |
| Higher productivity | [8,11-14] |
| and efficiency | [11-13,20,27,28] |
| Quality | |
| <u>DevOps Challenges</u> | |
| Lack of skills and | [11,12,20,21,24,25] |
| Knowledge | [5,6,13-15,17] |
| Resistance to | [2,5,6,17,19,20,23] |
| Change (Silos) | |
| Un-standardise Tools and High cost | |

### *3.1 Numerical Output*

The automated search described in 2.3 returned an initial collection of 358 primary studies. The detailed results of the digital libraries considered in this step are reported in Table 4 and filtered in this initial collection according to the inclusion and exclusion criteria stated in Table 2. Of these, 33 studies passed the two-step quality assessment, as presented in Figure 2.

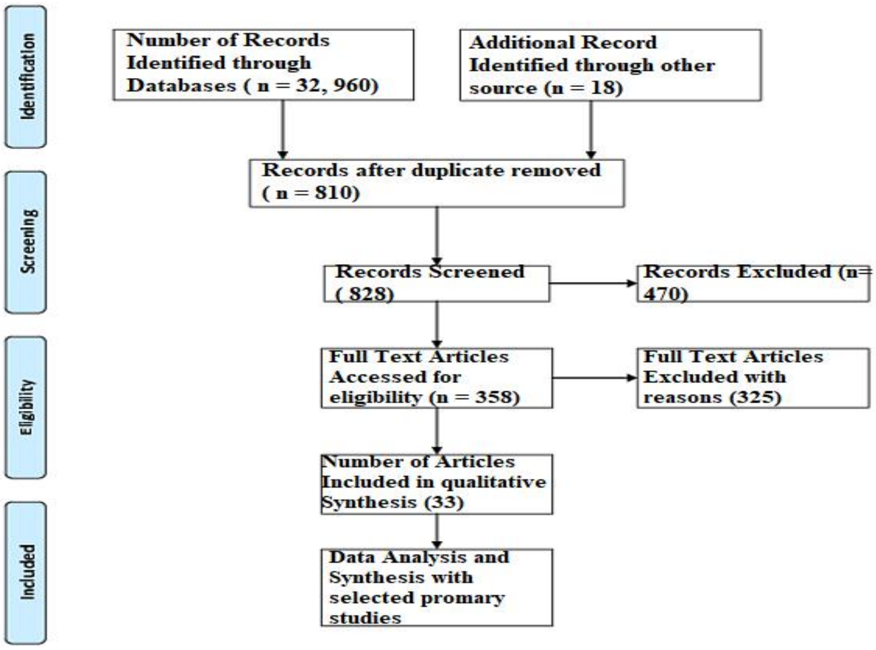


**Fig. 2.** Numerical Search Selection Flow Output Using the PRISMA Model

We launched the query search in July 2020 and applied both forward and backward snowballing iterations (see Section 2.6) on the selected 89 papers. In the snowballing step, 18 new peer-reviewed articles were identified, of which 33 passed the quality assessment selection from the total primary studies. The snowballing and quality assessment were iterated twice consecutively, generating a total of 368 (i.e. 350 + 18) studies. Iteration refers to the second iteration, the third iteration, and the fourth iteration that has not produced new results.

Another automated search was launched after the snowballing process, following the procedure presented in Section 2.3. As illustrated in Table 3, 33 primary studies were collected from all databases. This result confirmed that our snowballing systems were exhaustive; thus, we proceeded to the analysis and reporting stage.

*3.2 Quantitative Analysis*

Figure 3 presents the distribution of articles by the database. This allows us to find any article related to the topic we are researching.

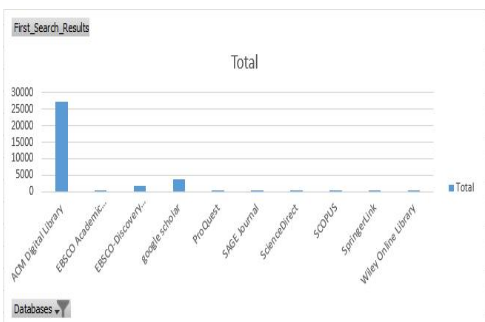


**Fig. 3.** Database-wise Articles Distribution

Figure 4 represents the number of articles per category.

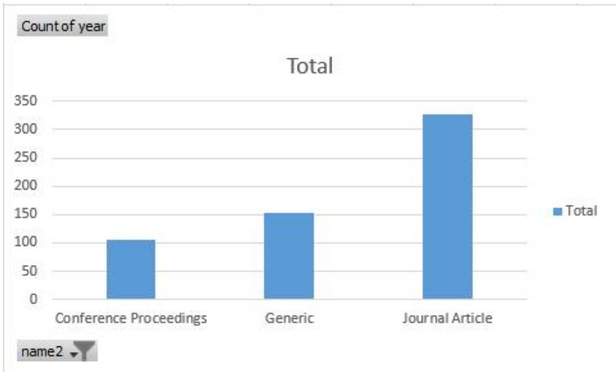


**Fig. 4.** Distribution Based on Categories

Figure 5 represents the number of articles per location.

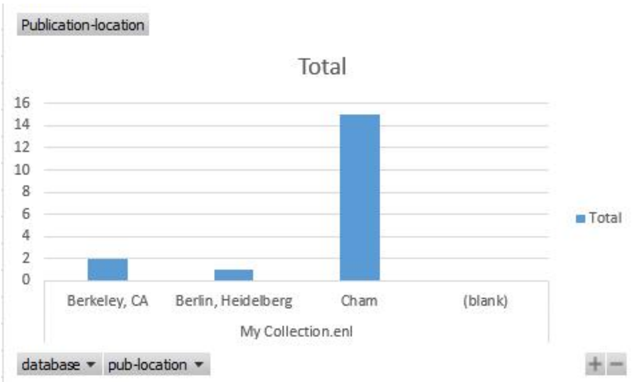


**Fig. 5.** Location-Wise Distribution of Articles

Figure 6 shows the number of articles published each year.

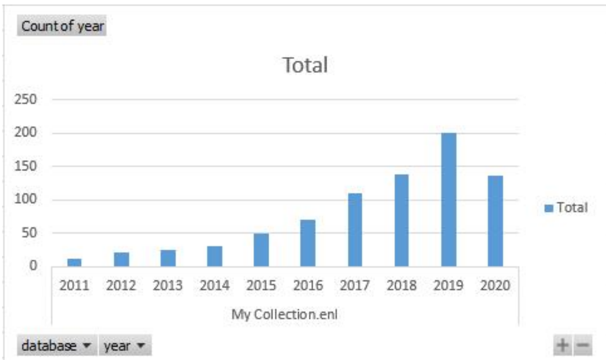


**Fig. 6.** Year-Wise Distribution of Articles

The following sections explain how research questions were answered.

### *3.2.1 DevOps guidelines/steps for adoption (RQ1)*

The adoption of DevOps can face obstacles since there is no straightforward method or framework for this process. Organisations might also want to adopt DevOps in different ways and approach their respective other purposes based on their current DevOps maturity level. This section will provide some general guidelines and suggestions related to the methods, strategies, and tools that aid the DevOps adoption process based on the selected primary studies.

*3.2.1.1 Continuous integration and continuous deployment pipelines*

Studies have highlighted the main objectives of DevOps adoption [12,19,20,28]. The goals are to achieve continuous delivery and successfully implement and practice continuous integrations in a deployment/delivery pipeline. The pipeline involves automated steps throughout the delivery process, from version control to testing the code to deploying it in production. Each action validates and pushes the code change when necessary and, if approved, sends the code to the next stage [49]. They halt the process if the code change fails to pass pipeline stages, and developers are notified to resolve the related issues.

One study by Ridzuan and Che'Rus [28] explained the continuous integration and continuous deployment/delivery stages to fully automate the delivery process, thus reducing the release cycle times, costs, and software development risks. However, the researchers defined the CI/CD pipeline as the last step of the supply chain in the software development process. Automated pipelines can be implemented in many ways using many different tools. Some suggestions presented by Toh, Sahibuddin and Mahrin [12] consist of the following steps: unit tests, platform tests, delivery to staging, application acceptance tests, deployment to production, and post-deployment tests, as presented in the pipeline stages. Two studies showed the same pipeline stages, including building, automated testing, and deployment [20,28]. Based on the previous reviews, the pipeline was a significant obstacle preventing implementation, which required a wide range of tools and tests. The entire process is monitored; we will discuss the monitoring in the next section [12,19,28].

*3.2.1.2 Monitoring*

Surveillance and monitoring are vital parts of DevOps and could apply to most aspects of the software development process. In DevOps, the monitoring process is always automated continuously [12,14,15,18,24,28,29]. Other researchers have claimed that monitoring is paramount as part of the continuous delivery process and is logged continuously aggregated during the coding phase [15]. Other researchers divided monitoring into two categories: infrastructure monitoring [14,30]. The infrastructure enhanced the planning and development processes to increase the value and business results. However, according to a preliminary study by Jabbari *et al.,* [14], monitoring automation is "the ability to monitor the applications and infrastructure without human intervention." The monitoring approach was intended to control the system's functionality and state and indicate to the developers if something in the system had abnormalities. Message alerts can be sent using a chat tool, such as Slack or Hip Chat [12,14,18]. A preliminary study by Amra and Safitri [18] presented a model for DevOps task categorisation and communication. The model, which monitors automation, is a system actor within the SDLC; workers monitor information captured from the source control system (SCS) and the state and functionality of all other systems within the SDLC. They are also notified if an issue occurs.

Similarly, according to another study by Zulfahmi Toh, Sahibuddin and Mahrin [12], dashboards are the primary service used for monitoring releases, users using the system, and users' initial country. Some others were also dashboards that enable special teams to monitor their part of the application. Finally, a preliminary study by Kolb and Rock [19] discussed monitoring to confirm whether a deprecated feature is still in use or whether it can be removed from the application.

*3.2.1.3 Tools*

The primary studies on DevOps tools present tools as an essential core factor in considering DevOps adoption [9,11,14,17,21,23,26,28,29]. DevOps tools are helpful throughout the development process and within the CI/CD pipeline. These primary studies mainly discussed tools used to control, communicate, monitor, test, and release. Collaboration tools are discussed in two primary studies [14,23]. Collaboration tools such as Slack and Hip Chat are typically used for DevOps communication by using the messaging approach to report alarms through notifications via monitoring tools. Preliminary studies mainly considered git-based tools, such as GitHub and Bitbucket, for version control. Git enables each developer to control their code in a local repository. The developer can then push the changes to a central repository or pull other developers' modifications [11,21,23,26].

The previous versions of the code are also available when needed if any changes have to be reverted. Monitoring tools, including New Relic, Graylog, and Kinesis, are also discussed by Al Smin *et al.,* [11]. These tools help monitor automation and report error signals to teams, as well as graphs and dashboards that represent the monitored data. Based on primary studies, testing can use Selenium or TestNG with the support of Jenkins's automation tool [11,26]. Selenium can also test for clicks, links, CSS, text, tag names, etc. When testing is completed, tools such as Ansible, Chef, Jenkins, Puppet, and Docker are used for automation, containerisation, delivery, and deployment. These are also some of the tools that enable continuous integration and the CI/CD pipeline [11,17,21,23,26].

*3.2.1.4 Adoption strategies*

Preliminary studies further showed that there are still many open questions about how organisations follow guidelines when adopting DevOps [1,4-6,10,11,23,28-30,32]. It is encouraging for DevOps practitioners to consider that DevOps adoption requires the support of top management. Furthermore, Mohammed Husien and Aljamali [32] explained that it does not happen in the beginning and that a "guerrilla" strategy can take place (i.e., acting outside the company's standard procedures). However, arguments for DevOps adoption can differ between engineers to managers. As such, convincing top management for a successful DevOps adoption and implementation. A very close integrations between the management, version control, automation tools, and deployment stages would simplify the process of establishing guidelines and promote compliant development strategies [29,30,32,33].

*3.2.2 DevOps challenges (RQ2)*

Adopting DevOps was clear regarding its benefits, as well as some challenges related to its adoption. Several articles provided an overview of the challenges of adopting DevOps [8,9,11-13,20,22,24,26-28]. These studies have discovered three main themes and challenges organisations face that want to adopt DevOps. The articles lacked expertise, skills, and knowledge of the individual implementing DevOps; strategies for overcoming obstacles to the resistance to change by the relevant stakeholders; and cost implications.

*3.2.2.1 Lack of standard guidelines and knowledge skills*

The literature indicates that DevOps lacks a precise method and guidelines or framework for its implementation. The results show that companies were hindered by uneducated personnel regarding DevOps, making it very difficult to adopt. However, the lack of clear guidelines, skills, and knowledge associated with DevOps is an extensive challenge discussed in many articles related to DevOps challenges [1-3,8-12,20,22-24,27]. These articles present an apparent problem with the lack of skills, knowledge, education, and clear guidelines on how to adopt DevOps. In research by Al Smin *et al.,* [11], the researchers discovered some required prerequisite skills and knowledge to develop, test, integrate, and deploy software to productions. In one of their cases, a company adopting DevOps had difficulties related to the technologies and platforms they used in their implementation. Even in other companies whose employees had a high technology and knowledge skillset, the DevOps practitioners had problems with the DevOps approach. In another setting, a primary study by Toh, Sahibuddin and Mahrin [12] revealed that recruiting the correct personnel with adequate skill sets is paramount. However, a lack of knowledge and required skill sets led to disastrous DevOps adoption and implementation:

*"Unqualified personnel having no required skills can slow the DevOps adoption journey because the competencies needed are missing at times of need."*

Four primary studies concluded that one of DevOps' main challenges is the lack of clear guidelines, understanding and expertise [20,22,24,27]. Finally, the researchers by Wohlin [27] established that DevOps adoption guidelines are lacking and can lead to delays in the software release cycle (Figure 1). Dlugi, Brunnert and Krcmar [10] provided similar guidelines for medical software that suggest the following:

i. fostering a community to change the culture to establish collaborative channels and merge traditional mindsets with agile (they are complementary, not in conflict)
ii. co-creating a standard DevOps IT platform while also supporting legal development
iii. improving, iterating, and gathering insights and key performance metrics
iv. growing processes over time, over-communicating, telling stories, sharing success, and showing data.

*3.2.2.2 Obstacles and resistance to change*

One of the challenges to adopting DevOps is the lack of valuable strategies, leading relevant stakeholders to resist change. Some DevOps-relevant stakeholders might resist DevOps adoption for fear of losing their jobs or their refusal to agree with the collaborative phenomenon. Different teams might have different orientation skills and perform other tasks that might negatively impact cooperation [12,20]. According to Nik Hassan *et al.,* [22], there are conflicting goals between development and operations teams. Developers tend to want new features and bugs fixed to be released rapidly. Simultaneously, the operations team wants to keep the releases to a minimum to ensure smooth operational activities to preserve the system's stability and reliability. From another perspective, senior management is resisting DevOps adoption. The literature provides some views on the problems encountered due to senior management. For example, according to "SIJMLV1_N1_PP1_10" [29], while the benefits of adopting DevOps are unclear, top management may resist DevOps adoption by questioning DevOps implementation's capability and wisdom.

Tools for DevOps automation is relatively expensive and need proper management, which may deter managers from understanding DevOps' value. DevOps adoption is sometimes affected at a particular hierarchical level and by an inflexible management style [13,22,24]. The researchers by Ali, Petersen and Tanveer [24] discussed the lack of productivity at the beginning of the adoption process, which may send a wrong signal as a problem for senior management. However, senior management can delay adopting DevOps due to such productivity issues.

*3.2.2.3 Methodological shortfalls & the cost of tools and automation*

The primary studies uncovered several challenges: the lack of standard methodological approach and tools for any particular organisation, cost implications, and insufficient knowledge of the automation process. Preliminary studies presented a lack of tools as a frequent challenge for adopting DevOps [1,5-7,9,12,13,24,26-28]. Due to a lack of standard DevOps practices, relevant stakeholders have difficulties finding the correct tools and methods for their organisations and understanding how to use them reliably for their continuous implementation [9,12,27]. The initial arrangement of the tools, experimentation, and deciding which tool to use are considered negative time factors [12]. DevOps tooling's relatively high cost is also considered a negative factor, and management may have difficulties justifying their investment in the DevOps journey. One preliminary study by Wohlin [27] pointed out that a lack of standardisation between process and tools hinders DevOps adoption. Therefore, devices can be a liability if handled poorly. The preliminary study by Ridzuan Idris and Che' Rus [28] stated:

*"The evidence of this study shows that the asset can become a liability if the control of the resources is managed wrongly. Respondents mentioned that the failure of resources control could lead to resource overhead during the integration of the source codes when all source codes have been deployed too often. As a result, this will jeopardise the Continuous Delivery Pipeline. In some cases, the automated test failed because the production environment is different and very complex to be executed with the automated test".*

Considering the statement above, the continuous deployment of features can be too complicated, leading them to fail continuous delivery pipeline tests.

*3.2.3 DevOps adoption advantages (RQ3)*

There are many benefits of adopting DevOps, according to the literature analysis. Similarly, in the literature, we identified eight (8) articles that mentioned these benefits [1-3,11-13,21,27]. These articles provide similar conclusions about DevOps' benefits. However, organisations should know DevOps benefits for those who want to adopt the DevOps journey.

*3.2.3.1 Frequent release cycle*

The frequent release cycle describes a method that begins with completing code and delivers the code to production. Concerning the DevOps phenomenon, the release time is often short, ranging from a few hours to a few weeks, possibly involving several daily releases. The six articles above agree that faster release cycles benefit from adopting DevOps [8,20].

A case study by Al Smin *et al.,* [11] discussed a shorter release cycle time. It was stated that *"An improved speed in the delivery of software changes was the most commonly perceived benefit of DevOps"*. This was the perspective presented by four out of five case companies in their study. Release times were reduced to a few days from several months. Other articles [12,13,27] also mentioned that DevOps reduces the release cycles and suggested that release cycles are an essential benefit of DevOps adoption. Similarly, the survey outcome results of (as presented in Figure 1 above) indicate the following [10,28]:

*"The majority of the respondents highlighted the advantage of the DevOps activities that they reduce the software cycle time according to what they believe DevOps practice can achieve."*

This assertion was also consistent with the other literature highlighting DevOps' benefits; one can confirm that one of DevOps' key advantages is a "shorter and faster release cycle."

*3.2.3.2 High throughput (output)*

Throughput explains several ways to measure the effectiveness of a process or system in converting inputs into outputs. High productivity is determined by dividing the result by the costs incurred or resources spent [50]. With software development, productivity is measured by backend code. The number of code lines can compare the number of regulations to another factor, such as the time required to implement the lines of principles, measured as lines of code per unit time. Some DevOps practices increase productivity, such as automation, collaboration, communication, and improved strategies implemented by all concerned teams. The primary study provided some support for improved productivity when adopting DevOps. Prior studies indicated that better strategies in breaking the silos of communication and less bureaucracy improved the productivity of development and operations teams [11,12,28]. However, the development and operations team must improve their communication to enhance collaboration, which would, in turn, increase productivity. When adopting DevOps, large portions of (if not all) bottlenecks are eliminated, significantly growing different teams' productivity. Morsy *et al.,* [13] claimed that continuous integration and feedback increase the productivity of development teams. Productivity is also enhanced by sharing knowledge across various groups to manage the shared understanding and track the knowledge needs throughout the development life cycle [9].

*3.2.3.3 Quality*

This research considered the quality of teams and the quality of codes and applications. The literature on DevOps' benefits unanimously indicates that code quality and application quality are essential benefits of DevOps [11-13,20,28]. Three primary studies conducted case studies and analysed companies that adopted DevOps [11,12,28]. They concluded that quality increases whenever DevOps practices are implemented. They also ascertained that they tend to produce higher quality codes, increasing assigning developers' responsibility for their respective codes.

Similarly, developers are more confident that their code will pass the tests and deploy into the production environment with more minor incremented releases. Two primary studies further discuss the QA of DevOps and conclude that DevOps can drive QA by enhancing communication and feedback loops, as presented in Figure 2 [13,20]. A study by Nik Hassan *et al.,* [21] reported that most companies feel that DevOps improves production quality and decreases the risks. The automation of processes, such as continuous testing and continuous deployment, substantially impacted the code's quality. Shorter and faster release cycles enabled deployment in smaller increments, increasing production quality since the risks and quality might be easy to control.

*3.3 Threats to Validity*

Validity and reliability reveal threats to internal-external, dependability, transferability, and empirical research study validity.

Internal validity has to do with the confidence of the analysed and reported results, and the study considered framework definition and adoption. We have benefited from the description of the classification framework and its usage for different research classifications. It is a threat to these studies. However, we make the classification data available to allow the research community to evaluate our results' validity.

Another threat to internal validity is framework inclusiveness. The reviewed classification framework might not include all areas and topics characterising DevOps guidelines research. If the risk is eliminated, the framework will be derived sequentially and incrementally. As we have previously explained, starting from the first draft framework obtained from reading only the title, abstract, and keywords, new subtopics have been added within each area after reading the whole paper.

External validity threats are derived from potential issues that prevent the absolute generalisation of the results. Only a subset of papers concerning DevOps guidelines was targeted in our study (i.e., we only considered papers published within a period of nine (9) years, from 2011 to July 2020. However, the results may not be well represented or represent the overall research in the field. We acknowledged that the risk of this is low because DevOps guidelines are a new research topic; as such, very soon, other investigations will likely include the most relevant research fields. One relevant threat was an automated search performed in July 2020. The area of DevOps is growing fast, and future studies would likely find more recent papers not included in the searched or used databases. This threat has been resolved by forwarding the snowballing approach, which gave us the chance to include many additional papers citing the primary studies with publication years of 2011–2020.

Finally, construct validity comprises threats related to the correspondence of measures managed to the related properties. Based on our approach, the following threats were highlighted.

We identified the primary studies. We initially defined a search string to identify prior studies in our approach. Some or most of the time, different search strings might have produced different results. However, this is a general threat to all systematic review approaches. We tried to overcome this issue first by redefining a very available search string to be as widespread and comprehensive as possible. The next threat to our primary study's identification was the consideration of digital databases. At the beginning of our search, we used the seven most popular databases for software engineering. Still, we thought searching on different sources might have yielded different results. This risk was considerably eliminated using various loop iterations of backward and forward snowballing as a search procedure to complement our search of originally selected databases. Lastly, we performed some verification tasks by launching another second automated search on three (3) more databases, and the results confirmed the validity of the search results previously conducted. We have seen it as very unlikely to miss all the relevant articles.

Selection of primary studies: The exclusion or inclusion of an article by reading the title, abstract, and keywords according to the inclusion and exclusion criteria was initially specified. Then, the whole paper was read based on a two-step quality assessment approach. There is a possibility that some reports were missed due to the defined inclusion/exclusion criteria or to the illustrated quality assessment checklist of the above selection procedure. Meanwhile, in representing such a selection procedure, we had strictly followed the guidelines for systematic reviews in software engineering by Pérez *et al.,* [51] and the selection procedures followed in similar studies [27].

### *3.4 Summary of the Findings*

The review identified DevOps benefits, challenges, and common DevOps guidelines for adoption. Our discussion of the findings highlights answers to the research questions in line with the six areas of the classification framework model proposed earlier. The main results are summarised in Table 6. The research started by identifying some general challenges to all aspects of DevOps adoption, which have a broad impact and, thus, return across the loop flow areas. The common challenges include the following:

*3.4.1 Lack of a standard methodology for DevOps adoption*

There was no transparent scientific approach to determining the DevOps capabilities. This challenge was described in studies concerning methodological evaluation and portability across different domains, such as medical, finance, embedded system domains, and other highly regulated environments and policy variations.

*3.4.2 Technology evolution constraints*

There are constraints concerning the continuous and rapid evolution of DevOps tools' technology. Transversally impacts all areas and the loop flow (refer to Figure 1). Practising activities will continuously face novel challenges [52] . It is always difficult to adapt to new technology constraints and conditions [53], especially considering the ongoing deployment problems in some working environments. Applying the DevOps methodology is not straightforward since it is highly regulated and often involves legal issues.

*3.4.3 Cost of tools*

The cost of tools ranges from the large dimensions and high complexity of DevOps systems to their different configurations. The high prices of these tools heavily impact their automation, continuous integration, continuous delivery, and constant testing.

*3.4.4 Policy restrictions (different teams' locations)*

Concerning different policy capabilities and cross-platform DevOps tools, the automation for provisioning and de-provisioning resources and services is another challenge that directly affects the continuous deployment policy perspective. It requires specific approach guidelines for different organisations.

*3.4.5 Security issues*

Security issues are of crucial concern in other domains and DevOps. Such issues arise in DevOps and entail even more difficulties than the traditional software development approach. The response to security issues was instantiated as “DevSecOps.”

**Table 6**
Summary of the main findings

| | | | |
|---|---|---|---|
| DevOps Challenges | The primary studies indicated that a lack of standardisation between process and tools significantly hinders DevOps adoption. There is no precise method, framework, or guidelines for its adoption. | Looking clearly at primary studies related to the DevOps computing discipline, there was no transparent scientific approach to determining DevOps' capabilities regarding process, practice, tools, and implementation. | Another frequent challenge discovered from the primary studies was the lack of standard methodological approach and tools for any particular organisation, cost implications, and insufficient knowledge of the automation process. |
| DevOps Adoption Benefits | Primary studies show that improving speed in the delivery of software changes was the most commonly perceived benefit of DevOps. | The primary studies indicated that the software's quality increases whenever DevOps best practices are implemented and adopted successfully. | The literature on the benefits of DevOps unanimously showed that code quality and application quality are essential benefits of DevOps. |
| Classification Adoption Guide | Documentation of projects and planning<br>Prepared expertise in the organisation<br>Ability to automate the process<br>Building an effective DevOps culture across all teams<br>Implementation of a code management strategy<br>Ability to implement the principles (continuous integration/development/deployment/delivery) | | |

## 4. Conclusions

The primary studies reviewed in the present work provide adequate information about the benefits and challenges of DevOps implementation. The guidelines to be followed were precisely identified based on the different modes of a report from the primary studies. The review identified DevOps benefits, challenges, and common adoption guidelines. The primary studies considered in this review show that DevOps adoption is currently being rolled out in software development worldwide. We examined the benefits, challenges, and guidelines of DevOps adoption. Throughout this process, we explored various publications on DevOps' benefits, challenges, and guidelines using multiple databases, including ACM Digital Library, Google Scholar, and ScienceDirect. The present review also revealed that the automation, sharing, and collaboration of the DevOps process are closely related to benefits and successes in software development based on common guidelines. We also found a direct link between DevOps implementation without guidelines and various challenges, such as insufficient knowledge of the automation process. There was no transparent scientific approach to determining DevOps' capabilities regarding processes, practices, tools, and implementation. The current review also revealed that DevOps is the fastest available response software development process. It is also essential that DevOps is accompanied by common guidelines. However, DevOps contributes to ensuring continuous software delivery and shortening time to market. Our research shows that research has focused primarily on process automation, continuous delivery, and integration. Further research is needed in many DevOps areas (e.g. measurement, the development of performance metrics of different stages, practice, and culture) to ensure the appropriateness of software reliability and quality factors such as efficiency, usability, and portability. Finally, other survey research questions categorised by different roles can be posed and answered by DevOps practitioners.

**Acknowledgement**
The authors would like to acknowledge the support from the School of Computer Sciences, USM for the publication of this paper.

**References**

[1] Dörnenburg, Erik. "The path to devops." *IEEE software* 35, no. 5 (2018): 71-75. https://doi.org/10.1109/MS.2018.290110337
[2] Hisham, Sakinah Muhamad, Norazlianie Sazali, Kumaran Kadirgama, and Devarajan Ramasamy. "A Comprehensive Review on Tribological Behaviour of Hybrid Nanocellulose-CuO as Nanolubricant for Piston Ring Cylinder Liner Application." *Journal of Advanced Research in Micro and Nano Engineering* 18, no. 1 (2024): 60-77. https://doi.org/10.37934/armne.18.1.6077
[3] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer. "What is DevOps? A systematic mapping study on definitions and practices." In *Proceedings of the scientific workshop proceedings of XP2016*, pp. 1-11. 2016. https://doi.org/10.1145/2962695.2962707
[4] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer. "What is DevOps? A systematic mapping study on definitions and practices." In *Proceedings of the scientific workshop proceedings of XP2016*, pp. 1-11. 2016. https://doi.org/10.1145/2962695.2962707
[5] Daneva, Maya, and Robin Bolscher. "What we know about software architecture styles in continuous delivery and DevOps?." In *Software Technologies: 14th International Conference, ICSOFT 2019, Prague, Czech Republic, July 26–28, 2019, Revised Selected Papers 14*, pp. 26-39. Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-52991-8_2
[6] Österberg, Gustaf. "A Systematic Literature Review on DevOps and its Definitions, Adoptions, Benefits and Challenges." (2020).
[7] Maroukian, Krikor, and Stephen R. Gulliver. "Leading DevOps practice and principle adoption." *arXiv preprint arXiv:2008.10515* (2020). https://doi.org/10.5121/csit.2020.100504
[8] Omar, Habsah, Nor Suhaila Abdul Ghani, Mohamad Sukeri Ismail, and Mohd Hazwan Hassim. "Readiness to Continue Online Learning Post Covid-19 Among Students of Politeknik Jeli Kelantan." *International Journal of Advanced Research in Future Ready Learning and Education* 35, no. 1 (2024): 42-51. https://doi.org/10.37934/frle.35.1.4251
[9] Artač, Matej, Tadej Borovšak, Elisabetta Di Nitto, Michele Guerriero, and Damian A. Tamburri. "Model-driven continuous deployment for quality DevOps." In *Proceedings of the 2nd international workshop on quality-aware DevOps*, pp. 40-41. 2016. https://doi.org/10.1145/2945408.2945417
[10] Dlugi, Markus, Andreas Brunnert, and Helmut Krcmar. "Model-based performance evaluations in continuous delivery pipelines." In *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, pp. 25-26. 2015. https://doi.org/10.1145/2804371.2804376
[11] Al Smin, Ahmed Moh A., Alkbir Munir Faraj Almabrouk, Sairul Izwan Safie, Mohd Al Fatihhi Mohd Szali Januddi, Mohd Fahmi Hussin, and Abdulgader Alsharif. "Enhancing solar hybrid system efficiency in Libya through PSO & flower pollination optimization." *Progress in Energy and Environment* (2024): 23-31. https://doi.org/10.37934/progee.27.1.2331
[12] Toh, M. Zulfahmi, Shamsul Sahibuddin, and Mohd Naz'ri 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*, pp. 173-177. 2019. https://doi.org/10.1145/3316615.3316619
[13] Fahmy, M., M. Morsy, H. Abd Elshakour, and A. M. Belal. "Effect of thermal insulation on building thermal comfort and energy consumption in Egypt." *Journal of Advanced Research in Applied Mechanics* 43, no. 1 (2018): 8-19.
[14] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer. "Towards a benefits dependency network for DevOps based on a systematic literature review." *Journal of Software: Evolution and Process* 30, no. 11 (2018): e1957. https://doi.org/10.1002/smr.1957
[15] Erich, Floris, Chintan Amrit, and Maya 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*, pp. 1-1. 2014. https://doi.org/10.1145/2652524.2652598

[16] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer. "What is DevOps? A systematic mapping study on definitions and practices." In *Proceedings of the scientific workshop proceedings of XP2016*, pp. 1-11. 2016. https://doi.org/10.1145/2962695.2962707
[17] Airaj, Mohammed. "Enable cloud DevOps approach for industry and higher education." *Concurrency and Computation: Practice and Experience* 29, no. 5 (2017): e3937. https://doi.org/10.1002/cpe.3937
[18] Amra, Siti, and Nelly Safitri. "Design and Construction of Steam Power Plant Prototype (Performance Analysis of Boiler System and Compress Cylinder)." *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences* 116, no. 2 (2024): 192-199. https://doi.org/10.37934/arfmts.116.2.192199
[19] Kolb, Stefan, and Cedric Röck. "Unified cloud application management." In *2016 IEEE World Congress on Services (SERVICES)*, pp. 1-8. IEEE, 2016. https://doi.org/10.1109/SERVICES.2016.7
[20] Lwakatare, Lucy Ellen, Terhi Kilamo, Teemu Karvonen, Tanja Sauvola, Ville Heikkilä, Juha Itkonen, Pasi Kuvaja, Tommi Mikkonen, Markku Oivo, and Casper Lassenius. "DevOps in practice." (2019).
[21] Díaz, Jessica, Daniel López-Fernández, Jorge Pérez, and Ángel González-Prieto. "Why are many businesses instilling a DevOps culture into their organization?." *Empirical Software Engineering* 26 (2021): 1-50. https://doi.org/10.1007/s10664-020-09919-3
[22] Kamarudin, Nurzatulshima, Nik Mawar Hanifah Nik Hassan, Mohd Mokhtar Muhamad, Othman Talib, Haryati Kamarudin, Norhafizan Abdul Wahab, Aidatul Shima Ismail, Haza Hafeez Borhan, and Nazihah Idris. "Unveiling Collaborative Trends in Fuzzy Delphi Method (FDM) Research: A Co-Authorship Bibliometrics Study." *International Journal of Computational Thinking and Data Science* 2, no. 1 (2024): 1-20. https://doi.org/10.37934/CTDS.2.1.120
[23] Desai, V., and H. A. Dinesh. "Efficient reputation-based cyber attack detection mechanism for Big Data environment." *Indian Journal of Science and Technology* 15, no. 13 (2022): 592-602. https://doi.org/10.17485/IJST/v15i13.2102
[24] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer. "Towards a benefits dependency network for DevOps based on a systematic literature review." *Journal of Software: Evolution and Process* 30, no. 11 (2018): e1957. https://doi.org/10.1002/smr.1957
[25] Brereton, Pearl, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. "Lessons from applying the systematic literature review process within the software engineering domain." *Journal of systems and software* 80, no. 4 (2007): 571-583. https://doi.org/10.1016/j.jss.2006.07.009
[26] Shameli, Kamyar, Zahra Izadiyan, Pooneh Kia, Hemra Hamrayev, Hossein Jahangirian, Aras Kartouzian, Emad Abbas Jaffar Al-Mulla, and Hassan Moeini. "Exploring the Potential of Gold Nanoparticles in Nanomedicine: A Comprehensive Analysis of Benefits and Limitations." *Journal of Research in Nanoscience and Nanotechnology* 11, no. 1 (2024): 1-15. https://doi.org/10.37934/jrnn.11.1.115
[27] Wohlin, Claes. "Guidelines for snowballing in systematic literature studies and a replication in software engineering." In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pp. 1-10. 2014. https://doi.org/10.1145/2601248.2601268
[28] Idris, Muhammad Ridzuan, and Ridzwan Che'Rus. "Validity and Reliability of a Rekatronik Module for Electronic Design in the Design and Technology (D&T) Subject." *Journal of Advanced Research in Computing and Applications* 34, no. 1 (2024): 28-36. https://doi.org/10.37934/arca.34.1.2836
[29] "SIJMLV1_N1_PP1_10".
[30] Kabir, Afrida, Faiyaj Kabir, Saief Newaz Chowdhury, and AR M. Harunur Rashid. "Design, Simulation and Fabrication of an Ergonomic Handgrip for Public Transport in Bangladesh." *Malaysian Journal on Composites Science and Manufacturing* 13, no. 1 (2024): 98-111. https://doi.org/10.37934/mjcsm.13.1.98111
[31] Abdullah, Nashimah, Wan Nur Zafirah Wan Razak, Nur Aliya Ezzaty Azali, Khairun Nasrin Azman, Khairunnisa Mohd Kharfizi, Siti Nur Syakinah Jansi, Nurru Anida Ibrahim, Salisa Abdul Rahman, and Siti Norbakyah Jabar. "Electric Vehicle Adoption: A Comparative Analysis in Malaysia and ASEAN Countries." *Semarak International Journal of Electronic System Engineering* 1, no. 1 (2024): 60-68. https://doi.org/10.37934/sijese.1.1.6068
[32] Husien, Nemah Sahib Mohammed, and Nagham Mahmood Aljamali. "Designation of Nano-Analytical Reagents and Evaluation of Nano-Applications." *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences* 116, no. 2 (2024): 27-40. https://doi.org/10.37934/arfmts.116.2.2740
[33] Rezaei, Ghasem, Hamed Gholami, Mohammad Reza Tootoonchi, and Mohammad Hossein Dehbozorgi. "Improving the Performance of Gas Extraction by Reducing the Shutdown Time Using an RCA-based Approach–A Case Study." *Journal of Advanced Research in Technology and Innovation Management* 7, no. 1 (2023): 1-14.
[34] Meixner, Kristof, Dietmar Winkler, and Stefan Biffl. "Towards combined process & tool variability management in software testing." In *Proceedings of the 13th International Workshop on Variability Modelling of Software-Intensive Systems*, pp. 1-6. 2019. https://doi.org/10.1145/3302333.3302339
[35] Gupta, Rajeev Kumar, Mekanathan Venkatachalapathy, and Ferose Khan 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)*, pp. 30-34. IEEE, 2019. https://doi.org/10.1109/ICGSE.2019.00020
[36] Toh, M. Zulfahmi, Shamsul Sahibuddin, and Mohd Naz'ri 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*, pp. 173-177. 2019. https://doi.org/10.1145/3316615.3316619
[37] Morales, Jose Andre, Hasan Yasar, and Aaron Volkman. "Implementing DevOps practices in highly regulated environments." In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pp. 1-9. 2018. https://doi.org/10.1145/3234152.3234188
[38] Adietya, Berlian Arswendo, Husein Syahab, Mahendra Indiaryanto, Wasis Dwi Aryawan, and I. Ketut Aria Pria Utama. "Influence Evaluation of Open Propellers with Boss Cap Fins: Case Studies on Types B4-70 and Ka4-70." *CFD Letters* 16, no. 9 (2024): 143-177. https://doi.org/10.37934/cfdl.16.9.143177
[39] Fazal-Baqaie, Masud, Baris Güldali, and Simon Oberthür. "Towards DevOps in Multi-provider Projects." In *CSE@ SE*, pp. 18-21. 2017.
[40] Vallon, Raoul, Bernardo José da Silva Estácio, Rafael Prikladnicki, and Thomas Grechenig. "Systematic literature review on agile practices in global software development." *Information and Software Technology* 96 (2018): 161-180. https://doi.org/10.1016/j.infsof.2017.12.004
[41] Lwakatare, Lucy Ellen, Terhi Kilamo, Teemu Karvonen, Tanja Sauvola, Ville Heikkilä, Juha Itkonen, Pasi Kuvaja, Tommi Mikkonen, Markku Oivo, and Casper Lassenius. "DevOps in practice." (2019).
[42] Nogueira, Ana Filipa, José CB Ribeiro, Mário A. Zenha-Rela, and Antoine Craske. "Improving la redoute's ci/cd pipeline and devops processes by applying machine learning techniques." In *2018 11th international conference on the quality of information and communications technology (QUATIC)*, pp. 282-286. IEEE, 2018. https://doi.org/10.1109/QUATIC.2018.00050
[43] Mandepudi, Snehitha. "Communication Challenges in DevOps & Mitigation Strategies." (2019).
[44] Lie, Martin Forsberg, Mary Sánchez-Gordón, and Ricardo Colomo-Palacios. "Devops in an iso 13485 regulated environment: a multivocal literature review." In *Proceedings of the 14th ACM/IEEE International Symposium on empirical software engineering and measurement (ESEM)*, pp. 1-11. 2020. https://doi.org/10.1145/3382494.3410679
[45] Senapathi, Mali, Jim Buchan, and Hady Osman. "DevOps capabilities, practices, and challenges: Insights from a case study." In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pp. 57-67. 2018. https://doi.org/10.1145/3210459.3210465
[46] Laukkarinen, Teemu, Kati Kuusinen, and Tommi Mikkonen. "Regulated software meets DevOps." *Information and Software Technology* 97 (2018): 176-178. https://doi.org/10.1016/j.infsof.2018.01.011
[47] Laukkarinen, Teemu, Kati Kuusinen, and Tommi 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)*, pp. 15-18. IEEE, 2017. https://doi.org/10.1109/ICSE-NIER.2017.20
[48] Lwakatare, Lucy Ellen, Teemu Karvonen, Tanja Sauvola, Pasi Kuvaja, Helena Holmström Olsson, Jan Bosch, and Markku Oivo. "Towards DevOps in the embedded systems domain: Why is it so hard?." In *2016 49th hawaii international conference on system sciences (hicss)*, pp. 5437-5446. IEEE, 2016. https://doi.org/10.1109/HICSS.2016.671
[49] Kamarudin, Nurzatulshima, Nik Mawar Hanifah Nik Hassan, Mohd Mokhtar Muhamad, Othman Talib, Haryati Kamarudin, Norhafizan Abdul Wahab, Aidatul Shima Ismail, Haza Hafeez Borhan, and Nazihah Idris. "Unveiling Collaborative Trends in Fuzzy Delphi Method (FDM) Research: A Co-Authorship Bibliometrics Study." *International Journal of Computational Thinking and Data Science* 2, no. 1 (2024): 1-20. https://doi.org/10.37934/CTDS.2.1.120
[50] Agoeng, Chandra, Nurul Dini Faqriah Miza Azmi, Hakimah Mat Harun, Nurzulaikha Abdullah, Wan Azani Mustafa, and Fakhitah Ridzuan. "Leveraging Correlation and Clustering: An Exploration of Data Scientist Salaries." *Journal of Advanced Research in Computing and Applications* 35, no. 1 (2024): 10-20. https://doi.org/10.37934/arca.35.1.1020
[51] Pérez, Jorge, Jessica Díaz, Javier Garcia-Martin, and Bernardo Tabuenca. "Systematic literature reviews in software engineering—Enhancement of the study selection process using Cohen's kappa statistic." *Journal of Systems and Software* 168 (2020): 110657. https://doi.org/10.1016/j.jss.2020.110657
[52] Morshidi, Azizan, Noor Syakirah Zakaria, Mohammad Ikhram Mohammad Ridzuan, Rizal Zamani Idris, Azueryn Annatassia Dania Aqeela, and Mohamad Shaukhi Mohd Radzi. "Artificial Intelligence and Islam: A Bibiliometric-Thematic Analysis and Future Research Direction." *Semarak International Journal of Machine Learning* 1, no. 1 (2024): 41-58. https://doi.org/10.37934/sijml.1.1.4158
[53] Abdullah, Nashimah, Wan Nur Zafirah Wan Razak, Nur Aliya Ezzaty Azali, Khairun Nasrin Azman, Khairunnisa Mohd Kharfizi, Siti Nur Syakinah Jansi, Nurru Anida Ibrahim, Salisa Abdul Rahman, and Siti Norbakyah Jabar.

"Electric Vehicle Adoption: A Comparative Analysis in Malaysia and ASEAN Countries." *Semarak International Journal of Electronic System Engineering* 1, no. 1 (2024): 60-68. https://doi.org/10.37934/sijese.1.1.6068



| Name of Author | Email |
|---|---|
| Usman Hamza | usmanhamza@student.usm.my |
| Nasuha Lee Abdullah | nasuha@usm.my |
| Sharifah Mashita Syed-Mohamad | s.mashita@umt.edu.my |