

EXPLORING THE BENEFITS, CHALLENGES AND GUIDELINES OF DEVOPS ADOPTION: A SYSTEMATIC LITERATURE REVIEW AND AN EMPIRICAL STUDY

USMAN HAMZA¹, SHARIFAH MASHITA SYED-MOHAMAD^{2*} AND NASUHA LEE ABDULLAH¹

¹*School of Computer Sciences, Universiti Sains Malaysia, 11800 Gelugor, Penang, Malaysia.* ²*Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia.*

*Corresponding author: s.mashita@umt.edu.my

ARTICLE INFO

Article History:

Received 26 May 2024

Accepted 9 September 2024

Published 20 October 2024

Keywords:

DevOps.

systematic literature review.

interview.

DevOps adoption.

DevOps benefits and

challenges.

ABSTRACT

This study aims to explore the benefits and challenges of DevOps adoption in the rapidly evolving landscape of DevOps and Information Technology (IT) firms. DevOps is a software development approach that emphasises communication and collaboration between software developers and IT operations teams, aiming to streamline processes and enhance software delivery. Despite the growing popularity of DevOps, there are several challenges to its adoption, including stakeholder confusion, a lack of clear processes and guidelines, and a lack of empirical studies that discuss the challenges of DevOps. To establish a foundation of understanding and provide insights into the benefits, challenges, and guidelines for DevOps adoption, this study uses a two-fold approach, including a systematic literature review and conducts semi-structured interviews involving six organisations of various sizes. The interviews aimed to obtain evidence of DevOps adoption in practice and to detail real scenarios and explain the role of each category during DevOps adoption. The study provides insights into the challenges faced by software organisations in adopting the DevOps culture and the benefits of DevOps adoption. The study also proposes DevOps adoption guidelines based on the findings. The findings contribute to the existing literature on DevOps adoption and provide valuable recommendations for software organisations.

2020 Mathematics Subject Classification:

2020 ACM Computing Classification Codes:

© UMT Press

Introduction

In the rapidly evolving landscape of DevOps and Information Technology (IT) firms, several emerging trends are poised to shape how organisations develop and deliver software solutions. DevOps, a software development approach that emphasises communication and collaboration between software developers and IT operations teams, aims to streamline processes and enhance software delivery [1]. It is crucial for frequently and reliably updating a system in an operational state, allowing organisations to release software faster and more frequently in response to market demands.

Despite the growing popularity of DevOps, its adoption faces several challenges [2,3,4,5,6]. Stakeholder confusion over where development ends and operations begin is a common hurdle, compounded by a lack of clear processes and guidelines [1]. Many organisations remain unaware of the true benefits and challenges of DevOps, with stakeholders often not collaborating or communicating effectively during software development [7]. Additionally, there is a notable lack of empirical studies discussing DevOps challenges, particularly those related to processes and guidelines for successful adoption [1].

This study aims to gain a deeper understanding of the benefits and challenges faced by companies during the adoption of DevOps, and to identify specific guidelines for its successful implementation. The study will explore the benefits and challenges faced by stakeholders and propose guidelines for DevOps adoption and implementation. These guidelines will help organisations implement DevOps smoothly, serving as a roadmap for those transitioning to DevOps or seeking to improve their software development processes. This study makes use of a two-fold approach to achieve the research objectives. The rest of the paper is organised as follows: Section 2 describes the background, section 3 explains the research methodology, section 4 defines and discusses the findings including limitations of the study, and section 5 concludes the article as well as provides directions for the future.

Background of the Study

Agile software development practices has revolutionised software projects with its iterative and collaborative approach, emphasising flexibility, collaboration, and customer satisfaction. Agile methodologies prioritise delivering working software in small iterations rather than providing a complete solution simultaneously, anticipating the need for flexibility and being pragmatic about the delivery of the finished product [7,8]. By embracing and adapting to change while delivering quality products, being agile enables the faster delivery of working software. Despite the use of Agile software development practices, empirical studies have identified the challenges of implementing Agile software development practices and the quality factors affected by it. Effective communication, continuous improvement, and proper documentation are emphasized in the article as necessary to overcome these challenges [7].

In contrast, DevOps practices such as continuous integration and deployment (CI/CD) are seen as a solution to the challenges of agile software development practices through the concept of Continuous Integration. DevOps emphasises collaboration, communication, and automation between the development and operations teams to enable faster and more reliable software delivery. Continuous integration and deployment are DevOps practices that enable faster and more reliable software delivery [1,9]. However, there is a lack of empirical studies that discuss the challenges of DevOps, particularly related to the process and guidelines for successful adoption [1].

DevOps has been defined in various ways across the literature, resulting in some ambiguity

in the field [2,10,11,12]. To clarify, we have categorised these definitions into collaborative, philosophical, and methodological perspectives. From a collaborative viewpoint, DevOps is described as an integrated approach that merges the principles and practices of both “development” and “operations” [2,13,10,11,12].

Furthermore, DevOps as a philosophical perspective emphasizes a cultural shift towards shared responsibility and continuous improvement. This involves instilling a mindset where teams are no longer siloed but work together throughout the software lifecycle [3]. Methodologically, DevOps focuses on automating processes, enhancing system reliability, and ensuring continuous delivery of value to end-users. This comprehensive approach addresses the need for speed and stability in software development and operations [4]. These various definitions of DevOps implies that there is still no common unified conceptualisation and that what prevails in the literature are different specific perspectives.

The DevOps movement aims to bridge the knowledge gap between project team entities by enforcing stringent processes to ensure real-time communications [10]. However, the approach has not yet matured to the point where many organisations have converged on how to begin their DevOps journey [11]. DevOps encourages collaboration between software development and operations teams, promoting a collaborative and unified method of delivering software. However, due to the rapid growth and high competitiveness of the IT industry, software developers are under pressure to finish and release new code as soon as possible, while operators strive to always keep production systems up-to-date and stable [4,12].

Methodology

In this section, the research objectives and questions are outlined. The study then explores two primary components: a systematic literature review and an empirical investigation. The systematic literature review aims to identify and analyse all relevant empirical evidence that addresses the research questions. This comprehensive approach allows the drawing of insights from a diverse array of studies and establishes the research. Additionally, semi-structured interviews with representatives from six organisations of various sizes was conducted to obtain practical insights and real-world perspectives on DevOps adoption.

Systematic Literature Review

This study follows the established guidelines for systematic reviews in software engineering research [13,14]. A systematic literature review (SLR) was used as an essential tool to identify pertinent research related to our specific research questions. In this study, the SLR was employed to collect and extract relevant information on the benefits, challenges, and practices of DevOps. The SLR process was segmented into three phases: planning the review, conducting the review, and presenting the review. The first two phases are detailed in the following sections, while the review results are discussed in Section 4.

Phase 1: Planning the Review

The plan is to discover current DevOps adoption guidelines and review the existing approach. The research questions that guide this study are as follows:

1. What are the benefits and challenges faced by stakeholders during the adoption of DevOps?
2. What are the guidelines for DevOps adoption and implementation for DevOps stakeholders?

Phase 2: Conducting the Review

Establishing the search protocol is the essential first step in a systematic literature review (SLR). Following the development of the research questions, targeted search terms and queries were created for use across publication databases. The literature search process involved an automated search through ten prominent digital libraries, along with multiple rounds of the snowballing method to guarantee thorough results. The process started with locating pertinent primary studies across these commonly used libraries:

- i. SCOPUS
- ii. EBSCO-Discovery Service
- iii. ACM Digital Library
- iv. ScienceDirect
- v. Wiley Online Library
- vi. SAGE Journal
- vii. Google Scholar
- viii. SpringerLink
- ix. EBSCO Academic Search Complete
- x. ProQuest

A set of pertinent keywords was identified for the searches. Considering abbreviations and synonyms, five inclusion and exclusion criteria were established, utilizing the logical operators “AND” and “OR” to construct an effective search string. The inclusion criteria encompass studies that present DevOps concepts, definitions, and methodologies; studies that outline DevOps strategies and practices; studies that discuss DevOps issues, goals, models, and their benefits; studies that detail DevOps principles; and studies that feature case studies related to DevOps strategies and guidelines.

Exclusion criteria include studies that do not specifically address solutions to DevOps challenges and practices, studies that are exclusively surveys, as well as editorials, abstracts, thesis panels, monographs, books, and book sections. Additionally, articles not written in English and studies published before 2011 were also excluded.

Next is the selection of the final list of articles. The selection process involved a three-point numeric scale: yes (two points), partial (one point), and no (zero points). The maximum number of points any study could be assigned was 10 points, and the minimum was zero points. An article had to receive at least five points to be included in the final study, and all items with four or fewer points were excluded from the research for being of low quality. The assessed articles were chosen based on their high relevance to the study, and 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, but to exclude articles that contributed little to the research. Table 2 depicts the acceptance or rejection of either “Yes” or “No”, respectively, provided by the individual research assessment scores.

Table 2: Sensitivity analysis

#	Items Checklist	Included (2 points)	Partial (1 point)	Excluded (0 points)
1	Was the research problem explicitly defined in the study?	Yes	1	No
2	Were the study’s initial research gaps and contributions clearly outlined?	Yes	1	No
3	Did the results concentrate on a specific relationship and were they validated?	Yes	1	No
4	Were the limitations and suggestions for future research clearly articulated?	Yes	1	No
5	Are the objectives and principles of the DevOps approach clearly articulated?	Yes	1	No

This study then adopted a backward snowballing strategy to identify new papers to include. The researchers first decided to consider all papers published from 2011–2020 that were not included in the initial search; then, the researchers applied the previously used sensitivity analysis to the other retrieved primary studies. The aim was to ensure that all relevant studies were included in the final list. Lastly, another literature search was performed to verify that all the relevant primary studies had been included. An automated search over the following electronic databases: Wiley Online Library, Ebsco Discovery Service, and Springer Link was also conducted. From this search, 18 relevant primary studies were found, confirming the validity, reliability, and completeness of the snowballing approach the search results were filtered, and duplicates were removed from the first assessment. Table 3 lists the search results.

Table 3: Search results

#	Databases	Initial search	First assessment
1	ACM Dig Lib	27,058	117
2	Ebsco Discov Service	1,570	83
3	Ebsco Academic Search Complete	18	12
4	Google Scholar	3,723	17
5	Scopus	14	3
6	SAGE Journal	1	2
7	ScienceDirec	109	23
8	Wiley Oline Lib	89	15
9	SpringerLink	233	77
10	ProQuest	1544	9
TOTAL		32, 960	358

Interviews

The other half of the two-fold approach of the study involved interviews. An exploratory interview-based study was conducted with eleven participants from six organizations of various sizes and domains to understand DevOps usage in practice. The study aimed to identify ways to successfully adopt the DevOps approach and validate the data identified from the SLR with empirical evidence. The semi-structured interview approach allows for flexibility in questioning while maintaining a thematic framework [15]. This approach combines elements of both structured and unstructured interviews, providing comparable and reliable data while allowing for follow-up questions and two-way communication.

Interview Instruments

The interview questions are designed as follows:

Section A: DevOps Practices and Guidelines

1. Are you currently practicing DevOps or planning to adopt it in your organization?
2. What benefits have you seen from adopting DevOps for software development projects?
3. What challenges did you encounter when implementing DevOps?
4. Is it important to have standardized guidelines for all DevOps practitioners in various organization?
5. With various automation tools from providers like Amazon, Facebook, Netflix, and Google, do you think a common automated standard methodology is needed?
6. What guidelines do you recommend for DevOps adoption and implementation for stakeholders?

Section B: Collaboration and Communication in DevOps

1. From your DevOps experience, should collaboration and communication be limited to QA, development, and operations teams?
2. Should other stakeholders be involved in collaboration and communication during development, especially during continuous integration and testing?
3. Can the efficiency of the development process increase if more stakeholders are involved?
4. How can a proposed DevOps stakeholder framework enhance collaboration and communication during development?

Data Sampling

This research uses purposive sampling, selecting organizations based on their DevOps-based software development projects. Eleven participants were interviewed, ten from Nigeria and one from Germany, each were assigned an individual ID's. Table 4 provides details about the interview

participants. Due to the Covid-19 pandemic, face-to-face interviews were not feasible, so data collection occurred over five months, from September 2020 to January 2021, primarily through phone calls. Each interview lasted between 45 minutes to one hour. All interviews were recorded and later transcribed for analysis. The main author coordinated all the interview sessions, which included an introduction to the research study and a question-and answer process was conducted by the research papers main author and the interviewees.

Table 4: Participant demographics

Interviewee ID	Location	Role	Organisation Size	Experience
A_Sok	Northern Nigeria	Software Developer	> 4,000	Two year
A_Gac	Northern Nigeria	Software Developer	> 3000	One year
A-Gen	Northern Nigeria	Software Developer	10,000	Three years
A_Eng	Northern Nigeria	Web Developer	> 10,000	Three years
D_Umr	Northern Nigeria	Software Developer	>10,000	Three years
I_10	Southern Nigeria	Software Developer	>150	One year
L_Sad	Northern Nigeria	Senior Specialist	> 5000	One year
M_Ger	Germany	Senior Software Engineer	50	Two years
R_Ore	Southern Nigeria	Software Developer	30	Seven months
T_Tec	Southern Nigeria	Software Developer	50	Six months
I_9	Northern Nigeria	Software Engineer	10	One year

Coding Process

Coding is defined as a word, phrase, or sentence that represents an aspect of data or captures the essence or future of data, and it implies that coding is a method of labelling data that can be applied to interview transcripts, documents, artefacts, and field notes [16], [17]. This paper followed guidelines for the coding process to extract relevant statements. Firstly, relevant information in the data is identified and coded with a phrase or sentence to capture its meaning. These codes are then organized into a “Node.” Next, similar codes are grouped together through a process called sorting. The research then combines different views from individual participants and identifies themes through a process called synthesis. The final step is theorising, where the research develops a theory based on earlier coded ideas and assigns themes to represent individual research questions. The themes are then placed in a node, described, classified based on similarities, and documented [18]. We identified themes are as follow:

- i. Practicing DevOps
- ii. DevOps Guidelines
- iii. Involving Stakeholders
- iv. Concurrency Model

- v. DevOps Benefits
- vi. DevOps Challenges

Findings and Discussions

The SLR identified several benefits and challenges of DevOps adoption, which are presented in below sections.

DevOps Benefits

The adoption of DevOps practices brings numerous benefits to software development and operations teams. This section highlights three primary advantages of DevOps: frequent release cycles, high throughput, and enhanced quality. Extensive literature and case studies support each benefit, demonstrating how DevOps can significantly boost an organisation's productivity and improve product quality.

Frequent Release Cycle

In DevOps, the frequent release cycle entails the rapid development and deployment of code. Release intervals are typically brief, ranging from hours to weeks, with multiple releases potentially occurring each day. Research consistently indicates that adopting DevOps leads to shorter release cycles [29,30,31,32,33,34,35,36,37].

High Number of Throughput (Output)

Throughput measures the efficiency of a system or process in converting inputs into outputs. However, evaluating developer productivity is challenging. One approach is to measure productivity by the number of lines of code produced relative to implementation time. DevOps practices like automation, collaboration, and improved strategies are known to boost productivity. Studies support this, showing that better communication and reduced bureaucracy enhance the productivity of development and operations teams [15,19,20,21]. Effective communication strategies, such as daily stand-ups, collaboration tools like Slack or Microsoft Teams, and automated notifications, help eliminate bottlenecks and improve productivity [22]. Continuous integration and shared knowledge further enhance productivity throughout the development lifecycle [23].

Quality

This study explores how DevOps impacts team quality, code quality, and application quality. The literature underscores the importance of code and application quality, with evidence showing that DevOps adoption improves these aspects through increased developer responsibility and frequent, smaller releases [39,40,41,42,43,44,45,46]. Additionally, DevOps enhances quality assurance by improving communication and feedback loops. Continuous testing and deployment allow for

incremental releases, which improves production quality by better managing risks. According to Mishra and Otaiwi [24] many companies find that DevOps enhances production quality and reduces risks, while practices like continuous integration and feedback increase development team productivity and facilitate knowledge sharing across groups.

DevOps Challenges

There are clear benefits to adopting DevOps, but there are challenges that organisations face when implementing DevOps. Several articles provided an overview of the challenges of implementing DevOps [25,26,27,28]. This study discovered three major themes and challenges that organisations faced when implementing DevOps as described below.

Lack of Standard Guidelines and Knowledge Skills

According to the literature, DevOps lacks a specific method and guidelines or framework for implementation, making adoption extremely difficult. The lack of clear guidelines, skills, and knowledge surrounding DevOps is a significant challenge that has been discussed in many articles related to DevOps challenges [15,26,27,29,30,31]. Companies are often trapped by uneducated DevOps personnel, which further complicate the adoption process.

In addition to the lack of clear guidelines, organisations face resistance to change from employees, management, or stakeholders who may be hesitant to adopt new practices. Implementing DevOps can have cost implications for organisations, including the cost of new tools, training, and other resources needed to implement DevOps practices effectively. There is also a lack of standard tools related to a specific organisation's need for automation.

In one of the case studies, a company implementing DevOps encountered difficulties due to the technologies and platforms they used, despite other companies with advanced technology and knowledge. In another setting, the study acknowledged that recruiting the right people with the right skills is critical [23]. A lack of knowledge and necessary skill sets results in disastrous DevOps adoption and implementation processes.

The lack of clear guidelines, skills, and knowledge surrounding DevOps has been identified as one of the primary challenges of implementing DevOps. The primary studies conclude that DevOps adoption guidelines are lacking, which can result in software release cycle delays [32,33,34,35].

Obstacles to Change

According to the literature, senior management involvement can also pose challenges to DevOps adoption. DevOps automation tools are relatively expensive and require proper management, which may deter managers from seeing the value of DevOps. A specific hierarchical level and rigid management style can also stymie DevOps adoption. One of the barriers to DevOps adoption is a lack of valuable strategies, which leads to resistance to change among all relevant stakeholders.

Some stakeholders may resist DevOps adoption out of fear of losing their jobs or because they disagree with the collaborative phenomenon. The teams may have different orientation skills and other tasks that may interfere with their ability to work together [36].

The literature reveals that development and operations teams have competing goals, with developers wanting new features and bug fixes to be released into production as soon as possible, while the operations team wants to keep releases to a minimum to maintain stability and reliability [37,38]. Additionally, senior management may oppose DevOps adoption is another issue that has arisen. The cost of DevOps automation tools and the need for proper management may deter managers from seeing the value of DevOps.

Furthermore, a specific hierarchical level and rigid management style can stymie DevOps adoption. The researchers [39,40,41] discuss the lack of productivity at the start of the adoption process, which may send the wrong signal to senior management. However, due to such productivity issues, senior management may postpone the adoption of DevOps. These findings suggest that there are significant challenges to DevOps adoption, including conflicting goals between development and operations teams and senior management's reluctance to invest in DevOps automation tools. Addressing these challenges will require a collaborative effort between teams and a shift in management style to promote DevOps adoption.

Methodological Short-Fall and Cost of Tools and Automation

The primary studies have identified the cost of tools as a common barrier to DevOps adoption [27,42,43,44]. Due to a lack of standard DevOps practice, relevant stakeholders are having difficulty identifying the right tools and methods for their organizations, as well as determining how to use them consistently for continuous implementations [25,45,46]. This lack of standardisation between processes and tools impedes DevOps adoption, as highlighted in a primary study [47]. The initial tool arrangement, experimenting, and deciding which tool to use are all considered time factors, which can further delay DevOps adoption. Additionally, the relatively high cost of DevOps tooling may make it difficult for management to justify the investment in DevOps. A review of the available literature suggests that addressing these challenges will require standardization of DevOps practices and processes, as well as a clear understanding of the benefits of DevOps tooling. Organisation's must invest in the right tools and methods for their specific needs, and management must be willing to support this investment to ensure successful DevOps adoption.

Interview Findings

The interview respondents apparently saw several benefits to implementing DevOps. They viewed DevOps as a means of increasing the number of features implemented and generating more releases, which is consistent with rapid delivery. Almost all of the experts interviewed for this study responded positively and agreed that involving everyone with a stake in an organisation in collaboration and

communication during the software development process is the only way for DevOps to maintain high efficiency, high software quality, and effective customer satisfaction. Tables 5 and 6 present the codes, inclusion criteria, and the identified themes for benefits and challenges of DevOps, respectively.

Table 5: Codes for DevOps benefits

Codes	Codes Inclusion Criteria
Continue software delivery	Frequency & Relevance
We minimised production cost	Frequency & Relevance
There is an Improved software performance.	Frequency & Relevance
We improved individual skills	Frequency & Relevance
Our Revenues have increased	Frequency & Relevance
We have a stabilized work environment.	Frequency & Relevance
Individual responsibilities	Frequency & Relevance
Fast and reliable bugs fixation	Frequency & Relevance
Faster deployment	Frequency & Relevance
Repetitive tasks Automation	Frequency & Relevance
We have massively reduced downtime	Frequency & Relevance

Table 6: Codes for DevOps challenges

Codes	Code Inclusion Criteria
Different tools setups for Dev and Ops teams	Frequency, relevance
Lack of sufficient expertise	Frequency, relevance
Individual workloads	Frequency, relevance
Silos issues/ Resistance to change	Frequency, relevance
Lack of management support	Frequency, relevance
Tools integration from a different domain	Frequency, relevance
Regulatory compliance	Frequency, relevance
Individual Learning capability	Frequency, relevance
Culture adaptation	Frequency, relevance
Lack of standard Tool Adoption	Frequency, relevance
Existing processes wastage	Frequency, relevance
Learning new programming languages or scripts configuration	Relevance
Lack of knowledge on how to migrate	Relevance
Integrating different tools	Relevance
Environments inconsistency	Relevance
Insufficient resources to configure environments	Relevance
Game of blame in failure	Relevance

Automation is promoted by DevOps, which has been shown to improve release quality. DevOps facilitated communication between developers and operations engineers, encouraging collaboration to improve the software development process. Respondents also thought that DevOps encourages real-time monitoring, which promotes quick feedback loops and an experimental culture that encourages more interaction with stakeholders.

However, communication gaps with other relevant stakeholders and organisational culture were among the issues identified by our research. A lack of knowledge and information sharing can obscure critical facts, and guidelines for information sharing can be useful, but changing an organisation's culture can be difficult. The size of the organisation or having organisational-wide support for the change was also considered important by some participants. Environmental constraints can also be challenging, and working habits that prioritize security, for example, can prevent some businesses from implementing DevOps practices. As discovered via SLR studies in this study, difficult-to-replicate technical environments exacerbate the difficulties. As DevOps evolves, so will its definition, practices, and tools, posing a long-term challenge. Research indicates that DevOps should not be tied to any specific must-have tools or communication practices, but rather should be aligned with the higher-level goals that an organization wishes to achieve. The responses of the participants represent their perspectives on DevOps, and the reported advantages and disadvantages are not universal.

All interview respondents agree that an organisation must implement DevOps correctly with laid down guidelines, and a centralized unit should be established to develop these guidelines and operating techniques. DevOps is a mindset, a cultural shift, where teams adopt new ways of working. Development and operations teams coalesce into a functional team that communicates, shares feedback, and collaborates throughout the entire development and deployment cycle. DevOps practices enable rapid collection and response to user feedback by real-time live monitoring and rapid deployment. DevOps allows organisations to break down silos, work with agility, collaborate, develop and challenge ideas, and diversify expertise.

Proposed DevOps Guidelines

Based on the systematic literature review (SLR) and semi-structured interviews conducted, an initial guideline for DevOps adoption have been proposed to provide necessary guidance about the factors that will provide insight for DevOps adoption. Organisations should consider the principles that include adoption methodology, practices, principles, strategies, benefits, and challenges, as depicted in Table 7. The adoption methodology involves involving and convincing relevant stakeholders, ensuring expertise availability, and selecting appropriate tools. DevOps practices include continuous maintenance, testing, monitoring, integration, and deployment. DevOps principles include automation, culture, measurement, sharing, and feedback exchange from all relevant stakeholders. DevOps strategies involve the requirement process, selecting and testing tools, and automation. The benefits of DevOps include monitoring the benefits and impacts on an organisation, while the challenges involve understanding the challenges and continuing the adoption.

Table 7: Proposed DevOps guidelines

Criteria	Description	Explanation
DevOps adoption methodology	<ul style="list-style-type: none"> Involve and convince the relevant stakeholders. Ensure expertise availability. Select appropriate tools. 	An organization should determine and plan for the adoption by involving all relevant stakeholders if expertise is available at their disposal as well as the tools required for the automation to avoid disaster when not properly arranged
DevOps practices	<ul style="list-style-type: none"> Continuous maintenance. Continuous testing. Continuous monitoring. Continuous integration Continuous deployment 	All relevant stakeholders should understand the basic DevOps practices in terms of required continuity and the required tools to support such capabilities
DevOps principles	<ul style="list-style-type: none"> Automation Culture Measurement Sharing Feedback exchange from all relevant stakeholders 	All relevant stakeholders should be aware of the automation of the entire process, silos should be eliminated, the performance should be continuously measured, and sharing feedback should be vice versa.
DevOps strategies	<ul style="list-style-type: none"> Requirement process Select and test tools Automation 	Understand the strategies that involve the requirement process, selecting the right tools, and automation, then moving for the adoption can be made.
DevOps benefits	<ul style="list-style-type: none"> Monitor the benefits and the impacts on an organisation 	Measure the benefits to organisations and the individual’s expertise.
DevOps challenges	<ul style="list-style-type: none"> Understand the challenges and continue the adoption 	Identify the challenges gives way to making the necessary adjustment to the entire implementation for better performance.

Conclusion

In conclusion, this study aimed to explore the benefits and challenges of DevOps adoption in the rapidly evolving landscape of DevOps and IT firms. Through a systematic literature review and semi-structured interviews with six organisations of various sizes, the study provided valuable insights into the challenges faced by software organisations in adopting the DevOps culture and the benefits of DevOps adoption. The study proposed DevOps adoption guidelines based on the findings, which can help organisations determine whether they will benefit from DevOps implementation and understand the challenges associated with it. The findings of this study contribute to the existing literature on DevOps adoption and provide valuable recommendations for organisations seeking to adopt DevOps practices. Future work could involve further research into the impact of DevOps on organisations and the development of more comprehensive guidelines for DevOps adoption.

Acknowledgement

This research is funded by the Talent and Publications Enhancement Research Grant (TAPE-RG) titled “Empirical Analysis of Software Maintainability Metrics in DevOps Environments” at Universiti Malaysia Terengganu.

Conflict of Interest Statement

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- [1] Azad, N., & Hyrynsalmi, S. (2023). DevOps critical success factors — A systematic literature review. *Information and Software Technology*, 157, 107150. <https://doi.org/10.1016/j.infsof.2023.107150>
- [2] Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., Kuvaja, P., Mikkonen, T., Oivo, M., & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information and Software Technology*, 114, 217–230. <https://doi.org/10.1016/j.infsof.2019.06.010>
- [3] Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps practices of continuous automation for machine learning. *Information*, 11(7), 363. <https://doi.org/10.3390/info11070363>
- [4] Van Belzen, M., Trienekens, J., & Kusters, R. (2019). Critical success factors of continuous practices in a DevOps context. *28th International Conference on Information Systems Development (ISD2019 Toulon, France)*.
- [5] Grande, R., Vizcaino, A., & Garcia, F. O. (2023). Is it worth adopting DevOps practices in Global Software Engineering? Possible challenges and benefits. *Computer Standards & Interfaces*, 87, 103767. <https://doi.org/10.1016/j.csi.2023.103767>
- [6] Almashhadani, M., Mishra, A., Yazici, A., & Younas, M. (2023). Challenges in agile software maintenance for local and global development: An empirical assessment. *Information*, 14(5), 261. <https://doi.org/10.3390/info14050261>
- [7] Cusumano, M. A., & Selby, R. W. (1998). *Microsoft secrets: how the world's most powerful software company creates technology, shapes markets, and manages people*. Free Press.
- [8] Fawzy, A. H., Wassif, K., & Moussa, H. (2023). Framework for automatic detection of anomalies in DevOps. *Journal of King Saud University - Computer and Information Sciences*, 35(3), 8-19. <https://doi.org/10.1016/j.jksuci.2023.02.010>

- [9] Zarour, M., Alhammad, N., Alenezi, M., & Alsarayrah, K. (2020). DEVOPS PROCESS MODEL ADOPTION IN SAUDI ARABIA: AN EMPIRICAL STUDY. *Jordanian Journal of Computers and Information Technology*, 0, 1. <https://doi.org/10.5455/jjcit.71-1580581874>
- [10] Riungu-Kalliosaari, L., Mäkinen, S., Lwakatere, L. E., Tiihonen, J., & Männistö, T. (2016). DevOps adoption benefits and challenges in practice: A case study. in *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24* (pp. 590–597).
- [11] Roy, R., Stark, R., Tracht, K., Takata, S., & Mori, M. (2016). Continuous maintenance and the future – Foundations and technological challenges. *CIRP Annals*, 65(2), 667-688. <https://doi.org/10.1016/j.cirp.2016.06.006>
- [12] Laukkanen, E., Itkonen, J., & Lassenius, C. (2016). Problems, causes and solutions when adopting continuous delivery—A systematic literature review. *Information and Software Technology*, 82, 55-79. <https://doi.org/10.1016/j.infsof.2016.10.001>
- [13] Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2006). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4), 571-583. <https://doi.org/10.1016/j.jss.2006.07.009>
- [14] Díaz, J., López-Fernández, D., Pérez, J., & González-Prieto, Á. (2021). Why are many businesses instilling a DevOps culture into their organization? *Empirical Software Engineering*, 26(2), Article 25. <https://doi.org/10.1007/s10664-020-09919-3>
- [15] Mandepudi, S. (2019). *Communication Challenges in DevOps & Mitigation Strategies* [Master Thesis, Blekinge Institute of Technology]. <https://www.diva-portal.org/smash/get/diva2:1368400/FULLTEXT02>
- [16] Harding, T. S., & Whitehead, D. (2013). Analysing data in qualitative research. In *Nursing & midwifery research: Methods and appraisal for evidence-based practice* (4th ed., pp. 141-160). Elsevier Mosby.
- [17] Quick, J., & Hall, S. (2015). Part two: Qualitative research. *Journal of Perioperative Practice*, 25(7-8), 129-133. <https://doi.org/10.1177/1750458915025007-803>
- [18] Brandão, C. (2014b). Book review: Qualitative data analysis with NVivo (2nd ed.). *Qualitative Research in Psychology*, 12(4), 492-494. <https://doi.org/10.1080/14780887.2014.992750+>
- [19] Guerrero, J., Certuche, C., Zúñiga, K., & Pardo, C. (2020). Trends in devops: A systematic mapping of the literature [Tendencias en devops: Un mapeo sistemático de la literatura],” *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, vol. 2020, no. E32, 2020.

- [20] Badshah, S., Khan, A. A., & Khan, B. (2020). Towards process improvement in DevOps: A systematic literature review. *ACM International Conference Proceeding Series*, pp. 427-433, 2020, <https://doi.org/10.1145/3383219.3383280>
- [21] Lucy Ellen, L. (2017). *DevOps adoption and implementation in software development practice: concept, practices, benefits and challenges.*. OuluREPO. <https://urn.fi/URN:ISBN:9789526217116>
- [22] Muñoz, M., Negrete, M., & Mejía, J. (2019). Proposal to avoid Issues in the DevOps Implementation: A systematic literature review. In *Advances in intelligent systems and computing* (pp. 666-677). https://doi.org/10.1007/978-3-030-16181-1_63
- [23] Albuquerque, A. B., & Cruz, V. L. (2018). Implementing DevOps in legacy systems. In *Advances in intelligent systems and computing* (pp. 143-161). https://doi.org/10.1007/978-3-030-00184-1_14
- [24] Morales, J. A., Yasar, H., & Volkman, A. (2018). Implementing DevOps practices in highly regulated environments. *ACM International Conference Proceeding Series, vol. Part F1477*, pp. 1-9, 2018, <https://doi.org/10.1145/3234152.3234188>
- [25] Wiedemann, A., Wiesche, M., Gewalt, H., & Krcmar, H. (2019). Implementing the Planning Process within DevOps Teams to Achieve Continuous Innovation. In *Proceedings of the Annual Hawaii International Conference on System Sciences*. <https://doi.org/10.24251/hicss.2019.841>
- [26] Daneva, M., & Bolscher, R. (2020). What we know about software architecture styles in continuous delivery and DevOps? In *Communications in computer and information science* (pp. 26-39). https://doi.org/10.1007/978-3-030-52991-8_2
- [27] Katal, A., Bajoria, V., & Dahiya, S. (2019). DevOps: Bridging the gap between development and operations. In *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, pp. 1-7. <https://doi.org/10.1109/iccmc.2019.8819631>
- [28] Cois, C. A., Yankel, J., & Connell, A. (2014). Modern DevOps: Optimizing software development through effective system interactions. *IEEE International Professional Communication Conference*, vol. 2015-Janua, pp. 1-7. <https://doi.org/10.1109/ipcc.2014.7020388>
- [29] Bolscher, R., & Daneva, M. (2019). Designing software architecture to support continuous delivery and DevOps: A systematic literature review. *ICSOF 2019 - Proceedings of the 14th International Conference on Software Technologies*. <https://doi.org/10.5220/0007837000270039>

- [30] Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. (2013). A grounded theory analysis of modern web applications: Knowledge, skills, and abilities for DevOps. In *RIIT 2013 - Proceedings of the 2nd Annual Conference on Research in Information Technology, 2013*, pp. 61-62. <https://doi.org/10.1145/2512209.2512229>
- [31] Artac, M., Borovssak, T., Di Nitto, E., Guerriero, M., & Tamburri, D. A. (2017). DevOps: Introducing infrastructure-as-code. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017*, pp. 497-498. <https://doi.org/10.1109/icse-c.2017.162>
- [32] Perez, J. E., Gonzalez-Prieto, A., Diaz, J., Lopez-Fernandez, D., Garcia-Martin, J., & Yague, A. (2021). DevOps Research-Based teaching using qualitative research and Inter-Coder Agreement. *IEEE Transactions on Software Engineering*, 48(9), 3378-3393. <https://doi.org/10.1109/tse.2021.3092705>
- [33] Cuppett, M. S. (2016). Automation and code control. In *DevOps, DBAs, and DBaaS*, Berkeley, CA: Apress (pp. 107-121). https://doi.org/10.1007/978-1-4842-2208-9_8
- [34] Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. *Computer Science Review*, 38, 100308. <https://doi.org/10.1016/j.cosrev.2020.100308>
- [35] De Baysier, M., Azevedo, L. G., & Cerqueira, R. (2015). ResearchOps: The case for DevOps in scientific applications. In *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*. <https://doi.org/10.1109/inm.2015.7140503>
- [36] Laukkarinen, T., Kuusinen, K., & Mikkonen, T. (2017). DevOps in regulated software development: Case medical devices. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Results Track, ICSE-NIER 2017*. <https://doi.org/10.1109/icse-nier.2017.20>
- [37] Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps capabilities, practices, and challenges: Insights from a case study. *ACM International Conference Proceeding Series, vol. Part F1377*, pp. 57-67. <https://doi.org/10.1145/3210459.3210465>
- [38] Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). DevOps adoption benefits and challenges in practice: A case study. In *Lecture notes in computer science* (pp. 590-597). https://doi.org/10.1007/978-3-319-49094-6_44
- [39] Shahin, M., & Babar, M. A. (2020). On the role of software architecture in DevOps transformation: An industrial case study. *Proceedings - 2020 IEEE/ACM International Conference on Software and System Processes, ICSSP 2020, vol. 10*, pp. 175-184. <https://doi.org/10.1145/3379177.3388891>

- [40] Ghantous, G. B., & Gill, A. (2017). DevOps: Concepts, practices, tools, benefits and challenges. *Proceedings of the 21st Pacific Asia Conference on Information Systems: “Societal Transformation Through IS/IT”*, PACIS 2017, pp. 96. <http://aisel.aisnet.org/pacis2017/96>
- [41] Gottesheim, W. (2015). Challenges, benefits and best practices of performance focused DevOps. *LT2015 - Proceedings of the 4th ACM/SPEC International Workshop on Large-Scale Testing, in Conjunction with ICPE 2015*, pp. 3. <https://doi.org/10.1145/2693182.2693187>
- [42] Leite, L., Rocha, C., Kon, F., Milojevic, D., & Meirelles, P. (2019). A survey of DevOps Concepts and challenges. *ACM Computing Surveys*, 52(6), 1-35. <https://doi.org/10.1145/3359981>
- [43] Jones, S., Noppen, J., & Lettice, F. (2016). Management challenges for devops adoption within UK SMEs. In *QUDOS 2016 - Proceedings of the 2nd International Workshop on Quality-Aware DevOps, co-located with ISSA 2016*, 2016, pp. 7-11. <https://doi.org/10.1145/2945408.2945410>
- [44] Trigo, A., Varajão, J., & Sousa, L. (2022). DevOps adoption: Insights from a large European Telco. *Cogent Engineering*, 9(1), Article 2083474. <https://doi.org/10.1080/23311916.2022.2083474>
- [45] Maroukian, K., & Gulliver, S. R. (2020). Leading DevOps Practice and Principle Adoption. *9th International Conference on Information Technology Convergence and Services (ITCSE 2020), May 30-31, 2020, Vancouver, Canada*. <https://doi.org/10.5121/csit.2020.100504>
- [46] Maroukian, K., & Gulliver, S. R. (2020b). Exploring the link between leadership and DEVOPs practice and principle adoption. *Advanced Computing an International Journal*, 11(4), 1-18. <https://doi.org/10.5121/acij.2020.11401>