Journal of Information Systems & Information Technology (JISIT)
ISSN: 2478-0677; Vol. 8 No.1 2023; [Pages 25 - 38]

The Complications and Solutions of Using DevOps in the discipline of Software Development

Sahabdeen Aysha Asra[1], Lakmal Rupasinghe[2]
Department Information Technology[1], South Eastern University of Sri Lanka
ashrasahabdeen005@seu.ac.lk
Faculty of Computing[2], Sri Lanka Institute of Information Technology Sri Lanka
ashrasahabdeen005@seu.ac.lk[1],lakmal.r@sliit.lk[2]

Abstract

When delivering respected software to clients during the development of software process, there are frequently disputes between operation and development. DevOps is suggested as a significant, recently developed idea to resolve the dispute between the operations team and development team. DevOps is becoming ever more popular among businesses and organizations. Even though DevOps is still a relatively new idea, it is important to comprehend potential problems and viable solutions. The research is conducted through literature review. Various number of articles are reviewed to develop this article. In this article we intend to analyze the challenges of DevOps during the software development process. Also, the mitigation strategies of those challenges as well.

Keywords: *DevOps, challenges, mitigation, software development*

1. Introduction

Development, IT operations, and Quality Assurance are all different divisions in conventional software firms. When development as well as operations are delivering great software to clients on a regular basis, disputes arise frequently. Operations suggest that developers do not regularly modify products, whereas developers are interested in swiftly delivering new additions or adjustments to clients. Operations also need better security and stability. Operations find it difficult to accept the frequent release of new editions. Such conflicts may impede the creation of software. Understanding areas of software engineering that are not DevOps is crucial for comprehending the breadth of DevOps. DevOps is not a brand-new division, and there is no DevOps approach or procedure either. While DevOps and Agile are similar, they also vary in key crucial ways. Agile represents shifting perspectives. However, it brings about alterations in company culture. Also, it is more like a conceptual framework as compared to Agile.

The goal of the study is to present a comprehensive articulation of DevOps application difficulties and associated mitigation solutions. Our paper used the

systematic review approach to identify how frequently DevOps will be able to comprehend the problems and the ways in which they might be mitigated. Additionally, the article expects that by sharing our research, DevOps practitioners will be better able to manage risks, reduce DevOps-related issues, and create budgets that take these challenges into account.

1.1 Objective of the study

In this study, we list the problems with applying DevOps to software development, along with solutions. The systematic literature review uses the gathered challenges and mitigating techniques.

The main objectives of the study are:

- Defining the DevOps.
- Identifying the problems with using DevOps when developing software.
- Identifying possible solution or mitigation strategies when problem arises during software development in DevOps

1.2 Research question

The following research questions are developed in accordance with the goal and objectives:

- RQ1: What problems with employing DevOps during software development?
- RQ2: How are they resolved by using strategies?

2. Literature Review

The current robotic software development industry occasionally employs the majority of the aforementioned DevOps methods [1]. However, it's neither done in the suggested manner nor is it as thorough as the suggested mapping. Even if most of the procedures are already occasionally used, this is only done when it is necessary. Additionally, they are often only utilized during a select few stages of the creation of robotic software, not throughout the full procedure [2]. Software development involves a great deal of difficulties. the lack of uniform KPIs (Key performance indicators) across DevOps as well as longer go-to-market times are caused by the collaboration, cooperation, and communication gap between Dev and Ops. Other problem include team conflicts, inefficiencies, and difficulties [3] in bringing about synergy in operation and development processes, management, people, knowledge, accountability, governance, and tools. Now, it takes at least 10 weeks to bring a change to company. Also organizations must concentrate on

DevOps culture rather than relying just on CI/CD processes to reap benefits [4].

There are risks using DevOps arise when developing software. It is crucial for corporate risk management that staff receive pertinent information regarding events and actions in order for them to fulfill their risk management and other duties. Communication is a key component of DevOps by nature. Collaboration and information sharing inside and across DevOps [5] teams are essential components of this approach. Socialization, externalization, combination, and internalization are the four processes through which information is shared. It is crucial to highlight that this framework places equal emphasis on knowledge sharing, which raises overall employee skill levels [6]. Knowledge sharing not only focuses on giving explicit information about events and actions for improved risk management. To mitigate the risk there are many strategies proposed. People Members of the team will unavoidably be impacted by changes in the development process and responsibilities. Like how these developments will need significant adjustments in how individuals think and act. Accordingly, the fundamental driver of the adoption of new technologies in agile technology is human behavior [7].

3. Methodology

According the RQ focuses on the difficulties of DevOps during software development and the literature-reported mitigation techniques. The important phrases are "DevOps," "challenge," “strategies” and “challenges” as these are the points. To include as many articles as possible, synonyms and alternative terms are suggested for each keyword. DevOps is a relatively new idea, and there aren't many academic articles on it, therefore searching is broad to locate as many pertinent publications. The following major factors have been considered to shortlist the downloaded articles which are selected from reputed publishers such as Springer, IEEE, Emerald, Sage and Inderscience. Only full-length papers are accepted.

- Recently published papers.
- Open access articles.
- Published in English language
- Articles in citation databases with a high index.

4. Results and discussion

The research found that the various risks were found during software development

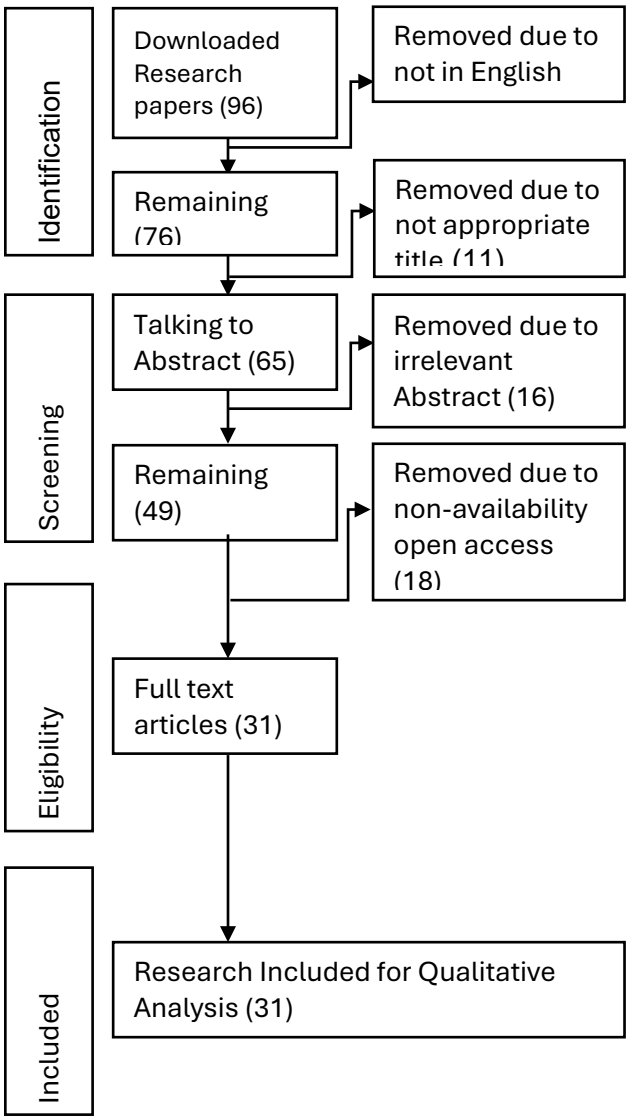

Figure 1 Arrangements of Research

in the field of DevOps. According to the previous papers the authors suggested some risk mitigation strategies in DevOps while developing software as well. In this sense, there are some specific risks and mitigation strategies that are analyzed and stated below.

References	Fields	Definitions
[8], [9]	Collaboration or Communication field	A successful DevOps approach has more to do with people than it does with technology. People are always difficulty when utilizing DevOps because they must collaborate in ways they have never done before, she said. It may be quite difficult to connect discrete development teams with operations teams while developing conventional software. On the surface, DevOps appears to be quite straightforward, but in practice, it can be challenging for software engineers as well as operators who come from various working contexts as well as have only had little prior contact.
[10]	Definition field	DevOps is a relatively new term that combines the terms "development" and "operations," making it difficult to explain or familiarize oneself with. Due to the vagueness of the term "DevOps," what is really done may not change all that much from what was done before it was given that name. Each organization has a different path to a successful DevOps implementation.
[11], [12]	Management field	Development and operations used to be two distinct departments with their own management structures. Development as well as operations must collaborate with DevOps, which necessitates a new management structure to oversee daily operations. The adoption process for DevOps may be hampered by the absence of an adequate management structure between development and operations. Large-scale projects can use DevOps expertise, but DevOps will be difficult since it has not been properly planned and controlled.
[13], [14]	Cultural field	Automated procedures require both technological assistance and strong rationale, such as a build plan. Therefore, it is unlikely that these methods might be employed to close the gap between operations and development. More than a quarter of individuals with DevOps expertise have not encountered implementation errors, but for other users, cultural change continues to be more difficult than technology. One of the main obstacles to implementing DevOps is the resistance of the development and operations teams to the change.

[15], [16]	Quality or evaluation field	There aren't any formal performance evaluation standards or metrics yet. DevOps assessment cannot be ensured as a result. Using DevOps presents a significant challenge but does not ensure high performance. This method increases the efficiency of development while allowing stakeholders to monitor current performance in real time. Stakeholders can assess the performance and compare it to their anticipated goals. This will make it easier for stakeholders and developers to evaluate current progress, find issues, and fix them as soon as feasible.
[17], [18], [14]	Field of tool	The DevOps community shares a lot of open-source artifacts, including modules, templates, and scripts to install middleware and application components. These artifacts may be utilized to automate the deployment of many types of apps due to their portability and the assistance of the DevOps community. However, DevOps artifacts are restricted to certain tools, making it difficult to use and combine them. Additionally, practitioners must put in extra work to understand and follow these artifacts.
[19], [20]	Adoptability	Even though the DevOps method places a focus on the advantages for individuals and teams, people are nevertheless curious about how DevOps would impact them personally. The enterprises should demonstrate to them how DevOps would enhance their job and handle the issues that matter to them the most. The management will benefit from this kind of staff motivation. Informing employees of the advantages DevOps would bring to their current job will motivate them to use DevOps.

4.1 Collaboration or Communication field *[8]*

DevOps is growing in popularity, and it seems like new DevOps tools are released every week. Although there are various tools to pick from, the development team and its members remain the only constants. The lack of communication and collaboration between development and operations will cause issues like team strife and inefficiencies. The communication between the operations and development teams is facilitated by this technique. Development and operations can set up a communication platform with appropriate management. Staff members can collaborate to solve problems and exchange ideas and experiences. When collaborating on ideas and professional experiences [21] aid in improving

communication, encouraging sentiments, and spotting problems early on for both the operations team and development team. "The technology will function if we can encourage individuals to collaborate and communicate more. To the greatest extent possible, DevOps will succeed if we can encourage individuals to share their ideas and professional experiences [22].

When thinking about cooperation inside DevOps, it's crucial to explore ways to stop teams from blaming one other when an issue arises. In the past, operations and development carried out their respective jobs independently. Collaboration between development and operations is necessary for DevOps adoption. They could accuse one another of abdicating their duties when issues occur. There must be a procedure to identify the underlying causes of performance issues to address the issue of shared blame and responsibility avoidance amongst teams. Teams can identify the root cause and accountable parties with the use of this method. Conflict between the development and operation teams is reduced by finding a solution to the issue of shared responsibility. Geographical distance makes it difficult to communicate face-to-face; effective communication cannot be achieved only through technological means. There is a possibility that development and operations work may be dispersed across many locations, making face-to-face contact impossible. The difficulty in communicating is caused by the varied time zones [8].

4.2 The field of definition

The organization may implement process, human, and technical improvements and innovations by defining the shared definition of DevOps [10] throughout the business. Many individuals are unfamiliar with the phrase "DevOps," and they mistakenly think that it refers to an IT position that is a cross between a developer and a systems administrator. Companies often establish a DevOps department as well as choose a manager who is knowledgeable about both operations and development because of this misunderstanding. Gaps in the use of DevOps are also a result of different perspectives on discipline. Another issue is that a lack of confidence in DevOps may result in a mindset that discourages the use of DevOps. There isn't a common definition for DevOps, nevertheless, both in academics and in the communities of practitioners. There are various insufficient or even incorrect interpretations of DevOps due to the absence of common definitions, which further complicates the field. DevOps is therefore difficult to understand at its foundation and difficult to apply effectively [13].

4.3 The field of management

Providing a methodical management strategy to handle DevOps needs tactic aids in

the effective management of DevOps needs and development. To provide a methodical method for handling DevOps needs, provides a concept for the entire DevOps knowledge management strategy. People may manage their DevOps requirements and increase productivity by using a methodical strategy. Managing DevOps knowledge is crucial for effective utilization. This technique improves the utilization of DevOps by developers and operators. Unstructured and semi-structured data sources are now used to disseminate DevOps expertise [11]. The productivity of developers and operators may be increased with methodical management. Due to proper administration, this method can also raise the caliber of the final output. The fundamental problem with the existing operations is that the operations team is under increasing amounts of work pressure. Adding more DevOps responsibilities and meetings to the operations team's already-heavy workload may cause employee fatigue. The organization that tends to use DevOps must make some shifts in order to deploy DevOps effectively and make it work. Following the mitigating behavior, the staff's tasks might be fairly distributed. Employees will be able to get rid of the unfavorable feelings that DevOps causes, which will encourage the adoption of the practice [12].

4.4 Culture

Putting incentives and approvals into effect and administering them is essential to create and manage awards and approval processes for teams. The managers might use this technique to inspire staff members. The personnel will thereafter be open to fresh alterations. The development schedule and effectiveness will be impacted by the team's passion. To meet cultural obstacles, goals and incentives must be adjusted and balanced. The developers are encouraged by this strategy to embrace DevOps. Increased staff passion will aid in promoting DevOps adoption. A key tactic for lowering anxiety and steadily constructing business cases is to communicate and recognize DevOps success in the development process. This procedure aids developers in comprehending their accomplishments and gives them the self-assurance to keep employing DevOps. Developers can learn about operations through real-world situations and via team interactions that break down the "wall". This method can facilitate cooperation by assisting developers in comprehending how operations operate. Additionally, this could make it easier for engineers to adopt DevOps. The development and operation teams may better understand one another and communicate experiences and ideas through this technique. There will be less of a separation between the two groups after this. Effective teamwork may boost DevOps adoption, product quality, and team productivity [13].

The burden for developers and operations may rise because of the application of DevOps. If there aren't any new development resources, the effort devoted to

genuine development may be divided to embrace DevOps. Another worry is that the excessive workload might make developers reluctant to utilize DevOps. DevOps in practice necessitates that development personnel have in-depth expertise in operations, and vice versa. However, there are several competencies that need to be enhanced for the operations, including understanding of the development process and code reviewing in addition to programming proficiency. Developers must learn the abilities to deploy in various environments for the development side. Some contend that it's challenging for operations and developers to manage the depth of both fields' expertise [23].

DevOps culture depends on having some interest in what other teams are doing. However, developers often don't care about operations tasks and vice versa. People in an organization always have an interest in their particular field, which is the 40 causes of this lack of interest. Learning knowledge and talents outside of their fields will take a lot of time, whether it be in development or operation. Employees are patient in learning new technologies and tools because they are committed to using DevOps. This procedure can aid in the adoption of DevOps by firms and enhance the effectiveness of workers' DevOps training. As a result, all DevOps developers may collaborate, and a platform is provided for all practitioners to exchange knowledge. Employee collaboration and communication may be improved with the use of this approach, which will boost DevOps adoption. The development of the DevOps tool industry and the expansion of DevOps[24] faced difficulties due to the complicated organizational structure and culture. The organization's complexity makes DevOps adoption challenging and raises the risk in application process. The adoption of DevOps may be hampered by staff members who are unwilling to adapt to new developments. Employee reluctance to accept the adjustment or the added effort will have an impact on their enthusiasm and drive. All of these issues arise while adopting DevOps [24].

4.5 Quality or evaluation

The crucial performance of current initiatives is aided by this strategy. This procedure can improve the quality of ongoing projects, evaluate the quality of ongoing projects, and spot issues early on. User behavior, deployment, and load all have a significant impact on performance. DevOps implementation presents performance challenges that are difficult to predict. DevOps implementation is extremely difficult due to this [25].

Creating a shared knowledge of operation, testing [25], and development approach can aid professionals in understanding one another's work and timely performance and product quality evaluation. A unified knowledge of development, testing, and

operation is crucial from the perspective of performance. From the perspective of performance, it's critical to develop a common vocabulary and performance measures that are clear to all teams. The development team can define standard performance indicators with the aid of this procedure. Common performance criteria aid in the evaluation of ongoing projects by developers and lessen inequality-related problems amongst groups. When creating applications in the context of DevOps and cloud services [26], it might be difficult to ensure security and dependability. DevOps complete automation may not yet be able to ensure the caliber of testing and deployment. Making quality assurance and testing work with continuous development and delivery will be a challenge for traditional software development teams adopting DevOps. Since DevOps demands continuous development and automation, security and compliance are more likely to suffer when there aren't any good security engineers on the team [27].

4.7 Tool

For practitioners to be effective, they must develop a deep understanding of the technology and orchestration of many tools and artifacts. Since a lot of open-source artifacts, including templates, modules, and scripts, are tied to certain providers or tools, making it challenging for others to reuse them and for them to be incorporated into the assets. We can build a framework for the automated transition of a unified standard to address this issue. For instance, according to, the authors suggest a standard for DevOps artifacts. To automate the transformation of current DevOps artifacts, this article develops model artifacts based on the TOSCA standard. No one tool fits all DevOps processes, despite what some manufacturers claim. Version control, continuous integration, configuration management [9], deployment, and monitoring are essential components of the DevOps tool chain. Numerous technologies, including GitHub [28], Puppet, Jenkins, Travis CI, MCollective, and SignalFx, are used in these procedures. The technology is still essential in today's DevOps application. Collecting crucial data from DevOps tools is highly vital and difficult when utilizing them [29].

DevOps automation is rapidly varying, making it challenging to select the best method for implementing the complete DevOps automation for a given application. Every DevOps-based development process needs a unique set of DevOps tools. It might be difficult to choose the best tools while utilizing DevOps. The compatibility of various software versions and hardware requirements present difficulties. There might be issues with interdependency or compatibility between different program versions. For DevOps tools, this is difficult. The old software may no longer be appropriate for the process employing DevOps since DevOps changed the way people previously worked. It's possible that outdated software lacks the necessary

features for DevOps. For development and operations teams to collaborate, synergy must be ensured throughout the process, among the staff, and in product governance. In order to guarantee the effective implementation of DevOps, it is crucial to develop synergy between the components outlined above [2].

4.8 Adoptability

The excitement of employees for learning new technologies and tools will improve as a result of educating them on the advantages of DevOps. The developers will learn about DevOps' benefits via this procedure. Informing workers on how DevOps would handle the issues that worry them the most with their current job will motivate staff to use DevOps. The interest of employees in learning new tools and technologies will rise by demonstrating the advantages of DevOps. Not only does DevOps require a precise definition, but it also requires a clear definition and adoption targets. Failure while implementing DevOps might result from unclear description and goals. If people's perceptions of the objectives of implementing DevOps varies, there may be a variety of necessary steps [29].

DevOps is not appropriate for processes and practices like lengthy testing durations or exact deployment schedules, which may be required by several customers. DevOps may not be appropriate for all clients, and clients may not desire DevOps. When considering communication, shared objectives and practices, decision-making [30], and systems thinking inside the business [31], DevOps has both a detrimental and beneficial impact on the organization's structure [29].

5. Conclusion

The problems and mitigation techniques of employing DevOps during software development are researched in this essay in the literature. The goal of this article is to present a thorough description of DevOps application difficulties and associated mitigation solutions. To uncover DevOps difficulties and mitigation techniques, the authors perform an organized literature study. The study’s objective is to give a thorough explanation of DevOps application challenges and related mitigating measures. People that commonly utilize DevOps will be able to understand the issues and the potential solutions because our research used the systematic review technique to discover the issues. The study also anticipates that by disseminating our findings, DevOps practitioners would be better able to manage risks, lessen DevOps-related problems, and develop budgets that account for these difficulties.

References

[1] S. Zaib and P. Lakshmisetty, "A systematical literature review and industrial survey in addressing the possible impacts with the continuous testing and delivery during DevOps transformation.," no. February, 2021, [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1531267.

[2] K. C. Ronanki, "Robotic Software Development using DevOps," no. June, 2021, [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1586512.

[3] M. Fernandes, S. Ferino, A. Fernandes, U. Kulesza, E. Aranha, and C. Treude, "DevOps Education: An Interview Study of Challenges and Recommendations," in *International Conference on Software Engineering*, 2022, pp. 90–101, doi: 10.1109/ICSE-SEET55299.2022.9794182.

[4] M. Rowse and J. Cohen, "A Survey of DevOps in the South African Software Context," vol. 0, 2021.

[5] J. Wettinger, V. Andrikopoulos, and F. Leymann, "Automated capturing and systematic usage of DevOps knowledge for cloud applications," *Proc. - 2015 IEEE Int. Conf. Cloud Eng. IC2E 2015*, pp. 60–65, 2015, doi: 10.1109/IC2E.2015.23.

[6] P. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, "A survey of DevOps concepts and challenges," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–35, 2019.

[7] C. Wang and C. L. Wang, "Adopting DevOps in Agile Challenges and Solutions," no. February, 2018.

[8] M. Strategies, "Communication Challenges in DevOps," 2019.

[9] M. S. Khan, A. W. Khan, F. Khan, M. A. Khan, and T. K. Whangbo, "Critical Challenges to Adopt DevOps Culture in Software Organizations : A Systematic Review," *IEEE Access*, vol. 10, pp. 14339–14349, 2022, doi: 10.1109/ACCESS.2022.3145970.

[10] and R. C. M. de Bayser, L. G. Azevedo, "ResearchOps: The case for DevOps in scientific applications," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 1398–1404.

[11] Z. O. Mishra, Alok, "DevOps and software quality: A systematic mapping," *Comput. Sci. Rev.*, p. 100308, 2020.

[12] M. A. B. Shahin, Mojtaba, "On the role of software architecture in DevOps transformation: An industrial case study," in *Proceedings of the International Conference on Software and System Processes*, 2020, pp. 175–184.

[13] A. S. Venkatajayasri, "AN EXTENSIVE RISK-MITIGATING FRAMEWORK FOR CONTINUOUS TESTING USING DEVOPS," no. December, p. 2020, 2020.

[14] and M. G. Srivastava, Divya, Madhushi Verma, Shashank Sheshar, "DevOps Tools: Silver Bullet for Software Industry.," in *Springer Nature Singapore*, 2022, pp. 105–118.

[15] A. Alnafessah, A., Gias, A. U., Wang, R., Zhu, L., Casale, G., Filieri, "Quality-aware devops research: Where do we stand?," in *IEEE access*, 2021, pp. 44476–44489.

[16] and M. H. H. Ibrahim, Mahmoud Mohammad Ahmad, Sharifah Mashita Syed-Mohamad, "Managing quality assurance challenges of DevOps through analytics," in *In Proceedings of the 2019 8th International Conference on Software and Computer Applications*, 2019, pp. 194–198.

[17] R. Luz, W. P., Pinto, G., Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study," *J. Syst. Softw.*, vol. 157, p. 110384, 2019.

[18] and R. P. Q. Solayman, Haleema Essa, "Seamless Integration of DevOps Tools for Provisioning Automation of the IoT Application on Multi-Infrastructures," in *In 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, 2023, pp. 1–7.

[19] B. F. Hemon, Aymeric, Barbara Lyonnet, Frantz Rowe, "From agile to DevOps: Smart skills and collaborations," in *Information Systems Frontiers*, 2022, pp. 927–945.

[20] and S. R. G. Maroukian, Krikor, "Leading DevOps practice and principle adoption," in *arXiv preprint arXiv:2008*, 2020, p. 10515.

[21] and M. S. Teixeira, Daniel, Rúben Pereira, Telmo Henriques, Miguel Mira Da Silva, João Faustino, "A maturity model for DevOps.," *Int. J. Agil. Syst. Manag. 13*, pp. 464–511, 2020.

[22] and S. L. Almeida, Fernando, Jorge Simões, "Exploring the benefits of combining DevOps and agile," *Futur. Internet*, vol. 14, no. 2, p. 63, 2022.

[23] H. Y. Hobeck, Richard, Ingo Weber, Len Bass, "Teaching DevOps: a tale of two universities," in *Proceedings of the 2021 ACM SIGPLAN International Symposium on SPLASH-E*, 2021, pp. 26–31.

[24] "Towards DevOps in the Embedded Systems Domain: Why is It So Hard?," in *Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 5437– 5446.

[25] S. R. Bertolino, Antonia, Guglielmo De Angelis, Antonio Guerriero, Breno Miranda, Roberto Pietrantuono, "DevOpRET: Continuous reliability testing in DevOps," *J. Softw. Evol. Process*, p. e2298., 2020.

[26] D. S. Battina, "Devops, A New Approach To Cloud Development & Testing," *Int. J. Emerg. Technol. Innov. Res. (www. jetir. org)*, pp. 2349–5162, 2020.

[27] D. T. Bonda, "Tools Integration Challenges Faced During DevOps Implementation," no. June, 2021.

[28] and S. H. Beetz, Florian, "GitOps: The Evolution of DevOps?," *IEEE Softw.*, vol. 39, no. 4, pp. 70–75, 2021.

[29] Y. Liu and Y. Zhou, "The Challenges and Mitigation Strategies of Using DevOps during Software Development," no. February, 2017.

[30] C. W. and A. Aurum, "Towards a decision-making structure for selecting a research design in empirical software engineering," *Empir. Softw. Eng.*, vol. 20, no. 6, pp. 1427–1455, 2015.

[31] H. Wiedemann, A., Forsgren, N., Wiesche, M., Gewald, H., & Krcmar, "Research for practice: the DevOps phenomenon," *Commun. ACM*, vol. 62, no. 8, pp. 44–49, 2019.