International Journal on Optimization and Applications
IJOA. Vol. 06, Issue No. 01, Year 2026, www.ijoa-journal.com

Implementing DevSecOps: A Systematic Literature Review on Integrating Security into DevOps

Rjanna Miki M. Balaybay, Jenny Ann T. Guyong, Justine Mae B. Macario
College of Information Technology and Computer Science
University of the Cordilleras
Baguio City, Philippines
rmb2105@students.uc-bcf.edu.ph, jtg6745@students.uc-bcf.edu.ph, jbm1737@students.uc-bcf.edu.ph

***Abstract*—The migration from DevOps to DevSecOps aims to address the growing needs for security in software development, but the integration of security itself into the workflow introduces new challenges that organizations often struggle to overcome. This study aimed to explore DevSecOps to gain a comprehensive understanding on the impact of its implementation as a security-integrated approach to software development. Through qualitative research design and Systematic Literature Review (SLR), existing studies related to the objectives were gathered and analyzed. In hand with this, using Thematic Analysis, the data was examined and patterns were extracted related to common barriers to successful DevSecOps adoption, its advantages, and strategies to mitigate the identified challenges. The findings revealed twelve recurring barriers across key aspects of software development, which are centered around people, processes, and technology. On the other hand, eight advantages were outlined, highlighting the ability of DevSecOps to reduce vulnerabilities and risks and improve the robustness of the system. Finally, five general strategies were found, emphasizing the role of automation in every strategy to improve security integration. These results highlight that the implementation of DevSecOps brings both advantages and disadvantages that must be considered. Therefore, its impact is dependent on how organizations adopt, implement, and utilize strategies to ensure a seamless and successful transition to DevSecOps.**

***Index Terms*—Agile software development, software development methods, software development process management, systems security, security and privacy.**

I. Introduction

Nowadays, modernization has pushed organizations to adopt advanced technologies in order to stay competitive and meet increasing demands for efficiency and innovation. This shift has transformed how businesses operate, with a growing reliance on digital tools and automated processes to streamline workflows and deliver products and services effectively. As software becomes an essential component of business operations, organizations must adopt development methodologies that enable them to build, deploy, and maintain systems efficiently. One such solution is Agile methodologies, which promote a flexible and iterative approach to software development, allowing faster software releases [1]. While Agile methodologies enhance development speed and adaptability, they fail to fully address the challenges of managing development and operations [2].

This limitation led to the emergence of DevOps, a critical software development technique that bridges the gap between development (Dev) and operations (Ops) teams to enhance collaboration, automate processes, and accelerate software delivery. Research demonstrates that DevOps adoption significantly improves productivity, reduces time-to-market, and enables organizations to respond rapidly to changing market needs [3]. For this reason, DevOps has become a cornerstone of modern agile software development [4]. However, a major drawback of DevOps is its tendency to prioritize speed over security until the later stages of the development lifecycle. Studies have identified misconfigurations, insecure code, and exposure to cyber threats as common risks in DevOps environments, often resulting from delayed security integration [4].

To address this issue, DevSecOps was introduced as a paradigm that integrates security practices into every phase of the DevOps lifecycle [5]. By embedding security into planning, development, testing, and deployment processes, DevSecOps aims to overcome the security limitations of DevOps while maintaining agility and efficiency. Nevertheless, integrating security practices into DevOps introduces certain trade-offs that may significantly alter development workflows. Unlike DevOps, which primarily emphasizes speed and delivery, DevSecOps incorporates tasks such as security assessments and compliance verification that may slow down development if not implemented effectively. This shift in workflow

may impact several key aspects of software development, including OPC (organization, people, culture), process capabilities, technology, and business strategy [6], [7].

Although DevSecOps provides a potential solution to the limitations of DevOps, it also introduces complexities that organizations may find difficult to manage. Many organizations struggle to successfully implement security practices while maintaining efficiency during the transition [7], [8]. By identifying these complexities, organizations can develop informed strategies that balance security with agility and ensure the development of secure and robust software systems.

For these reasons, this research aims to conduct a *Systematic Literature Review (SLR)* on DevSecOps to obtain a comprehensive understanding of the impact of implementing DevSecOps as a security-integrated approach to software development. Furthermore, the study seeks to identify both the advantages and disadvantages of DevSecOps in order to provide insights into more effective implementation strategies.

Specifically, the study addresses the following research questions:

1) What are the common barriers that prevent organizations from achieving successful DevSecOps implementation?
2) What are the key advantages that DevSecOps offers to organizations?
3) What strategies can be employed to address the challenges of successfully implementing DevSecOps?

II. Methodology

This study employed a qualitative research design using a *Systematic Literature Review (SLR)* as the primary research method. This systematic approach allowed for an extensive examination of existing literature to obtain valuable insights on DevSecOps and develop a comprehensive understanding of the topic in relation to the research questions.

The primary databases used to identify relevant literature included ACM Digital Library, IEEE Xplore, Springer, and ScienceDirect. These databases were selected due to their extensive collections of studies related to DevOps and DevSecOps.

To ensure the relevance and quality of the selected studies, specific inclusion criteria were established. The reviewed papers had to focus on DevSecOps or security practices within DevOps environments. Additionally, publications were limited to journal articles or conference proceedings published between 2019 and 2025. To ensure credibility, studies from predatory journals were excluded, and only publications written in English and available in full text were included in the review.

The main keywords used for the literature search included "DevSecOps" and "Secure DevOps" across all objectives. Additional keywords such as "challenges" OR "barriers," "advantages" OR "benefits," and "strategies" OR "solutions" were applied specifically to address the first, second, and third research objectives respectively.

After applying the filtering process, a total of 68 studies were selected for analysis. The researchers followed a structured process to organize and review the collected data thoroughly. Using the *Thematic Analysis Method*, the selected studies were systematically examined by identifying and labeling relevant features aligned with the research objectives. These features were then coded, analyzed for potential thematic patterns, and grouped into broader themes.

The application of thematic analysis provided a structured and systematic approach for examining the literature, enabling an in-depth exploration of the research domain. This process allowed the study to effectively evaluate the impact of integrating security into software development practices and address the research questions.

III. Findings and Discussions

A. Common Barriers to DevSecOps Implementation

To identify the common barriers that prevent organizations from successfully implementing DevSecOps, the researchers conducted a literature search using the keywords "DevSecOps," "Secure DevOps," "challenges," and "barriers" across several academic databases. The initial search yielded 875 results. After screening the studies for relevance to the research objective and applying the defined inclusion criteria, the number of selected papers was reduced to 27.

To ensure the reliability of the findings, only themes that appeared across multiple studies were considered. This approach ensured that the identified challenges represent widely recognized barriers in DevSecOps adoption.

The recurring themes identified from the analyzed studies are summarized in Table I.

TABLE I
THEMES OF THE COMMON BARRIERS TO DEVSECOPS IMPLEMENTATION

Themes	**Description**	**Citations**
Balancing Speed and Security	The extensive security measures add up to additional time that causes delays in deployment cycles.	[7] [9] [10] [11] [12] [13]
Resistance to Change	The migration to DevSecOps faces resistance because of the cultural shift required, where security becomes a responsibility shared by development and operations teams. Developers and operators who are accustomed to traditional methods may become reluctant to adopt the changes.	[8] [10] [14] [15] [16] [17] [19] [20] [21] [22]
Knowledge and Skills Gap	Many engineers are outdated or lack the necessary expertise to integrate security measures effectively into the development pipeline.	[7] [8] [10] [15] [16] [18] [21] [22] [23] [24]
Security Tools	The complexity of security tools combined with insufficient training of the team often results in tool selection and integration challenges.	[7] [10] [15] [16]
Complexity of Automation	The automation of security processes introduces an additional layer of complexity to workflows, especially when integrating advanced technologies such as AI/ML models that require specialized expertise.	[10] [21] [25]
Integration to Complex Infrastructure	Integrating security into complex infrastructures may create technical integration challenges and increase financial costs that demotivate organizations.	[7] [15] [26]
Team Collaboration	Developers and security teams often have conflicting priorities along with unclear communication, which can lead to instability in security implementations.	[7] [8] [14] [15] [17] [27] [28] [29]
Unrestricted Collaborations	Excessive collaboration can sometimes increase security risks because more people have access to sensitive information.	[30] [31]
Compliance Issues	Ensuring security measures while adhering to compliance requirements and regulations makes integration of security practices difficult, particularly when policies are inconsistent or poorly defined.	[8] [15]
Resource Constraints	Migration to DevSecOps requires significant financial and human resources, along with system modifications that may cause financial strain.	[7] [17] [22] [26]
More Complex Development Process	Multiple security measures introduced during development increase process complexity.	[13] [23]
Lack of Planning	Many organizations implement DevSecOps without a well-defined strategy and standardized methodology, which increases the risk of disruptions and inefficiencies in development.	[32] [33]

The findings reveal several themes that hinder successful DevSecOps adoption. Resistance to Change emerges as a primary barrier as integrating security into the workflow requires a significant cultural shift that teams accustomed to traditional methods tend to resist [8], [10], [14], [17]. This problem is further reinforced by the steep learning curve teams struggle to overcome, as found in the theme Knowledge and Skills Gap. They are found to lack expertise in security practices [15], secure coding standards [8], or the operation of advanced security tools [7]. This makes Balancing Security and Speed a major challenge, especially when additional security measures must be taken into account. As emphasized by Desai and Nisha [2021], DevOps emphasizes agility and delivery, but the integration of security into these practices introduces delays. The excessive security scans and additional automated processes to ensure security compromise deployment speed [12], [13]. Along with the integration of Sec into DevOps comes the Team Collaboration issues, often resulting from conflicting priorities of the security and DevOps teams [17], [29]. While the promotion of collaboration is indeed crucial for team management, Unrestricted Collaboration may rather expose sensitive information and increase security risks when teams collaborate too closely [30], [31].

Moreover, with an overwhelming number of options and a lack of expertise to make informed choices, organizations often struggle with the selection and integration of Security Tools. As addressed by several authors [7], [15], [16], selecting and configuring the right tools can get complex, especially for teams that lack training. Similarly, the Complexity of Automation can become technically challenging, especially in environments handling large volumes of data [10]. For organizations operating in cloud-native or hybrid infrastructures, Integration into Complex Infrastructure introduces both technical and financial challenges. This contributes to Resource Constraints, which are particularly evident for smaller organizations with limited budgets and personnel [11], [17].

Organizations also face Compliance Issues, as considering the adherence to security standards and regulatory requirements, while in the process of a cultural shift, can restrict the integration of security [15]. Additionally, the multiple security measures needed for the integration increase the effort required for security validation, leading to a More Complex Development Process [13], [23]. Finally, many implementations fail due to Lack of Planning, where pipelines that are built rapidly without proper engineering can lead to disruptions and inefficiencies that

may happen during the development process [32], [33]. These findings reveal that the barriers to the successful transition to DevSecOps are not isolated but are deeply intertwined, which affects one another if not resolved. Ultimately, these persistent barriers identified suggest that while DevSecOps provides a strong framework for security integration, it requires organizations to consider strategic plans to overcome these barriers and ensure success in implementation.

B. Advantages of adopting DevSecOps

The table below (Table 2) provides an overview of the positive impacts associated with adopting DevSecOps. To address the study's second objective, which focuses on evaluating these benefits, only 27 papers from 662 results were analyzed. These papers were identified using keywords such as "Secure DevOps," "DevSecOps," "advantages," and "benefits." Table II outlines the key themes derived from the reviewed literature, along with their descriptions.

Despite the challenges associated with its adoption, research indicates that the implementation of DevSecOps also offers several advantages by integrating security throughout the development process. One of the most cited benefits is how it leads to More Robust Systems. By embedding security early and into every stage of development, organizations can detect vulnerabilities sooner [18], [49], reduce risks [11], [22], and build stronger infrastructures [24], [26]. Not only does it reduce the risks of breaches, but it also introduces automated self-healing mechanisms [24] that address any detected issues without manual intervention. The reduction of delays as a result of automation leads to Operational Efficiency. As noted by Qasim and Bilal [2024], the automation of security practices minimizes any repetitive tasks, which results in faster detection of security issues, reduced rework, and improved productivity [12], [41]. Consequently, DevSecOps supports Faster Software Delivery, with the reduction of deployment delays reaching 40% [12].

Another important advantage is Enhanced Collaboration due to having a shared responsibility. The breaking down of traditional silos enforces collaborative culture, which speeds up feedback cycles and enhances deployment speed [36], [39], [48]. Moreover, by detecting issues earlier, DevSecOps helps in Cost Efficiency. Since issues are detected sooner, fixing them during development is considerably more cost-effective than resolving them after. The automation also helps in continuously fixing security problems during the development process [43]. Beyond technical efficiency, the agile method also supports Compliance Efficiency due to security controls in the pipeline that help accelerate adherence to regulatory requirements [18], [22], [49]. Automating compliance checks and embedding them directly into the development pipeline not only simplifies how organizations meet security regulations and industry standards but also helps them avoid penalties, reduce risks, and maintain trust with customers without slowing down development. Subsequently, DevSecOps helps companies to strengthen their Reputation Management by demonstrating security commitment to stakeholders and creating an edge over competitors through faster and more secure releases. As noted by Mao et al. [2020], a strong emphasis on security earns organizations customer trust and satisfaction, which is an essential element in a competitive market. Combining evidence from these studies, it can be said that adopting DevSecOps transforms traditional development by aligning speed, security, resources, collaboration, and compliance, which all provide value for both technical and business improvement.

C. Strategies for Implementing DevSecOps

In identifying the strategies that can be employed to address the challenges of implementing DevSecOps, the researchers conducted a literature search using the keywords "DevSecOps," "Secure DevOps," "strategies," and "solutions" to look for relevant literature. From the initial set of 789 results retrieved from academic databases, 55 research papers were relevant and were subsequently analyzed in relation to the third objective. The recurring themes under the strategies are summarized in Table III.

TABLE II
THEMES OF THE ADVANTAGES OF ADOPTING DEVSECOPS

Themes	**Description**	**Citations**
More Robust Systems	Embedding security practices throughout the development cycle enables organizations to detect vulnerabilities earlier and mitigate risks effectively.	[11] [18] [22] [24] [26] [34] [35] [40] [42] [43] [44] [46] [48] [50]
Reputation Management	This protects organizational credibility by preventing breaches and adherence to regulatory standards, which not only avoids financial penalties but also builds the trust of stakeholders.	[10] [17] [21] [36]
Operational Efficiency	The adoption optimizes workflow through automation, which eliminates manual bottlenecks, reduces human error, and allows faster delivery cycles.	[12] [16] [38] [41] [42] [45]
Faster Software Delivery	Integrating security checks creates a balance between speed and security, ensuring rapid releases without compromising safety.	[12] [16] [22] [43]
Enhanced Collaboration	Through shared tools, cross-training, and cultural alignment, different teams foster a unified approach to security.	[36] [39] [48] [50]
Cost Efficiency	Expenses are reduced through proactive security measures and automation.	[18] [37] [41] [43] [47] [50]
Compliance Efficiency	Adhering to regulatory requirements is simplified by automation and real-time monitoring.	[18] [22] [49]
Competitive Advantage	This enables companies and organizations to deliver secure, high-quality software faster than traditional approaches, which drives customer trust and market differentiation.	[17] [21]

TABLE III
THEMES OF THE STRATEGIES FOR IMPLEMENTING DEVSECOPS

Themes	**Description**	**Citations**
Automation and Integration	Automation and integration enable early vulnerability detection, consistent security measure implementation, and streamlined processes. These include the integration of proper automation into pipelines to ensure continuous security testing and seamless workflows.	[5] [18] [19] [20] [22] [25] [30] [32] [34] [35] [37] [38] [46] [51] [53] [56] [57] [60]
Governance and Compliance	This theme focuses on strategies that emphasize compliance and rules as essential for aligning security practices with industry standards while reducing manual effort.	[9] [31] [34] [39]
Collaboration and Communication	Collaboration and Communication points to breaking down silos between development, operations, and security teams critical for effective DevSecOps adoption.	[14] [27] [29] [31] [33] [42] [45] [48] [50] [53] [59] [62]
Tool Selection and Optimization	This refers to selecting the right tools that are vital for streamlining security processes while maintaining agility. Strategies focus on leveraging open-source tools, ensuring compatibility with existing workflows, and adopting advanced technologies like AI/ML for real-time threat detection.	[5] [7] [19] [20] [21] [24] [26] [28] [43] [44] [47] [52] [56] [57] [58] [59] [60] [67]
Risk Management	This refers to integrating threat modeling and continuous risk assessment from the start (shift-left approach) and throughout the development lifecycle. Strategies include automating these processes to identify vulnerabilities early and adapting to evolving threats effectively.	[21] [30] [40] [47] [49] [51] [54] [55] [60] [69] [71]

To overcome challenges in the integration of security within DevSecOps, organizations can adopt several strategic solutions. As one of the most cited strategies in literature, the findings reveal that Automation and Integration is central to DevSecOps as it ensures security measures across the development process. The integration of real-time scanning tools and services helps maintain security beyond deployment, helping organizations shift from reactive to more proactive approaches. As emphasized across numerous studies [18, 20, 38, 66], automation not only enhances efficiency but also reduces human error, lowers financial costs, and speeds up regulatory compliance, which ultimately makes it an essential aspect of a successful implementation of DevSecOps. Under this theme are practices such as Security as Code (SaC) [37] and Infrastructure as Code (IaC) [63] that treat security configurations and infrastructure as programmable objects, making them easier to audit and maintain. Moreover, with the integration of automation within the pipelines comes continuous Governance and Compliance. The role of AI can be seen in the automation of compliance checks to verify adherence to standards while maintaining speed. Desai and Nisha [2021] emphasize codifying security policies to ensure that the environment is regulated and that suitable security controls are in place. These compliance standards should be regularly updated to align with the latest security frameworks. Collaboration and Commu-

nication among the teams are another critical strategy to the successful execution of DevSecOps practices. Li and Zalialetdzinau [2022] point out that in DevSecOps, security is no longer isolated to a single department but is now embedded across all teams. Essentially, a successful collaboration entails behavioral change [14], cross-training [54], and constant communication between teams [62]. Training programs and appointing "Security Champions" or security specialists are also highly suggested to embed security practices in the team [45, 54]. A crucial addition is the feedback loop, which places channels for continuous feedback that improve coordination and address security challenges at the same time [33]. When teams are properly trained and collaborate well, they are better equipped for selecting the appropriate tools. Under the theme of Tool Selection and Optimization, studies recommend that the tools should be automated [44, 47, 52], up-to-date [5], and suitable to the infrastructure [7, 26] to be effectively utilized in rapid development cycles. Using the proper tools should be taken into consideration to streamline the processes while maintaining agility. Many studies also highlight the integration of Artificial Intelligence (AI) into tools to make it more efficient, though it still has to overcome ethical and adoption challenges as noted by Pakalapti et al. [2023]. Fundamental to every other solution is Risk Management, which involves integrating security considerations throughout the development lifecycle. A widely cited strategy under this theme is the shift-left approach, where security is incorporated early in the development [30, 49, 54]. Complementing this is continuous risk assessment, which automatically and regularly updates security measures to evolving threats [40, 55, 60]. Techniques like threat modeling further strengthen risk management by identifying potential vulnerabilities during the planning phase to ensure a robust foundation [51, 54, 55, 69]. Given these points, AI is essentially a core strategy in DevSecOps to enable efficient and more secure processes. These findings highlight that the implementation of DevSecOps requires a holistic transformation as its benefits can only be realized when supported by cultural, procedural, and governance changes.

IV. CONCLUSION

The study aimed to conduct a literature review to gain a comprehensive understanding of the impact of DevSecOps as a security-integrated paradigm to software development. Based on the findings, it can be concluded that the effect of the adoption is neither inherently positive nor negative. Rather, its success largely depends on how it is adopted and sustained within an organization. Its effectiveness is deeply influenced by the practices employed, as proper implementation strategies are needed to leverage the positive effects of DevSecOps on software development. This study was able to examine the common barriers, advantages, and strategies of adopting DevSecOps, focusing on its ability to provide security throughout the DevOps lifecycle while making sure operations are still running efficiently. The findings revealed twelve barriers that are deeply interconnected rather than isolated. These related issues disrupt collaboration, slow down deployment and delivery, and increase the risk of insecure products. With barriers identified, such as Team Collaboration, Resistance to Change, and Security Tools, the results imply that DevSecOps is not just simply a methodical process, but rather a more complicated approach that entails holistic changes in people, processes, and technology. Despite the multifaceted challenges, the study highlights eight advantages organizations experience during the adoption. With a more secure posture, the integration of early and continuous security enables faster detection of vulnerabilities, fewer issues to mitigate post-release, and overall, more robust systems. These results demonstrate that when integrated properly, it creates synergy between speed, security, collaboration, and business value. For a more effective implementation, the study identifies five general themes for strategies that can be applied to overcome the identified barriers. Results proved that the integration of automation is highly suggested in enhancing processes and reducing vulnerabilities. Moreover, cooperation and collaboration between the teams were critical factors in successfully adopting the methodology. When done right, DevSecOps not only mitigates risks but also improves compliance and lessens resource constraints. Ultimately, the impact of DevSecOps lies not in the methodology alone, but in how it is applied and how the organization utilizes available strategies to enable secure and agile development even in fast-paced environments.

V. RECOMMENDATION

The following suggestions from the methodological and objective perspective can be considered to improve the paper. From a methodological standpoint, expanding the database searches beyond the resources used can contribute to a more comprehensive literature review. Moreover, using gray literature or industry white papers, code documentation, and other technical reports can provide more insights into information overlooked in academic writing. Additionally, real-world validation by actual DevSecOps professionals through a mixed-methods approach can enhance the depth of the study. There should also be a focus on the long-term impact of implementing DevSecOps to review its efficacy. The

objectives standpoint can focus on the following topics: sector specificity and emerging technologies. The current study can be enhanced by focusing on specific industries and tailoring the DevSecOps methodology for each field. Furthermore, given the role of automation in DevSecOps as found in the results, concentrating on emerging technology, such as cloud-native architectures, quantum computing, and AI/ML in DevSecOps practices, can be valuable for future topics. To better facilitate the implementation of the DevSecOps methodology and to open opportunities for future research, the previously discussed recommendations are provided.

a) ACKNOWLEDGMENTS: We thank Mr. Efraim Jededdia Z. Pangan and Artificial Intelligence Chatbots for their support in completing this paper. With the lectures, material, and feedback provided by Mr. Pangan, we have successfully navigated the complexities of the research and refined the arguments presented within the paper. Multiple AI chatbots, namely ChatGPT, Gemini, DeepSeek, Perplexity, and Claude, have been used throughout this process. Leveraging their expansive data pools and file upload capabilities, their assistance and aid proved crucial for completing this work. Specifically, these chatbots facilitated a deeper understanding of the DevSecOps methodology and the Systematic Literature Review procedure, ultimately contributing to the rigor and comprehensiveness of our findings.

REFERENCES

[1] M. Doshi and P. Virparia, "Quality, speed, and collaboration in Agile vs. traditional models," *J. Electrical Systems*, vol. 20, no. 11, 2024.

[2] S. M. Masud, M. Masnun, A. Sultana, A. Sultana, F. Ahmed, and N. Begum, "DevOps enabled Agile: Combining Agile and DevOps methodologies for software development," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, Jan. 2022. http://dx.doi.org/10.14569/ijacsa.2022.0131131

[3] A. Wiedemann, M. Wiesche, and H. Krcmar, "DevOps in practice: A systematic literature review," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2020. https://doi.org/10.1002/spe.3096

[4] T. Leppänen, A. Honkaranta, and A. Costin, "Trends for the DevOps Security: A Systematic Literature Review," in *Proceedings of the 12th International Symposium on Business Modeling and Software Design (BMSD 2022)*, ACM, New York, NY, USA, 2022.

[5] C. Feio, N. Santos, N. Escravana, and B. Pacheco, "An empirical study of DevSecOps focused on continuous security testing," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Vienna, Austria, Jul. 8–12, 2024, pp. 610–617. http://dx.doi.org/10.1109/eurospw61312.2024.00074

[6] X. Zhao, T. Clear, and R. Lal, "Identifying the primary dimensions of DevSecOps: A multi-vocal literature review," *Journal of Systems and Software*, vol. 214, p. 112063, Aug. 2024. http://dx.doi.org/10.1016/j.jss.2024.112063

[7] R. N. Rajapakse, M. Zahedi, M. A. Babar, and H. Shen, "Challenges and solutions when adopting DevSecOps: A systematic review," *Information and Software Technology*, vol. 141, no. 6, p. 106700, Jan. 2022. http://dx.doi.org/10.1016/j.infsof.2021.106700

[8] M. A. Akbar, K. Smolander, S. Mahmood, and A. Alsanad, "Toward successful DevSecOps in software development organizations: A decision-making framework," *Information and Software Technology*, vol. 147, p. 106894, Jul. 2022. http://dx.doi.org/10.1016/j.infsof.2022.106894

[9] R. Desai and N. T. N., "Best practices for ensuring security in DevOps: A case study approach," *Journal of Physics Conference Series*, no. 4, Jul. 2021. http://dx.doi.org/10.1088/1742-6596/1964/4/042045

[10] A. Mustyala, "DevSecOps: Integrating security into the DevOps lifecycle," *ISAR Journal of Multidisciplinary Research and Studies*, vol. 1, no. 5, Nov. 2023.

[11] P. S. S. Patchamatla, "Security in DevOps: A DevSecOps Approach to Mitigating Software Vulnerabilities," *Recent Innovations in Wireless Network Security*, vol. 7, no. 2, pp. 14–16, Feb. 2025. https://doi.org/10.5281/zenodo.14921426

[12] V. Veeramachaneni, "A Systematic Review of DevSecOps: Bridging Security and Agile Development for Resilient Software Systems," *NeuroQuantology*, vol. 21, no. 7, pp. 1251–1255, Dec. 2023.

[13] A. Caniglia, V. Dentamaro, S. Galantucci, and D. Impedovo, "FOBICS: Assessing Project Security Level through a metrics framework that evaluates DevSecOps performance," *Information and Software Technology*, vol. 178, p. 107605, Feb. 2025. http://dx.doi.org/10.1016/j.infsof.2024.107605

[14] M. Sánchez-Gordón and R. Colomo-Palacios, "Security as Culture: A Systematic Literature Review of DevSecOps," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20)*, ACM, New York, NY, USA, 2020, pp. 266–269. https://doi.org/10.1145/3387940.3392233

[15] S. Afaneh, M. R. Al-Mousa, H. S. Al-hamid, B. S. AL-Awasa, M. Alia, H. Almimi, and A. A. Alkhatib, "Security Challenges Review in Agile and DevOps practices," in *2023 International Conference on Information Technology (ICIT)*, Aug. 2023. http://dx.doi.org/10.1109/icit58056.2023.10226018

[16] N. Qasim and M. Bilal, "DevSecOps: Integrating Security into IT Development and Operations," *Revista de Inteligencia Artificial en Medicina*, vol. 15, no. 1, Oct. 2024.

[17] X. Zhou, R. Mao, H. Zhang, Q. Dai, H. Huang, H. Shen, J. Li, and G. Rong, "Revisit security in the era of DevOps: An evidence-based inquiry into DevSecOps Industry," *IET Software*, vol. 17, no. 4, pp. 435–454, Jul. 2023. http://dx.doi.org/10.1049/sfw2.12132

[18] G. Bollieddula, "Challenges and Solutions in the Implementation of DevOps Tools & Security (DevSecOps): A Systematic Review," *Culminating Projects in Information Assurance*, no. 127, 2022.

[19] K. Zalialetdzinau, "Secure Change Management Process: On the Effectiveness of DevSecOps," *Journal of Computer Science and Information Technology*, vol. 10, no. 4, pp. 37–51, Dec. 2022. http://dx.doi.org/10.13189/csit.2022.100401

[20] Z. Ahmed and S. C. Francis, "Integrating security with DevSecOps: Techniques and challenges," in *2019 International Conference on Digitization (ICD)*, Sharjah, United Arab Emirates, Nov. 18–19, 2019, pp. 178–182. http://dx.doi.org/10.1109/icd47981.2019.9105789

[21] X. Ramaj, M. Sánchez-Gordón, V. Gkioulos, S. Chockalingam, and R. Colomo-Palacios, "Holding on to compliance while adopting DevSecOps: An SLR," *Electronics*, vol. 11, no. 22, p. 3707, Nov. 2022. https://doi.org/10.3390/electronics11223707

[22] R. Ticu-Jianu, "Continuous Resilience: DevSecOps Strategies for Cloud and Quantum Platforms," *Informatica Economică*, vol. 28, no. 4, pp. 63–73, Dec. 2024.

[23] V. Pendyala, "Evolution of integration, build, test, and Release Engineering into DevOps and to DevSecOps," in *Tools and Techniques for Software Development in Large Organizations: Emerging Research and Opportunities*, Jan. 2020, pp. 1–20. http://dx.doi.org/10.4018/978-1-7998-1863-2.ch001

[24] J. Alonso, R. Piliszek, and M. Cankar, "Embracing IaC through the DevSecOps philosophy: Concepts, challenges, and a reference framework," *IEEE Software*, vol. 40, no. 1, pp. 56–62, Jan. 2023. http://dx.doi.org/10.1109/ms.2022.3212194

[25] N. Pakalapati, S. Venkatasubbu, and S. M. Sistla, "The convergence of AI/ML and devsecops: Revolutionizing software development," *Journal of Knowledge Learning and Science Technology ISS*, vol. 2, no. 2, pp. 189–212, Aug. 2023. http://dx.doi.org/10.60087/jklst.vol2.n2.p212

[26] S. Nagasundari, P. Manja, P. Mathur, and P. B. Honnavalli, "Extensive review of threat models for DevSecOps," *IEEE Access*, vol. 13, pp. 45252–45271, Mar. 2025. http://dx.doi.org/10.1109/access.2025.3547932

[27] D. Ashenden and G. Ollis, "Putting the SEC in DevSecOps: Using social practice theory to improve secure software development," *New Security Paradigms Workshop 2020*, pp. 34–44, Oct. 2020. http://dx.doi.org/10.1145/3442167.3442178

[28] R. N. Rajapakse, M. Zahedi, and M. A. Babar, "Collaborative Application Security Testing for DevSecOps: An empirical analysis of challenges, best practices and tool support," *arXiv.org*, Nov. 22, 2022. https://doi.org/10.48550/arXiv.2211.06953

[29] M. A. Akbar and A. A. AlSanad, "Empirical investigation of key enablers for secure DevOps practices," *IEEE Access*, vol. 13, pp. 43698–43715, 2025. http://dx.doi.org/10.1109/access.2025.3549183

[30] D. Anjaria and M. Kulkarni, "Effective DevSecOps Implementation: A Systematic Literature Review," *Cardiometry*, vol. 11, no. 4, pp. 4931–4945, Nov. 2022. https://doi.org/10.18137/cardiometry.2022.24.410417

[31] S. Rafi, W. Yu, and M. A. Akbar, "Towards a Hypothetical Framework to Secure DevOps Adoption: Grounded Theory Approach," in *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering (EASE '20)*, ACM, New York, NY, USA, 2020, pp. 457–462. https://doi.org/10.1145/3383219.3383285

[32] T. Scanlon and J. Morales, "Revelations from an Agile and DevSecOps Transformation in a Large Organization: An Experiential Case Study," in *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering (ICSSP '22)*, ACM, New York, NY, USA, 2022, pp. 77–81. https://doi.org/10.1145/3529320.3529329

[33] J. Díaz, J. E. Pérez, M. A. Lopez-Peña, G. A. Mena, and A. Yagüe, "Self-service cybersecurity monitoring as enabler for DevSecOps," *IEEE Access*, vol. 7, pp. 100283–100295, Jul. 2019. https://doi.org/10.1109/ACCESS.2019.2930000

[34] L. Prates and R. Pereira, "DevSecOps practices and tools," *International Journal of Information Security*, vol. 24, no. 11, Nov. 2024. https://doi.org/10.1007/s10207-024-00914-z

[35] A. K. Sandu, "DevSecOps: Integrating Security into the DevOps Lifecycle for Enhanced Resilience," *Technology and Management Review*, vol. 6, no. 1, pp. 1–19, Feb. 2021.

[36] O. O. Abiona, O. J. Oladapo, O. T. Modupe, O. C. Oyeniran, A. O. Adewusi, and A. M. Komolafe, "The emergence and importance of DevSecOps: Integrating and reviewing security practices within the DevOps pipeline," *World Journal of Advanced Engineering Technology and Sciences*, vol. 11, no. 2, pp. 127–133, Mar. 2024. http://dx.doi.org/10.30574/wjaets.2024.11.2.0093

[37] S. Chittala, "Securing DevOps pipelines: Automating security in DevSecOps frameworks," *Journal of Recent Trends in Computer Science and Engineering*, vol. 12, no. 5, pp. 31–44, Nov. 2024. http://dx.doi.org/10.70589/jrtcse.2024.5.5

[38] K. Mittal, M. Sharma, M. Gupta, and K. Sheoran, "DevSecOps: A boon to the IT industry," *SSRN Electronic Journal*, Apr. 2021. http://dx.doi.org/10.2139/ssrn.3834132

[39] K. Boisrond, P. M. Tardif, and F. Jaafar, "Ensuring the integrity, confidentiality, and availability of IOT data in industry 5.0: A systematic mapping study," *IEEE Access*, vol. 12, pp. 107017–107045, Jul. 2024. http://dx.doi.org/10.1109/access.2024.3434618

[40] L. Prates, J. Faustino, M. Silva, and R. Pereira, "DevSecOps metrics," in *Lecture Notes in Business Information Processing*, Aug. 2019, pp. 77–90. http://dx.doi.org/10.1007/978-3-030-29608-7$_7$

[41] T. Chen and H. Suo, "Design and Practice of Security Architecture via DevSecOps Technology," in *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, Oct. 21–23, 2022, pp. 310–313. https://doi.org/10.1109/ICSESS54813.2022.9930212

[42] Dapshima, B. A., and S. K. Ahmad, "Evaluation and assessment of software security risks and vulnerabilities within the realm of secure DevOps," *International Journal for Multidisciplinary Research*, vol. 6, no. 4, Jul. 2024. https://doi.org/10.36948/ijfmr.2024.v06i04.25026

[43] G. Sanders, T. Morrow, N. Richmond, and C. Woody, "Integrating Zero Trust and DevSecOps," *Software Engineering Institute*, Feb. 2022. https://doi.org/10.1184/R1/19193099.v1

[44] E. Sermpezis, D. Karapiperis, and C. Tjortjis, "Integration of security in the DevOps methodology," in *2024 15th International Conference on Information, Intelligence, Systems & Applications (IISA)*, Chania Crete, Greece, Jul. 17–19, 2024, pp. 1–6. http://dx.doi.org/10.1109/iisa62523.2024.10786669

[45] M. Chen, B. Liang, and X. Lu, "The practice and application of a novel DevSecOps platform on Security," in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, Nanjing, China, Mar. 29–31, 2024, pp. 558–562. http://dx.doi.org/10.1109/ainit61980.2024.10581700

[46] A. Bahaa, A. Abdelaziz, A. Sayed, L. Elfangary, and H. Fahmy, "Monitoring real time security attacks for IOT systems using DevSecOps: A systematic literature review," *Information*, vol. 12, no. 4, p. 154, Apr. 2021. http://dx.doi.org/10.3390/info12040154

[47] M. Bedoya, S. Palacios, D. Díaz-López, E. Laverde, and P. Nespoli, "Enhancing DevSecOps practice with large language models and security chaos engineering," *International Journal of Information Security*, vol. 23, no. 6, pp. 3765–3788, Oct. 2024. https://doi.org/10.1007/s10207-024-00909-w

[48] M. Zaydi and N. Bouchaib, "DevSecOps practices for an agile and secure IT service management," *Journal of Management Information and Decision Sciences*, vol. 23, no. 2, pp. 134–149, Jun. 2020.

[49] K. K. Voruganti, "Implementing security by design practice with DevSecOps Shift Left Approach," *Journal of Technological Innovations*, vol. 2, no. 1, Feb. 2021.

[50] T. Li and K. Zalialetdzinau, "Attemps of scientific reflection on the role of e-learning of the future in the area of digital transformation: Nw opportunities and experiences with DevSecOps," *Futurity Research Publishing*, vol. 2, no. 4, pp. 52–63, Dec. 2022. http://dx.doi.org/10.57125/fed.2022.25.12.06

[51] J. de Kock and J. Ophoff, "Critical success factors for integrating security into DevOps environment," in *Proceedings of the 15th Dewald Roode Workshop on Information Research*. https://ifip.byu.edu/00000188-e1b8-d3db-afbf-e3bd83ff0000/drw-2023-paper-17

[52] M. Fu, J. Pasuksmit, and C. Tantithamthavorn, "AI for DevSecOps: A landscape and future opportunities," *ACM Transactions on Software Engineering and Methodology*, Jan. 2025. http://dx.doi.org/10.1145/3712190

[53] S. Tatineni, "Compliance and audit challenges in DevOps: A security perspective," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 10, Oct. 2023. http://dx.doi.org/10.56726/irjmets45309

[54] R. Mao, H. Zhang, Q. Dai, H. Huang, G. Rong, and H. Shen, "Preliminary findings about DevSecOps from Grey Literature," in *2020 IEEE 20th Conference on Software Quality, Reliability and Security (QRS)*, Macau, China, Dec. 11–14, 2020, pp. 450–457.

[55] R. M. Czekster, "Continuous risk assessment in secure DevOps," *arXiv.org*, Sep. 2024. https://doi.org/10.48550/arXiv.2409.03405

[56] B. S. Akula, "Vulnerability management in DevSecOps," *IRJMETS*, vol. 6, no. 4, Apr. 2024.

[57] S. Yautsiukhin, G. Dupont, A. Ginis, G. Iadarola, S. Fagnano, F. Martineli, C. Ponsard, A. Legay, and P. Massonet, "Product incremental security risk assessment using DevSecOps practices," in *Lecture Notes in Computer Science*, Feb. 2023, pp. 666–685. http://dx.doi.org/10.1007/978-3-031-23641-7$_3$2

[58] N. A. Bernardino, B. Sequeira, E. Piza, F. Henriques, F. Neves, and C. I. Reis, "Enhancing DevSecOps: Three custom tools for continuous security," in *2024 IEEE 11th International Conference on Cyber Security and Cloud Computing (CSCloud)*, Shanghai, China, Jun. 28–30, 2024, pp. 53–58. https://doi.org/10.1109/CSCloud62866.2024.00017

[59] N. O. Omoike, "DevSecOps in AWS: Embedding security into the heart of DevOps practices," *International Journal of Science and Research Archive*, vol. 13, no. 2, pp. 1309–1313, Nov. 2024. https://doi.org/10.30574/ijsra.2024.13.2.2306

[60] X. Ramaj, "A DevSecOps-enabled framework for risk management of critical infrastructures," in *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings (ICSE '22)*, ACM, New York, NY, USA, 2022, pp. 242–244. https://doi.org/10.1145/3510454.3517053

[61] S. T. Makani and S. Jangampeta, "DevOps security tools: Evaluating effectiveness in detecting and fixing security holes," *International Journal of DevOps*, vol. 1, no. 2, Jul. 2021. https://iaeme.com/Home/issue/IJDO?Volume=1Issue=2

[62] A. Parashar, A. Dwivedi, A. Kumar, and A. A. Khan, "DevSecOps: A case study on a sample implementation of devsecops," *International Journal of Engineering Applied Sciences and Technology*, vol. 5, no. 2, pp. 156–158, Jun. 2020. http://dx.doi.org/10.33564/ijeast.2020.V05i02.022

[63] H. Yasar and S. E. Teplov, "DevSecOps In Embedded Systems: An Empirical Study Of Past Literature," in *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22)*, ACM, New York, NY, USA, Art. 155, pp. 1–6, 2022. https://doi.org/10.1145/3538969.3544451

[64] H. Haverinen, T. Päivärinta, J. Vänskä, and H. Joutsijoki, "Information-centric adoption and use of standard compliant DevSecOps for operational technology: From experience to design principles," in *Lecture Notes in Business Information Processing*, Feb. 2024, pp. 400–415. http://dx.doi.org/10.1007/978-3-031-53227-6$_2$8

[65] N. Tomas, J. Li, and H. Huang, "An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps," in *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Oxford, UK, Jun. 3–4, 2019, pp. 1–8. https://doi.org/10.1109/CyberSecPODS.2019.8884935

[66] B. Gajbhiye, A. Aggarwal, and S. Jain, "Automated Security Testing in DevOps environments using AI and ML," *International Journal for Research Publication and Seminar*, vol. 15, no. 2, pp. 259–271, Jun. 2024. http://dx.doi.org/10.36676/jrps.v15.i2.1472

[67] M. Cankar, N. Petrovic, J. P. Costa, A. Cernivec, J. Antic, T. Martincic, and D. Stepec, "Security in DevSecOps: Applying Tools and Machine Learning to Verification and Monitoring Steps," in *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion)*, ACM, New York, NY, USA, 2023, pp. 201–205. https://doi.org/10.1145/3578245.3584943

[68] A. Sadovykh and V. Ivanov, "Enhancing DevSecOps with continuous security requirements analysis and testing," *Computer Research and Modeling*, vol. 16, no. 7, pp. 1687–1702, Nov. 2024. https://doi.org/10.20537/2076-7633-2024-16-7-1687-1702

[69] J. A. Morales, T. P. Scanlon, A. Volkmann, J. Yankel, and H. Yasar, "Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment," in *Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES '20)*, ACM, New York, NY, USA, Art. 63, pp. 1–8, 2020. https://doi.org/10.1145/3407023.3409186

[70] M. A. Akbar, S. Rafi, S. Hyrynsalmi, and A. A. Khan, "Towards People Maturity for Secure Development and Operations: A vision," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering (EASE '24)*, ACM, New York, NY, USA, 2024, pp. 528–533. https://doi.org/10.1145/3661167.3661238

[71] T. Okubo and H. Kaiya, "Efficient secure DevOps using process mining and Attack Defense Trees," *Procedia Computer Science*, vol. 207, pp. 446–455, Oct. 2022. https://doi.org/10.1016/j.procs.2022.09.079