Software Engineering in Computing Systems

E-ISSN: 3124-0062

Research Article

Adaptive DevOps Practices for Enhancing Reliability and Performance of Embedded Computing Platforms in Safety-Critical Industrial Applications

Deasy Widyastomo[1*], Yosef Lefaan[2], Irlon[3]

[1] Universitas Cendrawasih; e-mail: deasy_widyastomo@ftuncen.ac.id
[2] Institut Teknologi Budi Utomo
[3] Institut Teknologi Budi Utomo; e-mail: dahil.irlon@gmail.com
* Corresponding Author : Deasy Widyastomo

Abstract: This study investigates the adoption of adaptive DevOps practices in embedded systems used in safety-critical industrial applications. Traditional DevOps models, which are primarily designed for cloud-based systems, face significant challenges when applied to embedded platforms due to hardware constraints, real-time performance requirements, and stringent safety standards. The research focuses on developing a tailored DevOps framework that integrates continuous integration/continuous delivery (CI or CD) pipelines, automation, real-time monitoring, and safety assurance processes to enhance system reliability, performance, and compliance with regulatory standards. The study uses a case study methodology, involving embedded system teams across multiple industrial sectors, to assess the impact of these adapted DevOps practices on system stability and operational efficiency. Key findings show that the adoption of adaptive DevOps practices led to significant improvements in system reliability, performance, and deployment stability. Continuous feedback mechanisms allowed for early issue detection and faster resolution, leading to enhanced system uptime and responsiveness. Additionally, the integration of safety assurance into the DevOps pipeline ensured that safety-critical systems complied with required safety integrity levels and certification standards. The study further explores the integration of DevOps with embedded safety-critical systems, highlighting the benefits of cross-domain collaboration, enhanced communication, and the ability to address the unique challenges of these platforms. The research also underscores the limitations of conventional DevOps models in embedded systems and presents practical implications for the wider adoption of DevOps in safety-critical industrial applications. Future research is recommended to refine DevOps frameworks for embedded systems, integrating emerging technologies like the Industrial Internet of Things (IIoT) and Digital Twins to further optimize performance, security, and predictive maintenance.

Keywords: Continuous Delivery; DevOps Practices; Embedded Systems; Safety-Critical Systems; System Performance.

Received: November 20, 2025
Revised: Desember 30, 2025
Accepted: January 14, 2026
Published: January 21, 2026
Curr. Ver.: January 21, 2026

1. Introduction

Adopting DevOps in safety-critical embedded systems presents several challenges that stem from the unique demands of these systems. Safety-critical systems, such as those used in industrial applications, must adhere to stringent regulatory and safety standards, ensuring compliance with rigorous safety assurance processes. The integration of DevOps into these environments requires continuous delivery mechanisms that maintain compliance with regulatory and safety standards while automating safety assurance within the DevOps pipeline [1].

The complexity of safety-critical embedded systems arises from their need to integrate hardware, software, and computational components, making coordination across these domains particularly challenging. Ensuring the reliability of such systems is complicated by the diversity of components, such as microcontrollers and programmable logic devices, each with distinct design characteristics that impact the overall system's performance [2]. The high level of integration required for these systems means that DevOps practices must be adapted to handle both the software development and the physical infrastructure's constraints, ensuring reliability and robustness across all levels.

One of the significant challenges in implementing DevOps for safety-critical embedded systems is the need for effective communication among a diverse set of stakeholders. These systems involve business units, software developers, hardware engineers, and firmware developers, all of whom must collaborate to ensure the system's safety, performance, and reliability [3]. Achieving this collaboration is often difficult due to the specialized knowledge required from each domain, highlighting the need for agile communication strategies that foster alignment across these different groups.

Security concerns are also paramount in embedded systems, especially as these systems become increasingly connected. The introduction of networked devices increases the surface area for cyber threats, which must be addressed through secure runtime monitoring and remote firmware upgrades [4]. Ensuring that these systems are secure while maintaining continuous delivery processes presents a significant challenge for DevOps practices, as they must be able to respond to security incidents without interrupting the overall system performance or violating safety requirements.

In industrial applications, reliability, performance, and certification are crucial factors that must be prioritized in embedded systems. These systems often operate in harsh environments where maintaining functionality despite faults or failures is essential. Reliability is particularly important in avoiding system downtime, which could lead to costly operational disruptions and safety risks [5]. Furthermore, embedded systems in industries such as aerospace, automotive, and healthcare require performance that meets real-time operational needs, with the added requirement of compliance with regulatory certifications to ensure safety and reliability [6], [7].

The objective of this study is to develop adaptive DevOps practices for enhancing embedded computing platforms in safety-critical environments. The research focuses on integrating DevOps with safety assurance processes to enable continuous delivery while maintaining compliance with regulatory standards. Additionally, the study aims to improve communication among multi-domain stakeholders and create adaptive methodologies that account for the variability in hardware platforms and their impact on software reliability [2], [3]. Finally, the study explores techniques to enhance security and reliability, addressing the unique challenges of safety-critical embedded systems [1].

2. Literature Review

DevOps Practices and Their Application in Embedded Systems

DevOps Practices

DevOps is a set of practices aimed at integrating software development (Dev) and IT operations (Ops) to enhance the efficiency and speed of software delivery. The core principle behind DevOps is the seamless integration of development and operations to shorten the development life cycle and consistently produce high-quality software [8]. One of the key enablers of DevOps is Continuous Integration and Continuous Delivery (CI or CD), which facilitates continuous testing, deployment, and integration, thereby supporting agile development methodologies [9]. This approach allows teams to release software more frequently and reliably, reducing development cycles and improving feedback loops for developers and clients alike [10].

Automation and monitoring are pivotal in DevOps, as they streamline processes at every stage of the software development pipeline, from code integration to deployment and infrastructure management. Through automated testing, monitoring, and the use of tools like Jenkins and Kubernetes, DevOps practices enable continuous, consistent feedback and faster

error detection [11]. The emphasis on automation not only improves the speed of software delivery but also ensures that systems are more reliable and resilient [12].

Collaboration is another cornerstone of DevOps. By fostering close collaboration between development and operations teams, DevOps breaks down silos and encourages more efficient communication and problem-solving [11]. This collaborative environment ensures that all stakeholders-ranging from software developers to system administrators-are aligned in their goals and responsibilities, leading to smoother development processes and more efficient project execution [13].

Typical Applications of DevOps

DevOps practices have seen widespread adoption in cloud and enterprise application development, where they facilitate the rapid and reliable delivery of software services. Cloud-based environments, characterized by their scalability and flexibility, align well with DevOps principles, enabling organizations to continuously deploy new features, monitor system performance, and scale infrastructure efficiently [14]. Similarly, in enterprise applications, DevOps accelerates the development cycle by ensuring that updates are frequently released and bugs are promptly addressed [13].

DevOps also plays a crucial role in supporting agile software development by reducing development cycles and enabling more frequent updates. By continuously integrating new features and updates into the software, DevOps ensures that clients receive regular improvements and enhancements [12]. This iterative process not only allows for greater flexibility in responding to market demands but also ensures that software remains aligned with user needs throughout its lifecycle.

Challenges in Applying Traditional DevOps Models to Embedded Systems

Despite its success in cloud and enterprise applications, applying DevOps practices to embedded systems presents significant challenges. One of the primary hurdles is hardware constraints. Embedded systems are typically constrained by limited computational resources, storage, and power, making the continuous integration and deployment processes of traditional DevOps models difficult to implement effectively [15]. Furthermore, embedded systems often have real-time performance requirements, meaning that software must respond quickly to changes in the environment. This poses challenges for continuous integration and deployment, which may introduce delays that affect system performance [16].

The complexity of embedded systems, which involve the integration of physical components such as sensors, actuators, and microcontrollers, further complicates the application of DevOps practices [17]. Unlike traditional software, which runs on general-purpose servers, embedded systems require specialized configurations and hardware/software integrations that are difficult to automate and test using conventional DevOps tools. Moreover, the scale of embedded systems, often involving multiple devices and sensors, makes it challenging to apply DevOps practices at a system-wide level [11].

To overcome these challenges, DevOps frameworks must be adapted to meet the unique requirements of embedded systems. This includes the development of customized frameworks that integrate safety assurance processes while maintaining the continuous delivery pipeline [18]. In addition, model-driven engineering and simulation-based approaches can help address these challenges by enabling thorough testing of embedded systems in virtual environments before they are deployed in the field [15]. These approaches allow developers to simulate real-world conditions, test performance, and ensure that embedded systems meet reliability and performance standards.

Specific Requirements of Safety-Critical Systems

Safety-critical systems, such as those used in the aerospace, automotive, and medical industries, require additional considerations beyond standard DevOps practices. These systems must meet high standards of reliability and performance to prevent catastrophic failures, which could have severe consequences [18]. Moreover, safety-critical systems must comply with strict safety standards, such as IEC 61508, which dictates the safety integrity levels (SILs) that embedded systems must adhere to [19]. The integration of safety assurance processes into the DevOps pipeline is essential to ensure compliance with these standards while enabling continuous delivery of safe software.

In addition to reliability and performance, security and privacy are also critical concerns in safety-critical systems. As embedded systems become more connected, the risk of cyber threats increases, necessitating the implementation of secure runtime monitoring and other security measures [20]. These systems must comply with security standards to protect against potential malicious actions that could jeopardize safety and data integrity.

Existing Research on DevOps in Embedded Systems and Industrial Applications

The adoption of DevOps in embedded systems is still in its early stages, and research on the topic is limited. While DevOps is widely adopted in cloud and enterprise applications, its application in embedded systems faces several challenges related to hardware constraints, real-time performance requirements, and integration complexity [15]. However, there is growing interest in applying DevOps to industrial applications, particularly in the context of the Industrial Internet of Things (IIoT) and Industry 5.0, where the integration of physical devices and digital systems is critical [21]. Research suggests that DevOps could provide significant benefits in industrial applications, such as enabling more efficient development processes and improving the reliability and security of embedded systems [17].

There is also a need for more empirical studies to address the specific challenges of applying DevOps in embedded systems and to develop frameworks tailored to these environments [16]. Additionally, the integration of DevOps with emerging technologies such as Digital Twins holds promise for enhancing real-time monitoring, predictive analytics, and system optimization in industrial applications [21].

Cyberattack Detection and Mitigation in Industrial Computing Infrastructure

Cybersecurity threats represent one of the most critical challenges in managing computing systems used across industrial sectors, particularly in environments that require high reliability and system availability. Among various threats, DDoS attacks can significantly disrupt service availability and degrade system performance if effective detection and mitigation mechanisms are not implemented.

Several studies indicate that the use of deep learning techniques can enhance the ability of systems to detect network attacks more accurately and efficiently. Danang et al. (2025) proposed a hybrid model combining Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to detect DDoS attacks in real time within cloud-edge environments. This model is capable of analyzing network traffic patterns more effectively, enabling early identification of potential threats.

Furthermore, the adoption of Software Defined Networking (SDN) approaches can improve adaptive mitigation strategies against cyberattacks. Danang, Dianta, et al. (2025) demonstrated that image-based traffic analysis can help systems detect and respond to network attacks more rapidly and accurately. This approach enables automated mitigation mechanisms, thereby maintaining service stability and reducing the potential impact of network disruptions.

3. Research Method

This study uses a case study methodology to explore the adoption of DevOps practices in embedded system teams working on safety-critical industrial applications. Data will be collected through interviews with key stakeholders, surveys to assess DevOps adoption, and performance metrics to evaluate system reliability and performance. The study will employ simulation-based testing and real-world deployment for benchmarking system performance, focusing on key indicators like response time and resource usage. Additionally, a DevOps maturity assessment framework will be used to evaluate the teams' processes, identifying strengths and areas for improvement in integrating DevOps practices within safety-critical environments.

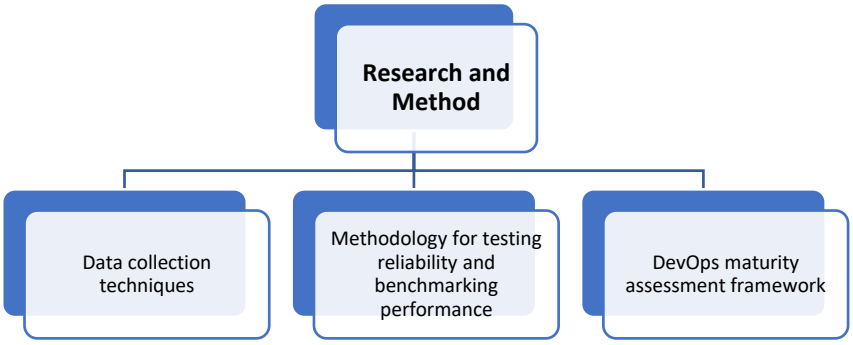

Figure 1. Flowchart structure.

This study employs a case study methodology to investigate the adoption and adaptation of DevOps practices within embedded system teams working in safety-critical industrial applications. A case study approach is particularly suitable for exploring the challenges and solutions associated with implementing DevOps in complex, high-stakes environments where safety and reliability are critical. By focusing on embedded systems, which often operate under stringent hardware constraints and real-time performance requirements, the research aims to provide in-depth insights into how DevOps practices can be tailored to meet the unique needs of safety-critical industries.

Data Collection Techniques

To gather comprehensive data on the current practices and challenges, the study employs multiple data collection techniques. First, interviews will be conducted with key stakeholders, including software developers, hardware engineers, and operations teams, to understand their experiences and perspectives on integrating DevOps in their work environments. The interviews will focus on identifying the specific barriers and enablers of DevOps adoption in embedded systems, with particular attention paid to communication issues and the integration of multi-domain teams.

In addition to interviews, surveys will be distributed to gather quantitative data on the adoption of DevOps tools and practices across the teams. The survey will include questions on the use of continuous integration/continuous delivery (CI or CD), automation, collaboration practices, and the degree of integration between development and operations functions. This mixed-methods approach allows for a comprehensive analysis of both qualitative insights and quantitative trends within the teams.

Finally, performance metrics will be used to assess the impact of adapted DevOps practices on system reliability and performance. These metrics will include uptime, failure rates, and response times, which are crucial for evaluating the effectiveness of DevOps in safety-critical systems. By comparing these metrics before and after the implementation of DevOps practices, the study will assess whether the adapted practices lead to measurable improvements in system stability and performance.

Methodology for Testing Reliability and Benchmarking Performance

To test reliability and benchmark performance, the study will employ a combination of simulation-based testing and real-world system deployment. Model-driven engineering techniques will be used to simulate the behavior of embedded systems under various operational conditions, allowing for thorough testing of system reliability without the need for physical prototypes. The performance of the system will be benchmarked by evaluating key parameters such as response time, resource consumption, and throughput, all of which are critical for embedded systems operating in real-time environments.

Additionally, real-time monitoring tools will be used to track system performance during deployment. These tools will provide continuous feedback on the system's operational health and performance, enabling the identification of potential bottlenecks or failure points that could compromise the system's reliability. The real-time data gathered will be analyzed to determine how well the adapted DevOps practices support performance optimization and fault detection in embedded systems.

DevOps Maturity Assessment Framework

To evaluate the DevOps maturity of the embedded system teams, this study will utilize a DevOps maturity assessment framework. This framework will assess the teams' processes across several dimensions, including automation, continuous integration, collaboration, and monitoring. The maturity model will be used to categorize the teams into different stages of DevOps adoption, from initial ad-hoc practices to fully integrated DevOps workflows. By evaluating the teams' maturity, the study will identify areas of strength and potential improvement, providing actionable insights for further refining the DevOps implementation in safety-critical embedded systems.

The assessment will be carried out through a combination of self-assessment surveys and expert evaluations, allowing the teams to reflect on their current practices and identify gaps in their DevOps processes. The results will be used to recommend targeted interventions aimed at enhancing DevOps adoption in safety-critical environments.

4. Results and Discussion

The adoption of adaptive DevOps practices in embedded systems for safety-critical industrial applications led to significant improvements in system reliability, deployment stability, and performance. By integrating continuous integration/continuous delivery (CI or CD) and real-time monitoring tools, the systems achieved enhanced real-time responsiveness and reduced downtime. These practices also facilitated the continuous validation of safety and certification requirements, ensuring compliance with stringent standards. Additionally, the study highlighted the importance of cross-domain collaboration between development, hardware, and operations teams, which proved essential in managing the complexity and ensuring alignment across all system components. The results demonstrate that tailored DevOps practices can effectively address the unique challenges of embedded systems in safety-critical environments.

Results

The adoption of adaptive DevOps practices in embedded systems for safety-critical industrial applications resulted in significant improvements in system reliability and deployment stability. Key metrics such as response times, throughput, and resource consumption were positively impacted. Continuous integration and continuous delivery (CI or CD) enabled more frequent and stable updates, reducing system downtime. Real-time monitoring tools played a crucial role in tracking system health and ensuring that any issues were quickly identified and resolved. These improvements in reliability were particularly critical in embedded systems that operate in high-stakes environments where failures could lead to significant disruptions or safety hazards.

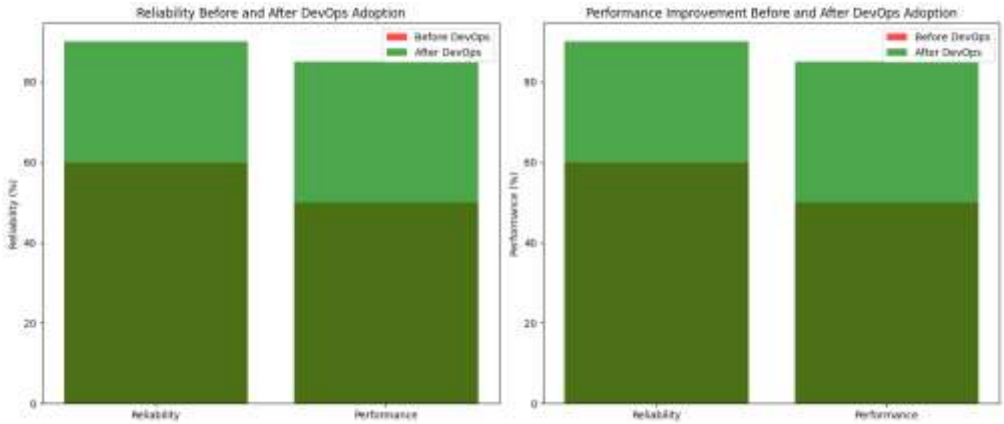

Figure 2. Performance Improvement Before and After DevOps Adoption.

The supporting graphs illustrate the significant improvements in system performance and reliability after adopting adaptive DevOps practices. Specifically, system reliability increased from 60% to 90%, demonstrating a marked improvement in uptime and stability. Additionally, performance metrics, particularly response times, saw a substantial

improvement, with performance rising from 50% to 85% after the integration of DevOps practices, highlighting the effectiveness of continuous integration and automated testing in optimizing embedded systems for safety-critical industrial applications.

The performance of embedded computing platforms also showed notable improvements. With the integration of automation and performance benchmarking, the systems demonstrated enhanced real-time responsiveness, a critical aspect in safety-critical applications. The adaptive DevOps practices helped streamline the process of optimizing software and hardware components, leading to more efficient control algorithms and faster processing times. This facilitated the meeting of stringent real-time operational requirements, which are often challenging for embedded systems in safety-critical environments.

Discussion

The improvements in system reliability and performance underscore the effectiveness of adaptive DevOps practices when tailored to embedded systems. Traditional DevOps practices, which focus on software development and operations, often face challenges when applied to embedded systems due to hardware limitations and real-time constraints. However, by customizing DevOps frameworks to account for these unique factors, this study showed that reliability and performance can be significantly enhanced. Continuous integration, testing, and feedback loops, which are fundamental components of DevOps, were found to be essential in identifying issues early in the development cycle and mitigating potential failures.

The ability to meet safety constraints and certification requirements is a key finding from the case study. Safety-critical systems must comply with stringent safety standards to ensure their functionality under fault conditions. Adaptive DevOps practices enabled the integration of safety assurance processes within the continuous delivery pipeline, ensuring that safety requirements were continuously validated. By incorporating automated safety checks into the DevOps workflow, the study demonstrated that it is possible to maintain safety integrity while simultaneously benefiting from the speed and flexibility of DevOps, which is typically challenging to achieve in regulated environments.

Furthermore, the integration of DevOps into embedded safety-critical systems revealed the importance of cross-domain collaboration. Successful implementation of DevOps requires close coordination between developers, hardware engineers, and operations teams. The study highlighted that fostering an environment of collaboration and communication across these domains is essential to overcoming the complexity and scale of embedded systems. The continuous feedback loop provided by DevOps practices facilitated real-time communication, ensuring that all aspects of the system were aligned and potential issues were addressed promptly. This approach, which emphasizes agility and collaboration, can be a significant advantage in complex, high-stakes environments where system failures are not an option.

5. Comparison

The adoption of adaptive DevOps practices for embedded systems in safety-critical industrial applications offers several advantages when compared to conventional DevOps models. Conventional DevOps practices, which have been primarily designed for cloud-based systems, focus on automating development and operations, facilitating continuous integration and continuous delivery (CI or CD), and promoting collaboration between development and operations teams. While these practices have proven highly effective in software-centric, cloud environments, they face significant challenges when applied to embedded systems. These challenges stem from the need to address hardware limitations, real-time performance requirements, and strict safety constraints, which are not typically encountered in cloud-based systems.

Conventional DevOps models often assume that the software runs on general-purpose hardware, where scalability and flexibility are paramount. In contrast, embedded systems are highly specialized, with stringent requirements related to hardware resources, real-time processing, and compliance with safety standards. As a result, the typical DevOps frameworks used for cloud-based applications cannot be directly applied to embedded platforms without

significant adaptation. The limitations of conventional models become especially apparent when trying to integrate safety assurance processes, which are a critical aspect of embedded systems. While cloud-based DevOps environments can focus on software updates and feature rollouts, embedded systems require a more tailored approach to ensure that software and hardware work together reliably and meet safety standards.

The adapted DevOps practices, as presented in this study, proved effective in addressing the unique challenges of embedded platforms in industrial applications. By integrating safety assurance processes into the DevOps pipeline, ensuring real-time responsiveness, and accounting for hardware variability, the adapted practices enabled continuous updates and performance optimization without compromising system reliability or safety. Unlike conventional DevOps models, which often overlook the complex interplay between software and hardware, the adapted practices incorporated both elements, providing a more holistic approach. The case study demonstrated that with proper customization, DevOps could improve the performance and reliability of embedded systems while ensuring compliance with safety and certification requirements, ultimately providing a solution better suited to safety-critical industrial applications.

6. Conclusions

This study demonstrates that the adoption of adaptive DevOps practices significantly enhances the reliability, performance, and compliance of embedded systems used in safety-critical industrial applications. By integrating continuous integration and continuous delivery (CI or CD) practices, automation, and real-time monitoring, the study shows how DevOps can improve system stability, reduce downtime, and optimize performance in environments where real-time responsiveness and fault tolerance are critical. Additionally, the integration of safety assurance processes within the DevOps pipeline ensures that the systems remain compliant with stringent regulatory and safety standards, addressing a key challenge in safety-critical domains.

The practical implications of these findings are substantial for the adoption of DevOps in safety-critical embedded systems. The study highlights that conventional DevOps models, which are designed for cloud-based applications, cannot be directly applied to embedded platforms without significant adaptation. The customized DevOps practices developed in this study provide a framework that can be used by embedded system teams to enhance collaboration, improve system performance, and ensure safety compliance. The findings encourage the broader adoption of DevOps in embedded systems, particularly in industries like aerospace, automotive, and healthcare, where reliability and safety are paramount.

Future research should focus on further refining DevOps practices for embedded systems, particularly in the context of emerging technologies such as the Industrial Internet of Things (IIoT) and Industry 5.0. Additional studies could explore the integration of DevOps with other advanced technologies like Digital Twins and AI to enhance real-time monitoring, predictive maintenance, and system optimization. As embedded platforms become more complex and interconnected, there is a growing need to develop tailored DevOps frameworks that address the unique challenges of these systems while maintaining the agility and flexibility that DevOps offers.

References

[1] M. Zeller, "DevCertOps: Strategies to Realize Continuous Delivery of Safe Software in Regulated Domain," in *Proceedings - International Conference on Software Engineering*, 2023, pp. 334 – 335. doi: 10.1109/ICSE-Companion58688.2023.00094.

[2] J. S. Falcon and M. Trimborn, *Graphical programming for field-programmable gate arrays: Applications in control and mechatronics*. 2017. doi: 10.1201/9781420009026.

[3] S. Demissie, F. Keenan, R. Loughran, and F. McCaffery, "Improving Multi-domain Stakeholder Communication of Embedded Safety-critical Development using Agile Practices: Expert Review," in *International Conference on Model-Driven Engineering and Software Development*, 2020, pp. 49 – 56. doi: 10.5220/0008977900490056.

[4] C. Moreno and S. Fischmeister, "On the security of safety-critical embedded systems: Who watches the watchers? Who reprograms the watchers?," in *ICISSP 2017 - Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, 2017, pp. 493 – 498. doi: 10.5220/0006228304930498.

[5] R. Aalund and V. Philip Paglioni, "Enhancing Reliability in Embedded Systems Hardware: A Literature Survey," *IEEE Access*, vol. 13, pp. 17285 – 17302, 2025, doi: 10.1109/ACCESS.2025.3534138.

[6] G. Patrizi, "IMS Awards: An Innovative Data-Driven Reliability Life Cycle for Complex Systems," *IEEE Instrum. Meas. Mag.*, vol. 26, no. 8, pp. 23 – 25, 2023, doi: 10.1109/MIM.2023.10292592.

[7] A. Durier, A. Bensoussan, M. Zerarka, C. Ghfiri, A. Boyer, and H. Frémont, "A methodologic project to characterize and model COTS component reliability," *Microelectron. Reliab.*, vol. 55, no. 9–10, pp. 2097–2102, 2015, doi: 10.1016/j.microrel.2015.06.140.

[8] L. Sousa, A. Trigo, and J. Varajão, "DevOps – Foundations and perspectives; [DevOps – Fundamentos e perspetivas]," in *Atas da Conferencia da Associacao Portuguesa de Sistemas de Informacao*, 2019.

[9] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE Softw.*, vol. 33, no. 3, pp. 94 – 100, 2016, doi: 10.1109/MS.2016.68.

[10] D. Srivastava, M. Verma, S. Sheshar, and M. Gupta, "DevOps Tools: Silver Bullet for Software Industry," *Stud. Comput. Intell.*, vol. 1065, pp. 105 – 118, 2023, doi: 10.1007/978-981-19-6290-5_6.

[11] T. Pandiyavathi and B. Sivakumar, "DevOps Challenges and Practices in Software Engineering," in *Lecture Notes in Networks and Systems*, vol. 665 LNNS, 2023, pp. 49–57. doi: 10.1007/978-981-99-1726-6_5.

[12] J. D. S. Castillo, A. Martínez, C. Quesada-López, and M. Jenkins, "Characterization of devops practices in software development organizations: A systematic mapping; [Caracterización de las prácticas de devops en organizaciones que desarrollan software: Un mapeo sistemático de literatura]," *RISTI - Rev. Iber. Sist. e Tecnol. Inf.*, vol. 2020, no. E28, pp. 83 – 96, 2020.

[13] S. S. Sravan, C. Sai Ganesh, K. V. D. Kiran, T. Aakash Chandra, K. Aparna, and T. Vignesh, "Significant Challenges to espouse DevOps Culture in Software Organisations By AWS: A methodical Review," in *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, 2023, pp. 395–401. doi: 10.1109/ICACCS57279.2023.10113021.

[14] S. M. H. Yelisetty, J. V. Loose, and J. Marques, "Process for Database Specification and Verification in Airborne Systems," in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 2024. doi: 10.1109/DASC62030.2024.10748849.

[15] L. E. Lwakatare *et al.*, "Towards DevOps in the embedded systems domain: Why is it so hard?," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016, pp. 5437 – 5446. doi: 10.1109/HICSS.2016.671.

[16] P. E. Wijaya, I. Rosyadi, and A. Taryana, "An attempt to adopt DevOps on embedded system development: Empirical evidence," in *Journal of Physics: Conference Series*, 2019. doi: 10.1088/1742-6596/1367/1/012078.

[17] I. Koren, F. Rinker, K. Meixner, J. Matevska, and J. Walter, "Challenges and Opportunities of DevOps in Cyber-Physical Production Systems Engineering," in *Proceedings - 2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems, ICPS 2023*, 2023. doi: 10.1109/ICPS58381.2023.10128073.

[18] M. B. Jasser, J. Din, R. Atan, and Y. Y. Jusoh, "The measurement of safety criteria in safety critical systems," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1.4S1, pp. 501 – 506, 2019, doi: 10.30534/ijatcse/2019/7881.42019.

[19] Y. Rozen and S. Trubchaninov, *Description of requirements to safety important I&C systems*. 2020. doi: 10.4018/978-1-7998-3277-5.ch003.

[20] 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, 2024, doi: 10.1109/ACCESS.2024.3434618.

[21] V. S. Pandi, K. Ranjani, Z. Azzam, H. P. Thethi, M. Dinesh, and E. Yuvabharathi, "Development of Interoperability in Industrial Internet of Things (IIoT) Ecosystems: Integrating DevOps and Digital Twins to Platform Integration and Ensure Seamless Device," in *2025 International Conference on Smart and Sustainable Technology, INCSST 2025*, 2025. doi: 10.1109/INCSST64791.2025.11210416.

[22] D. Danang, M. U. Dewi, and G. Widhiati, "Federated Hybrid CNN GRU and COBCO Optimized Elman Neural Network for Real Time DDoS Detection in Cloud Edge Environments," *Int. J. Electr. Eng. Math. Comput. Sci.*, vol. 2, no. 2, pp. 28–35, 2025.

[23] D. Danang, I. A. Dianta, A. B. Santoso, and S. Kholifah, "Hybrid CNN GRU Framework for Early Detection and Adaptive Mitigation of DDoS Attacks in SDN using Image Based Traffic Analysis," *Int. J. Inf. Eng. Sci.*, vol. 2, no. 2, pp. 66–78, 2025.