The Future of DevOps in Software Engineering

Onyeyili T.I, Ugwuanyi Gilbert,

# **Abstract**

One new approach that should be used in the software development life cycle is DevOps. The term "DevOps" suggests that the development and operations teams are integrated. By combining development and operations teams, DevOps has revolutionized software engineering by increasing productivity, automation, and continuous delivery. The many phases of the development lifecycle are integrated using this method. The current agile methodology has been expanded upon by DevOps. Continuous Integration, Continuous Delivery, Continuous Improvement, quicker feedback, and security are the goals of DevOps. The foundational elements of DevOps, adoption issues, models to enhance DevOps processes, and upcoming DevOps projects are all reviewed in this paper.

This study examines how DevOps will develop in the future, emphasizing new developments like cloud-native technologies, DevSecOps, AI-driven automation, and GitOps. Along with the move toward platform engineering, the growing significance of AI and machine learning in automation and predictive analytics is examined. According to the study's findings, DevOps will keep developing and provide more innovation, scalability, and security.

Keywords: DevOps, GitOps, Cloud, AI- automation, Technologies

# **1. 0 Introduction**

The procedures used throughout the software development lifecycle are crucial. Various teams will perform their respective roles at their respective levels in the traditional development lifecycle.   
Teams working independently lengthen the product life cycle and have worse communication (Lwakatare et al., 2015).

A software development methodology is a collection of steps that help create software. The end output of these efforts shows how the development process was carried out. The necessity to overcome the challenges and drawbacks of using traditional approaches in project management and implementation gave rise to agile methodology.

Shore and Warden (2021) stated that the Agile methodology ensures early and continuous delivery of software that is subject to evaluation by assuming short intervals between deliveries.

The impact on corporate culture, specifically the collaborative culture forced on developers and the ramifications of being entrenched in an Agile team, are the primary social and human aspects involved in the adoption of Agile methodology (Gregory & Taylor, 2019). According to Junker et al., two essential components of an effective Agile team are the dedication to the team's objectives and the continuous feedback given to all team members on the tasks being completed.

The DevOps (Development and Operations) movement, which strives to advance this line of thinking, originated as a result of the success of Agile methodologies. According to Luz et al. (2018) this movement aims to establish a collaborative culture between development and operations teams that will increase the flow of finished work by breaking the conventional culture of virtually minimal team interaction.

Notwithstanding its advantages, DevOps nevertheless has difficulties, such as fusing lifecycle management techniques with software engineering methodologies. More smooth integration of DevOps features into Agile approaches is necessary to address these issues.

By resolving innate inefficiencies in software development, artificial intelligence (AI) is anticipated to significantly contribute to the optimization of DevOps operations. Automation powered by AI may improve monitoring, anticipate possible errors before they happen, and strengthen continuous integration/continuous deployment (CI/CD) pipelines (Battina, 2021).

Wikipedia contributors (2025) explain that DevOps is a software development process that enhances software engineering cooperation, efficiency, and automation by fusing development (Dev) with operations (Ops). In order to facilitate continuous integration (CI), continuous delivery (CD), and continuous application monitoring, it seeks to dismantle the conventional silos that separate the development and IT operations teams.

The constant search for security, scalability, and efficiency has characterized the development of software engineering. These objectives have been greatly aided by DevOps, a paradigm that combines development (Dev) with operations (Ops) (Ebert et al., 2016). Organizations are adopting new DevOps methodologies as digital transformation picks up speed in order to satisfy growing demands for software delivery that is more dependable and efficient. This study looks at new developments that will influence DevOps in the future.

## **1.1 Fundamentals of DevOps**

**Cooperation and Communication:** Promotes shared accountability between the operations and development teams.

**Automation:** This technique makes use of tools to automate the development, testing, and deployment of software.

**CI/CD** or continuous integration and deployment, guarantees regular, incremental updates with less manual involvement.

**Monitoring & Feedback Loops:** This technique makes use of real-time logging and monitoring to swiftly identify and address problems.

**DevSecOps Security Integration:** Integrates security procedures into the DevOps workflow.

## **1.2 Advantages of DevOps**

According to Wikipedia contributors (2025), DevOps advantages includes:

**Quicker Software Releases:** By automating procedures, this approach shortens deployment timelines.

**Better Software Quality:** Ongoing testing guarantees improved performance and fewer defects.

**Scalability & Reliability:** Proactive monitoring increases system stability.

**Cost-Effectiveness:** Automation lowers downtime and resource waste.

**1.3 Frequently Used DevOps Tools**

* Ansible, Puppet, and Chef for configuration management; Jenkins, GitHub Actions, and GitLab CI/CD for continuous integration and delivery
* Datadog, ELK Stack, and Prometheus for monitoring and logging
* Containerization and Orchestration: Kubernetes, Docker

# **2.0 New Development Trends in DevOps**

New technologies, automation, and the growing need for quicker, more dependable software delivery are all driving DevOps' continuous evolution. The following are some of the most recent DevOps development trends:

## **2.1 Machine Learning and AI in DevOps**

By improving automation and predictive analytics, the combination of machine learning (ML) and artificial intelligence (AI) is transforming DevOps (Shahin et al., 2017). Large volumes of operational data can be analyzed, trends can be found, and troubleshooting can be automated with AI-driven DevOps, or AIOps. Intelligent warning systems, self-healing systems, and automated anomaly detection are decreasing the need for human intervention and increasing system dependability (Lombardi & Fanton, 2023).

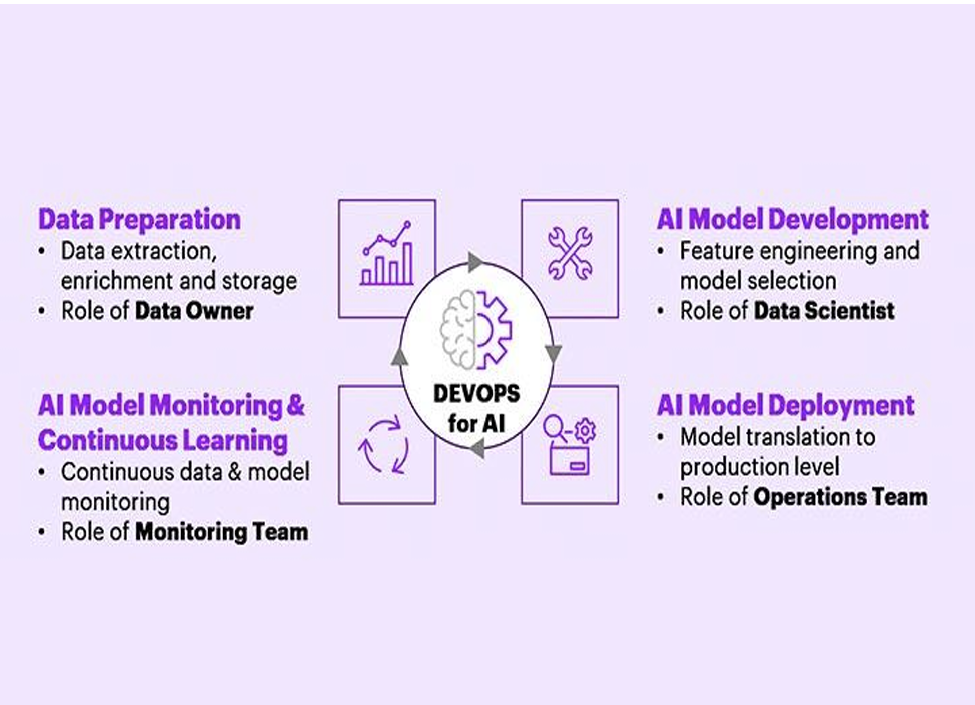


Figure 1: DevOps for AI

Source: (Battina, 2021)

DevSecOps, serverless computing, and AI-driven automation are all contributing to the evolution of DevOps. DevOps will be essential to contemporary software engineering and digital transformation as businesses continue to embrace cloud-native technology.

## **2.2 The Rise of DevSecOps**

DevSecOps emerged as a result of security becoming a key concern in DevOps. By incorporating security into the DevOps pipeline, this method guarantees automatic vulnerability scanning, compliance validation, and security checks (Myrbakken & Colomo-Palacios, 2017). Businesses are moving toward a "shift-left" security strategy, which lowers risks and compliance infractions by integrating security procedures early in the software development lifecycle.

In order to handle security issues in contemporary software development, DevSecOps developed from DevOps. For quick development cycles, traditional security models that depended on final-stage evaluations were insufficient. By integrating security early in development, DevSecOps moves security to the left (Shahin et al., 2017).

Adoption of DevSecOps is mostly driven by:

* the growing complexity of online dangers.
* the requirement to abide by industry rules.
* the need for software solutions that are secure by design.

## **2.3 Cloud-Native and Serverless Computing**

DevOps is being redefined by serverless computing and cloud-native development, which allow teams to create scalable and adaptable apps. Deployment and orchestration procedures are being streamlined by Kubernetes, Docker, and serverless platforms like AWS Lambda and Azure Functions (Mudadi & Lotriet, 2023). By enabling developers to autonomously distribute updates, these technologies enhance performance and agility by supporting microservices architecture.

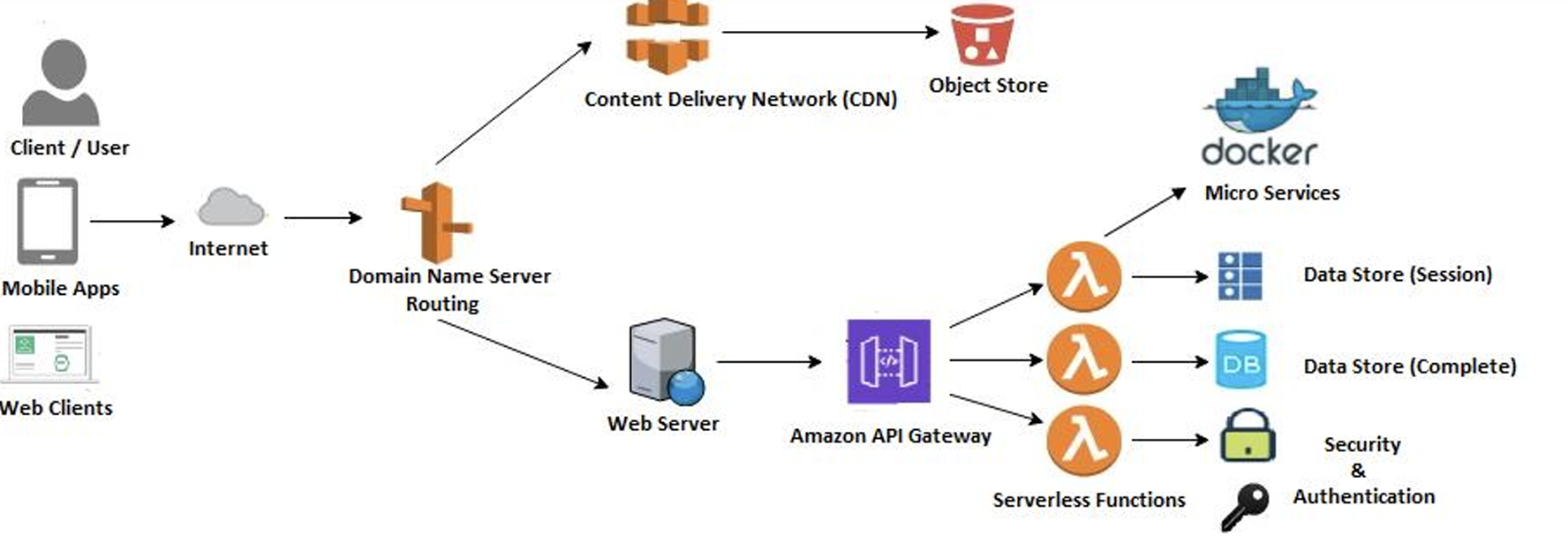


Figure 2: Serverless application architecture

Source: Venugopal and Reddy (2021)

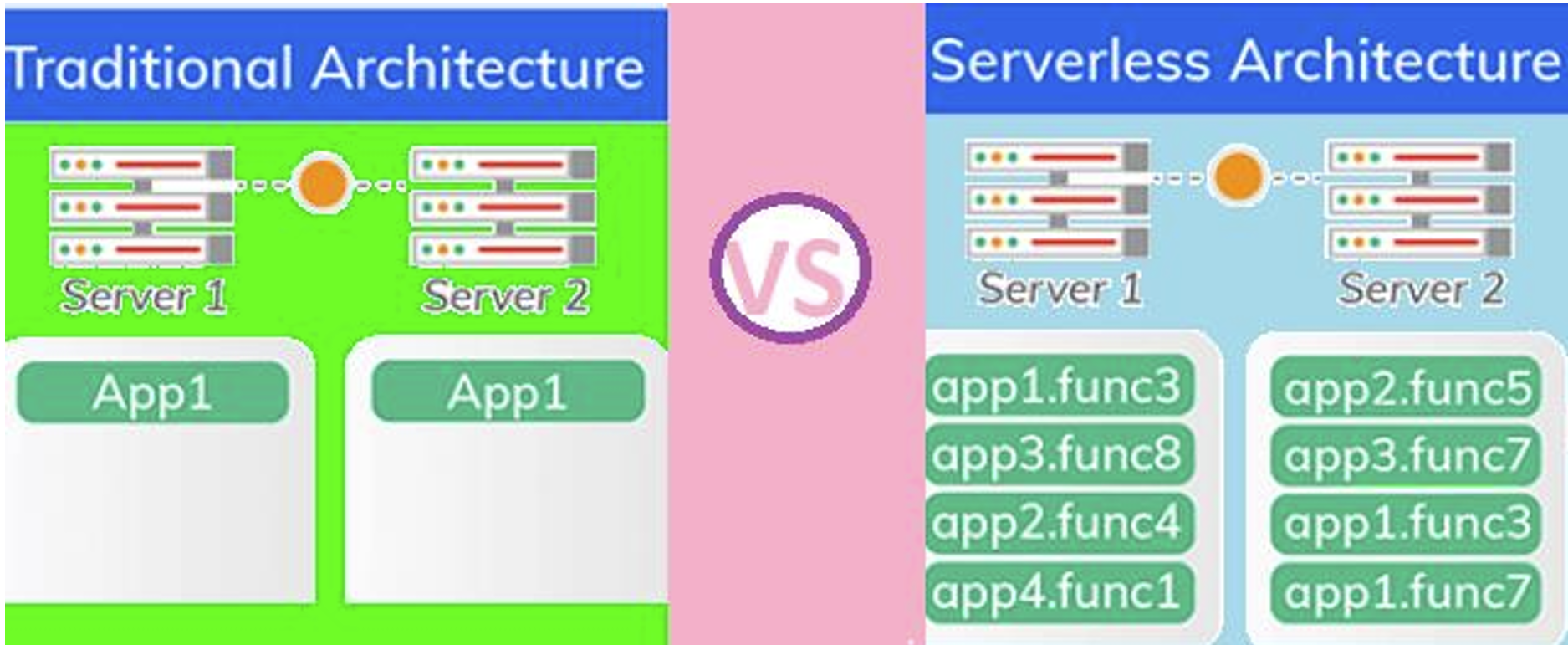
****

Figure 3: comparison of traditional and serverless architecture

Source: Venugopal and Reddy (2021)

## **2.4 GitOps and Infrastructure as Code (IaC)**

GitOps is emerging as a best practice for managing infrastructure and application deployment. By leveraging Git repositories as the single source of truth, teams can implement declarative infrastructure management and automate deployments (Shahin et al., 2017). Infrastructure as Code (IaC) tools such as Terraform and Ansible further enhance consistency and reproducibility, reducing configuration drift and operational overhead.The combination of DevOps and GitOps approaches represents a significant paradigm change in the ever changing field of software development. To enhance operational effectiveness, uphold adherence, and maximize Pipelines for Continuous Integration and Continuous Deployment (CI/CD)(Chinnam, 2024).

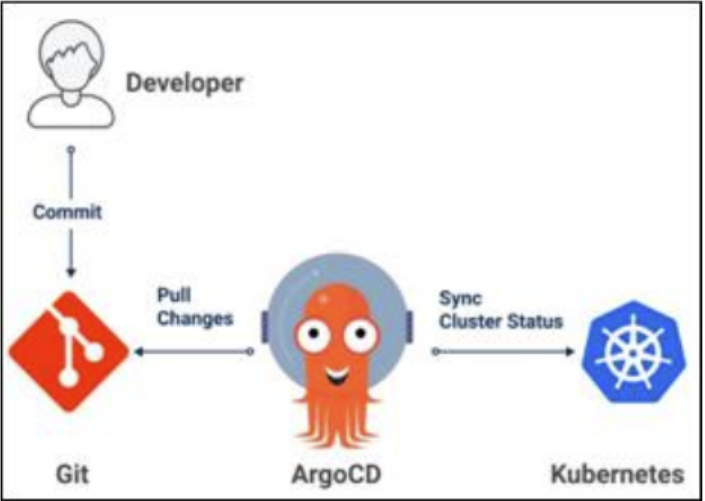


Figure 4: A simple architectural diagram of the GitOps model using Argocd

Source: (Chittibala, 2021)

Infrastructure as Code (IaC) is a DevOps practice that involves managing and provisioning infrastructure (such as servers, networks, and databases) through machine-readable configuration files, rather than manual processes. IaC allows developers and IT teams to automate the deployment, configuration, and management of infrastructure using code, making systems more scalable, consistent, and reproducible. Infrastructure-as-code (IaC) is the DevOps tactic of managing and provisioning infrastructure through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools (Guerriero et al., 2019).

## **2.5 Platform Engineering and Developer Experience**

The field of platform engineering is expanding with the goal of enhancing the DevOps developer experience. Internal developer platforms (IDPs) that offer self-service features, automation frameworks, and standardized workflows are being developed by organizations (Ebert et al., 2016). These platforms facilitate quicker onboarding, lessen cognitive burden, and increase developer productivity.

# **3.0 Challenges and Opportunities**

Although DevOps is constantly changing, organizations still encounter a number of implementation-related difficulties. But these difficulties also present fresh chances for development and creativity.

## **3.1 Challenges in DevOps**

Despite DevOps' many advantages, organizations nevertheless encounter a number of obstacles

When implementing and adopting it:

**Skill Gaps:** Constant learning and upskilling are necessary to meet the demand for proficiency in cloud-native technologies, AI, and security (Forsgren et al., 2018). Finding qualified experts who can integrate automation tools and manage DevOps pipelines is a challenge for organizations.  
**Cultural Shifts:** Implementing DevOps necessitates a change in the way an organization values cooperation, openness, and shared accountability. Adoption may be hampered by CEO buy-out, isolated teams, and resistance to change (Kim, Humble, & Debois, 2016).  
**Tooling Complexity:** It can be difficult to manage several DevOps tools and frameworks. For CI/CD, configuration management, monitoring, and security, organizations frequently employ disparate systems, which causes problems with integration and raises operational costs (Morrison, 2020).

**Security Issues:** If security is not integrated into the DevOps pipeline, quick deployments and continuous integration may reveal vulnerabilities. One of the biggest challenges is ensuring agility while adhering to security standards (Tashev, 2021).

**Scalability and dependability:** Managing massive distributed networks and preserving system dependability can become challenging as businesses grow their DevOps projects (Bass et al., 2015).

## **3.2 Opportunities in DevOps**

Despite its difficulties, DevOps gives businesses a lot of chances to improve software development and operations:  
**AI-Driven Automation:** By streamlining processes, anticipating errors, and enhancing software quality, AI and ML provide intelligent automation in DevOps. Operating hazards are decreased by self-healing systems and automated anomaly detection (Gokarna & Singh, 2020).  
**Cloud-Native DevOps:** Organizations may create scalable and robust apps while simplifying infrastructure management by implementing cloud-native technologies like Kubernetes and serverless computing (Weber, 2022).

**DevSecOps's Security-First Approach:** Organizations can lower the risk of cyber threats by minimizing vulnerabilities and ensuring compliance by including security measures early in the software development lifecycle (Tashev, 2021).

**Infrastructure as Code, or IaC,** improves efficiency and consistency in managing infrastructure across environments by enabling version-controlled, automated infrastructure deployment (Rahman et al., 2019).

**Continuous Governance and Compliance:** To guarantee regulatory compliance while preserving development speed, organizations can use automated compliance frameworks and policy-as-code (Guerriero et al., 2019).

# **4. Conclusion**

By using feedback loops, continuous testing, and real-time monitoring, DevOps adoption has demonstrated benefits in software quality. Future DevOps initiatives, however, must concentrate on incorporating DevSecOps principles, bolstering security inside CI/CD pipelines, and guaranteeing adherence to changing regulatory requirements (Perera et al., 2017).

More automation, stronger security integration, and greater adherence to Agile methodologies are key components of DevOps' future. AI and machine learning will be essential in helping companies refine their DevOps processes, which will make software development quicker, more secure, and more efficient.

Developments in AI, security integration, cloud-native development, GitOps, and platform engineering are driving the future of DevOps in software engineering. Software delivery pipelines are changing as a result of these trends, becoming more automated, secure, and scalable. In order to stay competitive in the ever changing technology landscape, firms that continue to implement DevOps must embrace these advancements.

# **References**

Almeida, F., Simões, J., & Lopes, S. (2022). Exploring the benefits of combining DevOps and Agile. *Future Internet, 14*(2), 63. <https://doi.org/10.3390/fi14020063>

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley.

Battina, D. S. (2021). *AI and DevOps in information technology and its future in the United States*. *International Journal of Creative Research Thoughts (IJCRT), 8*(1), 108-112. <https://ssrn.com/abstract=4004333>

Chinnam, S. K. (2024). GITOPS DEMYSTIFIED: A BEGINNER'S GUIDE TO STREAMLINING KUBERNETES DEPLOYMENTS. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET), 15(4), 481-494.

Chittibala, D. (2021). *GitOps: Revolutionizing configuration management in DevOps*. International Journal of Science and Research, 10(12), 1456–1462. [https://doi.org/10.21275/SR24203190625](https://doi.org/10.21275/SR24203190625%EE%88%86)

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. IEEE Software, 33(3), 94–100. https://doi.org/10.1109/MS.2016.68

Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The science of lean software and DevOps: Building and scaling high-performing technology organizations. IT Revolution.

Gokarna, M., & Singh, R. (2020). DevOps: A historical review and future works. IEEE Xplore. https://doi.org/10.1109/ICSME.2020.00092

Gokarna, M., & Singh, R. (2021, February). DevOps: A historical review and future works. *In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 366–371). IEEE. <https://doi.org/10.1109/ICCCIS51004.2021.9397204>

Gregory, P., & Taylor, K. (2019). *Defining agile culture: A collaborative and practitioner-led approach*. In *Proceedings of the IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (pp. 2). Montreal, QC, Canada, 27 May 2019. IEEE. <https://doi.org/10.3390/fi14020063>

Guerriero, M., Garriga, M., Tamburri, D. A., & Palomba, F. (2019). Adoption, support, and challenges of Infrastructure-as-Code: Insights from industry. *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 580-589. https://doi.org/10.1109/ICSME.2019.00092

Guerriero, M., Garriga, M., Tamburri, D. A., & Palomba, F. (2019). Adoption, support, and challenges of Infrastructure-as-Code: Insights from industry. 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), 580-589. https://doi.org/10.1109/ICSME.2019.00092

Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley.

Hüttermann, M. (2012). DevOps for developers. Apress.

Junker, T. L., Bakker, A. B., Gorgievski, M. J., & Derks, D. (2022). Agile work practices and employee proactivity: A multilevel study. *Human Relations*, *75*(12), 2189-2217.

Kim, G., Humble, J., & Debois, P. (2016). The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations. IT Revolution.

Kim, T., Lee, D., & Park, J. (2021). AI-powered DevOps: Automation and optimization. Journal of Software Engineering, 45(3), 122-135.

Lombardi, F., & Fanton, A. (2023). From DevOps to DevSecOps is not enough: CyberDevOps – an extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline. Software Quality Journal, 31(2), 619–654. https://doi.org/10.1007/s11219-023-09619-3

Luz, W. P., Pinto, G., & Bonifácio, R. (2018). Building a collaborative culture: A grounded theory of well-succeeded DevOps adoption in practice. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1-10). Oulu, Finland.

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of DevOps. *In International Conference on Agile Software Development* (pp. 212–217). Springer, Cham. <https://doi.org/10.1007/978-3-319-18612-2_18>

Morrison, J. (2020). Tooling challenges in DevOps: The need for an integrated approach. Software Engineering Journal, 38(2), 54-68.

Mudadi, A., & Lotriet, H. H. (2023). An analysis of DevOps’ impact on information technology organisations: A case study. The South African Journal of Industrial Engineering, 34(1), 155–167. https://doi.org/10.7166/34-1-2759

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: A multivocal literature review. In Software Process Improvement and Capability Determination (Communications in Computer and Information Science, Vol. 770, pp. 17–29). Springer. https://doi.org/10.1007/978-3-319-67383-7\_2

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: A multivocal literature review. In *Software Process Improvement and Capability Determination* (Vol. 770, pp. 17-29). Springer. <https://doi.org/10.1007/978-3-319-67383-7_2>

Perera, P., Silva, R., & Perera, I. (2017, September). Improve software quality through practicing DevOps. In *2017 seventeenth international conference on advances in ICT for emerging regions (ICTer)* (pp. 1-6). IEEE. DOI: [10.1109/ICTER.2017.8257807](https://doi.org/10.1109/ICTER.2017.8257807)

Rahman, A., Mahdavi-Hezaveh, R., & Williams, L. (2019). Where are the gaps? A systematic mapping study of Infrastructure as Code research. arXiv preprint arXiv:1807.04872.

Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. IEEE Access, 5, 3909–3943. <https://doi.org/10.1109/ACCESS.2017.2685629>

Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery, and deployment: A systematic review. *IEEE Access, 5*, 3909–3943. <https://doi.org/10.1109/ACCESS.2017.2685629>

Shore, J., & Warden, S. (2021). *The art of agile development*. O’Reilly Media.

Tashev, A. (2021). DevSecOps: Integrating security into DevOps practices. Cybersecurity Journal, 27(4), 201-219.

Venugopal, M. V. L. N., & Reddy, C. R. K. (2021). Serverless through cloud native architecture. Int. J. Eng. Res. Technol, 10, 484-496.

Weber, I. (2022). GitOps: A paradigm shift in software deployment. Software Development Trends, 29(1), 89-104.

Wikipedia contributors. (2025, March 9). DevOps. *Wikipedia, The Free Encyclopedia.* Retrieved from <https://en.wikipedia.org/wiki/DevOps>