Bulletin of Business and Economics, 13(1), 662-672
https://bbejournal.com
https://doi.org/10.61506/01.00253

Comprehensive Analysis of DevOps: Integration, Automation, Collaboration, and Continuous Delivery

Muhammad Moeez[1], Rashid Mahmood[2], Hamza Asif[3], Muhammad Waseem Iqbal[4], Khalid Hamid[5], Umair Ali[6], Nimra Khan[7]

Abstract
This paper highlights the significance of DevOps in various contexts and explores the intricacies of DevOps practices, their utility, and strategies for their application within software organizations. Existing research on DevOps is scarce and often lacks rigor. DevOps fosters a culture centered on cooperation, automation, scalability, knowledge dissemination, and the utilization of web services. The advantages of DevOps extend to both information systems development and operational efficiency, with favorable outcomes for web service development and quality assurance performance. In conclusion, our survey underscores the need for further research to quantitatively assess these findings.
Keywords: DevOps; integration; automation; requirement analysis; continuous delivery

1. Introduction

DevOps is an approach that merges traditional software roles, aiming to enhance communication and increase deployment frequency while maintaining software quality. The term "DevOps" was coined by software development expert Patrick Debois during the DevOps Days conference in 2009. It emerged as a solution to address the limitations of the agile software development methodology, which excelled in iterative and rapid code development but often fell short in quick code deployment.
DevOps bridged the gap between agile's emphasis on collaboration and change and the ITIL framework's focus on stable and predictable IT operations. This resonated with both IT administrators and software developers (Erich et al., 2017). The term "DevOps" typically refers to the evolving professional practice that fosters a collaborative relationship between development and IT operations, resulting in efficient workflow and enhancing the reliability, consistency, flexibility, and security of the development environment. The primary objective of DevOps is to reduce the time it takes for software to transition from creation to operation while upholding high-quality standards (Jha & Khan, 2018). IBM introduced the concept of Collaborative DevOps to describe systems connecting software development and IT teams. Furthermore, DevOps can be seen as treating infrastructure with the same processes as development. Initially, DevOps was associated with larger corporations like Netflix and Google, as well as startup companies (Rajapakse et al., 2021).
These organizations are commonly known as "cloud-native," "born-in-the-cloud," or "unicorns" in the DevOps community. Currently, DevOps is adopted by three distinct groups, evident in the rapid increase in release frequency (Karamitsos et al., 2020). DevOps is guided by its core principles encapsulated in the acronym CAMS: Culture, Automation, Measurement, and Sharing. These principles have a profound impact on modern software development lifecycle requirements.

- Culture: DevOps encourages the breakdown of silos and promotes improved inter-departmental communication (Arulkumar & Lathamanju, 2019). It facilitates better communication across teams, facilitating the removal of barriers.
- Automation: DevOps not only saves time but also prevents errors, ensures consistency, and fosters self-administration.
- Measurement: DevOps supports continuous deployment and delivery, aiding decision-making through clear and straightforward data, insights, defects, and experiences, allowing like-minded individuals to make informed choices.
- Sharing: DevOps recognizes the value of sharing tools, insights, mistakes, and experiences to benefit individuals with similar interests (Banica et al., 2017).

DevOps is an evolving field, where artificial intelligence is increasingly applied to assist in various aspects, ranging from incident management to code generation. However, several challenges must be overcome for AI for DevOps, or AIOps, to become a reality. These hurdles include the need for more intelligent automation, reduced waiting times, and improved translation of business requirements into technological solutions. Despite the growing popularity of DevOps, there is a notable lack of empirical research on its practical implementation, as highlighted in recent studies. Existing literature provides limited insights into the actual execution and methodologies of DevOps, as well as their effectiveness in supporting continuous software development, with only a small number of case studies available (Erich et al., 2014).
DevOps offers numerous advantages, including enhancements across the entire software delivery pipeline, such as validation and deployment, the breakdown of silos, improved communication among IT teams, faster time-to-market for software, responsive improvements based on feedback, reduced downtime, decreased manual tasks through automation, streamlined product development with increased responsibility and code ownership in development, and broader roles and skillsets (Wiedemann & Wiesche, 2020).
On the flip side, there are disadvantages associated with DevOps implementation. These include the proliferation of development and IT tools, the potential for unnecessary, fragile, or risky automation, the challenges of scaling DevOps across various projects and teams, the risk of hasty deployments due to a "fail-fast" approach, job generalization versus specialized knowledge, regulatory compliance concerns, especially in situations requiring role isolation, and the emergence of new job roles. These organizational

[1] Department of Software Engineering, Superior University, Lahore, Pakistan, muhammadmoeez64@gmail.com
[2] Department of Software Engineering, Superior University, Lahore, Pakistan., rmbhatti88@gmail.com
[3] Department of Software Engineering, Superior University, Lahore, Pakistan, hamzaasif008@gmail.com
[4] Ph.D. Associate Professor Department of Software Engineering, Superior University, Lahore, Pakistan, waseem.iqbal@superior.edu.pk
[5] Ph.D. Assistant Professor Department of Computer Science, NCBA &E University East Canal Campus, Lahore, Pakistan, khalid6140@gmail.com
[6] Department of Computer Science Alshifa Institute of Life Sciences Pakistan, umairwahla128@gmail.com
[7] Department of Software Engineering, University of Engineering and Technology Lahore, Pakistan, nimrainamkhan@gmail.com

and IT departmental changes must be addressed during DevOps deployment, and interviews with those involved in DevOps implementation have revealed a variety of challenges that have hindered the process or jeopardized the achievement of DevOps objectives (Noorani et al., 2022). Some of the key challenges faced by DevOps include.

1.1. Addressing the Challenge of Acquiring Adequate Technical Competence Among Employees

This issue encompasses two vital aspects: first, the task of recruiting new personnel possessing the necessary technical skills, and second, the process of enhancing and retaining the skills of the existing workforce. The deficiency of adequately skilled staff can significantly impede the adoption of DevOps due to a shortage of required competencies at the right time. These competencies encompass the ability to code, a comprehensive understanding of infrastructure, its configuration, deployment, post-deployment monitoring, infrastructure troubleshooting, and proficiency in utilizing supporting tools. According to the head of the infrastructure team, the skills needed are often in short supply, resulting in challenges in finding qualified job seekers and recent graduates. They expressed, "The skill set doesn't seem to be readily available," which led to recruitment being identified as "perhaps our greatest challenge" by the RQL. The training manager emphasized the complexity of upskilling the entire workforce so that each member can effectively handle operational issues. He referred to the present situation as a "bottleneck" in expanding DevOps adoption and training current staff in the utilization of new monitoring and automation tools and principles. The team perceives this challenge as stemming from a steep learning curve. The issue lies in the constant struggle to keep pace with the ever-increasing array of new tools and concepts. A developer also noted that even though they are accustomed to learning new technologies regularly, quickly acquiring operational technologies and related concepts to meet work demands remains a challenge (Khan et al., 2022).

1.2. Overcoming Resistance and Uncertainty

To effectively combat resistance to significant changes and uncertainties regarding how the transition to a DevOps work methodology will impact individuals in the future, motivation is a key factor. A developer expressed, "DevOps was not what I signed up for." Becoming productive team members and sharing knowledge took time for infrastructure specialists, and they faced particular challenges in embracing the mental shift of being on-call for managing emergent operational issues (Hermawati et al., 2021).

1.3. Adjusting Tools and Technological Infrastructure

The move towards DevOps and continuous deployment, driven by various strategic considerations, notably the shift to a cloud-based environment and microservices architecture, has proved to be a significant facilitator. This phase of the DevOps journey has been extremely demanding and challenging following a year of product redesign. The process of selecting, experimenting with, and configuring tools for the build pipeline, encompassing full-stack testing, automated deployment, and monitoring, has been strenuous and time-consuming. There is little room for respite due to the slow pace, substantial time investment, and technical intricacies involved. Numerous studies have been conducted in the realm of DevOps, leading to the emergence of several solutions, including:

A lot of research has been done on DevOps previously and there are a few solutions driven out through those researches, which are as follows:

1.4. Fostering Teamwork and Cross-Team Communication within Defined Guidelines

Promoting collaboration across teams, as opposed to working in isolated silos, emerged as a recognized best practice in various studies focusing on DevOps. One specific recommendation is the establishment of multidisci(Alsaber et al., 2021)involves close collaboration among development, security, and operations teams right from the initial stages of a project. However, it is crucial to carry out this enhanced cooperation within well-defined rules and guidelines. Clear delineation of the roles of each team member participating in cross-functional teams is a pivotal aspect of this practice.

1.5. Engaging Security Champions as Collaborators

In a DevOps environment with distinct team priorities, it is advisable to introduce the concept of security champions who can bridge the gaps. Security champions are individuals within teams with a primary focus on security and extensive security training. They play a crucial role in promoting and maintaining a strong security posture within the team (Hamid, Iqbal, Muhammad, Fuzail, et al., 2022).

1.6. Integrating Human Resource Management (HRM) Initiatives

DevOps cultural transformation may pose challenges for certain employees within an organization. To address these challenges, studies suggest implementing HRM initiatives in conjunction with DevOps transformation projects. These HRM programs should concentrate on addressing universal concerns, such as the fear of job displacement or a perceived loss of control over one's role (Hamid, Iqbal, Nazir, et al., 2022).

1.7. Training and Knowledge-Sharing Tools for Security Awareness

Enhancing the security awareness of team members to carry out their respective security-related responsibilities can benefit from the application of security training and knowledge-sharing techniques. For example, extensive security-related knowledge is essential for implementing and using security solutions effectively. The development team should be equipped with the expertise to differentiate between legitimate security issues and false positives when utilizing static analysis tools for vulnerability assessment (Hamid, Iqbal, Aqeel, Rana, et al., 2023).

1.8. Automated Identification of Vulnerabilities through Requirement Analysis

In alignment with CSE paradigms, research has proposed methods to automate the analysis of security requirements. Toolchains have been developed to automatically transform requirement documents into ontological models, analyze these models, and report the identified findings. The availability of toolchains enables developers to address security vulnerabilities at an early stage in the development process. This study aims to explore DevOps, the challenges encountered during its implementation, and the existing body of work on this subject. The motivation behind this paper is to investigate the issues surrounding DevOps and the diverse perspectives held by different stakeholders on this subject (Hamid, Iqbal, Aqeel, Liu, et al., 2023).

2. Literature Review

Ioannis Karamitsos and his colleagues in their research provide a clear understanding that the experimentation phase in the machine learning process might seem straightforward, but overlooking the deployment, design, and utilization of models can lead to complex and time-consuming approaches. These, in turn, involve maintenance, monitoring, and improvement efforts that are both costly and demanding. DevOps, short for development and operation, encompasses a set of applications and tools grounded in systems and software engineering. A central goal of DevOps is the transformation of organizational culture to foster increased collaboration and the formation of cross-functional teams.

At the core of achieving business strategy are two pillars, namely DevOps and Agile, which blur the lines between traditional operational and development teams. This convergence creates an environment where operations are continuously refined by a cross-functional team comprising operators and developers. These pillars play a vital role in realizing strategic business objectives. DevOps, in particular, aims to explore avenues for enhancing service quality and features to meet the needs of customers. The author also outlines three phases of continuous software engineering: planning development, operation, and business strategy. Within these phases, software development practices like continuous delivery and integration are employed. Continuous integration fosters reduced release cycles, enhanced software quality, and increased team productivity. Additionally, the push-based approach of continuous delivery offers benefits such as reduced deployment risks, cost savings, and swift user feedback. The automation-centric continuous delivery methodology streamlines development, testing, and deployment activities (Riasat et al., 2023).

V. Arulkumar and R. Lathamanju's research underscores the integration of software development practices into a cohesive approach that combines automation with cloud support and DevOps in the manufacturing sector. This amalgamation leads to the rapid development of software applications, reducing the development timeline from days to minutes. Automation serves as the linchpin of the DevOps approach, enabling the continuous automation of the software life cycle, which involves ongoing code analysis and testing. The resulting software is built and deployed within the developers' cloud environment. Depending on the volume of new applications to be released, each step in the continuous delivery process may be executed multiple times daily. Automation within DevOps greatly enhances development speed, accuracy, consistency, and delivery frequency, with automation representing a foundational principle of DevOps. Automation encompasses continuous integration, continuous testing, continuous deployment, and continuous delivery, leading to reduced human errors and significant time savings. These advantages make automation particularly crucial for smaller organizations (Hamid, Iqbal, Muhammad, Basit, et al., 2022).

In the context of software development and delivery, DevOps refers to a comprehensive automation approach. This research unites the realms of development and operations through automated development, deployment, and infrastructure monitoring. Additionally, Amazon Web Services is introduced as a provider of IT services in the form of web services, known as cloud computing, simplifying the development lifecycle through tools like Maven, which is a project management and comprehension tool. Logica Banana and their peers explore the distinctions between Agile and DevOps methodologies, with DevOps representing a novel approach that aligns with the principles of agile project management for software development and execution. The authors predict that DevOps is likely to replace its predecessors due to the advantages it offers businesses in terms of software project management efficiency and agility. Deshpande defines DevOps as "a software development approach that aims to integrate all software development processes, from development to operations, within the same cycle" (Mushtaq et al., 2021).

In the realm of software development, Agile methodology emerged as a highly adaptable and efficient alternative to traditional Waterfall and PMBOK techniques. Agile serves as the foundation for DevOps and offers substantial improvements in component delivery. Unlike conventional approaches such as PMBOK or Waterfall, which release the entire product at once, DevOps focuses on delivering software components incrementally, typically at the end of each sprint.

Furthermore, the evolution of Agile through DevOps is noteworthy. In Agile, component delivery to customers often experiences delays, while in DevOps, components are released immediately upon completion. Agile tends to lack pre-release quality assurance, whereas DevOps ensures quality through test automation. Agile typically operates without a strong emphasis on teamwork, while DevOps encourages collaboration between IT and development teams. The authors assert that DevOps is the most suitable approach for website development projects, as web pages can be viewed as deliverable components that can be developed, tested, and launched as they are completed (Continuous Delivery). Simultaneously, the database supporting the website can be developed in parallel. The authors progressed to the DevOps approach after initial experimentation with smaller projects (Banica et al., 2017).

Farid and colleagues elaborate on how DevOps enhances lean software development. They emphasize that lean software development lacks the integration between development and operations teams, which DevOps effectively facilitates. DevOps seamlessly incorporates operational components into the development process, maintains their currency during development, and reduces errors during deployment. Lean methodology aims to minimize resource waste in non-value-added products, while DevOps seeks to optimize the entire lifecycle by shortening release cycles and increasing software deliveries. DevOps emerged as a response to the convergence of development and operations teams, aiming to accelerate software deployment and responsiveness to changing consumer demands. By enabling quicker customer feedback loops, reducing overhead, and minimizing rework, DevOps enhances productivity and gives businesses a competitive edge through three dynamic capabilities. First, it encourages comprehensive collaboration across various stakeholders from business and software functions, accelerating ongoing planning and idea generation. Second, it emphasizes continuous delivery through automated software delivery processes and waste elimination. Third, it focuses on early problem identification and prompt notification to development teams, creating a feedback loop for continuous customer learning through monitoring and improving software-driven innovation. In this way, DevOps optimizes Lean software development, encompassing the entire lifecycle from development to operations environments, by identifying sources of waste and applying DevOps techniques to reduce and rectify them (Farid et al., 2017).

Floris Erich and fellow researchers investigate DevOps as a theoretical approach aimed at bridging the gap between information system development and operations. Their study involves a systematic mapping of DevOps research. They note that DevOps research is scarce, and the available studies often lack quality. Additionally, they identify that DevOps follows a conceptual

framework similar to Agile Software Development and that organizations must integrate DevOps concepts and principles into their methodologies. This integration may require organizational reconstruction. The authors also discover that a culture of cooperation, automation, information sharing, measurement, and web service utilization supports DevOps. DevOps improves performance in IS development and operations while enhancing the effectiveness of quality assurance and web service development. Knowledge, skills, and abilities (KSAs) serve as a framework for the US government's hiring decisions and can be applied to select the best candidates for positions within DevOps-centric organizations.
Traditionally, operations and development teams collaborated in creating new systems, with the operations department relying on the development team to create tools and applications. These tools aimed to improve elements such as performance, security, and operating system stability. This collaboration effectively merged the roles of employees, blurring the lines between development and operations staff. DevOps, through its emphasis on cooperation and tool utilization, strengthens quality assurance by fostering a closer working relationship between these teams. DevOps also enables quality assurance staff to gather more data, with Real-time User Monitoring as a valuable tool for early end-user issue detection (Erich et al., 2014).
In their paper, Anna Wiedemann and Manuel Wiesche discuss the essential skills required to build a proficient DevOps-focused IT team. IT organizations are implementing product-focused agile IT teams to cope with rapidly evolving environments, requiring adaptability from high-performing software development teams. DevOps provides the speed and flexibility necessary for continuous and rapid digital innovation development. To bridge the gap between software development and operations, DevOps promotes collaboration, automation, virtualization, and tool usage to prevent delays in the software delivery process. The authors then delve into 36 specific skills across seven categories, including full-stack development, analysis, functional, decision-making, social, testing, and advisory abilities. Their analysis reveals that effective DevOps team members need a blend of development, operations, and management skills, emphasizing functional and analytical abilities. However, not every organization can afford highly specialized individuals with above-average compensation expectations. Firms may need to revamp their employee development programs to equip their existing IT staff with full-stack development skills. They also point out that these skill categories can overlap and depend on each other, such as the interdependence of social and decision-making skills, as effective teamwork hinges on members' willingness to learn new abilities. These skills come together within a well-structured DevOps environment and should be present in every team member (Hussain et al., 2022).
Prixit Raj and Parul Sinha investigate the influence of Agile and DevOps on team dynamics and project management procedures in software development projects. Project success is influenced by the project management procedures. Agile and DevOps are altering the rules governing the execution and delivery of software projects, impacting scope management, quality management, and estimating procedures. These changes, in turn, affect project management practices, as shared accountability, automation, and feedback inherent in Agile and DevOps methodologies reshape team structures.
The development of Agile methodology aimed to address the shortcomings of the previous model and offers project teams numerous opportunities throughout the development lifecycle. The authors advise industries looking to implement Agile and DevOps approaches in their software development processes. They stress the need for project management procedures to adapt to the evolving methods of software product development, as Agile and DevOps work differently. These practices align more closely with client requirements, speeding up the software development process by identifying errors early, enhancing communication, reducing time and costs, and improving quality. Agile and DevOps empower teams to become more self-managed, providing timely and updated information for project management. To ensure successful implementation, organizations must address challenges related to team and management strategies that affect the teams' organizational and technical capabilities (Bhatti et al., 2023).
In their paper, Ineta Buena and Marite Kirikova explain DevOps and its adoption approach. They observe that the corporate landscape is evolving rapidly, particularly in IT and other industries. Customers now have higher expectations for application content, demanding better quality and quicker delivery. This shift has fueled the popularity of agile methodologies in software development. Additionally, the adoption of DevOps is influenced by the potential to use fewer resources for software development and maintenance through automation. However, adopting DevOps is not a simple process, as it often requires extensive changes to enterprise process organization and workflows. Successful DevOps implementation necessitates a clear strategy and plan that addresses all critical aspects. The authors identify four categories of DevOps adoption obstacles: lack of knowledge, insufficient support, technological implementation challenges, and difficulties in adapting organizational procedures.
Additionally, the process for adopting DevOps can be outlined as follows: Firstly, it should be guided by the DevOps Maturity Model, which involves assessing the current DevOps maturity level and selecting the desired level. Next, a prioritized list of DevOps practices should be compiled, taking into account the key priorities identified during the obstacle identification phase. These practices should align with the goals identified in the DevOps Maturity Model, bridging the gap between current and desired DevOps maturity levels. Moreover, an inventory of existing DevOps-related tools within the organization should be created, along with a list of relevant DevOps tools that support the identified DevOps practices (Rasool et al., 2023).
Khan and Shameem's paper delves into the challenges faced by DevOps. While DevOps processes significantly accelerate and automate the continuous delivery and deployment of software systems, the adoption of DevOps techniques is not without its difficulties. Many businesses struggle to keep pace with the rapid delivery and deployment characteristic of DevOps. Despite the importance of DevOps practices, it remains unclear why some software development companies face barriers or hesitance in their implementation. They aim to identify the challenging aspects that impact DevOps initiatives and establish a taxonomy of factors using AHP prioritization. The authors identify 16 factors that can pose challenges to the implementation of DevOps practices (Hamid et al., 2024).
These factors include a lack of a collaborative organizational culture, DevOps pipeline security, test automation, transitioning from legacy to microservices architecture, resistance to change, information and data sharing gaps, a lack of a unified DevOps maturity model, bureaucratic deployment procedures, absence of a clear DevOps transformation strategy, inadequate understanding of various DevOps tools, ambiguous DevOps definitions, poor team performance, insufficient DevOps training, lack of transparency,

and organizational vision. Among these, bureaucratic deployment procedures, lack of knowledge, data sharing, and DevOps pipeline security are given priority. The taxonomy they create serves as a robust framework to enhance the DevOps process by addressing major obstacles based on their significance and impact (Iqbal et al., 2024).

Pietrantuono and colleagues introduce DevOpRET, a strategy for continuous reliability testing in DevOps. DevOps is often perceived as the convergence of software development (Dev), quality assurance (QA), and operations (Ops). DevOps is defined as "a set of processes designed to reduce the time between committing a change to a system and having that change implemented in normal production while maintaining high quality." Two fundamental DevOps methodologies include monitoring and ongoing testing. Continuous testing involves short, automated testing cycles that provide immediate quality feedback in the context of continuous integration (CI) and an acceptance test phase to determine release readiness. The adoption of DevOps is primarily driven by business objectives, which are translated into measurable Key Performance Indicators (KPIs). Measurement plays a crucial role in DevOps' success (Ibrahim et al., 2023).

Notably, prior research has not extensively explored DevOps reliability. In response, the authors present the DevOpRET method, which integrates continuous software reliability testing into DevOps practice. This strategy, indicated by the acronym SRET, utilizes usage data to generate representative test cases and assess reliability before each release. DevOpRET leverages feedback from Operations to guide the acceptance testing phase conducted by the QA team before production release. DevOps relies on monitoring data to characterize the usage profile, reducing uncertainty about the operational profile pre-release. It employs data from operational monitoring to direct operational-profile-based testing as part of the acceptance testing process before each production release (Salahat et al., 2023).

Daniel Stahl, Torvald Martensson, and Jan Bosch shed light on the various definitions of DevOps. In the realm of software engineering, DevOps and continuous practices have gained increasing attention among practitioners and researchers. However, these terms are often used ambiguously and inconsistently. The authors examine the usage of these terms in published literature concerning continuous practices and DevOps. They argue that this ambiguity and miscommunication can be detrimental, as it hinders the objective evaluation of these practices, their impacts, and their relationships. The meanings and interpretations of these practices, such as continuous integration and continuous delivery or deployment, can vary significantly. DevOps, in particular, is known for its lack of clear and consistent definitions, with multiple interpretations and descriptions available (Fatima et al., 2023). DevOps can be defined as "a software development approach that seamlessly incorporates quality assurance practices with IT operations within the software development process." The authors analyze 35 publications on DevOps, revealing that some scarcely mention DevOps, let alone define it. DevOps is sometimes used as a buzzword, an engineering technique, or even as a strategy rather than a methodology. The lack of a consensus definition adds to the complexity. Some view DevOps as an approach that aims to automate the entire deployment process from source code in version control to production, emphasizing its technical aspects. Different people define DevOps differently, and this lack of clarity can be a source of confusion.

In a separate paper, Muhammad Shoaib Khan and colleagues highlight the significant considerations when implementing a DevOps culture. These considerations encompass challenges such as coordination and communication issues, a lack of expertise, complex infrastructure, inadequate management, absence of a DevOps strategy, and trust issues. DevOps is seen as a collection of techniques and a cultural movement aimed at enhancing collaboration and communication between development and operations teams. Despite its potential, including quicker production cycles, increased reliability, and stability, there is a dearth of literature and data on the fundamental concepts, processes, resources, and obstacles in implementing DevOps. The study identifies ten significant obstacles to adopting a DevOps culture, with culture, practices, and tools forming the three pillars of DevOps. Culture underpins these practices, and a range of tools is necessary for their implementation. However, implementing DevOps principles can be challenging, primarily due to the cultural shift between development and operations and the issues stemming from inadequate communication between these teams.

In the research by Dhaya Sindhu, the focus was on exploring the synergy between DevOps and cloud development and testing. The strong connection between DevOps and cloud technology is undeniable. As an increasing number of cloud projects integrate DevOps practices, this trend is set to continue. The advantages of applying DevOps in cloud application development are becoming increasingly evident. To remain competitive, companies must have the ability to rapidly deliver services or applications. Effective management practices and tools must be both fast and reliable. This necessitates the use of DevOps automation tools, whether in a cloud or non-cloud environment, to automate DevOps operations. However, experts predict that many predictable, repetitive tasks could be automated in the future. The application of intelligent automation (AI), which involves processes that adapt, learn, and improve over time, is becoming more feasible thanks to advances in AI and related fields. Algorithms are being developed to automate cognitive tasks. They also highlight that the application of AI in mobile robots has expanded the range of manual tasks that can be automated. Global enterprises are increasingly adopting DevOps to enhance their operational processes, with most operations being hosted on remote servers (cloud servers) for convenience and efficiency. The benefits of using cloud-based DevOps are expected to become more apparent over time.

In their research article, Anna Wiedemann and her colleagues express their concerns about the alignment between IT and business. Achieving alignment between IT and business at the organizational level is a top priority for businesses worldwide. The study focuses on the alignment of IT functions within the organization. Historically, IT tasks were divided into various components, but many organizations are now implementing cross-functional DevOps teams that integrate knowledge, tasks, and skills related to software development, planning, and management. This shift is driven by changing customer demands and the need to manage complex IT architectures. They propose a three-part model of intra-IT alignment, shedding light on how DevOps teams within the IT department can align development and operations. The model outlines various aspects of misalignment in the IT function, such as the tension between shorter development cycles that favor rapid feature delivery and the emphasis on maintaining stable operating systems with minimal changes. The study also suggests three approaches—individual componentization, multidisciplinary knowledge, and integrated responsibility—to address intra-IT misalignment in DevOps teams.

In another paper, Taryana and colleagues emphasize the development of software for managing the Internal Quality Assurance System (IQAS-HE) in Higher Education, which is currently a critical need. The primary challenge was automating software development to create reusable software for multiple developers. Automation of software development processes has become essential to bridge the gap between the operations team and the development team. The latest development necessitates automating software development processes to enhance productivity, accelerate product delivery, and enable direct operation. DevOps is a concept developed by practitioners to facilitate collaboration in managing software development, delivering software, and operating it in the intended environment.
DevOps is a model designed to eliminate the traditional divisions and barriers that often exist between operations and development teams in many businesses. The primary objective of DevOps was to enable continuous software development and deployment, with a focus on frequent and speedy releases. Additionally, software engineering is a critical discipline in business settings, and the system for implementing the Quality Assurance System, known as SPM-PT, is essential. Software development encompasses phases such as business modeling, analysis, programming, installation, design, product delivery, monitoring, operation, and evaluation, all of which require setting standards, implementing, monitoring, and evaluating SPM-PT. Both DevOps and SPM-PT address the same issue—meeting consumer demands for continuous delivery and improvement. They share a common organizational structure, combining the development and operations segments of the workplace in line with the DevOps cycle. The integration of SPM-PT into supply chain software using DevOps principles was crucial to addressing current challenges.

3. DevOps Tools

3.1. Version Control Systems

○ Git
○ Subversion (SVN)
○ Bitbucket

3.2. Continuous Integration Tools

○ Jenkins
○ Travis CI
○ CircleCI
○ pipelines

3.3. Configuration Management Tools

○ Ansible
○ Chef
○ Puppet

3.4. Containerization and Orchestration

○ Docker
○ Docker Compose
○ Kubernetes

3.5. Orchestration and Automation

○ Ansible
○ Puppet
○ Chef

3.6. Collaboration and Communication

○ Slack
○ Microsoft Teams

3.7. Monitoring and Logging

○ Nagios
○ ELK Stack (Elasticsearch, Logstash, Kibana)
○ Prometheus

3.8. Cloud Platforms

○ Amazon Web Services (AWS)
○ Microsoft Azure
○ Google Cloud Platform (GCP)

3.9. Infrastructure as Code (IaC)

○ Terraform
○ AWS CloudFormation
○ Azure Resource Manager (ARM) templates

All these tools fully depend on the software house's approach and cost.
DevOps encompasses a range of techniques designed to automate and amalgamate software development and IT operations (Ops) processes, fostering enhanced collaboration and productivity. The DevOps lifecycle involves several key processes and stages, which can vary based on specific methodologies and organizational needs. However, here is a general overview of the DevOps lifecycle:

1. Plan

Define Objectives: Establish clear goals and objectives for the development and operations teams. Collaborative Planning: Foster collaboration between development, operations, and other stakeholders for effective planning. Risk Assessment: Identify potential risks and challenges in the development and deployment processes.

2. Develop

Code Development: Developers write code collaboratively, often utilizing version control systems like Git for efficient code management. Continuous Integration (CI): Developers integrate their code changes into a shared repository multiple times a day. Automated CI tools build and test the code. Code Review: Peers review code changes to maintain code quality and identify issues early in the development process.

3. Test

Continuous Testing: Automated tests, including unit tests, integration tests, and acceptance tests, ensure code quality and identify bugs early. Test Automation: Automated testing tools are used to validate code changes and ensure they do not break existing functionalities.

4. Deploy

Continuous Deployment (CD): Automated deployment processes ensure that code changes that pass testing are automatically deployed to production or staging environments. Infrastructure as Code (IaC): Infrastructure configurations are managed as code, allowing for automated provisioning and management of infrastructure resources.

5. Operate

Monitoring and Logging: Continuous monitoring tools track application performance and collect data for analysis. Logs are generated and analyzed for troubleshooting. Incident Management: Rapid response to issues is facilitated through automated alerts and incident management systems. Feedback Loops: Feedback from monitoring and incidents is utilized to inform future development and operational improvements.

6. Monitor

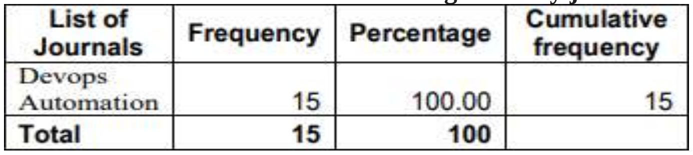
Performance Monitoring: Continuously monitor application performance, server health, and other relevant metrics to ensure optimal operation. User Experience Monitoring: Track user interactions and experiences to identify areas for improvement and enhance user satisfaction. Security Monitoring: Monitor for security vulnerabilities and breaches, ensuring the safety of data and applications.

7. Feedback

Post-Implementation Review: Evaluate the results of the deployment, considering factors such as performance, user feedback, and business outcomes. Iterative Development: Use feedback to make necessary improvements and iterate on the next development cycle, starting the process again. DevOps emphasizes automation, collaboration, and continuous feedback loops, enabling organizations to deliver high-quality software faster and more efficiently. It's important to note that DevOps is not a one-size-fits-all approach; organizations may tailor these processes based on their specific requirements and constraints.

4. Results and Discussion

Table 1: illustrates the distribution of articles categorized by journal names [2] [5].



List of Journals	Frequency	Percentage	Cumulative frequency
Devops Automation	15	100.00	15
Total	**15**	**100**	

Figure 1: depicts the breakdown of articles categorized by journal names.

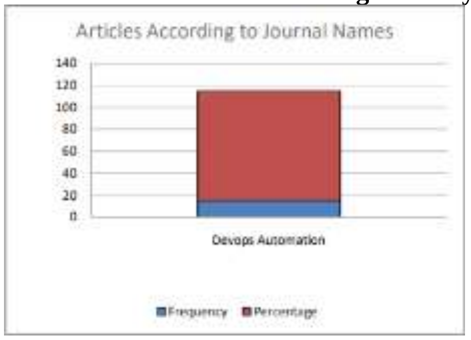
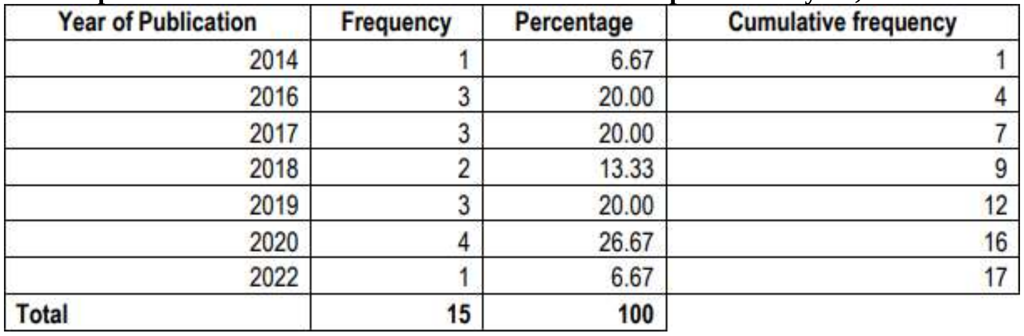


Table 2: presents the distribution of articles based on their publication year, sourced from.



Year of Publication	Frequency	Percentage	Cumulative frequency
2014	1	6.67	1
2016	3	20.00	4
2017	3	20.00	7
2018	2	13.33	9
2019	3	20.00	12
2020	4	26.67	16
2022	1	6.67	17
Total	**15**	**100**	

Figure 2: showcases the distribution of articles categorized by their publication year.

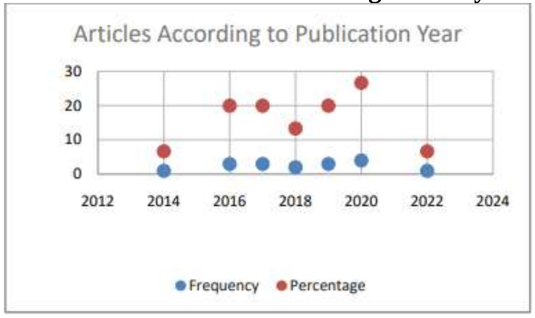


Table 3: provides an overview of articles distributed by their respective countries.

Countries	Frequency	Percentage	Cumulative frequency
UAE	1	6.67	1
Egypt	1	6.67	2
Riga, Latvia	1	6.67	3
Canada	2	13.33	5
Romania	1	6.67	6
China	1	6.67	7
Germany	2	13.33	9
Netherlands	1	6.67	10
Italy	2	13.33	12
Sweden	2	13.33	14
Korea	1	6.67	15
Total	**15**	**100**	

Figure 3: illustrates the distribution of articles based on their originating countries.

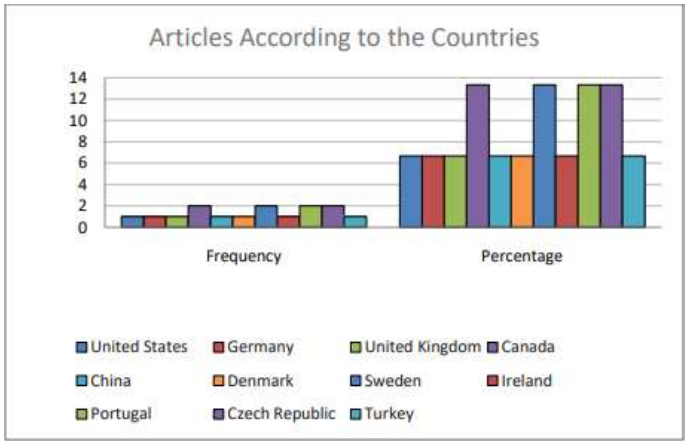


Table 4: outlines the distribution of articles concerning their research types.

Type of Research	Frequency	Percentage	Cumulative frequency
Quantitative	4	26.67	4
Qualitative	6	40.00	10
Mix method (quantitative & qualitative)	2	13.33	12
Theoretical	3	20.00	15
Total	**15**	**100**	

Figure 4: presents the distribution of articles categorized by their research types.

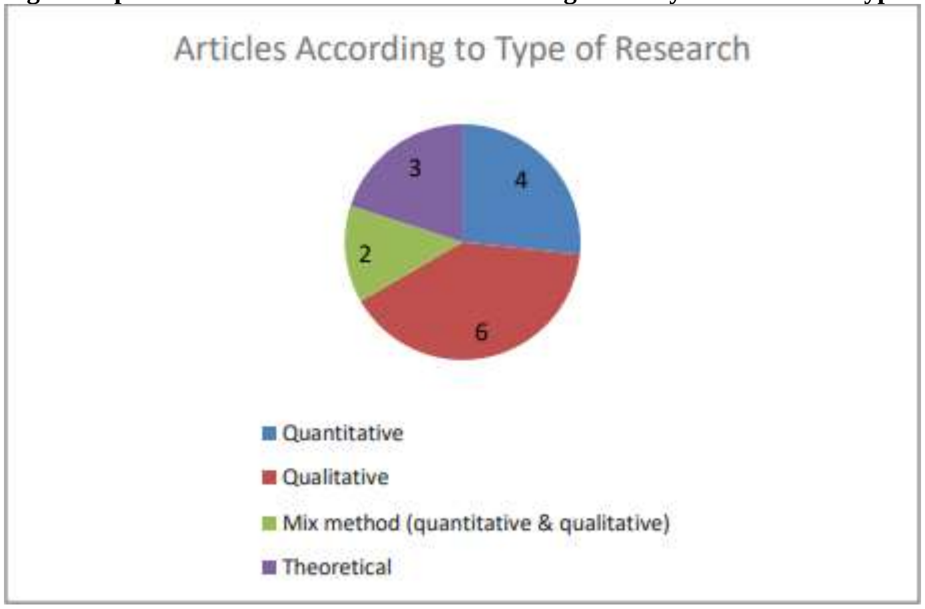


Table 5: Exhibits the distribution of articles according to their data collection tools.

Tools	Frequency	Percentage	Cumulative frequency
Survey	15	100.00	15
Total	**15**	**100**	

Figure 5: Displays the distribution of articles based on the data collection tools utilized .

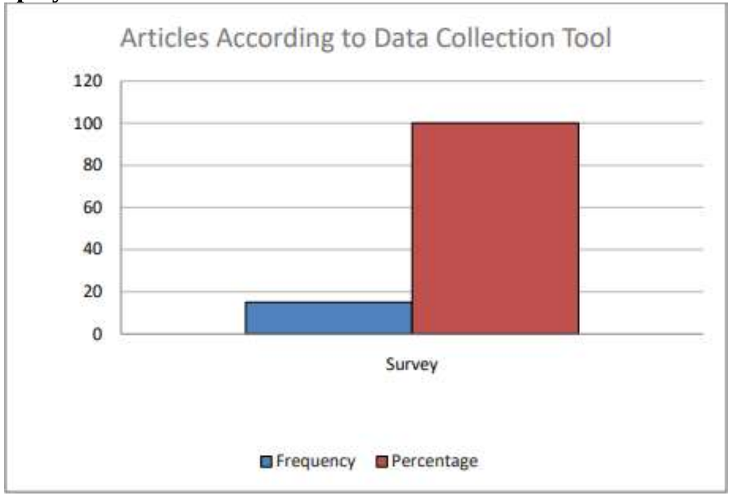
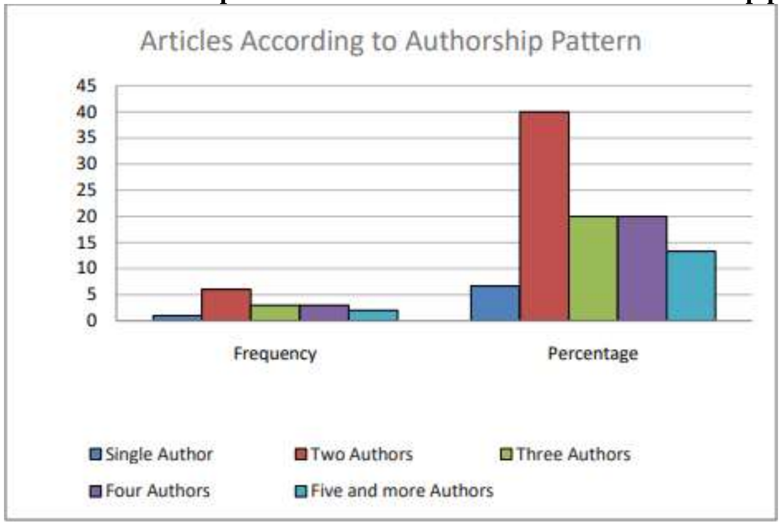


Table 6: Outlines the distribution of articles by their data authorship patterns.

Number of Contributors	Frequency	Percentage	Cumulative frequency
Single Author	1	6.67	1
Two Authors	6	40.00	6
Three Authors	3	20.00	3
Four Authors	3	20.00	3
Five and more Authors	2	13.33	2
Total	**15**	**100**	

Figure 6 illustrates the dispersion of articles based on the data authorship pattern.



5. Conclusion

DevOps represents a significant transformation in the realm of information system development. It plays a pivotal role in bridging the divide between end users, developers, and operations. This approach facilitates a continuous development process, enabling the frequent delivery of software. Furthermore, DevOps is instrumental in reducing downtime through the continuous execution of operations in the cloud.

References

Alsaber, L., Elsheikh, E., Aljumah, S., & Jamail, N. (2021). Perspectives on the adherance to scrum rules in software project management. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*, 360.

Arulkumar, V., & Lathamanju, R. (2019). Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method. *Procedia Computer Science*, *165*, 399–405.

Banica, L., Radulescu, M., Rosca, D., & Alina, H. (2017). Is DevOps another Project Management Methodology? *Informatica Economica*, *21*, 39–51.

Bhatti, S., Hamid, K., Bashir, A., zafar, zishan, raza, ahmad, & Iqbal, M. waseem. (2023). *Solutions, Countermeasures, And Mitigation Methods For The Rise Of Automotive Hacking*. *56*, 77–99.

Erich, F., Amrit, C., & Daneva, M. (2014). *Report: DevOps Literature Review*.

Erich, F., Amrit, C., & Daneva, M. (2017). A Qualitative Study of DevOps Usage in Practice. *Journal of Software: Evolution and Process*, *00*.

Farid, A. B., Helmy, Y. M., & Bahloul, M. M. (2017). Enhancing Lean Software Development by using Devops Practices. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *8*(7), Article 7.

Fatima, A., Iqbal, M. W., Hussain, D., Aqeel, M., & Akram, S. (2023). Usability Impacts Of Immersion-Based Online Task-Oriented Games Addiction In Youth. *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)*, *42*, 24.

Hamid, K., Ibrar, M., Delshadi, A. M., Hussain, M., Iqbal, M. W., Hameed, A., & Noor, M. (2024). ML-based Meta-Model Usability Evaluation of Mobile Medical Apps. *International Journal of Advanced Computer Science and Applications*, *15*(1).

Hamid, K., Iqbal, M. waseem, Aqeel, M., Liu, X., & Arif, M. (2023). *Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA)* (pp. 248–262).

Hamid, K., Iqbal, M. waseem, Aqeel, M., Rana, T., & Arif, M. (2023). *Cyber Security: Analysis for Detection and Removal of Zero-Day Attacks (ZDA)* (pp. 172–196).

Hamid, K., Iqbal, M. waseem, Muhammad, H., Basit, M., Fuzail, Z., † Z., & Ahmad, S. (2022). *Usability Evaluation of Mobile Banking Applications in Digital Business as Emerging Economy*. 250.

Hamid, K., Iqbal, M. waseem, Muhammad, H., Fuzail, Z., & Nazir, Z. (2022). Anova Based Usability Evaluation Of Kid's Mobile Apps Empowered Learning Process. *Qingdao Daxue Xuebao(Gongcheng Jishuban)/Journal of Qingdao University (Engineering and Technology Edition)*, *41*, 142–169.

Hamid, K., Iqbal, M. waseem, Nazir, Z., Muhammad, H., & Fuzail, Z. (2022). Usability Empowered By User's Adaptive Features In Smart Phones: The Rsm Approach. *Tianjin Daxue Xuebao (Ziran Kexue Yu Gongcheng Jishu Ban)/Journal of Tianjin University Science and Technology*, *55*, 285–304.

Hermawati, E., Budiastuti, H., Yulistiani, F., Adhitasari, A., Paramitha, T., & Hidayatulloh, I. (2021, January 1). *Implementation of Internal Quality Assurance System (IQAS) in Maintaining Excellent Accreditation Levels*.

Hussain, D., Rafiq, S., Haseeb, U., Hamid, K., Iqbal, M. waseem, & Aqeel, M. (2022). *Hci Empowered Automobiles Performance By Reducing Carbon-Monoxide*. *41*, 526–539.

Ibrahim, A., Al-Rajab, M., Hamid, K., Aqeel, M., Muneer, S., Parveen, M., & Saleem, M. (2023). *Usability Evaluation of Kids' Learning Apps* (p. 10).

Iqbal, M. W., Hamid, K., Ibrar, M., & Delshadi, A. (2024). Meta-Analysis and Investigation of Usability Attributes for Evaluating Operating Systems. *Migration Letters*, *21*, 1363–1380.

Jha, P., & Khan, R. (2018). A Review Paper on DevOps: Beginning and More To Know. *International Journal of Computer Applications*, *180*, 16–20.

Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps Practices of Continuous Automation for Machine Learning. *Information*, *11*(7), Article 7.

Khan, M., Khan, A., Khan, F., Khan, M., & Whangbo, T. (2022). Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review. *IEEE Access*, *10*, 1–1.

Mushtaq, M. S., Mushtaq, M., & Jamil, S. (2021). Design of Social Media Websites Acting as a Product of User's Virtual Needs and Expectations. *International Journal of Computer Science and Information Security*, *19*, 67–71.

Noorani, N. M., Zamani, A. T., Alenezi, M., Shameem, M., & Singh, P. (2022). Factor Prioritization for Effectively Implementing DevOps in Software Development Organizations: A SWOT-AHP Approach. *Axioms*, *11*(10), Article 10.

Rajapakse, R., Zahedi, M., Ali Babar, M., & Shen, H. (2021). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and Software Technology*, *141*, 106700.

Rasool, N., Khan, S., Haseeb, U., Zubair, S., Iqbal, M. waseem, & Hamid, K. (2023). Scrum And The Agile Procedure's Impact On Software Project Management. *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)*, *42*, 380–392.

Riasat, H., Akram, S., Aqeel, M., Iqbal, M. W., Hamid, K., & Rafiq, S. (2023). *Enhancing Software Quality Through Usability Experience And Hci Design Principles*. *42*, 46–75.

Salahat, M., Said, R., Hamid, K., Haseeb, U., Ghani, E., Abualkishik, A., Iqbal, M. W., & Inairat, M. (2023). *Software Testing Issues Improvement in Quality Assurance* (p. 6).

Wiedemann, A., & Wiesche, M. (2020, July 5). *Are You Ready For DevOps? Required Skill Set for DevOps Teams*.