International Journal of Basic and Applied Sciences, 14 (6) (2025) 63-70

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/1zwgxp78
Research paper

From Code to Cloud: Navigating The Future of Software Engineering and Testing Automation

Subramanya Shashank Gollapudi Venkata *

Software Engineer, McKinney, Texas, USA
**Corresponding author E-mail: Shashank.gvs@gmail.com*

Received: July 26, 2025, Accepted: September 12, 2025, Published: October 5, 2025

Abstract

This research explores how cloud technologies and mechanization are revolutionizing contemporary software engineering and examining practices. It evaluates the unification of cloud computing into advanced workflows, underscoring advancements in partnership, scalability, and implementation efficiency. The study additionally analyzes the role of automation in improving evaluation accuracy and speed, specifically through CI/CD pipelines and cloud-based testing contexts. Key obstacles such as tool merging, security worries, and technical skill gaps are recognized as critical obstacles in embracing cloud-based solutions. Additionally, the research showcases emerging trends encompassing AI-driven test automation, Infrastructure-as-Code (IaC), and DevSecOps, which are restructuring the future of software engineering. By evaluating current practices and innovations, this study delivers insights into enhancing software development life cycles using cloud and automation technologies, eventually aiming to assist in faster, more reliable, and scalable engineering outcomes. The findings add valuable recommendations for organizations seeking to revise their engineering operations in line with industry progression.

Keywords: *Cloud Computing; Scalability; DevOps; AI Testing; Automation; Continuous Integration (CI); Continuous Delivery (CD); and Software Quality Assurance.*

1. Introduction

Cloud technologies and automation are fundamentally reshaping software engineering and testing practices by enabling faster development cycles and improved collaboration, quality, and scalability. Cloud-based testing offers flexibility and the ability to test applications in diverse environments. Moreover, artificial intelligence-powered automation is further streamlining tasks, for example, code generation and testing. The future of software engineering and testing automation is being shaped by the developing adoption of AI, DevOps practices, and cloud-based solutions. AI helps recognize the most critical tests to run, grounded in historical data and code changes, which optimizes testing attempts and resource allocations. AS software systems grow in complexity and demand faster deployment cycles, traditional engineering approaches are being replaced by agile, scalable, and automated solutions. This research also evaluates how the integration of cloud technologies and automation is transforming software engineering practices and targeting efficiency. Thus, this research focuses on understanding the future trajectory of software engineering and testing automation by examining the role of cloud platforms and the challenges of automated testing. Thus, this research highlights technological trends shaping this shift. Also know that public cloud spending is projected to reach $679 billion by 2025, up from $563 billion in 2023. A software engineering background typically involves a strong foundation in computer science principles, which includes proficiency in programming languages and understanding of software development life cycles with tools like VC and CI/CD pipelines. Also, know that by 2027, over 80% of software development processes will involve AI assistance, requiring a significant upskilling of the engineering workforce.

2. Aim

To examine how cloud technologies and automation are reshaping software engineering and testing practices.

3. Objectives

RO-1: To evaluate the influence of cloud computing on modern software engineering workflows.
RO-2: To examine the role of automation in improving software testing efficiency and accuracy.
RO-3: To identify key challenges in integrating cloud-based testing tools in development pipelines.
RO-4: To evaluate future trends influencing software engineering and automated testing advancements.

4. Literature Review

Software engineering is moving rapidly from traditional development to cloud-based platforms. Cloud computing allows scalable, cost-effective solutions. It supports remote collaborations and speeds up development cycles. Testing automation is also becoming important. Manual testing is slow and prone to errors. Automated technologies like Selenium, JUnit, and TestNG are now widely used. The integration of cloud and automation helps with continuous integration and delivery (CI/CD). This enables developers to release code faster and more reliably. It explains that CI/CD pipelines depend on frequent testing and development. Cloud platforms like AWS, Azure, and Google Cloud provide strong CI/CD support (Koneru, 2025). AI and ML streamline various stages of the software development lifecycle, including code generation and deployment. Moreover, developers can spend more time on strategic tasks, complex systems design, and innovation than on manual coding and testing. AI-powered tools quickly identify potential issues and suggest improvements. Low-code/no-code platforms are gaining popularity. They also assist non-programmers in building applications using visual tools (Nelimarkka, 2023). This trend makes software development more accessible and faster.

Automated tests improve efficiency and accuracy, but frequent code changes break scripts, making maintenance costly (Giray, 2021). Cloud computing enhances scalability and collaboration, yet security and data privacy remain critical due to cyber threats and regulations. Future software engineers benefit from cloud-native environments, but must continuously adapt to evolving tools and frameworks to remain effective. Moreover, engineers need to adopt a mindset of continuous learning through microcredentials and online courses to stay updated. Cybersecurity and data privacy are critical as systems become more interconnected. Engineers demand to enable security into the development lifecycle (DevSecOps) and implement zero trust models, which are evolving data regulations such as GDPR, CCPA. DevSecOps adoption varies across industries due to cultural resistance, skill gaps, and integration complexity, leading to inconsistent implementation and slower realization of security benefits in development pipelines. Also know that the tech job market is slowly starting to pick up again, with the cloud engineering market anticipated to grow by 14% by 2031 (Masselos, 2025). Thus, the software engineers' market is expected to grow by 20% in the same duration. The 14% cloud engineering growth projection by 2031 lacks context; clarify data sources, methodology, and assumptions behind these estimates. Thus, software engineers and cloud engineers have different responsibilities and infrastructures. Software engineers' responsibilities include working on implementing new features, collaborating with product teams, and debugging existing software (Constantino et al., 2021). Hence, cloud engineers are managing systems security and reliability. Furthermore, cloud engineers are also designing and implementing cloud infrastructure.

Cloud-based testing tools face limitations such as high costs, integration challenges with legacy systems, dependency on internet reliability, and potential scalability issues across diverse enterprise environments.

5. Research Gap

A key research gap lies in the limited critical evaluation of cloud-based testing and DevSecOps adoption across industries. Existing studies emphasize benefits but overlook integration challenges, security risks, and sector-specific limitations, highlighting the need for deeper, comparative investigations to guide sustainable and secure software engineering practices.

6. Methods

The research employs a secondary research method that proves highly beneficial for exploring the evolving field of software engineering and testing automation (Arvanitou et al., 2021). This method helps this research to a wide range of existing literature, providing industry reports, papers, academic journals, and technological forecasts from globally recognized sources like Gartner and the Library. By reviewing secondary data, this research gains valuable insights into historical trends, current practices, and future projections concerning the incorporation of cloud technologies in software advancement. This research approach helps with a comprehensive comparative analysis of software engineering practices across different industries. Thus, it also allows researchers to identify cloud-native architecture. DevOps practices and automated testing tools are adopted in both developed and emerging markets (Aiyenitaju, 2024). Furthermore, this research using secondary data also enables the exploration of multiple case studies, providing companies like Microsoft, Google, Amazon, and emerging startups that showcase real-world applications from traditional code deployment to cloud-based solutions. Secondary research is crucial to help this research, making it suitable for studying technological patterns and innovation adoption over large sample sizes (Stornelli, 2021). Moreover, this research, through the analysis of previously conducted empirical studies and technical blogs, provides a well-rounded understanding of the challenges and opportunities software engineers face. Moreover, secondary data allows researchers to analyze ethical concerns, skill gaps, and security risks associated with cloud adoption and test automation. Therefore, this research, through specific secondary analysis, highlights a multidimensional understanding of how modern software engineering is being reshaped from code to cloud.

7. Results and Discussion

7.1. Impact of cloud computing on engineering workflows

The integration of cloud computing into software engineering workflows has crucially reshaped the process of how development teams operate, collaborate, and deliver software products (Lawal, 2023). This analysis reveals a marked improvement in efficiency, flexibility, and scalability across the entire software lifecycle. One of the most transformative impacts is development team collaboration. Cloud computing platforms enable developers, testers, and project managers to access shared environments, code repositories, and project documentation in real time. Common tools like GitHub, GitLab, and Atlassian's suite have become central to engineering workflows due to their integration with cloud-based environments (Surbhi Kanthed, 2025). Thus, the acceptance of cloud-based CI/CD pipelines has also resulted in faster development cycles. Automation of testing and integration has minimized human error, reduced time to market, and increased release frequency.

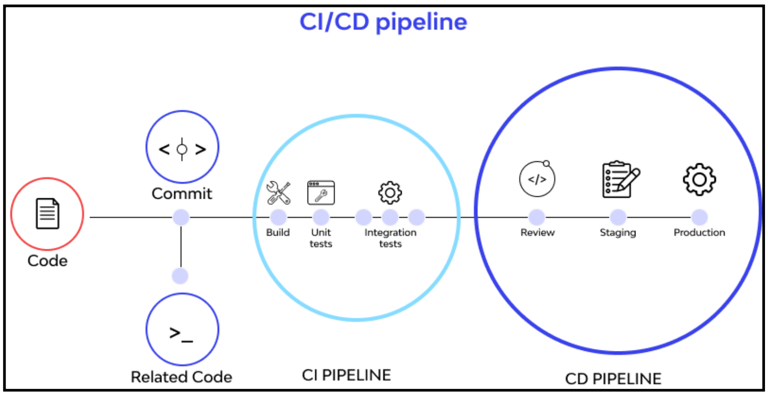

Fig. 1: CI/CD Pipelines.

Source: (Thorne, 2023).

The above includes the CI/CD pipeline, which supports cloud computing on engineering workflows. It visualizes how cloud-based CI/CD tools automate and streamline code commit stages for production (Thorne, 2023). The CI/CD segment develops speed and accuracy, while the CD segment, through review, staging, and production quality, ensures smooth, scalable deployment. This diagram reflects that transformation is provided by cloud platforms, increasing continuous delivery and real-time collaborations from specific analysis.

However, cloud computing presents challenges for software engineers. Security and compliance remain key concerns. As data moves to shared and, ensuring strong security systems, integrity, and compliance remain key concerns. As data moves to shared and, ensuring privacy, integrity with regulations like GDPR becomes critical. Therefore, cost management also emerged, as cloud services are served through pay-as-you-go models, a lack of oversight that results in unexpected expenditure, and prolonged usage of high-tier services (Ramesh, 2021). Therefore, vendor lock-in represents a strategic risk; migrating between providers can be an effect as high as costly and technically difficult.

7.2. Automation's role in testing efficiency and accuracy

The role of automation in developing testing efficiency and accuracy has become increasingly evident, particularly within cloud-based development environments. Cloud-enabled test automation technologies such as Azure DevTest Labs, AWS CodeBuild, and Google Cloud Test Lab offer dynamic test execution capabilities, which allow development teams to deeply test in virtualized, scalable environments (Aggarwal et al., 2022).

Table 1: Cloud-Based Automation Tools Enhancing Testing Efficiency and Accuracy

Tool/Service	Function	Cloud Provider	Key Benefit	Usage Stage
AWS CodeBuild	Build & test	AWS	Fast execution	CI phase
BrowserStack	Cross-browser testing	Third-party	Broad coverage	Integration tests
Azure DevTest Labs	Environment creation	Microsoft Azure	Scalable setups	Pre-deployment
Google Test Lab	Device testing	Google Cloud	Real-device feedback	Regression testing
Terraform	IaC provisioning	Open-source	Realistic environments	Test automation setup

Additionally, automation integrated within CI/CD pipelines triggers tests at specific stages, like post-commit or post-build, ensuring continuous feedback (D'Onofrio, 2023). This immediate assistance helps teams identify and isolate defects much earlier in the development life cycle, thereby improving test accuracy and reducing work. AI-driven testing frameworks like Selenium, JUnit, and TestNG enhance efficiency and accuracy. AI integration reduces manual testing by 40–60% (Virtuoso QA) and Google's Smart Test Selection optimizes test execution using ML, demonstrating AI's transformative impact on automated software testing (Team DigitalDefynd, 2025). The use of IaC also played a critical role. With platforms such as Terraform or CloudFormation, test environments are auto-provisioned to match production, ensuring realistic testing conditions without manual setup. Cloud automation also leverages machine learning-based test optimization, flaky test detection, and reduces unnecessary test runs and focusing on areas most likely to be affected by code changes. Furthermore, AI/ML smart test selection assists software engineers in optimizing their testing efforts by intelligently choosing which tests to run and reducing time and cost while improving software quality. AI/ML algorithms analyze historical test data to learn patterns and identify which tests are most effective at catching defects (Constantin, 2024). AI/ML algorithms prioritize test cases based on their potential to uncover critical issues so that high-risk areas are tested thoroughly (Sutharsan Saarathy, 2024). According to current market data from UK employees, it is known that if the experience level is above entry level, the software engineer's salary is £30,000-£40,000, and the cloud engineer's salary is £35,000-£45,000. Thus, a mid-level Software Engineer salary is £45,000 - £65,000 and a Cloud Engineer £50,000 - £70,000 (Xcede, 2025).

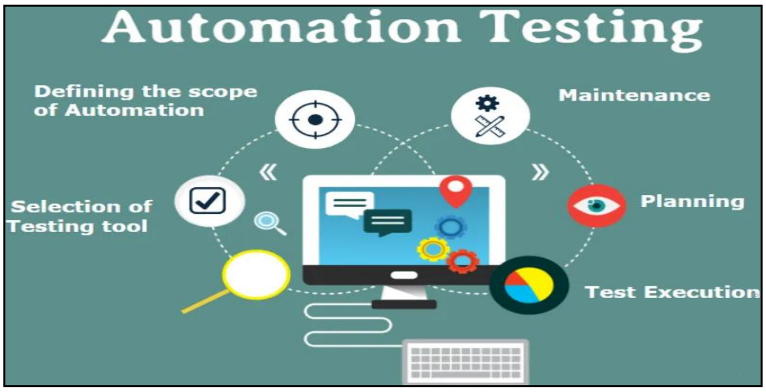

Fig. 2: Automation Testing.

Source: (Pedamkar, 2019).

Automation testing in traditional setups is resource-heavy and slower, while cloud-driven approaches enhance scalability, reduce manual effort, accelerate delivery, and support continuous integration and deployment pipelines. Automation testing plays a crucial role in the "from cloud to code journey by activating immediately after code is committed within CI/CD pipelines. Thus, it ensures rapid validation and early bug detection across development stages. Integrated with cloud services, automation supports parallel testing and scalable execution, which delivers while maintaining quality (Pedamkar, 2019). Integrated with cloud services, automation supports parallel testing and scalable execution, which accelerates delivery while maintaining quality. Thus, specific tools like AWS CodePipeline and Google Test Lab showcase automated workflows that streamline testing in real time (Boscain, 2023). The integration of predictive analytics and machine learning in cloud automation platforms supports smart test selection, which prioritizes tests based on code changes while maintaining test relevance.

7.3. Challenges in cloud-based testing integration

As organizations shift from traditional development models to cloud-based, automated software engineering environments, several critical challenges arise. Thus, obstacles impact productivity, cost, performance, and quality if not properly addressed. Below are the key challenges faced in this transformation. Cloud-based testing faces real-world challenges, including integration complexity, security risks, organizational resistance, and varied adoption across industries, impacting efficiency and reliability.

7.4. Tool integration complexity

Integrating a wide range of tools like version control, CI/CD pipelines, testing frameworks, and cloud platforms will be technically complex (Parmar, 2025). These technological tools often originate from different vendors and may not natively support each other. Furthermore, it ensures smooth communication between tools such as Jenkins, Selenium, and Git-like AWS to Azure, which requires custom scripting, APIs, or third-party middleware. Incompatibility problems lead to automation failures, delayed deployment, and broken workflows, especially in multi-cloud or hybrid environments.

7.5. Ethical and security considerations in cloud-based testing

Ethical and security considerations are crucial in cloud-based AI testing. Bias in AI-driven test case selection can lead to unfair outcomes or overlooked defects. Compliance with regulations like GDPR and CCPA is essential to protect sensitive data (Wong, Chong, and Aspegren, 2023). Cloud environments also pose risks of unauthorized access, data breaches, and vulnerabilities in shared infrastructure. To address these concerns, organizations should implement robust ethical frameworks, encryption protocols, secure DevOps practices, regular security audits, and continuous monitoring, ensuring both responsible AI use and data privacy.

7.6. Test environment parity

Maintaining parity between test and production performances is a recurring challenge. Even in cloud setups, discrepancies in configuration, data, and dependency versions cause software to behave differently across environments. Cloud environments are dynamic, which significantly differentiates between development and production, leading to discrepancies that impact the reliability of tests (Vankayalapati, 2025). As cloud applications become more complex and distributed, achieving true environment parity becomes increasingly difficult, potentially leading to unexpected behavior in production. Data security and privacy are also paramount in the cloud, and ensuring that testing environments are secure while still providing realistic scenarios is crucial. Automating testing processes and scaling test environments to handle developing workloads are essential for efficient cloud software development. Cloud-native and serverless architectures introduce new complexities in testing, which also require specialized knowledge and tools to ensure proper functionality (Gade, 2022).

7.7. Security and compliance risks

Automation and cloud-based workflows provide new security and compliance risks. Thus, automated scripts may unintentionally expose sensitive data or misconfigured access controls, such as GDPR. Cloud-based testing environments often provide third-party tools and services, which develop the attack surface (Nguyen, 2022). Implementing secure coding practices and encoded data flows. Maintaining coding practices, encrypted data flows, entry audits, and adherence checks within mechanized pipelines is essential but often overlooked during fast-paced deployments. Misconfigured permissions, exposed credentials in test scripts, and a lack of encryption lead to data access. However, regulatory compliance becomes harder to enforce across distributed cloud systems. Security testing is embedded in CI/CD workflows and provides automated cloud systems (Reddy, 2022).

7.8. Cost overruns

Cloud services operate on pay-as-you-go models, which can lead to unexpected expenses if resources are not managed properly. Automated test scripts that run repeatedly and unused virtual machines can accumulate costs quickly. Cost overruns in software engineering are a common problem with many projects, with their initial budgets (Sandaa, 2021). Unexpected technical challenges and resource constraints can significantly impact the budget. Thus, cost monitoring tools exist, but integrating them exceeds their allocated resources. Cost monitoring tools exist, but integrating them into DevOps pipelines and enforcing usage policies remains a crucial challenge.

7.9. Emerging trends in software engineering practices

Machine learning is developing embedded testing tools to enable intelligent test case selection and flakiness detection. This minimizes test redundancy and speeds up fault localization. Testing is being combined earlier in the progress lifecycle. IaC tools, for instance, Terraform and AWS CloudFormation, allow teams to automate environment provisioning, ensuring consistency and speeding up advancement across development and production.

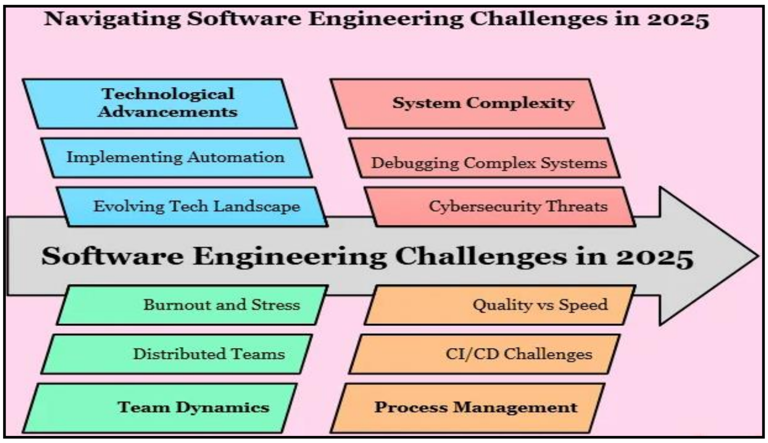

Fig. 3: Navigating Software Engineering Challenges.

Source: (Vadapalli, 2025).

Figure 3 illustrates key software engineering challenges, contrasting traditional limitations with cloud-driven approaches that enhance scalability, automation, and adaptability. Modern software engineering faces challenges like compromised code quality, complex debugging, and mounting technical debt, which are crucial in cloud-based, automated environments. Thus, poor code quality leads to bugs and maintenance issues, while debugging distributed systems like microservices increases resolution time and impacts user performance. Technical debt, if ignored, hinders scalability and slows future development. Solutions implementing CI/CD pipelines, automated testing, and TDD, and refactoring (Klotins et al., 2022). Tools like Jenkins, SonarQube, GitHub Actions, and Sentry help maintain quality and observability. Addressing these challenges ensures stability, efficiency, and maintainability in the code-to-cloud engineering journey.

7.10. Advancements in automated testing technologies

Testing is an essential aspect of software development that ensures dependability. However, creating and implementing all-inclusive test cases is particularly challenging in complex systems with many interdependencies. Engineers must offer various testing methodologies, involving unit testing and integration testing, to guarantee that all key points of the data are meticulously assessed (Giray, 2021). Moreover, the challenges of testing are intensified by the need for uninterrupted testing in agile ecosystems. Thus, including feedback from stakeholders in the development process is necessary for generating successful software. This task is complicated and challenging, as feedback differs extensively in terms of clarity and relevance. Engineers sift through feedback to recognize actionable understandings while also considering the technical implications with stakeholders to ensure that their needs are adequately dealt with without jeopardizing the integrity of the software engineers' performance. Another key development is the rise of low-code/no-code testing tools, which permit non-technical consumers to design automated tests employing visual workflows (Roksana, 2025).

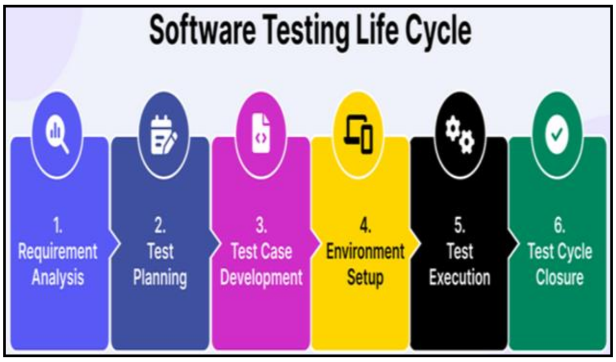

Fig. 4: Software Testing Life Cycle.

Source: (Priya Sr Faculty, 2018).

The Software Testing Life Cycle contrasts traditional manual stages with cloud-driven automation, emphasizing faster execution, accuracy, and continuous integration benefits. AI-powered tools are conquering traditional challenges in software testing by employing machine learning to simplify test creation. These tools offer such things as faster test execution and improved quality guarantee to predict prospective problems before they arise. AI grows test coverages and recognizes intricate patterns that individual testers might neglect, leading to higher-quality software releases (Priya Sr Faculty, 2018). AI-powered testing tools execute large numbers of test cases in a short time, thus decreasing the time necessary for testing. This speeding up allows development teams to meet tight deadlines and accelerate time to market. Moreover, AI-powered tools spontaneously adjust test scripts in response to modifications in the programs, which saves time. This flexibility maintains continuous testing even as the program evolves. Therefore, AI assists in achieving a high test scope by implementing a large number of assessments that involve border cases in less time.

7.11. Comparative insights: traditional vs cloud-driven approaches

Cloud computing has crucially reshaped software engineering workflows by offering scalable infrastructure and on-demand resources. Traditional engineering relied heavily on localised environments and manual configurations that often led to delays and inconsistencies in software development (Asghar, 2024). In contrast, cloud platforms like AWS, Google Cloud provide seamless environments in which teams develop, test, and deploy software continuously. One of the most notable impacts of cloud adoption is the acceleration of DevOps practices. Thus, the use of containerization in cloud environments has made software deployment more consistent across different stages of prosperity. However, this shift also provides challenges like data security, latency issues, and dependency on internet connectivity. Nonetheless, the transition to cloud-based engineering workflows is largely viewed as positive, enabling organizations to innovate faster and manage resources efficiently (Yarram, 2023). Therefore, this shift provides challenges like data security, latency issues, and dependency on internet connectivity. Nonetheless, the transition to cloud-based workflows is largely viewed as positive for innovating faster and having more efficient resources. Additionally, cloud-based test labs now offer on-demand infrastructure.

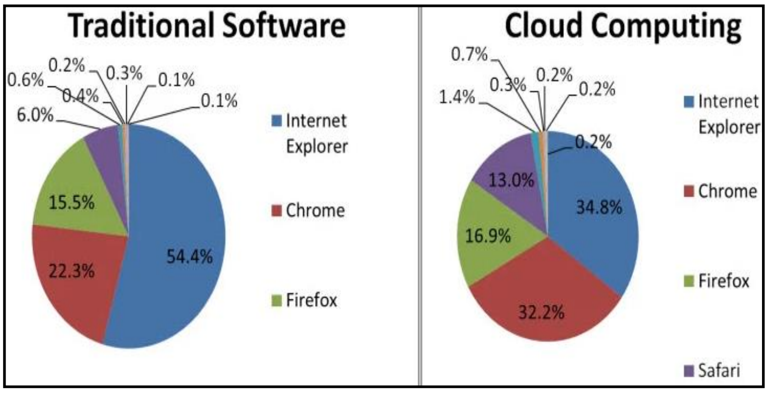

Fig. 5: Traditional Software and Cloud Computing.

Source: (Aztechit, 2023).

Figure 5 compares traditional software and cloud computing, highlighting scalability, efficiency, and collaboration advantages of cloud-driven approaches. The above pie charts compare browser usage in traditional software versus cloud computing environments. In traditional software, Internet Explorer dominates with 54.4%, followed by Chrome at 22.3% and Firefox at 15.5%. Thus, under cloud computing, Chrome leads at 32.2%, with Internet Explorer dropping to 34.8% and Firefox rising to 16.9% (Aztechit, 2023).

7.12. Synthesis of findings and future implications

The synthesis of findings and key insights from the research reveals how cloud technologies and automation are transforming software engineering practices. Cloud computing develops scalability, collaboration, and accessibility, and streamlines workflows across geographically dispersed teams. Automation, particularly in testing, improves accuracy, reduces manual effort, and accelerates delivery timelines (Babar, 2024). Therefore, challenges like integration complexity, security concerns, and the need for specialized skills persist. Thus, for future implications, institutions must invest in reskilling teams to leverage the full capability of cloud-indigenous resources and automation frameworks. Additionally, enhancing robust governance and security frameworks implies that embracing these advancements is not optional but essential for remaining competitive (Kolade et al., 2025). Therefore, future software engineering will rely heavily on cloud-native architectures (software designed specifically to run in cloud environments, using microservices and containers) and intelligent automation. Organizations should invest in training for advanced tools that provide AI-driven testing and DevOps practices. Future research should focus on AI-driven testing in niche domains such as IoT and the development of standardized protocols for multi-cloud environments to strengthen resilience and sustain innovation.

8. Conclusion

This study emphasizes the transformative impacts of cloud technologies and automation on software engineering and testing practices. Cloud computing enables greater scalability and efficiency, while automation develops accuracy and reduces manual effort in testing environments. Together, these advancements increase agile workflows and faster deployment. However, long-term sustainability of cloud-based workflows and ethical implications of AI-driven testing remain important research gaps. The merging of these technologies signals a shift toward DevOps, continuous implementation, and AI-assisted evaluation as standard approaches. For institutions to remain competitive, investing in workforce reskilling and robust infrastructure is fundamental. Moreover, future studies should explore how emerging technologies such as quantum computing and edge computing may influence cloud and automation adoption. The findings suggest that cloud-based systemization is not just a trend but a foundational shift in how software is created, tested, and sustained. Future studies should persist in examining AI-driven innovations, ethical aspects, and hybrid models to ensure sustainable and resilient software engineering solutions in a changing digital landscape. Future research should also examine hybrid models, AI ethics, and sustainable practices to ensure resilient software engineering solutions in an evolving digital landscape.

References

[1] Aggarwal, K., Bögel, E., La Rosa Betancourt, M., Collier-Wright, M., Brake, M., Dörr, O., Yalcin, B.C., Richard, A. and Olivares Mendez, M.A. (2022). *Enabling Elements of Simulations Digital Twins and its Applicability for Information Superiority in Defence Domain*. [online] orbilu.uni.lu. Available at [Accessed 19 June. 2025].

[2] Aiyenitaju, K. (2024). The Role of Automation in DevOps: A Study of Tools and Best Practices. *Theseus.fi*. [online]. Available at http://www.theseus.fi/handle/10024/876681 [Accessed 19 June. 2025].

[3] Arvanitou, E.-M., Ampatzoglou, A., Chatzigeorgiou, A. and Carver, J.C. (2021). Software engineering practices for scientific software development: A systematic mapping study. *Journal of Systems and Software*, 172, p. 110848. Available at [Accessed 19 June. 2025]. https://doi.org/10.1016/j.jss.2020.110848

[4] Asghar, A. and Anazagasty, E. (2024). AI-Driven Infrastructure Automation: Leveraging AI and ML for Self-Healing and Auto-Scaling Cloud Environments. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, [online] 5(1), pp.32–43. Available at [Accessed 19 June. 2025]. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I1P104.

[5] Aztechit (2023). *Key Differences Between Cloud Computing vs. Traditional*. [online] www.aztechit.co.uk. Available at: https://www.aztechit.co.uk/blog/cloud-computing-vs-traditional [Accessed 19 June. 2025].

[6] Babar, Z. (2024). A Study of Business Process Automation with Devops: A Data-Driven Approach to Agile Technical Support. [online] 04(04), pp.01-32. Available at https://doi.org/10.63125/3w5cjn27.

[7] Balogun, A.Y. (2025). Strengthening Compliance with Data Privacy Regulations in U.S. Healthcare Cybersecurity - East Asian Archive. *Go2articles.com*. [online] Available at http://authors.go2articles.com/id/eprint/1654/1/Balogun1812025AJRCOS130092.pdf [Accessed 19 June. 2025].

[8] Boscain, S. (2023). *AWS Cloud: Infrastructure, DevOps techniques, State of Art.* [online] web thesis.biblio.polito.it. Available at: https://webthesis.biblio.polito.it/26672/ [Accessed 19 June. 2025].

[9] Constantin Aliferis and Simon, G. (2024). Lessons Learned from Historical Failures, Limitations and Successes of AI/ML in Healthcare and the Health Sciences. Enduring Problems, and the Role of Best Practices. *Computers in health care*, pp.543–606. Available at [Accessed 19 June. 2025]. https://doi.org/10.1007/978-3-031-39355-6_12.

[10] Constantino, K., Souza, M., Zhou, S., Figueiredo, E. and Kästner, C. (2021). Perceptions of open-source software developers on collaborations: An interview and survey study. *Journal of Software: Evolution and Process*. Available at https://doi.org/10.1002/smr.2393.

[11] D'Onofrio, D. (2023). CI/CD Pipeline and DevSecOps Integration for Security and Load Testing. *OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information)*. [online]. Available at https://doi.org/10.2172/2430395.

[12] Gade, K.R. (2022). Cloud-Native Architecture: Security Challenges and Best Practices in Cloud-Native Environments. *Journal of Computing and Information Technology*, [online] 2(1). Available at: https://universe-publisher.com/index.php/jcit/article/view/3 [Accessed 19 June. 2025].

[13] Giray, G. (2021). A software engineering perspective on engineering machine learning systems: State of the art and challenges. Journal of Systems and Software, p.111031. Available at https://doi.org/10.1016/j.jss.2021.111031.

[14] Giray, G. (2021). A software engineering perspective on engineering machine learning systems: State of the art and challenges. *Journal of Systems and Software*, p.111031. Available at https://doi.org/10.1016/j.jss.2021.111031.

[15] Klotins, E., Gorschek, T., Sundelin, K. and Falk, E. (2022). Towards cost-benefit evaluation for continuous software engineering activities. *Empirical Software Engineering*, 27(6). Available at https://doi.org/10.1007/s10664-022-10191-w.

[16] Kolade, T.M., Aideyan, N.T., Oyekunle, S.M., Ogungbemi, O.S., Dapo-Oyewole, D.L. and Olaniyi, O.O. (2025). Artificial Intelligence and Information Governance: Strengthening Global Security, through Compliance Frameworks, and Data Security. *SSRN Electronic Journal*. [online]. Available at https://doi.org/10.2139/ssrn.5044032.

[17] Koneru, K. (2025). Optimizing CI/CD Pipelines for Multi-Cloud Environments: Strategies for AWS and Azure Integration. *The Eastasouth Journal of Information System and Computer Science [online], 2(03), pp.* pp.288–310. Available at https://doi.org/10.58812/esiscs.v2i03.534.

[18] Lawal, K. (2023). Strategic Approach to Facilitating Communication and Collaboration Between Software Development Teams and End Users in the Cloud. *International Journal of Advances in Engineering and Management (IJAEM)*, [online] 5, p.348. Available at.

[19] Masselos, J. (2025). *Experts Predict Tech Job Market Trends for 2025*. [online] Toggl Blog. Available at: https://toggl.com/blog/tech-job-market [Accessed 19 June. 2025].

[20] Nelimarkka, P. (2023). *Emerging low-code/no-code paradigm: Evaluating adoption, opportunities, and cyber security challenges in the infor- mation technology sector*. [online] www.theseus.fi. Available at: https://www.theseus.fi/handle/10024/798496 [Accessed 19 June. 2025].

[21] Nguyen, M. and Debroy, S. (2022). Moving Target Defense-Based Denial-of-Service Mitigation in Cloud Environments: A Survey. *Security and Communication Networks*, 2022, pp. 1–24. Available at https://doi.org/10.1155/2022/2223050.

[22] Parmar, T. (2025). Implementing CI/CD in Data Engineering: Streamlining Data Pipelines for Reliable and Scalable Solutions. *SSRN Electronic Journal*. [online] Available at https://doi.org/10.2139/ssrn.5190570.

[23] Pedamkar, P. (2019). *Automation Testing? | Automation Testing Applications and Tools*. [online] EDUCBA. Available at: https://www.educba.com/automation-testing/ [Accessed 19 June. 2025].

[24] Priya Sr Faculty (2018). *Software testing steps - Software Testing- Software development life cycle*. [online] Available at: https://www.h2kinfosys.com/blog/software-testing-steps/ [Accessed 19 June. 2025].

[25] Ramesh, A., Pradhan, V. and Lamouche, H. (2021). Understanding and Analysing Resource Utilization, Costing Strategies and Pricing Models in Cloud Computing. *Journal of Physics: Conference Series*, 1964(4), p. 042049. Available at https://doi.org/10.1088/1742-6596/1964/4/042049.

[26] Reddy, V. and Jayaram Immaneni (2022). Optimizing CI/CD in Healthcare: Tried and True Techniques. *International Journal of Emerging Research in Engineering and Technology*, [online] 3(2), pp.28–38. Available at https://doi.org/10.63282/3050-922X.IJERET-V3I2P104.

[27] Roksana Jahan Tumpa and Naeni, L. (2025). Improving decision-making and stakeholder engagement at project governance using digital technology for sustainable infrastructure projects. *Smart and Sustainable Built Environment*. Available at https://doi.org/10.1108/SASBE-10-2024-0451.

[28] Sandaa, K. (2021). What Causes Schedule Overrun in Complex Engineering Projects? *Ntnu.no*. [online]. Available at no.ntnu:inspera:78072291:25569516 [Accessed 19 June. 2025].

[29] Stornelli, A., Ozcan, S. and Simms, C. (2021). Advanced manufacturing technology adoption and innovation: A systematic literature review on barriers, enablers, and innovation types. *Research Policy*, 50(6), 104229. Available at https://doi.org/10.1016/j.respol.2021.104229.

[30] Surbhi Kanthed (2025). From Code to Cloud: The Role of GitOps, GitHub, and GitLab in Modern DevOps. *Journal of Technological Innovations*, [online] 6(1).

[31] Sutharsan Saarathy, Suresh Bathrachalam and Rajendran, B. (2024). Self-Healing Test Automation Framework using AI and ML. *International Journal of Strategic Management*, 3(3), 45–77. Available at https://doi.org/10.47604/ijsm.2843.

[32] Team DigitalDefynd (2025). AI in Software Testing [5 Case Studies] [2025]. [online] DigitalDefynd. Available at: https://digitaldefynd.com/IQ/ai-in-software-testing-case-studies/?utm_source=chatgpt.com [Accessed 11 Sep. 2025].

[33] Thorne, J. (2023). *What is a CI/CD pipeline?* [online] blog.openreplay.com. Available at: https://blog.openreplay.com/what-is-a-ci-cd-pipeline/ [Accessed 19 June. 2025].

[34] Vadapalli, P. (2025). *Redirect Notice*. [online] Google.com. Available at: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.upgrad.com%2Fblog%2Fsoftware-engineering-

challenges%2F&psig=AOvVaw3Tl5LvaMBCyukN5wXz9Yxv&ust=1750443895705000&source=images&cd=vfe&opi=89978449&ved=0CBcQjhxqFwoTCKibyuaN_o0DFQAAAAAdAAAAABAE [Accessed 19 Jun. 2025].

[35] Vankayalapati, R.K. (2025). Composable Infrastructure: Towards Dynamic Resource Allocation in Multi-Cloud Environments. *SSRN Electronic Journal*. Available at https://doi.org/10.2139/ssrn.5121215.

[36] Wong, R.Y., Chong, A. and Aspegren, R.C. (2023). Privacy Legislation as Business Risks: How GDPR and CCPA are Represented in Technology Companies' Investment Risk Disclosures. Proceedings of the ACM on human-computer interaction, [online] 7(CSCW1), pp.1–26. Available at: https://doi.org/10.1145/3579515.

[37] xcede (2025). *Cloud Engineer vs Software Engineer, Making the right career choice in 2025 | Xcede*. [online] Xcede Technology Recruitment Specialists. Available at: https://www.xcede.com/blog/cloud-engineer-vs-software-engineer-making-the-right-career-choice-in-2025 [Accessed 19 June. 2025].

[38] Yarram, S. and Rao, B.S. (2023). *Predictive Test Automation: Shaping the Future of Quality Engineering in Enterprise Platforms*. [online] Ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5132329 [Accessed 19 June. 2025].