



A SURVEY ON QUALITY ASSURANCE APPLICATION IN AGILE SOFTWARE DEVELOPMENT TOOLS, TECHNIQUES, AND TRENDS

Sachin Manekar¹

¹ Assistant Professor, Department of Computer Sciences and Applications, Mandsaur University, Mandsaur
Sachin.manekar@meu.edu.in

Abstract: The development-world issue of the ability of these teams around the world to deliver quality software at a high rate of speed in a rapidly digitised world, with customer needs ever-evolving. The survey has a glance at the practices, tools, and trends in quality assurance (QA) of Agile software development teams. Explore the main principles of Agile, including the iterative nature of the development process, flexibility, and feedback, and ensure that QA is not treated as a post-development stage. The paper has covered a broad range of QA tools within the Agile environment. In addition to conventional automation tools (e.g., web and API testing), the paper also notes that AI-based testing solutions are becoming increasingly popular and can, e.g., detect visual regressions and perform predictive analytics. Furthermore, discuss the fact that QA methods are not only going to become the next fads in the software market but also expanding to the manufacturing and automotive, as well as aerospace sectors, being informed by such phenomena as Industry 4.0, IoT, and supply chain traceability. A list of recent empirical research publications shows that Agile, combined with current QA practices, results in a higher defect identification rate, shorter deployment cycles, and improved overall product quality. The survey provides an in-depth understanding of modern QA practices aligned with Agile methodologies for developing reliable software and other high-quality products.

Keywords: Quality Assurance, Agile Software Development, Test Automation, Test-Driven Development, AI-based Testing, Continuous Integration.

1 INTRODUCTION

The software business is a dynamic environment in the modern world, which requires a rapid reaction to constantly evolving needs. Moreover, the contemporary world has been characterized by globalization, and therefore, businesses are no longer limited to a specific location, but have opened up their world to endless opportunities [1]. This increased globalization, mass production and constantly evolving demands in which rapid solutions are required, offers numerous challenges such as the inability of the organization to match the rapid and constantly growing demands [2]. The software program is well aligned with the client's needs through the use of software testing and quality assurance methodologies [3]. There is however a significant difference between the two conceptions. Testing is performed once the application has been developed or in the case of the static testing once the software requirements have been identified and registered in the relevant record [4]. Quality assurance is the term that is used to denote a group of methodologies that can be applied during the life cycle of an application, and it starts with specification of the requirements at the beginning and ends with the delivery of the finished program to the client [5]. The following steps help to guarantee the quality of application and make sure that it corresponds to the established required sets and criteria.

In such a dynamic nature of software development, offering robust quality assurance (QA) is a key to ensuring reliable and well performing systems. The introduction presents the novel approach to enhancing software testing practices by integrating Agile development practices with Artificial Intelligence (AI) [6]. The iterative nature of Agile is a concept that encourages continuous enhancement and flexibility, whereas the AI has more sophisticated features, including test automation, predictive analytics, and smart defect detection. A software quality assurance (SQA) is a collection of procedures that guarantee the properness and thoroughness of software development procedures [7]. SQA is an activity based on the process which evaluates and structures the processes that create software services or products. SQA provides the use of proactive staff functions, which prevent defects in services or goods, and software quality control provides the use of reactive line functions to detect and fix errors. Software testing is in fact called software quality control (SQC). Software testing is performed on products and then they are released to the live environment [8]. This study exploratory path is based on the identification and implementation of such technologies within different Agile systems with a focus on their potential in combination to transform quality assurance processes [9]. The given study aims at delivering a holistic knowledge of the ability of Agile to jointly enhance the quality and efficiency of software development in current development settings through the documentation of practice-based uses and consequences of the integration.

1.1 Structure of the Paper

The structure of the paper is the following: Section 2 deals with an overview of the Agile Software Development (ASD). Section 3 discusses QA tools in agile software development. Section 4 discusses evolving trends in quality assurance testing. Recent studies on quality assurance practices in Agile Software Development are detailed in Section 5. The literature study is present in Section 6. Section 7 summarises the overall conclusion of Agile-based Quality Assurance.

2 OVERVIEW OF AGILE SOFTWARE DEVELOPMENT (ASD)

The term "agile" may indicate either "moving quickly" or "flexible," which is really how it works. Put simply, agile development is a contemporary method that prioritises self-organization, inspection, and monitoring; it also aims to rapidly provide high-quality software while also addressing customer demands and aligning them with business goals [10]. Agile software development is sometimes characterised as incremental and iterative due to the fact that each phase is revisited at various points in the life cycle [11]. The following is a discussion of agile methods:

2.1 Extreme Programming (XP)

Extreme programming is an approach to development that is lightweight. It was successful because it prioritised happy customers over punctual product delivery. As a methodology, Extreme Programming places an emphasis on cooperation and provides a basic, yet sufficient, means of facilitating the growth of group work styles [12]. Figure 1 shows the XP system in action.

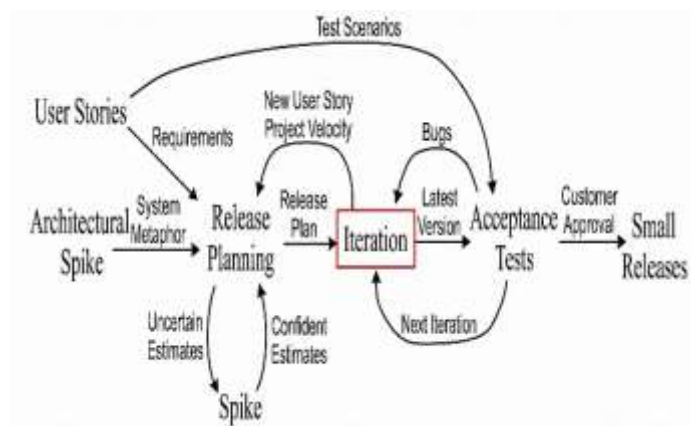


Figure 1: Extreme Programming Methodology

The five principles of XP are as follows: teamwork, honesty, openness to criticism, and simplicity. However, remote teams cannot use XP. As a result of XP's support for shared ownership of the code, no one person owns a module. This speeds up development and helps with mistake and fault detection while coding, which in turn improves software efficiency.

2.2 Scrum

Scrum is another methodology for developing software that is lightweight. Scrum introduced the idea of empirical process control—the only real distinction between theory and practice—instead than the other way around [13]. In Scrum, the projects are broken into small working units that are referred to as the sprints that are normally one to three weeks [14]. Scrum methodology phases are shown in figure 2.



Figure 2: Scrum Methodology

There are three responsibilities in Scrum. The product owner's responsibilities include communicating the product's vision to the development team and acting as a customer advocate by outlining needs and setting priorities. The scrum master acts as a go-between for the team and the product owners on matters of communication. Team members are accountable for completing the tasks, and the seven-person group represents a variety of functional areas.

2.3 Feature Driven Development

The very definition of a "feature" is a "small piece of valuable functionality," so the term. Regular, quantifiable, and functional outputs are the goals of the model-driven software development technique called Feature Driven Development (FDD) [15]. Figure 3 shows the iterative process that this method is designed for usage by big project teams.

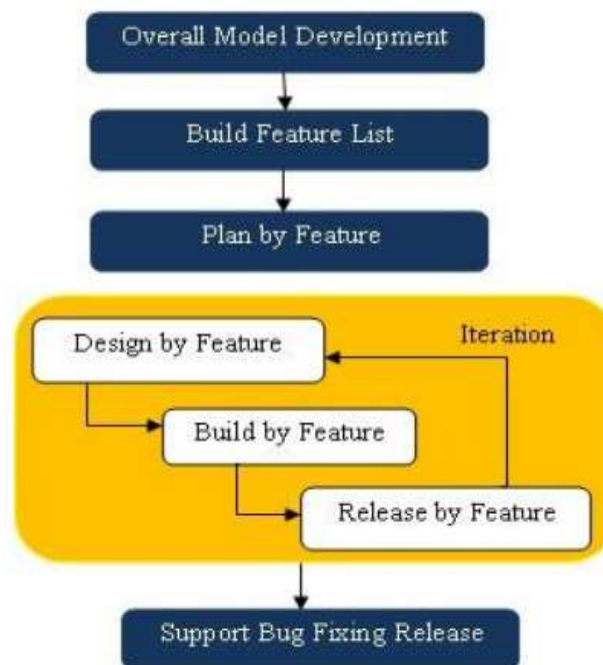


Figure 3: FDD Methodology

3 QA TOOLS IN AGILE SOFTWARE DEVELOPMENT

Quality Assurance (QA) is important in Agile development that involves the delivery of working software in short and repeated sprints. Agile, in contrast to more traditional approaches, emphasizes testing at every stage of development. This helps teams find bugs early on and maintain product quality across the board. To prove this point, there are a number of types of QA tools in use: automation tools, CI/CD tools, and AI-based QA tools[16]. The tools guarantee constant testing, quick feedback, and increased reliability of the developed product.

3.1 Automation tools

Automatic testing is made possible by an automated testing tool. An intrinsic structure determines the baseline for automation for a certain product type [17].

- **Selenium:** One open-source solution for automating tests for all kinds of web and real-time apps is Selenium. Given that Selenium was developed with the intention of automating the user interface of web browsers, it follows that scripts written in programming languages may complete tasks that were previously handled manually [18]. Automated web application testing was the original intent and use of Selenium; however, it can perform any kind of automated interaction.
- **JMeter:** An Apache product, Apache JMeter is a load testing tool for evaluating and assessing the performance of online applications. The Apache JMeter framework is a tool for evaluating web applications. An examination of server performance under excessive demand may be done with Apache JMeter [19].
- **Postman:** Manual and automated API testing are both made much easier using Postman, a popular platform for developing and testing APIs. It is now more important than ever to thoroughly validate APIs due to the widespread adoption of microservice architecture. An essential tool for QA, Postman provides an easy-to-use interface for submitting requests, checking replies, and automating testing processes [20].

3.2 CI/CD tools

CI/CD software One of the first continuous integration and continuous delivery (CI/CD) solutions was Jenkins, an open-source automation server. The loosely designed nature of Jenkins and broad set of plug-ins enabled development teams to automate different points of the software development lifecycle [21].

- **Jenkins:** Continuous integration tool: Jenkins is a tool that helps developers to create software that is more reliable and predictable. Maintaining a consistent version of the public code requires regular contributions of updated code to the central repository from developers [22].
- **GitLab CI/CD:** GitLab CI/CD provides a combined package of dynamic environment management features, which makes it possible to publish the code in a continuous flow with the help of automated testing, rollback, and quick cycles of reiteration. The popularity of GitLab in the automation and DevOps environment can be explained by its customization, cloud-native architecture, and in-depth integration into the Docker and Kubernetes environment [23].
- **GitHub Actions:** GitHub Actions is one of the well-known tools to apply CI/CD, and it is a service offered by GitHub that enables developers to automatize their development processes [24]. GitHub Actions has the ability to build, test and deploy automatically to assist with the development workflows.

3.3 AI-based QA tools

Assurance of software quality AI is a rapidly growing area of software development that is making a big splash. It utilizes the power of artificial intelligence (AI) technologies like ML and NLP to automate testing, expand test coverage, and improve the accuracy of test findings [25].

- **Testim:** The Tricentis company acquired Testim, a software testing platform that provides numerous white papers and data sheets available in PDF format that talk about test automation, AI in testing and product capabilities.
- **Applitools:** Applitools uses Visual AI to automate visual testing, making sure that apps work properly across all devices and browsers. Utilising artificial intelligence, it is able to spot visual inconsistencies that more conventional methods may overlook [26].
- **MABL:** MABL ensures the decentralised implementation of learnt rules and can encode critical global information into latent states during training. At the top level, MABL learns an agent's global latent state; at the bottom level, it uses this information to learn the agent's latent state [27].

4 EVOLVING TRENDS IN QUALITY ASSURANCE TESTING

Various sectors rely on quality assurance (QA) testing to ensure the reliability, efficiency, and overall awesomeness of their goods and services. The principles and techniques of QA have long been associated with software development, but they now have many other potential uses as well [28]. Some new tools and techniques for quality assurance testing in contexts other than software have emerged recently:

4.1 Manufacturing and Industry 4.0

A prime example of how quality assurance testing has progressed from its software roots to become an integral part of product assurance is the manufacturing industry. Industry 4.0 has had significant impacts on manufacturing because it incorporates digital technology, automation, and data-driven processes. This has increased testing to QA to other areas [29]. QA testing processes are evolving dynamically in this environment in order to align with the purpose of Industry 4.0. The objective is to ensure that the items which come out of the manufacturing facility are of the best standards.

4.2 Automotive Excellence

Quality assurance testing has also been critical in the automobile sector for achieving product safety, dependability, and performance. Every aspect of a contemporary automobile, from its software-based systems to its components and manufacturing methods, must undergo rigorous quality control. The automobile industry has relied on crash testing as a quality control measure for quite some time [30]. In the motor industry, however, the range of tests in modern QA procedures has increased. This is done in order to ensure that the car can withstand the harsh conditions of normal usage. The car manufacturing sector is particularly dependent on quality assurance in a supply chain besides testing of vehicles. The automotive supply chain is adopting new advanced technologies, including RFID (Radio Frequency Identification) and blockchain to achieve higher levels of traceability and transparency. This lets out the chances of using poor quality parts in production.

4.3 Achieving Aerospace Excellence

QA testing is one of the most important activities since the aerospace industry is known to be very precise and reliable. It is essential that aerospace projects be made to secure the safety of the passengers and crew members because of the high costs involved. This is one point in which there is no error margin [31][32]. Continuous testing during the QA is a complicated procedure with numerous aspects within the aerospace industry. This includes testing the structures, materials and systems. Some of these parts include wings,

fuselages, landing gears and control surfaces. The multifaceted nature of aircraft requires a lot of simulations and redundancy checks during QA testing. The primary objective is to make sure that the key systems are able to continue functioning as usual even when there is a failure.

5 AGILE-BASED SOFTWARE QUALITY ASSURANCE TECHNIQUES

The goal of quality assurance (QA) in an Agile development environment is to make software testing more efficient, effective, and adaptable by applying Agile concepts and practices [33]. This is executed in the following way:

5.1 Agile Principles and Testing Integration

- Iterative Development: The iterative method makes sure that this testing becomes a part of each development cycle and the quality of the software can be continuously and regularly evaluated [34].
- Continuous Feedback: This feedback loop allows detecting and fixing defects early, and minimizing the chances of severe problems appearing further into the development cycle.
- Adaptive Planning: This flexibility guarantees that the application can remain high-quality even when user needs or project scopes change.

5.2 Testing Practices within Agile Frameworks

- Test-Driven Development (TDD): This is to make sure that the code is of the required standards and assists in ensuring high code quality throughout its existence [35].
- Behavior-Driven Development (BDD): It includes composing test scenarios using plain language, and thus becoming easier to understand by developers, not to mention the non-technical stakeholders, which fits well with the Agile emphasis on user stories and requirements.
- Continuous Integration (CI) and Continuous Deployment (CD): This is due to the fact that the defects could be detected and resolved quickly due to this frequent release cycle, enhancing the overall quality of the software [36].

5.3 Collaboration and Communication

- Cross-Functional Teams: This cooperation promotes better communication, assures the alignment of testing with development requirements, and allows understanding the requirements and possible problems more easily.
- Daily Stand-Ups: The meetings give testers a chance to discuss their problems, receive real-time response and liaise with developers to rectify any flaws at an early stage.

5.4 Automation and Tools

- Test Automation: Reduced manual testing labor and rapid iteration are two additional benefits of automated testing.
- Tools and Frameworks: CI/CD, issue tracking, and test management systems streamline testing and provide valuable metrics for test coverage and quality.

5.5 Quality Metrics and Continuous Improvement

- Defect Tracking: Test coverage, defect resolution time, and defect density are some metrics that may be used to measure the efficacy of QA and to direct future testing endeavours.
- Retrospectives: Teams benefit from this method because it allows them to fix problems that arise during the sprint and improve their testing tactics on the fly.

5.6 Agile development and quality assurance factors

Numerous parameters within agile methodologies, when correctly applied, can ensure the quality of the product. In summary, certain of these parameters are presented in Table 1 [37].

Table 1: Software quality assurance factors [38]

Compatibility	Compatible software refers to the capacity to operate and function harmoniously without any inconsistencies. Elements of compatible software can be seamlessly integrated with other components.
Cost effectiveness	Cost-effectiveness refers to a system's capacity to be concluded within a specified budget.
Correctness	System performance in accordance with established specifications.
scalability	The capacity of hardware to maintain operational performance despite alterations in system volume or size.
Functional stability	The term "software functionality" refers to how well a piece of software meets both explicit and implicit needs when used in a certain way.

Efficiency	Efficiency is defined as the capacity to optimise the utilisation of all available system resources.
Maintainability	Software must be capable of evolving and adapting to better meet the requirements of new users.
Rate of Delivery	High delivery rates to customers and constructive feedback on software contribute to improved software quality.
Reusability	Software components should be developed to enable their utilisation in other software applications.
Verifiability and Validation	Verifiability and validation refer to the process of ensuring the accuracy and effectiveness of the developed system.
Integrity	Capacity to safeguard information against unauthorised access.
Extendibility	Flexibility is to adapt to new conditions and changes in a system.
Ease of use	The client is actively involved in the development process and gives continual input in agile approaches, which results in a product that is easier to use.

6 LITERATURE REVIEW

This section offers a concise synopsis of the information in Table 2 and a thorough overview of the research titled "Quality Assurance Practices in Agile Software Development."

Bao (2025), gives the process of quality assurance, which is divided into three major processes, namely, planning process, execution process and summary process, and each process contains several activities. Through the application of this process in software projects of city commercial banks, SQA breaks the conventional practice of traditional quality assurance, supervises and guarantees the whole process of project and project quality from all directions and perspectives. SQA makes a complete digital measurement assessment on the execution status of software projects in real time, and evaluates the projects from the schedule, quality, problems, evaluation, risk and scale. SQA reports the project status to relevant leaders and stakeholders every week, and coordinates and handles the problems and risks found in time. Through the above work, SQA has greatly improved the efficiency and quality of software projects [39].

Chhatria *et al.* (2025), applies quantitative analysis, using time-series data and survey responses from software development teams to evaluate key performance indicators associated with each trend. The analysis, conducted using SPSS software, examines correlations between the adoption of modern methodologies and outcomes such as reduced development time, enhanced security, and overall software quality. The results have shown that Agile and DevOps practices can be effective in shortening the deployment cycle, whereas the integration of machine learning can facilitate predictive analysis and enhance the detection of errors and minimise the maintenance requirements. The relevance of this study in practice is the importance of adopting the new practices to promote innovation, increase project success rates, and guarantee strong software products. These insights aim to support software companies and developers in adopting strategic trends for sustainable and effective engineering solutions [40].

Galindo-Francia and Aucchuasi (2024), Customer satisfaction was achieved during testing as a consequence of an improvement in application review, reduction of failures, and prevention of errors. Hence, the MAC approach may be used across the organisation. It is a procedure that includes good practice principles, making it suitable for use in future projects. The technique reduced five metrics: customer satisfaction, amount of testing mistakes, total process cost, time for error review, and delivery time. Overall, the MAC technique was useful in enhancing the project's quality assurance process, which led to ongoing development [41].

Pathiraja (2024), investigate software quality assurance (SQA) methods used by Sri Lankan software businesses registered with the BOI for both Waterfall and Agile IS projects. In addition, the paper can help software businesses registered with the BOI in Sri Lanka with their Waterfall and Agile IS projects by identifying important variables in SQA processes, as well as problems, SQA practices, and useful advice [42].

Deshpande *et al.* (2023), the software sector in Bangalore, India, and the ideas and quality standards used for software projects, along with the processes for keeping these standards up to date. The prevalence of software development practices that adhere to the CMMI standards is investigated in this study. Find out how many businesses are measuring up to global standards and how many are assessing the quality of their projects. What found shows that some Bangalore-based companies are using software quality assurance methods [43].

Tokdemir and Uğuz (2022) use text mining to uncover the popular subjects that researchers are interested in by exploring the research done in the ASD domain over the past 20 years. Eight areas explored further in the ASD field as a consequence of this study's findings. As far as popularity, funding, and citations go, the top three are software development process, agile development, and team communication [44].

Table 2: Summary of recent study for Quality Assurance Practices in Agile Software Development

Author	Research Focus	Key Insights	Challenges	Limitations
Bao (2025)	The study examines a structured SQA process	The SQA model improves project quality through real-	Continuous full-cycle monitoring and	Findings relate only to city commercial banks,

	applied in commercial bank software projects.	time digital evaluation and weekly reporting.	coordination among stakeholders remain demanding.	limiting applicability. wider
Chhatria et al. (2025)	The research evaluates Agile, DevOps, and ML integration using quantitative methods.	Modern methodologies reduce development time, improve security, and enhance software quality.	Adoption requires significant training, organizational change, and technical readiness.	Results depend on survey data, creating potential response bias and limited representation.
Galindo-Francia & Aucahuasi (2024)	The study focuses on implementing the MAC methodology to improve testing and QA processes.	The methodology reduces errors, costs, review time, and increases customer satisfaction.	Applying the MAC approach consistently across different teams can be challenging.	Findings are based on a single company, limiting generalization to other contexts.
Pathiraja (2024)	The research identifies SQA practices in Waterfall and Agile IS projects in Sri Lanka.	Early defect detection enhances reliability, compliance, and long-term customer retention.	Differences in SQA needs between Agile and Waterfall create implementation difficulties.	The study covers only BOI-registered companies in Sri Lanka, restricting broader relevance.
Deshpande et al. (2023)	The study investigates CMMI-based quality standards in Bangalore software companies.	Many organizations follow CMMI guidelines to meet international quality expectations.	Maintaining compliance with global standards requires resources and skilled personnel.	The findings are limited to the Bangalore region and may not reflect other locations.
Tokdemir & Uğuz (2022)	The research explores ASD trends over two decades through text-mining analysis.	Themes such as development process, agile development, and team communication come up frequently.	Rapid evolution of Agile topics makes long-term categorization difficult.	The study uses only academic sources, excluding real-world industry practices.

6.1 Research gap

There are still many blanks in the extensive literature on Agile software development quality assurance approaches. Available studies are mostly specific to the industry or region or organization, which makes it hard to generalize their study as it can be seen in the summarized studies presented in Table 2. The majority of the works mention the gains made with the help of the structured SQA models, current methodologies, or special frameworks, but fail to present the actual evaluation of the variety of Agile settings. Also, the literature does not provide in-depth information about the integration of emerging technologies, including automation, AI-based QA tools, and predictive analytics, within Agile processes at the consistent basis. The issues associated with the scalability, cross-team coordination, and the adaptation of QA practices in a hybrid (Agile-Waterfall) environment are also not properly investigated. This also brings in the necessity of conducting wider and multi-context studies that explore the practicality, effectiveness and sustainability of QA strategies in different organizational and technological environments.

7 CONCLUSION AND FUTURR WORK

The evolution to Agile approaches has fundamentally altered the approach to quality assurance of software development. QA activities e.g. test automation, CI and CD, TDD, BDD, cross-functional collaboration are directly engaged in the iterative sprints, defects are detected and fixed as soon as possible, teams are able to accommodate requirement changes swiftly and the quality of the code and the product can be maintained at a high level. The use of AI-based testing tools is also a magnificent assist to the process because it presents the opportunity to test more areas, be more precise in predicting defects, and consume less manual labor. Moreover, the effects of these QA practices are extending to other software sectors like manufacturing, automotive, and aerospace which also use the same quality assurance paradigm during the Industry 4.0 and IoT transformations. Combined with this shift, this indicates that QA is becoming more of a proactive, collaborative component of development and production processes and is essential to the trust, safety, and customer satisfaction guarantee, which cuts across the sectors.

Further research can contemplate in detail the issues of widely implementing AI-powered QA tools, especially in terms of the quality of training data, compatibility of different platforms, and the increase of maintenance. Studies over time measuring metrics such as defect density, deployment frequency, and post-release failures in different Agile teams will be instrumental in determining the long-term effects of integrated QA, which are currently unquantified. Besides that, investigating the QA practice for the new paradigms like microservices, serverless architectures, and continuous delivery in heavily regulated industries will uncover the scalability and flexibility of present-day QA strategies

REFERENCES

- [1] G. Waja, J. Shah, and P. Nanavati, "Agile software development," vol. 5, no. 12, pp. 73–78, 2021.

- [2] R. Hasan and N. U. Eisty, "Documentation Practices in Agile Software Development : A Systematic Literature Review," 2023.
- [3] S. P. Kalava, "Enhancing Software Development with AI-Driven Code Reviews," *North Am. J. Eng. Res.*, vol. 5, no. 2, pp. 1–7, 2024.
- [4] A. Bhanushali, "Ensuring Software Quality Through Effective Quality Assurance Testing : Best Practices and Case Studies," vol. 26, no. 26, pp. 1–18, 2023.
- [5] V. Kumar, "Quality Assurance And Quality Control," no. 11, pp. 2357–2368, 2024.
- [6] W. Ajayi and M. Gbadegesin, "Implementing Quality Assurance in an Agile Software Development Process," vol. 7, no. 9, pp. 1951–1958, 2022.
- [7] B. R. Cherukuri, "Maintenance of Web Development Standard for Multiple Devices with Serverless Computing through Cross Browser Affinity Using Hybrid Optimization," in *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, IEEE, Feb. 2024, pp. 1855–1859. doi: 10.1109/IC2PCT60090.2024.10486499.
- [8] K. Patel, "An Analysis of Quality Assurance Practices Based on Software Development Life Cycle (SDLC) Methodologies," *J. Emerg. Technol. Innov. Res.*, vol. 9, no. 12, pp. g587–g592, 2022.
- [9] F. Azzam, "Quality Assurance in Agile Software Development - Literature Review," 2022. doi: 10.13140/RG.2.2.35304.75522.
- [10] L. S. Awasthi, A. K. Rai, and K. S. Awasthi, "Agile Software Development Issues : A Review," vol. 3, no. 10, pp. 2000–2011, 2016.
- [11] V. Prajapati, "Advances in Software Development Life Cycle Models : Trends and Innovations for Modern Applications," *J. Glob. Res. Electron. Commun.*, vol. 1, no. 4, pp. 1–6, 2025.
- [12] M. Yasvi, "Review On Extreme Programming-XP," 2019.
- [13] M. Menghnani, "Modern Full Stack Development Practices for Scalable and Maintainable Cloud-Native Applications," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 2, pp. 1206–1216, 2025, doi: 10.5281/zenodo.14959407.
- [14] J. D. S. Pinto and S. Leme, "Analysis of project management principles with the Scrum framework in systems development : a case study in a public organization," vol. 21, no. 2, pp. 1–17, 2024.
- [15] S. Rahmad, K. Sofi, and J. Shadiq, "Selection of Feature Driven Development (FDD) Model in Agile Method for Developing Information System of Mosque Management," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 4, pp. 127–136, 2022, doi: 10.47709/cnahpc.v4i2.1469.
- [16] S. Garg, "AI-Driven Innovations in Storage Quality Assurance and Manufacturing Optimization," *Int. J. Multidiscip. Res. Growth Eval.*, vol. 1, no. 1, pp. 143–147, 2020, doi: 10.54660/IJMRGE.2020.1.1.143-147.
- [17] A. Bhanushali, "Impact of Automation on Quality Assurance Testing : A Comparative Analysis of Manual vs . Automated QA Processes," vol. 26, no. 26, pp. 39–54, 2023.
- [18] R. S. Jyoti Devi, Kirti Bhatia, "International Journal of Advanced Research in Computer Science and Software Engineering," vol. 7, no. 5, pp. 855–862, 2017.
- [19] D. A. S. C. Ashwin K Chandrasekhar, "Comparative Analysis Of Load Testing Tools," vol. 9, no. 6, pp. 758–764, 2021.
- [20] A. Rani and R. Nair, "Journal of Artificial Intelligence , Machine Learning and Data Science Postman for API Testing : A Comprehensive Guide for QA Testers," 2022.
- [21] S. T. Makani, "The Evolution Of Cidc Tools In Devops From Jenkins To Github Actions," vol. 13, no. 2, pp. 166–174, 2022.
- [22] P. Narang, "Automated Continuous Deployment of Software Projects with Jenkins through DevOps-based Hybrid Model," 2023.
- [23] E. Journal and P. K. Koppanati, "Implementing Dynamic Environments with GitLab CI / CD," vol. 9, no. 12, pp. 175–179, 2022.
- [24] V. N. June, A. R. Faqih, A. Taufiqurrahman, J. H. Husen, and M. K. Sabariah, "Empirical Analysis of CI / CD Tools Usage in GitHub Actions Workflows," vol. 3, no. 2, 2024.
- [25] S. A. Khan, N. T. Oshin, M. Musfique, M. Nizam, and I. Ahmed, "AI based Software Testing," pp. 1–20, 2024.
- [26] R. S. Narendar Kumar Ale, "Available online www.jsaer.com Cloud-based Automation Testing and Tools Leveraging AI and ML," vol. 10, no. 2, pp. 230–233, 2023.
- [27] A. Venugopal and S. Milani, "MABL : Bi-Level Latent-Variable World Model for Sample-Efficient Multi-Agent Reinforcement Learning," 2023.
- [28] A. Bhanushali, "Evolving Trends in Quality Assurance Testing : A Comprehensive Review of Methodologies and Tools," vol. 26, no. 26, 2023.
- [29] S. Zadjali, A. Ullah, and B. Obrenovic, "Impact of Industry 4.0 in Manufacturing Sector," *Int. J. Manag. Sci. Bus. Adm.*, vol. 7, pp. 25–33, 2021, doi: 10.18775/ijmsba.1849-5664-5419.2014.76.1003.
- [30] S. K. D. S. Grover, "Flaky Test Automation and Mitigating Test Crashes in Agile Releases," *Int. J. Comput. Exp. Sci. Eng.*, vol. 11, no. 03, pp. 5887–5897, 2025.
- [31] G. I. Pop *et al.*, "Process Integration and Quality Enhancing Aerospace Industry Efficiency and Sustainability : Process Integration and Quality Management in the Context of Industry 4 . 0," 2023, doi: 10.20944/preprints202310.1532.v1.
- [32] S. K. Tiwari, "Quality Assurance Strategies in Developing High-Performance Financial Technology Solutions," *Int. J. data Sci. Mach. Learn.*, vol. 05, no. 01, pp. 323–335, 2025, doi: 10.55640/ijdsml-05-01-26.
- [33] P. Barahate, S. Atul, P. Bhangare, S. Nivrutti, P. Jadhav, and V. Kailas, "Enhancing Software Testing Quality Assurance Using Agile Development Environment and Artificial Intelligence With Research Travelogue," pp. 464–468, 2024, doi: 10.48175/IJARST-19443.

- [34] F. Temole, "Agile Integration in Software Development : Principles , Practices , and Challenges," vol. 11, no. 1, pp. 18–29, 2025.
- [35] J. Yu, S. Zhu, J. Zhang, Y. Chen, N. Wu, and Y. Mei, "An Agile Testing Framework of Four Quadrants An Agile Testing Framework of Four Quadrants," 2021, doi: 10.1088/1742-6596/1792/1/012004.
- [36] Pooja Chandrashekar, "A Survey of Tools, Techniques, and Best Practices: CI/CD Integration in DevOps Workflows," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 1366–1376, Jul. 2023, doi: 10.48175/IJARSCT-11978V.
- [37] M. A. Subih *et al.*, "Comparison of agile method and scrum method with software quality affecting factors," *Int. J. Adv. Comput. Sci. Appl.*, 2019, doi: 10.14569/ijacsa.2019.0100569.
- [38] Mousei Mahdi, "Review on Role of Quality Assurance in Waterfall and Agile Software Development," *J. Softw. Eng. Intell. Syst.*, vol. 5, no. 2, pp. 90–97, 2020.
- [39] W. Bao, "Software Quality Assurance Process and Engineering Practice of City Commercial Banks," in *2025 4th International Conference on Computer Technologies (ICCTech)*, 2025, pp. 74–78. doi: 10.1109/ICCTech66294.2025.00023.
- [40] B. Chhatra, H. Dharua, S. K. Tarai, and S. Panda, "Emerging Trends in Software Engineering: Implications for Development and Efficiency," in *2025 International Conference on Emerging Systems and Intelligent Computing (ESIC)*, 2025, pp. 13–18. doi: 10.1109/ESIC64052.2025.10962758.
- [41] J. Galindo-Francia and W. Auccahuasi, "Implementation of Best Practices based on the MAC Methodology for Software Development Quality Assurance," in *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2024, pp. 1727–1733. doi: 10.1109/IDCIoT59759.2024.10467386.
- [42] A. L. C. D. Pathiraja, "Software Quality Assurance Practices Towards Waterfall and Agile Information Systems Projects of BOI Registered Software Companies in Sri Lanka," in *2024 9th International Conference on Information Technology Research (ICITR)*, 2024, pp. 1–4. doi: 10.1109/ICITR64794.2024.10857716.
- [43] M. V. Deshpande, P. A. Soitkar, D. R. Tripathi, and S. B. Agarmore, "Ensuring Web Application Quality: The Role of Software Testing as a Form of Quality Assurance," in *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, 2023, pp. 1–6. doi: 10.1109/ICTBIG59752.2023.10456277.
- [44] G. Tokdemir and S. Uğuz, "Research Trends in Agile Software Development," in *2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, 2022, pp. 1–5. doi: 10.1109/ETCEA57049.2022.10009873.