International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
e-ISSN: 2455-2585
Volume 10, Issue 10, October-2024

A Review on Software Quality Assurance (QA): Emerging Trends and Technologies

Kishan Patel[1]
[1]*Independent Researcher, kpatelscholarnetwork@gmail.com*

***Abstract*—Software Quality Assurance (QA) is essential for delivering reliable and high-performing software systems. As software complexity and the demand for high-quality products grow, advanced QA practices become increasingly important. This study explores recent advancements in QA, focusing on the evolution of QA processes, the impact of early QA interventions, and the integration of emerging digital technologies. It delves into the use of QA audits as educational tools, the application of agile methodologies, and specific challenges such as face detection in social media. The research highlights several key findings: early QA efforts significantly reduce bug counts and overall bug-fixing efforts; QA audits can enhance educational outcomes and improve software quality through pedagogical approaches; and agile methodologies can effectively improve software quality by adapting to changing requirements and continuous feedback. Additionally, the study introduces a QA system for face detection in Arabic social media, demonstrating high accuracy and contributing valuable insights to computer vision and social media analysis. The study concludes with recommendations for refining QA practices and identifies directions for future research to address ongoing challenges and leverage technological advancements.**

***Index Terms*—Software Quality Assurance, QA Processes, Digital Technologies, Agile Methodologies, QA Audits.**

I. INTRODUCTION

Quality assurance is utilised by both testers and quality assurance specialists in the IT sector and is an essential component of the development process. Ensuring quality and reliability are complementary concepts. First and foremost, dependability is a promise of improved cybersecurity, dependability, and failure prevention. When a software system is deemed a "high confidence" or "high integrity system", and its failure could have extremely negative consequences, the overall warranty—which covers hardware, software, and human factors—is the most important quality requirement to ensure the system's main functionality[1].

The goal of software testing and quality assurance is to ensure that the software program is of a high standard and satisfies customer requirements. However, there is a significant distinction between these two concepts. Testing takes place after the creation of the application or, in the case of static testing, following the definition and documentation of the software requirements in the relevant document. From the initial requirements gathering to the final handoff to the customer, quality assurance is an ongoing process that guarantees an application of high standards throughout its development [2].

Furthermore, automation ensures consistency in testing operations, which helps to reduce the impact of human errors. On top of that, it makes it possible to run tests next to each other and streamline regression testing. Implementing automation in software testing leads to the delivery of high-quality software with fewer bugs in the end [3][4]. The primary goal of using automated software testing is to accomplish certain tasks. Examining the many testing tools used in the business, learning effective methods for creating automated test cases, and becoming an expert at scripting and executing these tests efficiently are all part of this[5].

Software development is always a challenging process; adding new features and maintaining quality becomes much more of a challenge as software grows in size. You may strive to enhance the quality of software in a testing environment. Many companies invest over half of their time and money into software testing due to quality concerns.

This study explores various technologies in software quality assurance (SQA) to ensure high-quality software delivery. It examines the evolution of QA processes, the impact of early QA efforts, and the role of digital technologies. The study also investigates QA audits as educational tools, agile methodologies, and domain-specific QA challenges like face detection in social media, aiming to offer insights and recommendations for enhancing SQA practices:

- Demonstrates through a simulation model how early QA efforts can significantly reduce the number of bugs and overall bug-fixing efforts, providing a quantitative basis for emphasising QA activities in the early phases of software development.
- Highlights an application and monitoring of quality standards within the Bangalore software industry, revealing the adherence levels to CMMI guidelines and the overall impact on software quality, thus serving as a case study for quality assurance practices in a specific region.
- Introduces the concept of using QA audits as a pedagogical tool to enhance students' learning of software quality assurance techniques, suggesting a positive correlation between audit-based learning and the ability to develop quality software products.

- Explores the implementation of agile methodologies in software development, identifying factors such as employee behaviour and mutual relations that critically influence the success of agile practices and providing practical insights for improving software quality through agile practices.
- Develop a quality assurance system for face detection in Arabic social media using the heart detection technique, achieving high accuracy rates and creating a publicly available dataset, thus contributing to a field of computer vision and social media analysis.

A. Structure of this paper

The outline of the paper is as follows: Section II covers the evolution of Software Quality Assurance (QA). Section III details digital technologies for quality assurance. Section IV examines the recent trends in testing and quality assurance. Section V reviews recent QA trends. Section VI presents a literature review and identifies research gaps, and VII offers recommendations for conclusions and future work.

II. OVERVIEW OF SOFTWARE QUALITY ASSURANCE

Software Quality Assurance (QA) has evolved significantly since its inception. In the early days of software development, QA was primarily focused on manual testing and debugging, with little formal structure or methodology. The concept of QA gained prominence in the 1970s and 1980s with the advent of structured programming and the establishment of formal software engineering practices. This era saw the introduction of systematic testing techniques, like unit testing, integration testing, and system testing[6].

Software QA is a methodical process that checks for compliance with software standards and processes and reviews internal control systems. It plays a crucial role in the software development lifecycle and project management[7]. In keeping with the specified time and financial limitations, the process's goal is to guarantee compliance with standards, lessen risk, evaluate internal controls, and enhance quality.

Software QA is an ongoing process that includes every step of a project's software development life cycle. Its usefulness should extend beyond its measurement of the final result or its use at the very end of software development. It has to begin at the outset of the project and continue all the way until the program is decommissioned or terminated. SQA is a continual procedure and evaluation since quality cannot be added to a final product at this point; it can only be patched. Provide the foundational knowledge of software quality and quality assurance in this section[8].

- **Quality**

Software engineering differs from other engineering fields, like manufacturing, in that quality is not limited to established criteria but rather should be adapted to meet the unique needs of every customer and the standards of the organisation. "Meeting requirements" and "fitness for use" are the terms used to describe quality in software engineering.

- **Quality management**

The term "quality management" refers to the practice of organising and directing efforts towards the goal of achieving a certain standard of quality. A company's culture must include this practice rather than treat it as an afterthought.

A. Key Concepts and Principles of QA

Software Quality Assurance encompasses a range of activities and practices aimed at ensuring that software meets specified requirements and is free of defects. Key concepts and principles include [9][10]:

- **Quality Control (QC):** QC encompasses the operational procedures and actions used to meet quality standards. The main objective is to find flaws in the finished product by inspecting and testing it [11].
- **Quality Assurance (QA):** The term QA describes the methodical processes put in place within the quality system to guarantee that the software will meet quality standards. Process-oriented tasks like process improvement, auditing, and definition are included.
- **Verification and Validation (V&V):** While validation makes sure the finished product satisfies user needs and expectations, verification makes sure the program complies with requirements at every step of development.
- **Continuous Improvement:** QA emphasises the continuous improvement of processes and practices to enhance software quality. This principle is often implemented through methodologies such as Plan-Do-Check-Act (PDCA) and the use of quality metrics and feedback loops.
- **Risk Management:** Identifying and mitigating risks is a crucial aspect of QA. This involves assessing potential risks to software quality and implementing measures to address them proactively.

B. Traditional QA Methodologies and Practices

Traditional QA methodologies and practices have laid the foundation for modern QA approaches. Some of the most significant traditional methodologies include:

- **Waterfall Model**: Software development in the Waterfall model follows a strict linear progression, with each stage having to be finished before moving on to the next. QA activities in this model are typically concentrated in the testing phase after development is complete.

- **V-Model:** The V-Model is a variation on the Waterfall methodology that places more emphasis on the interdependence between testing and development. We ensure that QA is incorporated throughout the SDLC by having a testing phase that corresponds to each development phase.
- **Spiral Model:** The Spiral paradigm integrates systematic risk assessment and minimisation with iterative development. Every iteration includes quality assurance tasks, with an emphasis on testing driven by risk and frequent feedback.
- **Capability Maturity Model (CMM):** CMM offers a structure for assessing and enhancing software development procedures. With QA procedures integrated at every level, it outlines five stages of maturity, starting with initial (ad hoc processes) and ending with optimised (continuous process improvement).

III. DIGITAL TECHNOLOGIES FOR QUALITY ASSURANCE

Digital technology for quality assurance in the construction sector may be broadly classified into four areas: data collecting, decision-making, collaboration, and transparency and security. This depends on how well the technologies perform when applied to the study.

A. Data Collection Technologies

Throughout QA, data collection is crucial for interpreting results and determining whether a service or product is up to par. Manual data collection during QA in the construction process relied on on-site documentation and in-person observation in the past.

B. Decision-Oriented Technologies

The QA process relies heavily on decision-making about the project's quality after data collection [12]. The quality level that an ongoing construction project has achieved is determined by further examining data that was gathered throughout the QA procedures. It becomes tiresome and difficult to manually understand the vast amounts of data that are generated in a large project.

C. Collaborative Technologies

The many stakeholders and parties engaged in meeting the demands of clients need close cooperation throughout the QA procedures in the construction sector. Construction services must adhere to all government rules, standards, contractual agreements, specifications, and requirements, which may be achieved via collaboration. In order to guarantee that the final product satisfies the specified quality standards, all parties involved in the project exchange relevant data and information [13].

D. Transparency and Security-Related Technologies

Data and information transfers must be conducted in an open and secure manner. Important data pertaining to building site quality checks and inspections must remain unaltered. The ability to make an informed choice and guarantee substantial improvement depends on having access to unmodified data on the quality of goods or services [14][15]. Therefore, security is critical to guarantee the openness of data acquired about services and goods during QA.

IV. RECENT TRENDS IN TESTING AND QUALITY ASSURANCE

The rapidly evolving technological landscape and the ever-increasing complexity of software systems have propelled the software testing and quality assurance (QA) industry to new heights of excellence. Some of the most current tendencies in this industry are as follows:

A. Greater Emphasis on Security and New Technology

New technology trends are becoming more apparent, with mobile testing, cloud computing, and Service-Oriented Architecture (SOA) all seeing notable increases. A whopping 55% of all tests in 2013 were conducted using mobile devices, up from 31% in 2012, according to the 2013–2014 worldwide quality report [16]. Nevertheless, a substantial number of organisations, specifically 56%, lack comprehensive protocols for conducting mobile testing.

B. Higher Levels of Automation

Agile testing teams, a proliferation of Testing Centres of Excellence (TCoEs), and the ever-increasing need for quicker time-to-market have all prompted testing teams to pursue automation. This is something that happens in regression analysis, unit testing, and load testing, among other places.

C. Testing Continuous Integration

According to this procedure, new code is integrated on a regular basis and testing is done in an environment that is similar to production. The tester can assess the efficacy of a modification, find out whether it really satisfies the end user's expectations, and spot issues early on using this form of testing.

D. Testing independent software

An increasing number of businesses rely on specialised QA services to do testing as a result of the growing importance of QA. This is attributable, in part, to the expertise that specialised QA organisations like TCoE capabilities provide [17] [18]. As a result, collaborating with them enables the business to avoid the challenges associated with finding qualified QA personnel and developing a sophisticated QA, both according to the procedures followed and the technology used.

E. Testing in the Cloud (Virtualization and Cloud Computing)

Experts in the field of information technology anticipate a rise in the use of cloud-based testing as cloud computing continues to permeate the IT mainstream, and more than 26 percent of software applications are expected to be hosted on cloud platforms in 2015. This fundamental difference makes cloud infrastructure a simple option compared to other test environments, which may be expensive and difficult to set up and maintain [19][20].

F. Agile Development Environment Testing

Businesses are working hard to create a comprehensive testing plan that integrates the right testing tools and is consistent with the agile development process [21]. Businesses nowadays are increasingly focused on cutting down on testing time before moving on to delivery, which means continuous testing is becoming more prevalent. As a means to consistently provide information on the software's development in relation to its features and the value it offers to businesses, testers will often use testing in production[22][23].

V. LITERATURE REVIEW

This section highlights that most research in this area has focused on using arithmetical approaches to trial challenges related to SQA, particularly in the context of emerging trends and technologies.

In this paper, Nakahara, Monden and Yucel, (2021) provide a software quality assurance simulation model to prove, via quantitative means, the benefits of incorporating QA into software development from the beginning. The suggested model may depict the connection between the quantity of bugs at each stage, the quantity of QA work, the anticipated quantity of detectable bugs, and the quantity of effort to correct problems. By simulating several QA tactics in a certain software development setting, the model helps to find the optimum strategies for improving software quality while reducing the amount of work needed for QA and bug fixing[24].

In this research, Deshpande et al., (2023) presents the concepts and quality standards that are applied to software projects in the Bangalore software industry, as well as how these standards are monitored and maintained. We will examine how many software companies use the CMMI guidelines while producing software in this research. How many companies evaluate the quality of their initiatives, and how many are meeting international standards. The discoveries exhibit that specific organisations in Bangalore are utilising software quality assurance strategies[25].

The hypothesis presented in this work-in-progress (WIP) paper, Wade, Hammond and Anwar, (2023) is that using audits as a pedagogical tool helps students acquire SQA
procedures and will enable them to implement such techniques in real-world projects. They also postulate that there would be a favourable association between students' knowledge of SQA gained via audits and their capacity to create high-quality goods. The purpose of this work is to contribute to the existing research on audits as a teaching tool in an undergraduate SE course at Texas A&M University, a major R1 university, by examining how these audits have been implemented in practice[26].

In this paper, Lochan, (2021) consider using agile approaches to generate software quicker while also improving software quality so that the product meets its quality requirements. Choosing and adapting any approach is determined by the project type and participants. Employee behaviour, mutual relationships, and excitement are all aspects that may have a significant impact on the effectiveness of method implementation. As a result, all of these parameters were assessed using the agile technique. People are unquestionably at the heart of agile approaches. To present the best technique for implementing the agile methodology. Finally, this document investigates every problem impacting software quality assurance and provides all conceivable solutions. Practitioners may find the study's findings valuable in evaluating certain assessment instruments and application quality features[27].

In this paper, Y. Lu et al., (2022) provide a model for evaluating software quality that takes into account four factors: effectiveness, stability, sufficiency, and compliance, all of which are relevant to testing safety-critical software. Some measure aspects are present in every facet. Analytic Hierarchy Process gives each piece a weight. Every one of the components and their relative weights take the features of mission-critical software into account. Applying the fuzzy comprehensive evaluation approach, the evaluation result is achieved by combining the weights and the measure values. The outcome of a software testing quality assessment is affected by the kind of program that is being tested. In light of the software's characteristics, we adjust the values of a few measures to fix this issue. The technique is validated with various real tasks, and the results demonstrate its effectiveness and practicality[28].

In this paper, Nakamichi et al., (2020) provide an MLS quality metric determination process that is requirements-driven. The following are the primary contributions of this paper: (1) defining MLS's unique quality attributes and expanding a quality characteristic of ISO 25010, which specifies traditional software quality, to incorporate them. (2) The aims of the system being developed dictate the quality attributes and measurement techniques for MLS. Hence, a process is needed to identify needs or problems to be decided in the requirements specification in order to derive these. They show the efficacy of the proposed approach by comparing its generated quality criteria and assessment techniques with those recommended by developers based on the research[29].

In this paper, Tsuda et al., (2019) create the Waseda Software Quality Framework (WSQF), a SQuaRE-based comprehensive framework for evaluating software quality that concretises many of the product quality and quality-in-use assessment techniques first outlined in the SQuaRE series. We determined the state of software product quality by using the WSQF on 21 commercially available ready-to-use software products. The final result is a thorough benchmark that takes into account things like the quality measurement values' trends, the interactions between different quality characteristics, the quality-in-use relationship with product quality, and the product contexts inside an application's boundaries in relation to the quality characteristics[30].

In this study, Mohialden, Hussien and Albahadily, (2022) developed a framework that uses the OpenCV Python library and the Haar detection technique to assess the veracity of social media face recognition for Arabic stars. They used a number of testing methodologies on 64 photos to establish and discover quality metrics, such as the error rate of face recognition in the photographs, in our system. Our efforts culminated in the creation of a publicly accessible dataset called Yasmin Nadia Arabic Social Media Images. You may discover this collection on Kaggle and GitHub; it contains faces for Arabic social media celebrations. The developed approach achieves a detection accuracy of above 90% in the majority of cases[31].

Table I: Summary of the related work on SQA

Ref	Methodology	Dataset	Limitations & future work
[24]	Simulation model of SQA to demonstrate QA effort in early phases	PROMISE Repository	Needs empirical validation with real-world data; future work could involve integrating more variables to simulate different software development contexts.
[25]	Concepts and quality standards applied in the Bangalore software industry	NASA Metrics Data Program (MDP) Repository	Limited to Bangalore; future work could expand to other regions and industries to generalise findings.
[26]	Use of audits as a pedagogy technique to improve students' SQA learning	Code Quality Dataset	Work-in-progress; future work involves further empirical studies to establish the effectiveness of audits as a pedagogical technique and its long-term impact on students' industry readiness.
[27]	Exploration of agile methodologies and factors affecting their implementation	Bugzilla and JIRA Repositories	The study is qualitative and might lack generalizability; future work could include quantitative analysis and case studies from various industries.
[28]	Quality evaluation model for safety-critical software using AHP and fuzzy comprehensive evaluation	Practical projects involving safety-critical software	Limited by the subjectivity of weight assignments, future work could refine the model with more comprehensive data and explore automation of the evaluation process.
[29]	Requirements-driven method to determine quality characteristics of MLS	Empirical study involving functional correctness and maturity of MLS for enterprise	Focuses on MLS; future work could adapt the method to other software types and explore its application in different industries.
[30]	SQuaRE-based comprehensive software quality evaluation framework (WSQF)	21 commercial ready-to-use software products	Limited by the scope of evaluated products; future work could involve expanding the framework to cover more products and different software development methodologies.
[31]	Haar detection quality assurance for Arabic celebrity face recognition on social media	Yasmin Nadia Arabic Social Media Images dataset (64 images)	Limited by the small dataset size; future work could involve expanding the dataset and testing with different face detection algorithms to improve accuracy and generalizability.

A. Research gaps

Using a variety of approaches, such as simulation models, audits as instructional tools, agile techniques, quality assessment models, and empirical research, the articles together cover many facets of software quality assurance (SQA). While each research offers insightful recommendations for improving software quality, they all have certain drawbacks. The first paper's simulation model necessitates empirical confirmation, but the second paper's limited generalizability is due to its concentration on Bangalore. There is insufficient proof in the work-in-progress audit document regarding long-term advantages. Agile methodology research is qualitative, requiring quantitative analysis. Although useful, the SQuaRE-based framework and the quality assessment model for safety-critical software need improvement to reduce subjectivity and increase scope. Because it is unique to MLS, the requirements-driven approach for MLS quality characteristics has to be modified for wider use.

VI. CONCLUSION AND FUTURE WORK

Ensuring high-quality software delivery is essential in today's competitive and technology-driven landscape. This study highlights the critical role of Software Quality Assurance (SQA) in delivering software that meets customer expectations and maintains reliability. By exploring traditional and modern QA methodologies, the impact of digital technologies, and recent trends in testing, the research underscores the importance of integrating QA practices throughout the software development lifecycle. The use of automation, agile methodologies, and advanced digital technologies such as cloud computing and security-related innovations represents significant strides in enhancing QA processes.

Future work should focus on further integrating emerging technologies into QA practices to address evolving challenges. This includes expanding research on an application of AI in QA, improving automation tools for complex testing scenarios, and developing methodologies to enhance security and transparency in software quality processes. Additionally, there is a need to explore domain-specific QA challenges and tailor approaches to address unique requirements in various industries. Continued exploration and adaptation will be essential to keep pace with rapid technological advancements and ensure software quality remains robust and reliable.

REFERENCES

[1] D. Carrizo and A. Alfaro, "Quality assurance method in a software development methodology: A practice approach," *Ingeniare*, 2018, doi: 10.4067/S0718-33052018000100114.

[2] I. Ushakova, Y. Skorin, and A. Shcherbakov, "Methods of quality assurance of software development based on a systems approach," in *CEUR Workshop Proceedings*, 2021.

[3] H. M. Ali, M. Y. Hamza, and T. A. Rashid, "A Comprehensive Study on Automated Testing with The Software Lifecycle," *J. Duhok Univ.*, 2023, doi: 10.26682/csjuod.2023.26.2.55.

[4] J. Thomas, "Optimizing Nurse Scheduling : A Supply Chain Approach for Healthcare Institutions," pp. 2251–2259, 2024.

[5] S. R. Thota and S. Arora, "COLLABORATIVE FILTERING AND KNOWLEDGE GRAPHS FOR DATA DISCOVERY," no. 05, pp. 8679–8692, 2024.

[6] K. Patel, "Quality Assurance In The Age Of Data Analytics: Innovations And Challenges," *Int. J. Creat. Res. Thoughts*, vol. 9, no. 12, pp. f573–f578, 2021.

[7] R. Agarwal, P. Nayak, M. Malarvizhi, P. Suresh, and N. Modi, "Virtual quality assurance facilitation model," in *Proceedings - International Conference on Global Software Engineering, ICGSE 2007*, 2007. doi: 10.1109/ICGSE.2007.46.

[8] S. Arora and S. R. Thota, "Ethical Considerations and Privacy in AI-Driven Big Data Analytics," no. May, 2024.

[9] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V Weber, "for Software , Version 1 . 1 Software Engineering Institute," *Office*, 1993.

[10] V. Rohilla, M. Kaur, and S. Chakraborty, "An Empirical Framework for Recommendation-based Location Services Using Deep Learning," *Eng. Technol. Appl. Sci. Res.*, 2022, doi: 10.48084/etasr.5126.

[11] S. Arora and S. R. Thota, "Automated Data Quality Assessment And Enhancement For Saas Based Data Applications," *J. Emerg. Technol. Innov. Res.*, vol. 11, pp. i207–i218, 2024, doi: 10.6084/m9.jetir.JETIR2406822.

[12] T. Olsson, K. Wnuk, and T. Gorschek, "An empirical study on decision making for quality requirements," *J. Syst. Softw.*, 2019, doi: 10.1016/j.jss.2018.12.002.

[13] E. Westerveld, "The Project Excellence Model®: Linking success criteria and critical success factors," *Int. J. Proj. Manag.*, 2003, doi: 10.1016/S0263-7863(02)00112-6.

[14] W. Lu, L. Wu, J. Xu, and J. Lou, "Construction E-Inspection 2.0 in the COVID-19 Pandemic Era: A Blockchain-Based Technical Solution," *J. Manag. Eng.*, 2022, doi: 10.1061/(asce)me.1943-5479.0001063.

[15] M. K. Vinita Rohilla, Sudeshna Chakraborty, "Artificial Intelligence and Metaheuristic-Based Location-Based Advertising," *Sci. Program.*, no. 1, 2022.

[16] V. Rohilla, S. Arora, P. S. Nirwan, and V. Purohit, "Recommendation System Using Location-Based Services," *SSRN Electron. J.*, 2021, doi: 10.2139/ssrn.3833939.

[17] S. Poulding, J. A. Clark, and H. Waeselynck, "A principled evaluation of the effect of directed mutation on search-based statistical testing," in *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 2011. doi: 10.1109/ICSTW.2011.36.

[18] P. Khare, "Enhancing Security with Voice : A Comprehensive Review of AI-Based Biometric Authentication Systems," vol. 10, no. 2, pp. 398–403, 2023.

[19] S. Poulding and H. Waeselynck, "Adding contextual guidance to the automated search for probabilistic test profiles," in *Proceedings - IEEE 7th International Conference on Software Testing, Verification and Validation, ICST 2014*, 2014. doi: 10.1109/ICST.2014.42.

[20] S. A. Pranav Khare*1, "THE IMPACT OF MACHINE LEARNING AND AI ON ENHANCING RISK-BASED IDENTITY VERIFICATION PROCESSES," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 06, no. 05, pp. 1–10, 2024.

[21] M. Katara and A. Kervinen, "Making model-based testing more agile: A use case driven approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007. doi: 10.1007/978-3-540-70889-6_17.

[22] S. A. and S. R. Thota, "Using Artificial Intelligence with Big Data Analytics for Targeted Marketing Campaigns," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 3, pp. 593–602, 2024, doi: DOI: 10.48175/IJARSCT-18967.

[23] S. Arora and P. Khare, "AI/ML-Enabled Optimization of Edge Infrastructure: Enhancing Performance and Security," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, pp. 230–242, 2024.

[24] H. Nakahara, A. Monden, and Z. Yucel, "A Simulation Model of Software Quality Assurance in the Software Lifecycle," in *Proceedings - 22nd IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2021-Fall*, 2021. doi: 10.1109/SNPD51163.2021.9704927.

[25] 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.

[26] P. Wade, T. Hammond, and S. Anwar, "Work in Progress Paper: Learning Software Quality Assurance using Audits," in *Proceedings - Frontiers in Education Conference, FIE*, 2023. doi: 10.1109/FIE58773.2023.10342642.

[27] M. Lochan, "Elevating Software Quality Role Using Angle Methodology," in *Proceedings of International Conference on Technological Advancements and Innovations, ICTAI 2021*, 2021. doi: 10.1109/ICTAI53825.2021.9673334.

[28] Y. Lu, C. Li, S. Wang, Y. Liu, and J. Dai, "A Quality Evaluation Method for Software Testing about Safety-Critical Software," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2022. doi: 10.1109/ICSESS54813.2022.9930258.

[29] K. Nakamichi *et al.*, "Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation," in *Proceedings of the IEEE International Conference on Requirements Engineering*, 2020. doi: 10.1109/RE48521.2020.00036.

[30] N. Tsuda *et al.*, "WSQF: Comprehensive Software Quality Evaluation Framework and Benchmark Based on SQuaRE," in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, 2019. doi: 10.1109/ICSE-SEIP.2019.00045.

[31] Y. M. Mohialden, N. M. Hussien, and H. K. Albahadily, "Quality Assurance of Facial Recognition Software for Arabic Social Media Celebrities," in *4th International Conference on Current Research in Engineering and Science Applications, ICCRESA 2022*, 2022. doi: 10.1109/ICCRESA57091.2022.10352456.