Computer Science, Engineering and Technology
Vol: 2(4), December 2024
REST Publisher; ISSN: 2583-9179
Website: https://restpublisher.com/journals/cset/
DOI: https://doi.org/10.46632/cset/2/4/6

Software Quality Assurance Techniques of Models and Practices

Elayaraja Subbaiah
Teknatio Inc, United States of America.
Corresponding Author Email: elayarajaconnect@gmail.com

Abstract: *Software quality assurance (SQA) is essential for reliable software in fast-paced development. ISO 9001 is used by over one million companies in 170+ countries; CMMI (Capability Maturity Model Integration) defines five maturity levels and integrates with Agile, with CMMI-based improvement showing measurable gains in schedule, cost, product quality, and ROI. Test-driven development (TDD) has been associated with 40–90% lower pre-release defect density in industry studies; behavior-driven development (BDD) improves requirement clarity and collaboration. Automated testing and CI/CD maintain stable codebases; the DORA framework (deployment frequency, lead time, change failure rate, recovery time) links delivery performance to business outcomes. This paper reviews SQA models and practices and summarizes evidence from recent industry surveys and the 2024 DORA report.*

Index Terms: *Software quality assurance (SQA), CMMI, ISO 9001, agile development, test-driven development (TDD), continuous integration (CI), DevOps, quality metrics.*

1. INTRODUCTION

Quality assurance (QA) is central to modern software development. ISO 9001 is used by over one million companies in more than 170 countries and focuses on risk-based thinking, customer satisfaction, and continual improvement. CMMI (Capability Maturity Model Integration), developed by Carnegie Mellon University's Software Engineering Institute, defines five maturity levels—Initial, Managed, Defined, Quantitatively Managed, and Optimizing—and integrates with Agile. CMMI-based process improvement has demonstrated measurable gains in schedule performance, cost performance, product quality, and return on investment. ISO/IEC 25030 (SQuaRE) provides a quality requirements framework for systems and software. Waterfall, Agile, and DevOps offer different QA views. Waterfall stresses documentation and defined testing; Agile stresses iterative testing and feedback; DevOps unites development and operations and relies on CI/CD and automated testing for rapid feedback. Strong QA reduces production bugs and shortens release cycles. A quality-centric culture—where developers and managers share responsibility—helps find issues early and keeps QA part of the full lifecycle.
Metrics (defect density, test coverage) guide improvement. Automation speeds regression testing; ML and AI are changing testing; DevSecOps integrates security into QA.

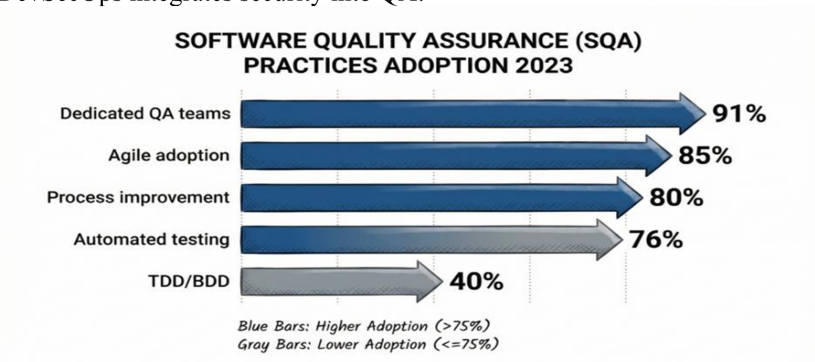

FIGURE 1. SQA practices adoption (2023): dedicated QA teams, Agile, automation, TDD/BDD.

2. LITERATURE REVIEW

Literature on SQA stresses standards and frameworks. ISO/IEC 25010 defines quality characteristics for stakeholder satisfaction; ISO/IEC 25030:2019 (SQua RE) updates the quality requirements framework for systems and software engineering. The main distinction is that ISO 9001 sets requirements for a quality management system while CMMI measures process maturity; integrating ISO 9001:2015 with CMMI 3.0 has shown significant benefits for quality and efficiency in software projects. Defined guidelines reduce defects and improve product quality. Automated testing reduces human error and improves consistency. Context (team, scope) shapes which techniques work best. Combining static and dynamic testing finds more defect types across the lifecycle. DevOps and Agile reshape SQA: CI/CD, rapid feedback, and collaboration. User acceptance testing (UAT) keeps end-user needs central to development. Empirical research and systematic data collection validate which SQA techniques work best in practice.

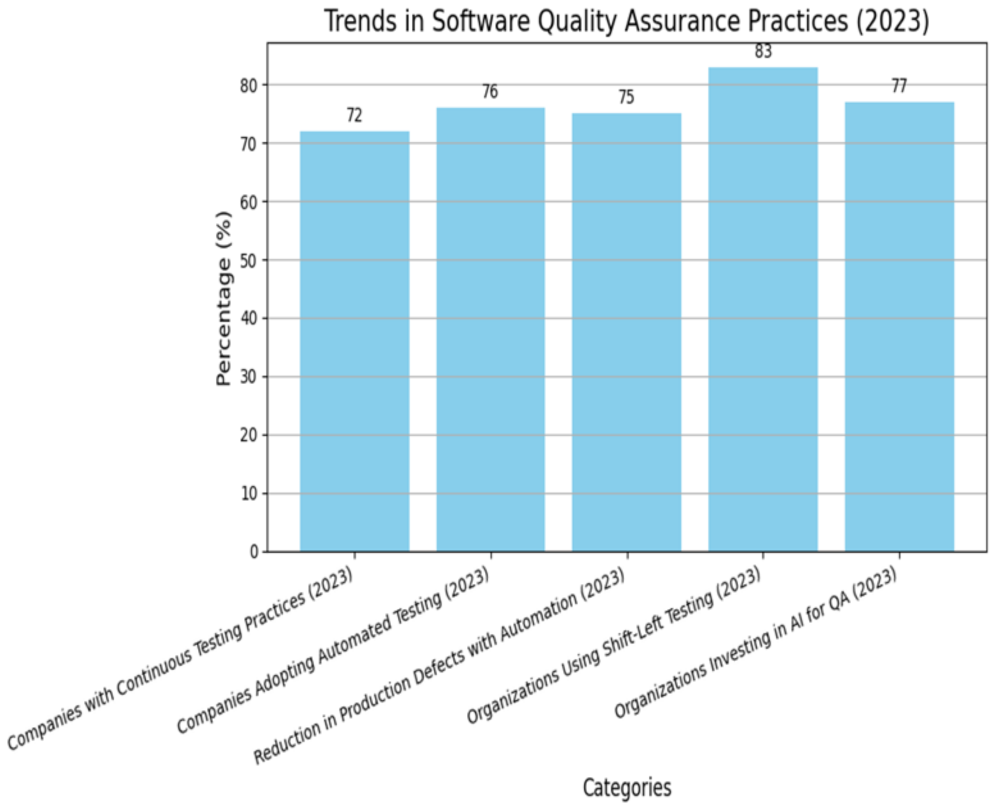

FIGURE 2. Continuous testing, automation, and defect reduction in SQA (2023).

3. METHODOLOGY

Effective SQA requires a systematic methodology that combines models and frameworks; the approach must be reproducible. Evidence supports early testing: a Microsoft Research study with three Microsoft teams and one IBM team found pre-release defect density decreased between 40% and 90% when using TDD compared with similar projects without TDD. A systematic review of 27 studies reported 76% identified increased internal quality and 88% increased external quality with TDD versus Test Last Development. Fixing defects in production can cost up to 200 times more than fixing them in the requirements phase, justifying investment in early testing. A base model (e.g. V-Model, Agile) embeds QA in the SDLC. Qualitative tools (peer reviews, walkthroughs) and quantitative metrics measure and improve quality early. Quantitative metrics (testing, defect density, satisfaction surveys) support quality decisions; combining them with user feedback raises quality. Risk-based QA prioritizes testing by impact; teams identify high-risk areas early and allocate resources accordingly.

FIGURE 3. Risk-based QA methodology and resource allocation.

4. RESULTS

SQA models improve quality and efficiency. The 2024/25 DORA (DevOps Research and Assessment) report surveyed over 32,000 professionals worldwide and links software delivery performance to business outcomes. DORA's four metrics are deployment frequency, lead time for changes, change failure rate, and time to restore service; elite performers deploy on demand, achieve lead times under one day, keep change failure rate at 0–15%, and restore service in under one hour. Continuous integration, loosely coupled teams, and fast code reviews improve delivery and operations; generative culture and psychological safety are among the strongest predictors of performance. Agile within SQA enables early defect detection and lowers late-cycle fix costs. Agile improves adaptability, quality, and satisfaction; it is linked to higher productivity. Compared with Waterfall, Agile projects show fewer critical bugs after deployment; continuous testing and feedback reduce failure risk. Katalon 2023 (3,000+ QA engineers) found 79% report cost savings from test automation; teams with high test coverage are over three times more likely to increase deployments 50–100%. TDD can increase initial development time by 15–35%, but the long-term ROI is positive due to fewer production defects. Automated testing enables repeatable regression tests and faster time-to-market (e.g. ~30% over manual testing). Nearly 90% of organizations prioritize DevOps; integrating SQA with DevOps promotes shared responsibility, fewer defects, and QA as a continuous process. The 2024 DORA report notes that throughput and stability remain steady among high-performing teams despite organizational turbulence, and that teams with high-quality documentation are more than twice as likely to meet reliability targets. Deployment frequency and lead times improve when QA is in the pipeline.

5. DISCUSSION

SQA mixes traditional (Waterfall, V-Model) and agile practices. This mix creates both challenges and opportunities for improving quality. SQA choice affects outcomes; automated testing in agile frameworks finds more defects than manual processes and aligns with CI/CD. The 2024 DORA report cautions against using DORA metrics to compare teams or set inappropriate targets; metrics should guide improvement rather than drive competition. DevOps streamlines workflows and supports collaborative QA; psychological safety and team identity are strong predictors of delivery performance. Organizational maturity shapes SQA: advanced firms use metrics and analytics; others rely on basic practices. SQA actions should fit context. Training and skills support quality culture. Aligning SQA with organizational goals improves performance; clear metrics linked to business goals support tracking and resource decisions. ML and AI can automate regression testing and bug detection; human roles and oversight will need to evolve with training and workforce planning. Ethics matter: SQA must address privacy and security; techniques should meet specs and protect users. Ethical awareness in SQA maintains public trust.

6. CONCLUSION

SQA combines many methods to improve reliability and performance. Modern models use automation and real-time metrics for a proactive approach. Agile and TDD support iterative work, frequent testing, and early defect fixing. Strong SQA improves customer satisfaction and release speed. AI and ML improve testing accuracy and speed; predictive strategies fix issues before they reach users. These technologies support CI/CD pipelines. Combining established practices with new tech and continuous learning keeps quality high. SQA is a system of practices and tools tailored to context. Viewing quality broadly aligns development with business goals. Collaboration among developers, testers, and stakeholders finds issues early; clear communication strengthens complex systems. Organizations need an agile SQA strategy tailored to project and team. Feedback and analysis keep methods relevant as technology changes. QA is continuous; success depends on culture and investment in training and resources. Quality at every stage reduces defect costs and improves competitiveness. Continuous improvement and adapting models to new technology support lasting success in quality management.

REFERENCE

[1]. Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A. S., Nayyef Al-dabbagh, B. S., Fadhel, M. A., et al. (2023). A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. Journal of Big Data, 10, 46. https://doi.org/10.1186/s40537-023-00727-2
[2]. Torous, J., Bucci, S., Bell, I. M., Kessing, L. V., Faurholt-Jepsen, M., Whelan, P., & Carvalho, A. F. (2021). The growing field of digital psychiatry: current evidence and the future of apps, social media, chatbots, and virtual reality. World Psychiatry, 20(3), 318–335. https://doi.org/10.1002/wps.20883
[3]. Nassif, A. B., Abu Talib, M., Nasir, Q., & Dakalbab, F. Q. (2021). Machine learning for anomaly detection: A systematic review. IEEE Access, 9, 78658–78700. https://doi.org/10.1109/access.2021.3083060
[4]. Berrocoso, J. V., Arroyo, M. D. C. G., Videla, C. B., & Morales-Cevallos, M. B. (2020). Trends in educational research about e-learning: A systematic literature review (2009–2018). Sustainability, 12(12), 5153. https://doi.org/10.3390/su12125153
[5]. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6, 60. https://doi.org/10.1186/s40537-019-0197-0
[6]. Alowais, S. A., Alghamdi, S. S., Alsuhebany, N., Alqahtani, T., Alshaya, A., Almohareb, S. N., Aldairem, A., et al. (2023). Revolutionizing healthcare: The role of artificial intelligence in clinical practice. BMC Medical Education, 23(1), 689. https://doi.org/10.1186/s12909-023-04698-z
[7]. du Sert, N. P., Ahluwalia, A., Alam, S., Avey, M. T., Baker, M., Browne, W. J., Clark, A., et al. (2020). Reporting animal research: Explanation and elaboration for the ARRIVE guidelines 2.0. PLoS Biology, 18(7), e3000411. https://doi.org/10.1371/journal.pbio.3000411
[8]. Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access, 8, 21980–22012. https://doi.org/10.1109/access.2020.2970143
[9]. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, 20. https://doi.org/10.1186/s42400-019-0038-7
[10]. Saelens, W., Cannoodt, R., Todorov, H., & Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. Nature Biotechnology, 37(5), 547–554. https://doi.org/10.1038/s41587-019-0071-9
[11]. Kesavan, E. (2022). Empirical research in software testing in fuzzy TOPICS method. REST Journal on Data Analytics and Artificial Intelligence, 1(3), 51–56. https://doi.org/10.46632/jdaai/1/3/7
[12]. Dwivedi, Y. K., Kshetri, N., Hughes, L., Slade, E., Jeyaraj, A., Kar, A. K., Baabdullah, A. M., et al. (2023). Opinion paper: "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. International Journal of Information Management, 71, 102642. https://doi.org/10.1016/j.ijinfomgt.2023.102642
[13]. Polese, M., Bonati, L., D'Oro, S., Basagni, S., & Melodia, T. (2023). Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys & Tutorials, 25(3), 1376–1411. https://doi.org/10.1109/comst.2023.3239220
[14]. Andrews, D. O., Sarmiento, M., Gliklich, R., Leavy, M., Dreyer, N., Jacobs, V., Cutler, M., et al. (2020). Registries for evaluating patient outcomes: A user's guide. Agency for Healthcare Research and Quality. https://doi.org/10.23970/ahrqepcregistries4
[15]. Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M. D., & Hahn, A. (2015). Guide to industrial control systems (ICS) security. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-82r2
[16]. Cao, H. (2021). Comparative analysis of software development methodologies. Applied and Computational Engineering. https://www.semanticscholar.org/paper/8641a6a11e8b36470dd04a6fae85605d49350645
[17]. Kumar, S. (2021). A study of tools and techniques used in software project. International Journal of Scientific Research in Engineering and Management. https://www.semanticscholar.org/paper/fb62e79b6d23b15741ce995510e724ab4d523500
[18]. dos Santos, C. H., & Esteves, B. B. (2022). O impacto do quality assurance dentro de uma empresa: A importância dos testes de software, planejamento, metodologia e automação. Revista FT. https://www.semanticscholar.org/paper/6ef58c753509372531715babb2be4118a161bb15
[19]. Vashishtha, A., & Sharma, S. K. (2019). Estimation of software reliability testing using machine learning techniques. 2025 International Conference on Intelligent Control, Computing and Communications (IC3). https://www.semanticscholar.org/paper/4cc385688ec3ec8acc552728d5b780ddd07b6e9b
[20]. Mollashaik, A. S. (2020). Enterprise test data management: A comprehensive framework for regulatory compliance and security in modern software development. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. https://www.semanticscholar.org/paper/e0e3c22e230ba22be51a8b22265a170fab3adbde
[21]. Krichen, M. (2023). Convolutional neural networks: A survey. Computers, 12(8), 151. https://doi.org/10.3390/computers12080151.