NAVIGATE YOUR NEXT
— in Insurance —

POINT OF VIEW

QUALITY ENGINEERING: BUILDING TRUST FOR DIGITAL TRANSFORMATION

Infosys®
Navigate your next

FOREWORD

Many testing organizations today see a new trend when it comes to ensuring quality – the shift from 'quality assurance' to 'quality engineering'. While 'quality engineering' certainly sounds more inspiring, how is it different from QA? More importantly, how can an organization move from QA to QE? Let us take a closer look to see how it works.

PARADIGM SHIFT IN QUALITY

The activities of organizations with a mature quality engineering practice are remarkably more sophisticated and technology-driven compared to quality assurance. For instance, there is pervasive use of automation techniques such as AI/ML-driven cognitive automation and robotic process automation across the entire software testing lifecycle in test design, test execution, test governance, test data, and environment management. Open-source tools and cloud-first approaches are commonly adopted to enable continuous service improvements and optimize costs. There is strong focus on user experience and customer-centricity through persona-based testing approaches.

QE organizations primarily use AI/ML techniques to pinpoint ambiguities in requirements and identify the impacted test cases that ought to be tested for a particular requirement. AI/ML also helps with code coverage analytics, optimizing test suites, automating monitoring and reporting, and implementing auto healing techniques. Overall, it translates to reduction in testing effort and testing cycle times and increase in quality.

Finally, QE organizations institute mechanisms of instantaneous feedback to identify issues as they originate anywhere in the software development lifecycle and act quickly to address the root cause of the issue.

Dimension	Traditional Quality Assurance Organization	Matured Quality Engineering Organization
Testing approach	Driven by requirements and design	Focuses on user experience and customer-centricity through persona-based approaches
Test design	Manual processes where testers write test cases based on requirements/user stories	Adopts 'test first' approaches such as TDD/BDD to fully automate test design and ensure 100% requirement coverage
Test automation	A script-based approach where automation is limited to execution	Enables end-to-end automation across the entire testing lifecycle. Leverages autonomous testing and robotic or business process-based automation
Test environments	Mostly on-premises	Provisions 'infrastructure as code' test environments on-demand either on cloud or on-premises using a pay-per-use model to optimize cost
Test data	Limited to synthetic or masked production data provisioned through manual or semi-automated processes	Uses fully automated on-demand test data that is integrated into the CT pipeline
Continuous service improvement	Insights are based on human intelligence	Uses AI/ML-driven intelligent automation techniques to enable 'fit for purpose' testing and optimize test effort and cost
Governance	Largely manual	Leverages a fully-automated, continuous testing process with self-healing and automated gating and reporting mechanisms
Workforce	Conversant in testing processes and writing and executing automation scripts	Deploys a sizeable SDET population comprising hands-on developers who can write scripts to completely automate the pipeline
User experience	Manually-driven where business users perform ad-hoc testing	Implements a unified testing process to integrate user experience into the CT pipeline. Leverages techniques like 'crowd testing' to support functional teams in simulating real-world experiences
Tools	Commercial-off-the-shelf (COTS) and on-premises	Open source tools and a cloud-first approach
Organizational structure	Mostly centralized	Product-aligned teams with a shared services QA team for specialized competencies such as performance, security, TDM, and automation

QUALITY ENGINEERING TOOLS THAT BUILD TRUST

What sets 'quality engineering' apart as a practice is how it drives trust in the organization and fosters collaboration. Rather than being linear and testing the software after it is built, quality engineering software development lifecycle (SDLC) stages move in an infinity loop fashion that connects building, coding, planning, testing, deploying, operating, and monitoring releases with a mechanism to provide instantaneous feedback. Thus, achieving quality engineering across the SDLC and QE is the collective responsibility of the entire IT organization, and not just of the testing organization.

Organizations that adopt quality engineering practices and automate the continuous testing pipeline benefit from reduced testing cost of over 30%, reduced testing cycle times by over 40%, test coverage of nearly 100%, high quality with near-zero defects, and accelerated release deployment to production.

BENEFITS OF ADOPTING QUALITY ENGINEERING PRACTICES

30% Reduction in testing cost	>40% Reduction in testing cycle times
100% Test coverage	~0 Defects & high quality

INFOSYS SUCCESS STORIES

Infosys has been implementing quality engineering practices and delivering superior business outcomes for many of its clients. The following success stories demonstrate the effectiveness of this approach.

American financial services firm uses quality engineering to rank among top 5 in Android Play Store – A leading US-based financial services company had embarked on a major digital transformation project and wanted to increase the number of deployments, reduce fallout, and improve the user experience. Infosys built a fully-automated CT pipeline with an AI/ML engine to identify and run test scripts, trigger self-healing mechanisms for issue resolution, and automate reporting to stakeholders.

The quality engineering approach delivered by Infosys helped the client reduce testing time by 60-80%, optimize head count by 40%, achieve up to 95% test automation, and increase the number of releases from four per year to weekly deployments.

Following the transformation, the company's mobile app was ranked among the top 5 in the Android Play Store.

American insurer gets agile with automated pipelines – Infosys helped an American insurance major transform from a legacy-based waterfall model to an agile model of delivery. Some of the key highlights included modernizing mainframe applications to microservices architecture, migrating from on-premises to cloud architecture, creating fully automated pipelines for critical applications, and optimizing the testing workforce by 50% with SDETs.

Adopting quality engineering helped the client reduce defect leakage by 40%, reduce production fallouts by over 30%, improve stability of applications, and reduce technical debt to less than 5%.

Conclusion

Moving to quality engineering helps an organization test early, test fast, and test often, thereby reducing the cost of quality and the testing cycle time significantly. It enables faster deployment and superior user experience, so organizations can build trust in their applications and fast-track their digital transformation programs.

ABOUT THE AUTHOR

Author

Sudhakar Viswanathan

Sudhakar Viswanathan is a Senior Director with the Financial, Healthcare, and Life Sciences (FHL) vertical in Infosys. He has over 25 years of experience and heads the assurance portfolio for insurance clients in North America and financial services clients in Canada. Sudhakar has helped several organizations succeed in their digital and quality engineering transformation journeys.

Infosys®

Navigate your next

For more information, contact askus@infosys.com