Journal of Information Systems Engineering and Management
2025, 10(61s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Platform Thinking in Quality Engineering: Scaling Enterprise Testing through Reusable Service Platforms

Shiva Krishna Kodithyala
Bread Financial, USA

ARTICLE INFO

Received: 07 Sept 2025

Revised: 18 Oct 2025

Accepted: 27 Oct 2025

ABSTRACT

Platform thinking of quality engineering is a paradigm shift in the organization's approach to software testing and quality assurance at the enterprise level. This radical approach reinvents the concept of quality engineering capabilities as standardized and reusable services instead of single project-specific implementations. Organizations build scalable platforms to achieve consistency in quality practices across a wide range of applications by building quality platforms that offer self-service test environments, automation frameworks as services, and integrated observability solutions. The platform model resolves key inefficiencies in classic models, where redundant testing infrastructures and poor methodology ensure huge operational costs. Organizations will gain both flexibility and consistency through the standardization of the core quality capabilities, as well as offering adaptation mechanisms to meet the project-related requirements. This article explores the development of quality engineering into platform-based models, considers the main characteristics of successful quality platforms, discusses the implementation issues, and determines the strategic advantage of regarding quality engineering as a service-based field throughout the enterprise.

Keywords: Quality Engineering Platforms, Self-Service Test Environments, Automation Frameworks, Standardization and Flexibility, Enterprise Quality Transformation

1. Introduction

Organizations today are under growing pressure to create high-quality applications at unparalleled speed and volume. Conventional quality engineering (QE) practices, being project-specific and hence isolated from one another, cannot keep pace with such demands. A shift in thinking toward platform thinking is a more productive approach by envisioning QE capabilities as services that can be reused instead of separate, project-specific processes.

Current enterprise transformation efforts underscore the importance of adopting this strategy. As recorded in Infosys' in-depth report on quality engineering platforms, companies switching to platform-based models of quality have seen a drastic enhancement of their testing effectiveness and overall software delivery abilities [1]. This approach allows development teams to procure quality services on demand, like they would subscribe to cloud infrastructure or other development tools. The platform approach institutes quality engineering as a standard service layer in the enterprise architecture, offering uniform testing practices, automated validation frameworks, and ongoing quality monitoring across various business units and application portfolios.

The shift to quality platforms is more than an act of technology; it redefines fundamentally how organizations think about and execute quality practices. Based on Gartner's research into nascent quality engineering methodologies, platform thinking overcomes key challenges in modern software delivery by developing economies of scale for quality capabilities [2]. This eliminates the conventional model whereby every project team builds independent testing infrastructure, automation frameworks, and quality processes. Rather, they are enterprise assets, standardized and optimized for reuse across contexts. The platform method creates a quality service catalog that can be used by teams throughout the development life cycle, maintaining uniform quality practices without re-creating solutions for every new project.

When properly implemented, quality platforms form a bedrock for the expansion of quality practices across multiple technology environments. The Infosys quality engineering platform framework illustrates how firms can create self-service test environments, reusable automation frameworks, and integrated observability solutions that support both legacy systems and newer cloud-native applications [1]. The convergent solution to quality engineering allows firms to have standard consistency while being technology diverse. The platform model is specifically useful to businesses with complex application portfolios across a variety of technology generations that offer standardized quality strategies that cut across legacy monolithic applications, microservices architecture, and API ecosystems, as well as cloud infrastructure.

2. The Evolution of Quality Engineering

Quality engineering has historically existed within the confines of single projects, with groups creating tailored testing solutions for particular uses. Although effective for a single project, this model generates huge inefficiencies when extrapolated to an entire enterprise. Duplicate test infrastructures, conflicting methodologies, and repeated effort translate into increased expense and inconsistent quality results.

The development of quality engineering mirrors changes in overall software development processes. As companies have moved on from waterfall methodology to agile and DevOps methodologies, quality engineering has also moved on from separate testing phases to integrating quality continuously across the development lifecycle. Traditional ways of software testing tended to form specialized quality teams working independently of development, leading to late feedback and quality bottlenecks. Industry research by Idealink Technology indicates that testing activities usually absorb between 25-40% of overall development efforts, with enterprise organizations often having several overlapping testing infrastructures within business units [3]. This duplication not only adds to the cost of operation but also causes heterogeneous quality standards within the organization's application portfolio.

The shift towards platform-based quality engineering is a natural course of this trend. A study conducted and published in ResearchGate's extensive research on digital transformation suggests that companies moving from project-oriented to platform-based operational paradigms attain higher scalability and consistency in their technology competence [4]. This change directly relates to quality engineering, where platform thinking creates standardized quality services available across various development teams. Platform thinking forms the basis for continuous quality practices woven throughout the software delivery life cycle. By making quality engineering a common service instead of a project-oriented function, organizations achieve economies of scale along with maintaining standardized quality levels across various applications. This change is in line with overall digital transformation initiatives, making quality engineering a strategic driver of faster software delivery as opposed to being a compliance gate or delivery bottleneck.

Metric	Traditional QE (Project-Centric)	Platform-Based QE
Testing Infrastructure	Redundant across business units	Centralized and shared
Quality Standards	Inconsistent	Standardized
Feedback Cycle	Delayed	Continuous
Development Integration	Isolated teams	Integrated throughout the lifecycle
Scalability	Limited (linear with projects)	High (economies of scale)
Delivery Impact	Often a bottleneck	Strategic enabler

Table 1: Evolution of Quality Engineering: From Project Silos to Enterprise Platforms [3, 4]

3. Platform Thinking: A Transformative Approach

Platform thinking redefines quality engineering as a service-based field. It refers to QE capabilities as common, reusable assets that organizations can use across the enterprise. Organizations can standardize testing routines, drive faster delivery cycles, and enhance the overall quality of the products by using quality platforms.

The platform-based model is a paradigm shift in the delivery and consumption of quality capabilities in enterprise settings. Instead of developing independent testing solutions per project, organizations create a centralized quality platform with standardized services delivered through defined interfaces. This type of architectural strategy mirrors the overall industry trends toward service-oriented and microservice architectures. Research conducted on ResearchGate that investigates software quality metrics in agile environments provides evidence that organizations applying standardized quality frameworks to more than one team attain more uniform quality results and considerable decreases in test cycle time than those applying project-specific strategies [5]. The research points to the way standardized quality platforms create uniform measurement structures so that organizations can place quality metrics from various projects side by side and spot systemic opportunities for improvement that are not always obvious within the confines of the project.

Platform thinking reaches beyond technical architecture into organizational change. As Infosys' in-depth study of digital operating models illustrates, the shift towards platform-based capabilities means radical changes in organizational design, governance mechanisms, and delivery practices [6]. This shift directly translates to quality engineering, where companies have to change from dispersed quality teams operating for individual projects to centralized platform teams building and operating enterprise quality services. The platform model sets up quality competencies as a service layer in the enterprise architecture that offers standardized interfaces by which development teams tap into testing environments, automation platforms, and quality insights. This shift involves close monitoring of organizational change management, as groups need to adjust to using quality services instead of developing custom solutions per project. Organizations achieve economies of scale by making quality engineering a platform capability while providing consistent quality practices across various application portfolios.

Capability	Project-Specific Implementation	Platform-Based Service
Test Environments	Custom per project	Self-service provisioning
Automation	Individual frameworks	Standardized service APIs
Test Data	Project-specific datasets	Centralized management
Measurement	Varied metrics	Consistent frameworks
Governance	Isolated	Enterprise-wide
Integration	Manual connections	Standardized interfaces
Team Structure	Distributed QE teams	Centralized platform teams
Resource Model	Dedicated per project	Shared services

Table 2: Quality Engineering Transformation: From Project Assets to Enterprise Services [5, 6]

4. Core Elements of a Quality Engineering Platform

4.1 Self-Service Test Environments

Classic environment provisioning tends to introduce bottlenecks in the development pipeline. Quality platforms use self-service functionality where teams can provision standardized test environments on demand. These environments are consistent across projects without queue times and minimize dependencies on specialized infrastructure teams. The self-service model also allows the development teams to initiate testing sooner in the development cycle, allowing the shift-left testing practices. The modern applications rely on infrastructure-as-code and containerization methods to deliver environment similarity at various development levels.

Studies from Leiden University that examine the adoption of self-service test environments within enterprise organizations illustrate how automated provisioning of the environment significantly minimizes testing bottlenecks alongside enhanced resource utilization [7]. The study illustrates how businesses moving from conventional environment management practices to self-service platforms registered huge gains in terms of environment provisioning times and increased consistency throughout testing phases. This shift allows development teams to trigger testing processes without relying on specialized infrastructure teams, essentially eliminating a key constraint in the delivery pipeline. The self-service style enables the latest continuous integration practices by enabling teams to provision environments on demand using standardized interfaces instead of creating tickets and waiting for manual provisioning.

4.2 Automation Frameworks as Services

Instead of developing automation frameworks from scratch for every project, platform thinking encourages building strong, reusable automation services offering standardized solutions for API testing, UI automation, performance testing, security validation, and accessibility compliance. By exposing these capabilities as services with clearly defined interfaces, organizations minimize implementation time while preserving consistent quality practices among teams.

A wide-ranging report found in Wiley's Software Testing, Verification and Reliability journal illustrates that companies using shared automation frameworks create substantial gains in automation coverage and resource utilization relative to teams building project-specific automation products [8]. The study measures the extent to which centralized automation services eliminate duplicate effort across teams while creating uniform testing practices. Platform-based automation frameworks have standardized components for generic test scenarios so that teams can concentrate on test cases specific to applications instead of framework building. This reduces the technical intensity of test automation and converts the same from being a complex technical challenge needing automation expertise into a standardized service accessible to all QE teams, irrespective of their automation capability.

Capability	Traditional Approach	Platform Approach	Key Benefits
Environment Provisioning	Manual ticket-based	Self-service portal	Reduced provisioning time
Environment Configuration	Manual setup	Infrastructure-as-code	Consistency across projects
Environment Availability	Limited by capacity	On-demand scaling	Eliminates waiting periods
Test Automation	Project-specific	Shared services	Higher automation coverage
Automation Expertise	Required per team	Centralized capability	Democratized automation
Component Reuse	Limited	Extensive	Reduced implementation time
Technology Support	Varies by team	Standardized	Consistent quality practices
Maintenance Effort	Duplicated	Centralized	Reduced operational costs

Table 3: Quality Platform Capabilities: Self-Service Environments & Automation Services [7, 8]

5. Benefits of Platform-Based Quality Engineering

5.1 Scalability

Platforms for quality functions scale very well across many projects and teams without commensurate growth in QE resources. Once core services are in place, new projects can benefit from pre-existing capabilities, minimizing both effort and implementation time. The platform approach shifts quality engineering from a linear resource model, where testing capacity increases directly with the number of projects, into a more lean shared-services model. Evidence in a study published in the Journal of Systems and Software that investigated systematic testing methodologies proves that companies adopting formalized testing processes achieve more efficient testing on various projects [9]. This multi-vocal literature review establishes how systematic testing frameworks allow companies to have consistent quality processes and extend across many projects without corresponding increases in resources. The platform approach makes quality engineering a common service layer, permitting new projects to inherit existing quality capabilities instead of creating tailored solutions for every project.

5.2 Consistency

Standardized testing and quality assurance methodologies build uniform practices within the organization. The uniformity increases reliability, cross-team collaboration, and reduced adherence to internal and external standards. The platform model sets uniform quality languages, processes, and measurements among different development teams, building a basis for enterprise-quality governance. Through the creation of uniform quality interfaces for quality services, organizations ensure uniform implementation of vital quality practices irrespective of team members or technology stack.

5.3 Cost Efficiency

Platform strategies eliminate the wasted effort and avail economies of scale, thus lowering the cost of quality overall. Funds hitherto spent on keeping duplicate testing infrastructures can now be diverted toward improving the platform for quality as well as innovation. Studies by Lab Manager on establishing good business cases prove that bringing redundant technologies together into common platforms generally provides substantial cost savings over having to keep several stand-alone systems [10]. This study is consistent with quality engineering revolutions, whereby companies often find that 30-40% of testing budgets are spent on duplicative automation frameworks and test infrastructures from one project to another. The platform model eliminates this waste through the development of common quality services, which in effect reduces the overall cost of quality and increases the resource use within the enterprise.

Benefit Category	Metric	Traditional Approach	Platform Approach
Scalability	Resource scaling	Linear with projects	Shared service model
	New project onboarding	Requires setup from scratch	Leverages existing capabilities
Consistency	Quality standards	Varies by team	Enterprise-wide standardization
	Methodology	Team-dependent	Unified approach
	Metrics	Project-specific	Comparable across the portfolio
Cost Efficiency	Resource allocation	30-40% on duplicate infrastructure	Consolidated investment
	Maintenance effort	Distributed across projects	Centralized platform team
	Automation framework	Multiple implementations	Single shared framework
	Infrastructure	Duplicated environments	Shared environment services

Table 4: Quantifiable Benefits of Platform-Based Quality Engineering [9, 10]

6. Challenges and Considerations

Although platform thinking has very high advantages, there are a number of challenges that organizations will have to go through during implementation. Throughout the process of making a transition between project-oriented quality practices and platform-oriented ones, it is necessary to plan the transition and help the organization successfully reach the intended results.

Balancing standardization and flexibility represents a primary challenge in quality platform implementation. Organizations must establish standardized quality practices and interfaces while accommodating unique requirements across diverse projects and technology stacks. CTOsync analysis of technology standardization strategies proves that winning platform implementations create apparent boundaries between standardized core capabilities and adaptable extension points [11]. This study determines how organizations can develop effective technology platforms by standardizing basic capabilities and offering adaptation mechanisms for project-based requirements. For quality platforms in particular, this balance is generally achieved through standardizing primary testing procedures, environmental requirements, and quality measures and permitting flexibility in implementation specifics, technology-specific testing practices, and reporting structures. Organizations that succeed in achieving this balance build platforms providing consistent advantages without strict constraints that inhibit functionality in varying technology environments.

Organizational change management poses equally daunting challenges during platform transformation. Moving from project-based quality teams to centralized platform models involves quite radical cultural and structural changes. Project Management Institute research illustrates that successful change management is an essential success factor in transformational programs, with those organizations that adopt holistic change programs achieving much higher success rates than those that concentrate on technical implementation [12]. This study emphasizes how successful platform transformation goes beyond technology deployment to include overhauling cultural resistance, aligning incentive systems, and creating new operating models facilitating platform adoption. For quality platforms in particular, organizations need to overcome resistance from both testing experts who are used to having control of their testing environments and development teams with ingrained testing processes. Effective transformations overcome such challenges with inclusive governance strategies, phased implementation plans, and transparent communication of the benefits of the platform to all stakeholders.

7. Case Study: Enterprise Transformation

A multinational financial services institution applied a quality platform strategy, shifting from project-based testing initiatives to common quality services. This transformation is a complete case study of implementing a platform-based quality engineering solution in a complex enterprise environment. As per a study by Qualitest analyzing digital transformation in banking and financial services, quality engineering platforms are being used as critical enablers for financial institutions undergoing digital transformation while addressing regulatory compliance and security expectations [13]. The report brings to the fore how financial services firms experience different quality challenges, such as extensive compliance needs, complicated integration environments, and increased security issues. Platform-based solutions resolve these issues through the standardization of quality practices with the stringent validation needed for financial systems.

The financial services firm deployed a number of core platform quality components based on industry best practices. These were containerized test environments made available through self-service portals, automatable test frameworks for UI and API testing, centralized test data management capabilities, and unified quality dashboards offering real-time metrics. This deployment was after the architectural styles outlined in Tata Consultancy Services' quality engineering analysis of digital banking recognized platform-based quality strategies as essential pillars for speeding up financial service delivery [14]. The study shows how quality engineering platforms facilitate banks to improve customer experience via digital channels with security and stability requirements that are inherent in banking systems.

The change yielded quantifiable improvements in several aspects of quality engineering performance. In less than 18 months of deployment, the organization achieved 40% improvement in environment provisioning time, 60% reduction in effort spent on automation framework development, 30% increase in defect detection rates, and considerable boosts in release cycles in several business units. These results are consistent with the performance gains reported in studies of Qualitest and TCS on quality engineering revolutions in financial services. The case study shows how platform-based quality engineering sets up a stage for faster delivery in intricate enterprise settings while ensuring the quality standards required in mission-critical financial environments.

Future advancements may integrate AI-driven analytics and self-healing automation into QE platforms, enabling predictive quality and fully autonomous release validation, a natural progression of this platform thinking model.

Conclusion

Platform thinking is a critical change in quality engineering, where the approach to quality engineering is no longer project-oriented but is now delivered as a service across an enterprise. This is a mode of operation that is in line with the new age software development practices, whereby organizations are able to sustain the quality levels whilst addressing the call to deliver at scale much faster. When quality engineering is defined as a shared service layer in the enterprise architecture, the organization will develop economies of scale, eliminate redundant efforts, and implement consistent quality practices within a wide range of application portfolios. The platform model turns quality engineering into a constraint that may be delivered as a source of strategic acceleration of the software delivery. Those organizations that manage to overcome the challenges of implementation, such as balancing flexibility with standardization, handling organizational change, initially investing in an application, and building an effective governance framework, have a chance to provide software of a higher quality in a more efficient way in the whole package of applications. With development practices constantly shifting towards microservice-based, cloud-native, and continuous delivery approaches and models, quality platforms offer the platform they require to tie quality engineering into these new models in a seamless manner.

References

[1] Infosys Limited, "Quality Engineering Platform: Accelerating Digital Transformation,". [Online]. Available: https://www.infosys.com/it-services/validation-solutions/documents/quality-engineering-platform.pdf

[2] Sam New and Hope Warrilow, "Market Guide for Quality Management System Software," Gartner Research, 2024. [Online]. Available: https://www.gartner.com/en/documents/5678455

[3] Idealink, "Understanding the Cost of Software Testing in Software Development: A Complete Breakdown," 2025. [Online]. Available: https://idealink.tech/blog/understanding-software-testing-costs-development-breakdown

[4] Adolfo M. Carreno, "The Evolution from Project-Based to Continuous Digital Transformation: A Strategic Approach to Sustainable Growth," ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/385040861_The_Evolution_from_Project-Based_to_Continuous_Digital_Transformation_A_Strategic_Approach_to_Sustainable_Growth

[5] Charles Ikerionwu and Nwandu Caesar Ikenna, "Quantifying Software Quality in Agile Development Environment," ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/publication/354478280_Quantifying_Software_Quality_in_Agile_Development_Environment

[6] Infosys Limited, "Digital Operating Model for the AI-first Enterprise,". [Online]. Available: https://www.infosys.com/navigate-your-next/digital-operating-models.html

[7] Robert Blinde, "DevOps Unravelled: A Study on the Effects of Practices and Technologies on Organisational Performance," Leiden University, 2022. [Online]. Available: https://theses.liacs.nl/pdf/2021-2022-BlindeRobert.pdf

[8] Yuqing Wang et al., "Improving test automation maturity: A multivocal literature review," Wiley, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/stvr.1804

[9] Hamad I. Alsawalqah, Sungwon Kang, and Jihyun Lee, "A method to optimize the scope of a software product platform based on end-user features," Journal of Systems and Software, Volume 98, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0164121214001861

[10] Sherri L. Bassner, "Building a Successful Business Case Around an ROI Calculation," Lab Manager, 2023. [Online]. Available: https://www.labmanager.com/building-a-successful-business-case-around-an-roi-calculation-30659

[11] CTOsync, "Balancing Standardization and Flexibility in Technology: 13 Examples," 2025. [Online]. Available: https://ctosync.com/qa/balancing-standardization-and-flexibility-in-technology-13-examples/

[12] Project Management Institute, "Enabling Organizational Change Through Strategic Initiatives," 2014. [Online]. Available: https://www.pmi.org/learning/thought-leadership/pulse/enable-change-strategic-initiatives

[13] Qualitest Group, "How Quality Engineering Can Ensure Financial Services Safely Achieve Digital Transformation," 2023. [Online]. Available: https://www.qualitestgroup.com/insights/blog/how-quality-engineering-can-ensure-financial-services-safely-achieve-digital-transformation-ebook/

[14] B Ganesh Seshan, "Quality Engineering in Digital Banking," TCS. [Online]. Available: tcs.com/what-we-do/industries/banking/white-paper/quality-engineering-improve-digital-banking