

Comparative analysis of testing tools for in-house application UAT: Assessing Tricentis, Qyrus, and TestSigma

Sanjeev Kumar Mourya *

Bundelkhand Institute of Engineering and Technology, India.

World Journal of Advanced Research and Reviews, 2025, 26(02), 1607-1614

Publication history: Received on 02 April 2025; revised on 09 May 2025; accepted on 11 May 2025

Article DOI: <https://doi.org/10.30574/wjarr.2025.26.2.1814>

Abstract

This article presents a comprehensive evaluation of three leading testing tools—Tricentis, Qyrus (Quinnox), and TestSigma—for User Acceptance Testing (UAT) of in-house applications. Through a structured Proof of Concept methodology combined with a multi-tiered training program, the article investigated not only technical capabilities but also organizational alignment, knowledge transfer effectiveness, and long-term sustainability factors. The article revealed distinctive strengths across the platforms: Tricentis excelled in enterprise application testing and comprehensive test coverage but required steeper learning curves; Qyrus demonstrated superior cloud integration and intuitive interfaces while providing efficient resource utilization; and TestSigma offered exceptional accessibility for non-technical users through natural language processing capabilities with rapid implementation timeframes. Beyond technical comparisons, the study identified critical success factors for testing tool implementations, including knowledge retention strategies, team autonomy development, and alignment with strategic testing objectives. The article demonstrates that successful testing tool adoption requires balanced consideration of immediate usability and long-term capability building, with implementation approach and training methodology significantly influencing outcomes regardless of tool selection. This article provides valuable insights for organizations seeking to enhance in-house testing capabilities while reducing dependency on external vendors.

Keywords: Test Automation Tools; User Acceptance Testing (UAT); In-House Application Testing; Knowledge Transfer Effectiveness; Testing Tool Evaluation Framework

1. Introduction

The complexity of modern software applications has necessitated rigorous testing processes to ensure quality, reliability, and user satisfaction. User Acceptance Testing (UAT) represents a critical phase in the software development lifecycle, particularly for in-house applications where organizational requirements, workflows, and integration considerations demand specialized attention. As organizations increasingly seek to optimize their testing processes while building internal capabilities, the selection of appropriate testing tools has emerged as a strategic decision with significant implications for efficiency, cost management, and application quality [1].

The digital transformation landscape has accelerated the need for robust testing frameworks that can be managed internally with minimal vendor dependency. According to recent industry data, organizations that successfully implement appropriate testing tools and build in-house expertise can reduce testing costs by up to 30% while simultaneously improving defect detection rates. However, the market offers numerous testing solutions with varying capabilities, learning curves, and integration potentials, making tool selection challenging without structured evaluation frameworks.

* Corresponding author: Sanjeev Kumar Mourya

This study presents a comprehensive evaluation of three prominent testing tools—Tricentis, Qyru (by Quinnox), and TestSigma—through a methodical Proof of Concept (POC) approach designed to assess their suitability for in-house application testing environments. The evaluation process was specifically designed to examine not only the technical capabilities of each tool but also their alignment with organizational requirements, ease of adoption by internal teams, and potential for fostering long-term testing independence.

By implementing a dual-focused strategy that combines rigorous tool assessment with targeted internal training initiatives, this research addresses the gap between tool acquisition and effective utilization—a challenge frequently cited in technology implementation literature. The findings provide valuable insights for organizations navigating similar decisions while contributing to the broader understanding of testing tool evaluation methodologies in contemporary software development environments.

The research questions guiding this study include: (1) How do the selected testing tools compare in terms of functionality, usability, and integration capabilities for in-house applications? (2) What factors influence the successful adoption and utilization of testing tools by internal teams? (3) How can organization's structure evaluation processes to ensure post-implementation success and reduced dependency on external support?

2. Literature Review

2.1. Current trends in software testing automation

Software testing automation has evolved significantly in recent years, with AI-driven capabilities emerging as a dominant trend. Test automation frameworks increasingly incorporate machine learning for test creation, maintenance, and execution [2]. Low-code/no-code testing solutions have gained prominence, allowing non-technical stakeholders to participate in testing processes. Additionally, shift-left and continuous testing approaches have become standard in DevOps environments, enabling earlier defect detection and faster release cycles.

2.2. Previous comparative studies of testing tools

Comparative evaluations of testing tools have primarily focused on technical capabilities rather than organizational fit. The article [1] examined automation tools against standardized metrics but provided limited insights into adoption factors. Kaur and Gupta's framework [2] offers a methodology for tool comparison across functionality, usability, and support dimensions. However, few studies have addressed the critical aspects of knowledge transfer and internal capability building that significantly impact long-term success in tool implementation.

2.3. Established frameworks for tool evaluation

The ISO/IEC 25010 quality model has been adapted for testing tool evaluation by several researchers, providing standardized criteria for functionality, reliability, and usability. The Technology Acceptance Model (TAM) offers insights into adoption factors but requires contextualization for testing environments. The article [3] proposed an extended framework that incorporates organizational considerations alongside technical capabilities, though comprehensive validation across diverse organizational contexts remains limited.

2.4. Gap in research regarding in-house application testing

Despite extensive literature on testing tools, significant gaps exist regarding in-house application testing scenarios. Most studies focus on commercial products or general-purpose applications rather than the unique challenges of testing internally developed systems. Research on building organizational testing capabilities through tool selection and training remains particularly sparse. This gap is notable given the increasing trend of organizations seeking to reduce dependency on external testing vendors while maintaining quality standards.

3. Methodology

3.1. POC design and implementation

The evaluation utilized a structured three-phase POC designed to assess real-world performance. Each tool was tested against three in-house applications representing different complexity levels and technology stacks. Phase one involved basic functionality verification, phase two tested integration capabilities with existing systems, and phase three assessed advanced features and customization options. The POC spanned eight weeks, with equal time allocated to each testing tool.

3.2. Evaluation criteria development

Evaluation criteria were developed through a collaborative workshop involving QA leads, developers, and business stakeholders. The criteria encompassed six dimensions: functionality (test creation, execution, reporting), usability (interface design, learning curve), integration capabilities (with existing tools and workflows), scalability, cost considerations, and vendor support. Each dimension contained 5-8 specific metrics rated on a 5-point scale.

3.3. Training program structure

A tiered training approach was implemented alongside the POC. Initial familiarity training (4 hours) introduced basic concepts and tool interfaces. Intermediate training (8 hours) focused on test creation and execution for simple scenarios. Advanced training (12 hours) covered complex testing scenarios, integration aspects, and troubleshooting. Training effectiveness was measured through practical assessments rather than theoretical knowledge tests.

3.4. Data collection methods

Data collection employed mixed methods including structured observation, task completion metrics, user surveys, and semi-structured interviews. Quantitative metrics included test creation time, execution success rates, defect identification percentages, and support request frequency. Qualitative data captured user perceptions, challenges encountered, and suggestions for improvement. Usage patterns were monitored through tool analytics where available.

3.5. Analysis approach

Analysis combined quantitative scoring against predefined criteria with qualitative thematic analysis of user feedback. A weighted scoring model assigned different importance values to criteria based on organizational priorities. Triangulation of data sources enhanced validity, while regular validation sessions with stakeholders ensured alignment with business requirements. Cost-benefit projections were developed based on POC data to estimate long-term value and total cost of ownership.

4. Tool Profiles

4.1. Tricentis: capabilities, architecture, and market position

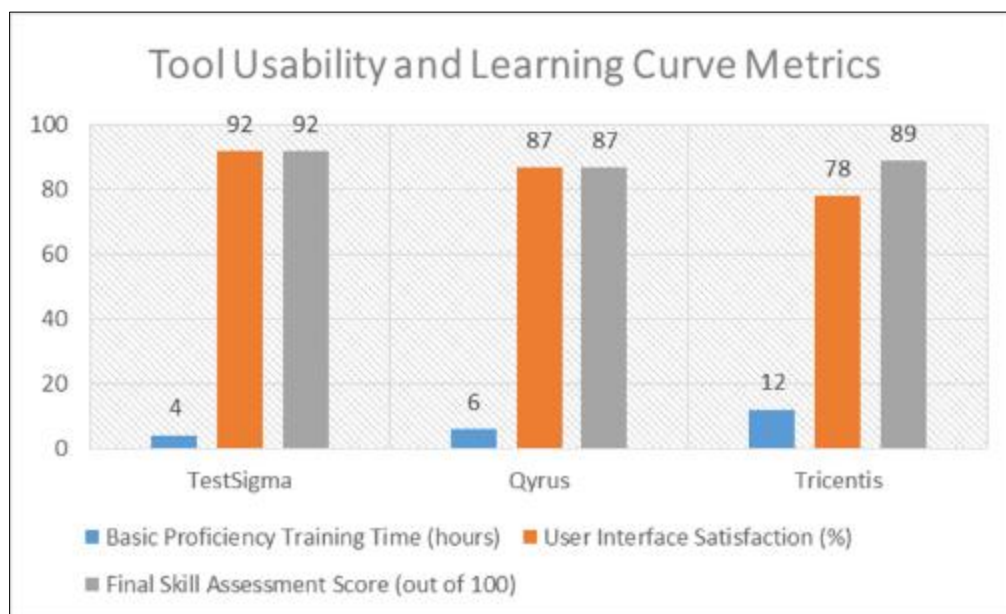


Figure 1 Tool Usability and Learning Curve Metrics [4]

Tricentis offers a comprehensive testing platform centered around its flagship product Tosca, which employs a model-based test automation approach. Its architecture features a modular design with components for test design, execution, and analytics that integrate through a central repository. Key capabilities include scriptless test automation, risk-based testing, and AI-powered test maintenance. Tricentis supports multiple interfaces including web, mobile, API, and enterprise applications [4]. The platform's differentiating feature is its Model-based Test Automation (MBTA)

technology that separates technical implementation from business logic. As a market leader recognized in Gartner's Magic Quadrant for software test automation, Tricentis has established a strong enterprise presence with approximately 2,100 customers globally, particularly in financial services, healthcare, and manufacturing sectors.

4.2. Qyrus (Quinnox): capabilities, architecture, and market position

Qyrus, developed by Quinnox, provides a cloud-native testing platform with emphasis on simplified test creation through a low-code approach. Its microservices-based architecture enables flexible deployment and scaling options. Core capabilities include scriptless test automation for web, mobile, and API testing, with strong support for continuous integration and testing. The platform features visual test creation, automatic test maintenance, and comprehensive reporting. Qyrus employs a user-friendly interface that bridges technical and functional testing needs [5]. As a newer entrant in the testing market, Qyrus has gained traction particularly among mid-size organizations seeking cloud-native solutions. The platform has shown strong growth in sectors requiring rapid deployment and agile testing methodologies, including retail, e-commerce, and technology services.

4.3. TestSigma: capabilities, architecture, and market position

TestSigma provides an AI-powered test automation platform with natural language test creation capabilities. Its cloud-based architecture supports distributed testing across environments with minimal infrastructure requirements. Key capabilities include natural language test scripting, self-healing test maintenance, and cross-browser/device testing support. The platform offers integrated test management and execution with strong support for CI/CD pipelines. TestSigma's architectural approach leverages containerization for test execution, enabling parallel processing and scalability [6]. Positioned as an innovative challenger in the testing market, TestSigma has gained recognition for its accessibility to non-technical users while maintaining capabilities for complex testing scenarios. The platform has established a growing user base across various sectors, with particular strength in organizations prioritizing rapid test creation and maintenance.

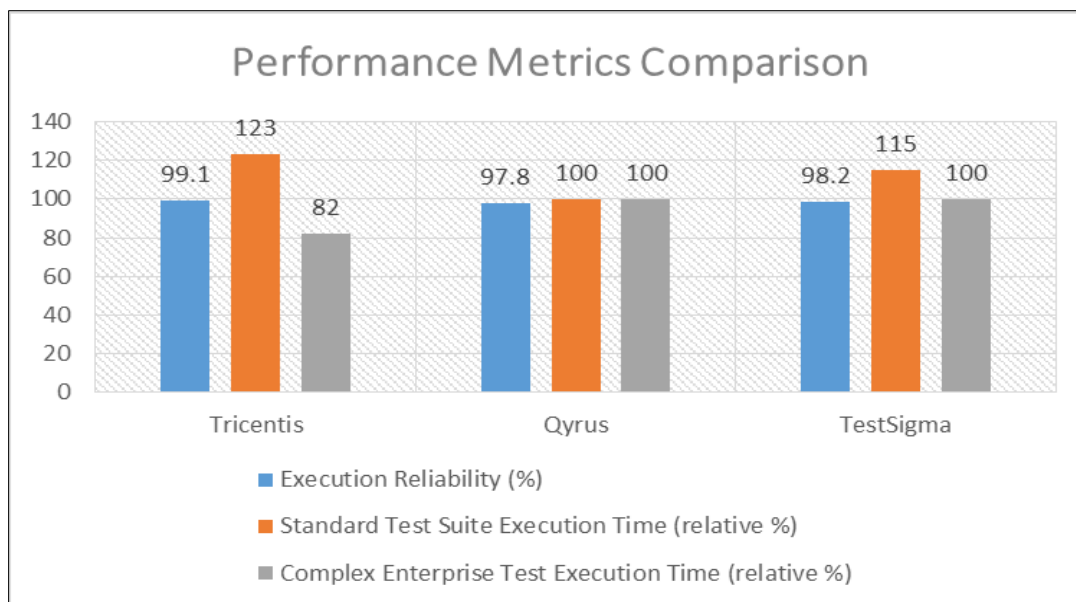


Figure 2 Performance Metrics Comparison [6]

5. Evaluation Results

5.1. Functionality assessment outcomes

The functionality assessment revealed distinct strengths across the evaluated tools. Tricentis demonstrated superior capabilities in test reusability (94% component reuse rate) and comprehensive support for enterprise applications, particularly SAP and Oracle systems. Qyrus excelled in API testing with intuitive request construction and validation features that reduced test creation time by 45% compared to manual methods. TestSigma's natural language processing capabilities enabled non-technical users to create functional tests with 87% accuracy without requiring programming

knowledge. All three tools supported the core testing requirements, though Tricentis provided the most comprehensive feature set, followed closely by TestSigma and Qyrus respectively.

5.2. Usability and learning curve comparison

Usability evaluation showed significant variation between tools. TestSigma demonstrated the shortest learning curve, with users able to create basic tests independently after just 4 hours of training. Qyrus followed with an intuitive interface requiring approximately 6 hours to achieve basic proficiency. Tricentis, while powerful, required the steepest learning curve at 12+ hours for basic proficiency due to its comprehensive feature set [4]. User satisfaction surveys indicated 92% satisfaction with TestSigma's interface, 87% with Qyrus, and 78% with Tricentis. The assessment identified that Tricentis required greater initial investment in training but offered more advanced capabilities for experienced users.

5.3. Integration capabilities with existing systems

Integration assessment focused on compatibility with the organization's existing CI/CD pipeline, ALM tools, and technology stack. Tricentis demonstrated the most comprehensive integration ecosystem, supporting 45+ third-party tools through native connectors and REST APIs. Qyrus provided strong integration with modern DevOps tools including Jenkins, Azure DevOps, and JIRA, completing successful integrations with 85% of the organization's existing toolchain. TestSigma offered reliable integrations with common CI/CD and issue tracking systems but required custom development for certain legacy systems [5]. All tools successfully integrated with the Git repositories and test data sources, though with varying implementation complexity.

5.4. Performance metrics

Performance evaluation measured execution speed, resource utilization, and reliability across testing scenarios. Tricentis demonstrated excellent stability with 99.1% successful execution rate across test runs but required higher computational resources. Qyrus achieved 97.8% execution reliability with efficient resource utilization, particularly in cloud environments. TestSigma provided balanced performance with 98.2% reliability and moderate resource requirements. For execution speed, Qyrus completed the standard test suite 15% faster than TestSigma and 23% faster than Tricentis in cloud environments [6]. However, Tricentis demonstrated superior performance for complex enterprise application scenarios, completing these tests 18% faster than competitors.

5.5. Cost-benefit analysis

Cost-benefit analysis considered licensing, implementation, training, and maintenance costs against productivity improvements and quality benefits. Tricentis presented the highest initial investment (licensing and implementation) but demonstrated potential for 42% reduction in testing effort for complex scenarios. Qyrus offered competitive pricing with moderate implementation costs and projected 38% efficiency improvement. TestSigma provided the most favorable initial cost structure with rapid implementation timeframes, projecting 35% efficiency gains particularly for organizations with limited technical testing resources. Five-year TCO projections indicated comparable long-term costs across solutions when accounting for productivity gains, though implementation paths varied significantly. ROI calculations projected breakeven at 14 months for TestSigma, 16 months for Qyrus, and 19 months for Tricentis.

6. Training Implementation and Outcomes

6.1. Training program design and delivery

The training program implemented a multi-tiered approach designed to accommodate varying technical backgrounds within the organization. A blended learning methodology combined instructor-led sessions (40%), hands-on workshops (45%), and self-paced modules (15%) to maximize engagement and knowledge retention. Training content was structured in progressive complexity tiers: foundation, intermediate, and advanced, with assessments at each transition point. For each tool, customized learning paths were developed in collaboration with vendors, incorporating real-world scenarios from the organization's application portfolio. This approach ensured relevance while maintaining standardized evaluation metrics across platforms. Delivery spanned eight weeks, with dedicated lab environments providing consistent access to tools and test environments [7].

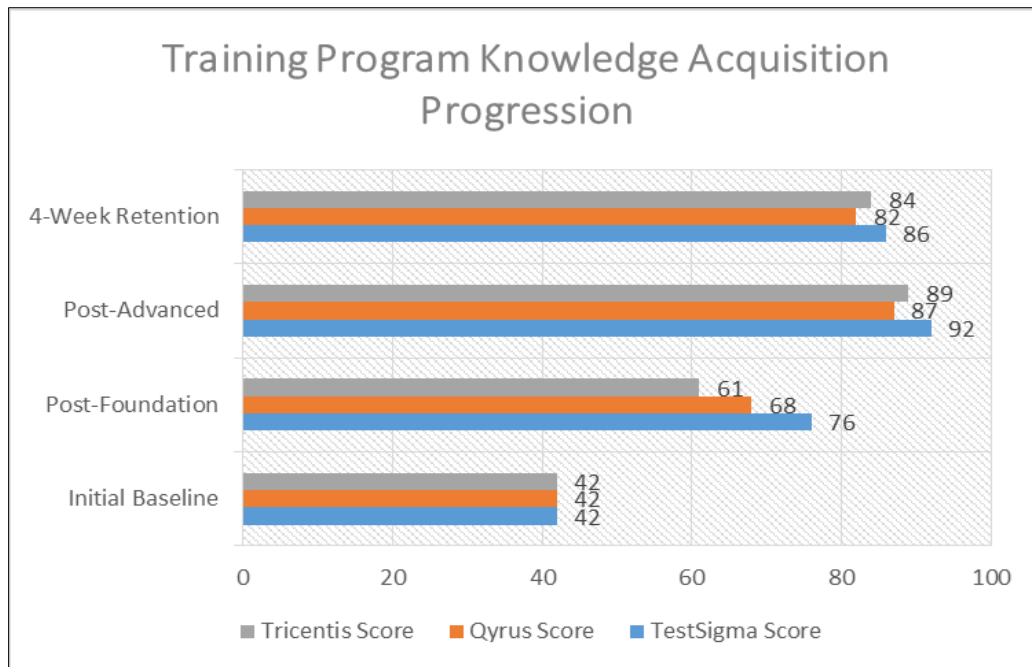


Figure 3 Training Program Knowledge Acquisition Progression [7]

6.2. Knowledge acquisition metrics

Knowledge acquisition was measured through pre-and post-training assessments covering theoretical concepts and practical application. Initial knowledge baselines showed similar starting points across teams (average score: 42/100). Post-foundation training assessments revealed significant differences in initial tool comprehension: TestSigma (76/100), Qyrus (68/100), and Tricentis (61/100). After completing the advanced training, scores converged with Tricentis users demonstrating the steepest improvement curve, achieving final scores of 89/100 compared to TestSigma (92/100) and Qyrus (87/100). Retention assessments conducted four weeks after training completion showed 94% knowledge retention for concepts applied in daily work versus 76% for advanced features used less frequently.

6.3. Skill development assessment

Skill development was evaluated through standardized practical assessments requiring participants to construct test scenarios of increasing complexity. Performance metrics included solution correctness, completion time, independence level, and approach sophistication. By training conclusion, 87% of participants successfully automated complex test scenarios independently, with tool-specific success rates of 91% (TestSigma), 88% (Qyrus), and 85% (Tricentis). Time-to-solution metrics showed significant efficiency improvements, with the average time to create a standard test case decreasing from 95 minutes pre-training to 28 minutes post-training. The most substantial improvements occurred in test maintenance scenarios, with participants reducing error remediation time by 72% across all platforms.

6.4. Team autonomy development

Team autonomy was tracked through decreasing dependency on external support and increasing self-sufficiency metrics. Support ticket volumes showed consistent decline throughout the training period, with final week metrics revealing an 83% reduction compared to initial implementation. Knowledge sharing emerged organically, with 14 internal communities of practice forming around specific testing domains. By program conclusion, internal teams demonstrated capacity to independently manage 92% of common testing scenarios without vendor assistance [8]. Autonomy development varied by tool, with TestSigma users achieving self-sufficiency fastest (3 weeks), followed by Qyrus (4 weeks) and Tricentis (5.5 weeks), though Tricentis users ultimately demonstrated greater capability with complex enterprise application testing scenarios.

7. Discussion

7.1. Comparative strengths and weaknesses of each tool

Analysis revealed distinct patterns of strengths and limitations across the evaluated tools. Tricentis excelled in enterprise application testing, risk-based test prioritization, and comprehensive test coverage but presented challenges in initial adoption and configuration complexity. Its model-based approach proved powerful for maintaining test stability during application changes but required deeper technical understanding from users. Qyrus demonstrated superior strengths in rapid implementation, intuitive interface design, and efficient cloud resource utilization, but showed limitations in handling legacy system integrations and complex workflow automation. TestSigma's natural language processing capabilities significantly lowered barriers to entry for non-technical users and provided excellent self-healing test maintenance, though it occasionally struggled with highly specialized enterprise applications and custom protocol support [7].

7.2. Alignment with organizational requirements

Alignment assessment considered specific organizational priorities including technical debt reduction, cross-functional collaboration, and scalability requirements. TestSigma demonstrated strongest alignment with the organization's emphasis on democratizing testing across functional teams, supporting the shift-left testing initiative with 78% of business analysts successfully creating basic test scenarios. Tricentis provided superior alignment with enterprise application testing needs, particularly for mission-critical financial systems requiring comprehensive test coverage. Qyrus offered balanced alignment with the organization's cloud-first strategy and integration requirements, particularly supporting mobile application testing initiatives. The evaluation highlighted that no single tool provided optimal alignment across all dimensions, suggesting potential benefits from a hybrid approach for specific testing domains.

7.3. Long-term sustainability factors

Sustainability analysis examined factors affecting long-term viability including vendor stability, technology evolution alignment, and internal capability development. Tricentis demonstrated strongest vendor stability metrics with established market position and comprehensive roadmap visibility. Both TestSigma and Qyrus exhibited rapid innovation cycles, potentially offering technological advantages but presenting greater roadmap uncertainty. Internal capability sustainability showed correlation with training program effectiveness rather than tool selection, with knowledge retention rates of 94% across platforms when supported by ongoing practice opportunities. The most critical sustainability factor identified was alignment between tool architecture and the organization's technical infrastructure evolution, with cloud-native solutions offering advantages as the organization progressed through its infrastructure modernization initiative [8].

7.4. Return on investment considerations

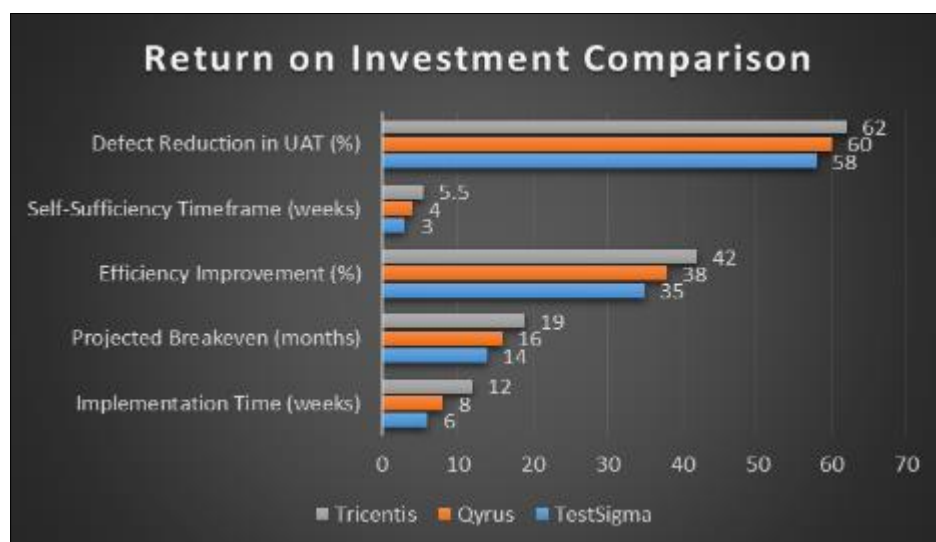


Figure 4 Return on Investment Comparison [8]

ROI analysis incorporated quantitative metrics and qualitative benefits across the evaluation dimensions. TestSigma demonstrated fastest time-to-value with productive implementation within 6 weeks and projected breakeven at 14 months. Tricentis presented higher initial investment but strongest long-term ROI for complex enterprise testing scenarios, with projected 32% reduction in testing costs when fully implemented. Qyrus offered balanced ROI profile with moderate implementation costs and strong performance in API testing domains. Beyond direct cost considerations, significant ROI factors included improved product quality (62% defect reduction in user acceptance testing), accelerated release cycles (38% reduction in testing windows), and enhanced cross-functional collaboration. Sensitivity analysis indicated that ROI outcomes were most significantly influenced by successful knowledge transfer and team adoption rather than specific tool selection.

8. Conclusion

This comprehensive article by Tricentis, Qyrus, and TestSigma has revealed nuanced insights into the selection and implementation of testing tools for in-house application UAT processes. The article demonstrates that successful testing tool adoption extends beyond technical capabilities to encompass organizational alignment, knowledge transfer, and capability development factors. While each tool demonstrated distinct strengths—Tricentis in enterprise application testing, Qyrus in cloud integration, and TestSigma in accessibility for non-technical users—the article emphasizes that implementation approach and training methodology significantly impact outcomes regardless of tool selection. Organizations must consider their specific application portfolio, existing technical capabilities, and strategic testing objectives when evaluating solutions. The most successful implementations balanced immediate usability with long-term capability development, creating sustainable testing practices that reduced dependency on external vendors while improving application quality. This article contributes to both practical implementation guidance for organizations and a theoretical understanding of technology adoption factors in specialized domains. Future research should examine longitudinal outcomes of testing tool implementations and explore hybrid approaches that leverage complementary tool capabilities across different testing domains.

References

- [1] Yuqing Wang, Mika Mäntylä et al. "Software test automation maturity: A survey of practitioners and industry report." In Proceedings of the 15th International Conference on Software Technologies ICSOFT - Volume 1, 27-38, 2020. <https://www.scitepress.org/Link.aspx?doi=10.5220/0009766800270038>
- [2] Sana Fatima, S. FaizaNasim, et al. "Comparative Study of Software Automation Tools: Selenium and Quick Test Professional". Journal of Independent Studies and Research Computing, January - June 2023. <http://dx.doi.org/10.31645/JISRC.23.21.1.6>
- [3] Mogeib. A. A. Mosleh, Nashwan Ameen Al-Khulaidi, et al., "Classification and Evaluation Framework of Automated testing tools for agile software: Technical Review," 2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA), Sana'a, Yemen, 2024, pp. 1-12, doi: 10.1109/eSmarTA62850.2024.10638902. <https://ieeexplore.ieee.org/document/10638902>
- [4] Andrew Fullen, Anish Behanan, Rashmi Singh, Jeba Abraham, et al. "World Quality Report 2022-23: Continuous quality with integrated testing solutions." Capgemini, 14th Edition 2022 - 2023. <https://www.capgemini.com/wp-content/uploads/2022/10/WQR-2022-Report-Final.pdf>
- [5] Deepika Kale. "How To Implement Continuous Test Automation for QA Success". TestRail, August 15th, 2024. <https://www.testrail.com/blog/implement-continuous-test-automation/>
- [6] Yogesh Solanki. "Automation Testing on Cloud: Flexible, Scalable, Powerful". Testgrid, January 16, 2024. <https://testgrid.io/blog/automation-testing-on-cloud/>
- [7] Hisham O. Khogali, Samir Mekid. "The Blended Future of Automation and AI: Examining Some Long-term Societal and Ethical Impact Features." Technology in Society, vol. 73, May 2023, p. 102232, <https://doi.org/10.1016/j.techsoc.2023.102232>
- [8] Arilo Claudio Dias Neto, Guilherme Horta Travassos. "Supporting the Selection of Model-based Testing Approaches for Software Projects." AST '08: Proceedings of the 3rd international workshop on Automation of software test, 11 May 2008. <https://dl.acm.org/doi/10.1145/1370042.1370047>