IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

The Impact of Software Quality Assurance Practices on the Competitiveness of the Technology Sector

Drogunova Yulia
Bachelor's Degree, Dostoevsky Omsk State University, Omsk, Russia

Abstract: *This article examines the impact of software quality assurance practices on the competitiveness of companies in the technology sector. The study focuses on analyzing the relationship between the maturity of software quality processes and business metrics such as customer acquisition cost, customer lifetime value, retention rate, and net promoter score. Modern approaches – including Agile, DevOps, and Shift-left testing – are described, highlighting their role in accelerating innovation while ensuring digital stability. The article presents examples of companies of various sizes, demonstrating how the maturity level of quality assurance practices influences operational efficiency, release frequency, and customer loyalty.*

Keywords: software, software quality, Agile, DevOps, Shift-left testing, technology company competitiveness, development.

I. INTRODUCTION

In the context of intensifying digital transformation, software quality has become a central factor in the resilience and competitiveness of technology companies. Growing user expectations of the availability, speed, and security of digital goods are placing more and more pressure on IT organizations to ensure product quality and shorten development cycles. Modern quality assurance (QA) practices like test automation, Shift-left testing (earlier QA activity integration), and the use of Agile and DevOps approaches are setting new benchmarks for digital product development and deployment. Under such an environment, QA process maturity serves not only as a measure of the effectiveness of development within the company but also as a key business performance driver, influencing a company's ability to adopt innovations rapidly, retain customers, and provide stable service.

The purpose of this study is to identify and compare the relationship between software QA process maturity and the business firms' level of competitiveness in the technology sector. The study examines factors such as time-to-market, digital infrastructure resilience, innovation capacity, and customer loyalty.

II. MAIN PART. CONTEMPORARY QA PRACTICES

Software QA practices develop hand in hand with more general shifts in digital product development paradigms. Traditional waterfall development models have increasingly given way to agile ones designed to compress delivery cycles, improve responsiveness to market needs, and improve the user experience. In this context, Agile approaches, DevOps practices, and Shift-left testing as a concept have become most prominent – each of them contributing to further improvement in QA process maturity and, in turn, technology companies' competitiveness.

Agile, as a software development methodology, is based on the principles of flexibility, iteration, and continuous communication among team members (fig. 1).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

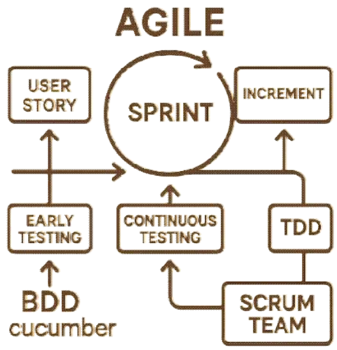

Fig. 1 Agile workflow and QA practice integration in the iterative development process

Agile in QA promotes the practice of early testing, continuous testing, and test-driven development (TDD). Importance is given to short development sprints, during which each user story is followed by acceptance criteria and automated test cases. Methods like Behavior-Driven Development (BDD) are also prevalent, allowing teams to formalize requirements in the form of natural language scenarios (e.g., with Cucumber), thus enhancing collaboration between developers, testers, and business analysts [1]. Incorporating QA experts into Scrum teams ensures continuous validation of product increments, minimizing the chances of defect accumulation in subsequent project phases.

According to a 2025 survey by Forrester [2], 95% of the respondents guaranteed that Agile methodology remains relevant and in great demand. Of these, 61% are applying Agile practices for more than five years and 24% for the last three years – showing long-term application across the industry. However, only 33% guaranteed that Agile adoption had actually led to the anticipated rise in business value. One of the most striking trends is the shift from project teams to product teams (43%), followed by growing interest in AI and TuringBots to make teams more productive and innovative.

DevOps is a set of cultural and technical practices that enable end-to-end automation of software deployment, testing, and monitoring processes (fig. 2).

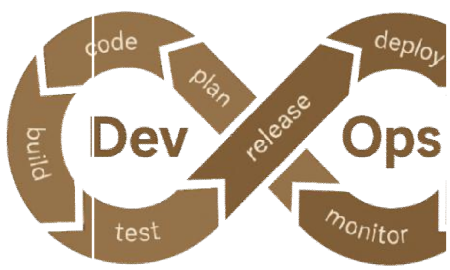

Fig. 2 DevOps cycle: integration of development and operations processes

In terms of QA perspective, DevOps embraces the strategy of Continuous Integration (CI) and Continuous Delivery/Deployment (CD), and every code commit is automatically processed through a verification pipeline [3]. The process includes the running of unit tests, static code analysis, and integration as well as end-to-end tests in sandboxed environments. It utilizes containerization (Docker) and orchestration (Kubernetes) in order to reproducibly control the environment, which is a very critical step for reproduction defects and release reliability. Metrics such as Mean Time to Detect (MTTD) and Mean Time to Recovery (MTTR) also become critical QA maturity metrics in the DevOps environment. Tools that support IaC also enter the picture, allowing for configuration compatibility verification and proper deployment as part of CI/CD pipelines.

The global DevOps industry is, based on analysis by The Business Research Company, going to grow from $12,54 billion in 2024 to $15,06 billion in 2025 at an CAGR of 20,1% [4]. According to the forecast of 2029, the market is valued at $38,11 billion. All these figures prove there to be an increasing trend of implementing DevOps processes through the board and the guarantee to double software delivery velocity, quality, and reliability.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, July 2025
Impact Factor: 7.67

Shift-left testing is a strategic approach that involves moving testing activities as early as possible in the software development lifecycle (fig. 3).

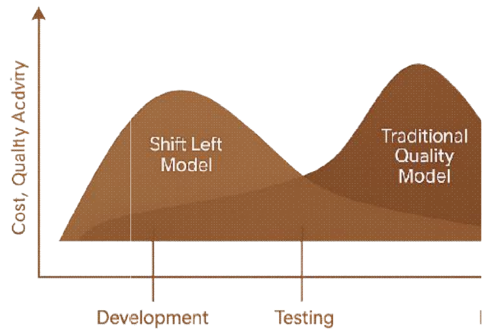

Fig. 3 Comparison of QA models: Shift-left vs. traditional approach

Technically, Shift-left testing appears in forms such as requirement-driven static analysis, test case modeling before code writing, and including quality checks when designing architecture and API during API planning and architectural design phases. SonarQube and ESLint are some of the tools that provide real-time validation of code and enable QA engineers and developers to track technical debt together. Service virtualization is also widely utilized – this approach allows teams to simulate the behavior of not-yet-implemented or not-available system components, thereby allowing critical integration scenarios to be tested early on even before full implementation [5]. As a result, organizations are able to detect defects earlier and enhance the stability of architectural decisions through early validation of design hypotheses.

As per the 2024 Thinksys QA trends report, Shift-left testing adoption resulted in 40% of the companies noticing a reduction in overall testing costs through earlier detection of defects and 57% noticing improved product quality [6].

One of the key enablers of these strategies is test automation. New technologies allow automated functional, regression, performance, and integration testing, which eliminates significant time investments and minimizes the impact of human error.

III. QUALITY ASSURANCE MATURITY MODEL AND BUSINESS METRICS

The measurement of software QA process maturity is a systematic approach that tries to diagnose and enhance production and management practices in IT organizations [7]. Various maturity models exist, with the most popular being the CMMI (Capability Maturity Model Integration) and the TMMi (Test Maturity Model Integration) models.

The CMMI model is applied to evaluate the general maturity of software development processes, and QA is one of its most important elements. It defines five levels of maturity, ranging from the initial stage – ad hoc and unstable processes – to the optimizing stage, which involves continuous improvement based on metrics and analytical feedback. CMMI is typically applied for general evaluations at the organizational level, allowing for consistent process improvement across departments.

The TMMi model, however, is a special one that specifically addresses testing process maturity alone. It too classifies five maturity levels from non-existing formalized test practices to fully integrated and optimized by means of automation and management through measures. TMMi is extremely clear in describing the steps involved in improving QA activities and therefore is specially relevant to teams willing to improve software test quality and efficiency methodically.

In the context of this study, we examine how QA maturity affects not only technical development metrics but also key business metrics that determine the competitiveness of a technology company. CAC (Customer Acquisition Cost) is a metric that reflects the average cost of acquiring a new customer and is likely to decrease with greater product stability and reduced defects. LTV (Customer Lifetime Value) is the total revenue generated by a customer over their lifetime with the product and is directly influenced by the quality of the user experience. Retention rate is the proportion of users who remain with the product over time and grows with fast, consistent updates. NPS (Net Promoter Score) measures willingness of customers to recommend the product and is a measure of customer satisfaction that QA facilitates directly in terms of system stability, security, and quality overall (table 1).

TABLE I: QA MATURITY LEVELS AND BUSINESS IMPACT [7, 8]

QA maturity level	QA process characteristics	Impact on business metrics
1. Initial	No formal QA processes, manual testing, reactive approach.	High CAC due to frequent errors; low LTV; unstable retention; negative NPS.
2. Managed	Basic test documentation, defect tracking, limited automation.	Partial CAC reduction; improved retention via critical defect resolution.
3. Defined	Standardized testing processes, QA integrated with SDLC, partial CI/CD.	Reduced CAC and increased LTV; stable retention; positive but moderate NPS.
4. Quantitatively managed	Coverage metrics, >70% automation, integrated with DevOps and monitoring.	Lower CAC, higher LTV; high retention; increased trust reflected in NPS.
5. Optimizing	Continuous QA improvement, use of AI/ML in testing, full CI/CD pipeline.	Minimal CAC; maximum LTV; high retention; consistently high NPS.

At the initial level of QA maturity, practices are often limited to manual testing conducted at the final stage of the software development lifecycle. It results in large probabilities of defect in released products, heavy costs of rework, degraded customer experience, and thus high CAC and low customer retention. Digital solution stability also diminishes the NPS, indicating inadequate user loyalty.

At managed and defined levels, testing processes are standardized, systematic defect tracking is enforced, and bug-tracking tools such as JIRA and Bugzilla are applied. Elements of CI/CD pipelines are implemented, reducing the occurrence of critical defects to production. This has a positive effect on customer retention and reduces indirect support costs, thereby optimizing CAC.

Quantitative management escalation is characterized by adopting complete test automation (for instance, using Selenium, TestNG, Cypress), possessing end-to-end CI/CD pipelines (for instance, through Jenkins, GitLab CI, Azure DevOps), and measuring quality based on the critical metrics such as test coverage, Mean Time to Detect (MTTD), Mean Time to Repair (MTTR), and release stability. Through systematic quality management, companies can continuously improve LTV by raising customer confidence and more users without proportionally increasing costs.

During the optimizing level, sophisticated analytics solutions are added including machine learning predictive testing, dynamic test prioritization, and adaptive testing in real-time. This optimizes QA efficiency, enables user experience personalization, and enables fast release of new features. Therefore, CAC is minimized to its lowest value, customer retention makes all-time records, and LTV and NPS are regular measures of a company's business and QA maturity.

Hence, a clear correlation is observed: with an increase in the maturity of QA processes, operational costs decrease, customer satisfaction enhances, and digital solution business value overall grows. This emphasizes the strategic importance of QA, not just influencing software quality but also the economic viability and growth prospects of tech firms.

IV. COMPARING THE PERFORMANCE OF IT COMPANIES BASED ON QA MATURITY LEVELS

To determine the pragmatic correlation between software QA process maturity and the performance of IT firms, comparative analysis of corporate practices was conducted. The study includes U.S.-based tech companies of varying sizes – multinational corporations, mid-size firms, and startups. The goal of the analysis is to illustrate how QA maturity influences key business metrics such as CAC, Customer LTV, Retention rate, and NPS. The information is based on publicly available industry reports, tech blogs, and company case studies, which enables establishing a direct relationship between QA maturity and resilience of digital business operations.

Google uses a QA strategy based on deep embedding of DevOps practice, CI/CD pipelines, and Site Reliability Engineering (SRE) culture where the QA activities are distributed among reliability engineers and developers. Internal tools like Bazel for auto-testing and Google Cloud Operations Suite (previously Stackdriver) for monitoring and

ISSN: 2581-9429
Impact Factor: 7.67

predictive analysis are heavily utilized. Although some of these, like Mean Time to Recovery (MTTR), are not disclosed, pervasive automation and incident management minimize downtime to the barest essentials. This is further confirmed by the NPS, which, according to SurveySensum statistics for 2024, stands at 58 – on par with industry leaders – and also indicates high user loyalty towards Google Cloud and Google Workspace solutions.

Microsoft has extensive utilization of test automation and Shift-left practices within its Agile and DevOps strategies. The Azure division is fairly typical, having excellent integration of QA activities into CI/CD pipelines via Azure DevOps. Important quality control metrics are code coverage (with the target of attaining a minimum 70% coverage of modified lines in pull requests), deployment frequency (attaining many deployments per day), mean time to recover (under an hour), and user activity metrics. On the DORA metrics, such performance falls into the «elite» maturity level of DevOps.

Atlassian, the creator of tools such as Jira, Confluence, and Bitbucket, actively implements unified QA practices. Its ecosystem welcomes behavior-driven development (BDD) practices thanks to the support of tools such as CucumberStudio, AssertThat, and Behave Pro, along with UI and REST test automation with Jira-friendly platforms such as AIO Tests and Zephyr. There is an extreme emphasis on embedding QA procedures in DevOps pipelines: Atlassian official documentation highlights the organization's strategy to achieve nearly complete code coverage by automated testing (up to 100%) and provide ongoing QA visibility through Jira and Confluence. There are also testing metrics that are used, such as daily execution status, test cycles, and tester activity, enabled through plugins like Zephyr. The availability of information on adopted tools and process tracking indicates a mature and structured QA system. This contributes to product release stability, increases the confidence of B2B customers, and helps reduce maintenance and support costs.

As a mid-sized company with experience in the development of project management and team communication software, Basecamp utilizes mostly manual testing with minimal automation, aligned with its development process based on the Shape Up methodology. QA processes focus on user interface validation and regression management, with new features tested in production environments with limited users. This approach offers satisfactory product stability and allows for an acceptable level of customer retention. However, a lack of scalable automation slows new feature introduction pace, limits LTV growth, and lowers competitiveness in the area of time-to-market.

Another illustrative example is Trello, a company that specializes in visual project management tools. Prior to its integration with Atlassian, Trello operated with minimal QA formalization, relying mainly on manual testing and sporadic automated checks, primarily during acceptance phases. The low maturity level of QA resulted in limited quality control under scalable loads, which constrained the safe expansion of functionality without increasing the risk of defects. Despite having a loyal user base, startups like Trello often face customer retention challenges during periods of growth if their QA practices do not evolve to more mature levels.

These examples show that QA maturity has a direct correlation to operational effectiveness and ability to scale. Companies with mature QA are able to rush innovations to market while maintaining product stability, which equates to reduced CAC, increased retention and LTV, and high NPS. On the other hand, organizations that do not have QA maturity suffer from delayed releases, costly bug-fixing, and even loss of user trust. QA is therefore no longer a support function but a competitive asset in the digital economy.

V. CONCLUSION

Software QA process maturity impacts the competitiveness of technology companies in a multitude of ways. Improving QA maturity significantly impacts internal operational metrics – defect resolution rate, reduced time-to-market, and reduced technical debt – along with external business metrics, such as reduced CAC, higher customer LTV, and improved retention and loyalty (Retention rate and NPS). Companies that integrate QA into their engineering and product development plans have a greater tolerance towards failures, the ability to publish more frequently, and faster innovation without compromising on digital service stability.

During an era of relentless digital evolution, QA is not an afterthought but a strategic asset on which a firm's capacity to develop sustainably and evolve rests. Agile, DevOps, Shift-left efforts, and automated tests – are supported by data-

DOI: 10.48175/IJARSCT-28486

driven quality – validate a culture of technological stewardship and continuous improvement. And thus, QA investments not only enhance a firm's technical maturity but also translate into the driving force behind long-term market performance.

REFERENCES

[1]. V. Smoliarchuk, Management strategies and models in conditions of high market volatility: application of Agile and Scrum in industrial business, The Eurasian Union of Scientists. Series: Economic and legal sciences, 2025, vol. 1, no. 126, pp. 16-19.

[2]. The state of Agile development 2025: it's still relevant, with benefits and challenges / Forrester // URL: https://www.forrester.com/report/the-state-of-agile-development-2025-its-still-relevant-with-benefits-and/RES181977 (date of access: 18.05.2025).

[3]. Tyagi, Intelligent DevOps: Harnessing Artificial Intelligence to Revolutionize CI/CD Pipelines and Optimize Software Delivery Lifecycles, Journal of Emerging Technologies and Innovative Research, 2021, vol. 8, pp. 367-385.

[4]. DevOps Global Market Report 2025 / Business Research // URL: https://www.thebusinessresearchcompany.com/report/devops-global-market-report (date of access: 18.05.2025).

[5]. S. A. Vaddadi, R. Thatikonda, A. Padthe, P. Arnepalli, Shift left testing paradigm process implementation for quality of software based on fuzzy, Soft Computing, 2023, pp. 1-13.

[6]. QA Latest Trends 2024 / Thinksys // URL: https://thinksys.com/qa-testing/qa-trends-report/ (date of access: 18.05.2025).

[7]. A Blazhkovskii, Collecting metrics for continuous platform monitoring, Universum: technical sciences: electronic scientific journal, 2025, vol. 3, no. 132, pp. 10-15.

[8]. P. Ismailov, A review of reinforcement learning algorithms for real-time problem solving // Professional Bulletin: Information Technology and Security, 2024, no. 2/2024, pp. 3–6.

[9]. S. Bolgov, Automation of business processes using integration platforms and backend technologies, International Research Journal of Modernization in Engineering Technology and Science, 2024, vol. 6, no. 12, pp. 3847-3851.