

A Comparative Survey of Software Testing Techniques: Manual vs Automated Testing

Aasirbad Biswal

Department of AI

FET.Jain Deemed-to-be University

Bengaluru-562112

aasirbad.biswal06@gmail.com

Dr. K S Arvind

Department of CSE

FET.Jain Deemed-to-be University

Bengaluru-562112

Ks.arvind@jainuniversity.ac.in

Abstract—Software testing is a fundamental element of Software Quality Assurance (SQA). With increasing system complexity, both manual and automated testing methods have been extensively adopted. This survey presents a comprehensive review of manual and automated testing approaches, analyzing their methodologies, tools, advantages, limitations, and real-world operations. The study highlights that manual testing is effective for exploratory and usability testing, whereas automated testing provides high efficiency, scalability, and accuracy for repetitive tasks. Also, emerging trends such as AI-based testing, DevOps integration and cloud-based testing are transforming modern testing practices. The survey concludes that a hybrid approach combining both methods is essential for achieving optimal software quality.

Keywords— Software Testing, Manual Testing, Automated Testing, Software Quality Assurance, Test Automation, DevOps, AI Testing

I. INTRODUCTION

Software systems are increasingly being deployed in critical disciplines such as healthcare, finance, transportation, and e-commerce. In such environments, even minor defects can lead to severe consequences, including financial loss, security breaches, or system failure. Software testing is a methodical process used to estimate software functionality, detect defects, and verify that the system meets specified conditions. Traditionally, testing was performed manually, which

needed significant human effort and time. Still, manual testing is often inefficient for large-scale systems due to its limitations in speed and scalability.

With advancements in technology, automated testing has surfaced as an important result that enables faster execution, improved accuracy, and continuous testing. Modern development practices such as Agile and DevOps further emphasize the need for automated testing to support rapid development cycles and continuous integration.

This survey aims to give a detailed comparison of manual and automated testing methods, highlighting their strengths, weaknesses, and applicable use cases in modern software development.

II. BACKGROUND AND OVERVIEW OF SOFTWARE TESTING

Software testing can be broadly classified into two categories:

- 1) **Manual Testing:** Manual testing involves executing test cases manually without automation tools. It relies on human observation and is generally used in early-stage testing and usability evaluation.
- 2) **Automated Testing:** Automated testing uses tools and scripts to execute test cases automatically. It is extensively used for regression testing and large-scale systems.

III. MANUAL TESTING

Definition: Manual testing is a process where testers execute test cases manually without automation tools to identify defects and verify functionality.

Techniques: Exploratory Testing, Usability Testing, Ad-hoc Testing, Black-box Testing.

Disadvantages: Time-consuming, Prone to human error, Limited scalability.

Applications: UI/UX confirmation, Early-stage development, Small-scale operations.

IV. AUTOMATED TESTING

Definition: Automated testing involves using tools and scripts to execute test cases automatically, improving effectiveness and consistency.

Techniques: Data-driven testing, Keyword-driven testing, Behavior-driven development (BDD), Script-based automation.

Advantages: Faster execution, High accuracy, Reusability of scripts, Suitable for regression testing.

Disadvantages: High initial cost, Maintenance overhead, Requires specialized expertise.

Applications: Regression testing, Performance testing, CI/CD channels.

V. TESTING TOOLS

A. Manual Testing Tools

Manual testing tools help testers in managing test cases, tracking defects, and organizing the testing workflow efficiently. These tools do not automate execution but significantly improve collaboration and documentation.

1) JIRA: JIRA is one of the most extensively used issue tracking and project management tools developed by Atlassian. It is considerably used in Agile and Scrum environments.

2) Bugzilla: Bugzilla is an open-source bug tracking system used to manage and cover software defects throughout the development lifecycle.

3) TestRail: TestRail is a test case management tool designed to organize, manage, and track testing conditioning.

B. Automation Testing Tools

Automation testing tools are used to execute test cases automatically, reducing manual effort and improving testing effectiveness.

1) Selenium: Selenium is one of the most popular open-source automation testing frameworks used for web applications.

2) Cypress: Cypress is a modern front-end testing framework designed for fast and dependable web testing.

3) JUnit: JUnit is a unit testing framework primarily used for Java applications.

4) TestNG: TestNG is an advanced testing framework inspired by JUnit, offering more powerful features for test execution.

VI. RELATIVE ANALYSIS

Parameter	Manual Testing	Automated Testing
Execution Speed	Slow	Fast
Accuracy	Moderate	High
Cost	Low Initial	High Initial
Scalability	Limited	High
Human Effort	High	Low

VII. USE CASE ANALYSIS

Manual Testing: Exploratory testing for defect discovery, UI/UX confirmation, Small-scale systems.

Automated Testing: Regression testing, Large-scale systems, Continuous integration channels.

VIII. CHALLENGES

Software testing, especially automated testing, faces several practical challenges that impact the software. These must be addressed to ensure effective testing in modern software development environments.

1) Automation Script Maintenance: Automation scripts require continuous updates whenever there are changes in the application, such as UI variations, feature updates, or workflow changes.

2) Tool Compatibility Issues: Different testing tools may not integrate easily with each other or with development environments.

3) Skill Requirements: Automated testing requires specialized knowledge, unlike manual testing. Testers must be familiar with programming languages and testing frameworks.

4) Handling Dynamic UI Changes: Modern web applications frequently have dynamic user interfaces that change constantly, making it difficult for automation scripts to locate elements.

IX. FUTURE TRENDS

A. AI-Based Testing: Self-healing scripts and intelligent test generation.

B. DevOps Integration: Continuous testing and faster release cycles.

C. Cloud Testing: Scalable environments and cross-platform testing.

X. EMERGING RESEARCH DIRECTIONS

With the increasing complexity of software systems, traditional testing approaches are evolving rapidly. Emerging technologies such as Artificial Intelligence (AI), low-code platforms, and autonomous systems are transforming the future of software testing by making it more intelligent, effective, and scalable.

1) AI-Driven Testing Systems: AI-driven testing systems use machine learning and deep learning techniques to enhance the testing process intelligently. These systems analyze historical test data and user behavior to automatically generate test cases.

2) Low-code Automation Platforms: Low-code automation platforms enable testers to create automation scripts with minimum or no programming knowledge.

3) Autonomous Testing Frameworks: Autonomous testing frameworks represent the next generation of

testing systems that can independently design, execute, and maintain test cases without human intervention.

XI. DISCUSSION

Manual testing provides flexibility and human intuition, making it suitable for exploratory and usability testing. Still, it lacks scalability. Automated testing, on the other hand, ensures speed and accuracy but requires specialized expertise and maintenance. A combination of both methods is essential for achieving effective software testing.

XII. CONCLUSION

This survey presents a comprehensive comparison of manual and automated testing methods in software quality assurance. Manual testing remains crucial for exploratory and usability testing, while automated testing offers efficiency, scalability, and reliability. A hybrid approach combining both methods is recommended for modern software development environments to achieve high-quality software systems.

REFERENCES

- [1] G. J. Myers, C. Sandler, and T. Badgett, "The Art of Software Testing," 3rd ed., Wiley, 2011.
- [2] B. Beizer, "Software Testing Techniques," Dreamtech Press, 2003.
- [3] R. S. Pressman, "Software Engineering: A Practitioner's Approach," McGraw-Hill, 2014.
- [4] I. Sommerville, "Software Engineering," 10th ed., Pearson, 2016.
- [5] M. Fewster and D. Graham, "Software Test Automation," Addison-Wesley, 1999.
- [6] IEEE, "IEEE Standard for Software Test Documentation (IEEE 829)," 2008.
- [7] M. Gupta and R. K. Bathla, "Comparative Study of Software Testing Techniques," IJSRST, 2022.
- [8] B. Kumari et al., "A Comparison Between Manual and Automated Testing," JETIR, 2018.

[9] A. Bhanushali, "Comparative Analysis of Manual vs Automated QA," IJAER, 2020.

[10] H. M. Ali et al., "A Comprehensive Study on Automated Testing," arXiv, 2024.

[11] F. Wang et al., "Requirements-Driven Automated Testing: A Survey," arXiv, 2025.

[12] J. Wang et al., "Automated Unit Test Case Generation: A Review," arXiv, 2025.

[13] M. Harman et al., "Search-Based Software Testing," IEEE ICST, 2015.

[14] G. Fraser and A. Arcuri, "EvoSuite: Automatic Test Generation," IEEE, 2014.

[15] IEEE COMPSAC, "Software Test Automation Advances," 2021.

[16] F. Ricca et al., "AI-Assisted Software Testing Review," 2025. :

[17] J. Itkonen et al., "Exploratory vs Scripted Testing," IEEE, 2007.

[18] N. Juristo and A. M. Moreno, "Basics of Software Engineering Experimentation," Springer, 2013.

[19] A. Bertolino, "Software Testing Research: Achievements and Challenges," IEEE, 2007.

[20] P. Ammann and J. Offutt, "Introduction to Software Testing," Cambridge University Press, 2016.