ISSN: 2754-6705

Journal of Mathematical & Computer Applications

SCIENTIFIC Research and Community

Review Article

Open Access

Quality Assurance / Software Testing – Traditional to Modern

Bhupinder Paul Singh Sahni

USA

ABSTRACT

This Paper focusses on the challenges faced by Traditional Software Testing and learnings from it which paved way to modern-day Software Quality Assurance/ Software Testing techniques. With the growth of the digital world and in a new era of Technology, modern-day Software Quality Assurance has come a long way. Not only the Software Quality Assurance need to ensure quality of the Product meeting Customer's requirements, but also need to deal with things like cyber-security, Performance issues and the most important challenge of being cost effective with minimal issues. Has software testing evolved along with time and requirements? This paper answers these questions by going deep into different components added to modern-day Software Quality Assurance.

***Corresponding author**

Bhupinder Paul Singh Sahni, USA.

Received: July 14, 2022; **Accepted:** July 20, 2022, **Published:** July 27, 2022

Keywords: Software Testing, Software Quality Assurance, Agile Methodology, DevOps, Automation Testing, Security Testing, Cloud based Testing

Introduction

Software Testing in the modern-day world has come a long way from the initial years when Software Quality Assurance/ Software Testing was just done as Manual Testing and there were huge turnaround times to delivering the Software. In the Technology dominated world, Software Quality Assurance is a multifaceted process now that involves various approaches, tools, and methodologies. Software Quality Assurance still plays a crucial role in Software Development process across various domains and industries, but it has taken a lot more different form from traditional years. A software development process is also called a Software Development Life Cycle (SDLC). While software development processes differ, the goal of IT organizations remains the same: to provide higher-quality Software at the lowest cost while meeting deadlines. To get higher-quality Software, Software Quality Assurance/ Software Testing needs to be robust.

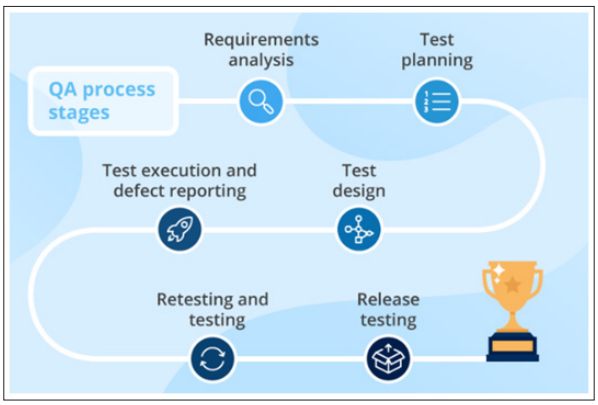
This paper presents the short-comings and challenges faced with Traditional Testing, how it shaped the way for modern day Testing processes and techniques. This includes Automation, Cloud based Testing, integration of Development and Operations leading to DevOps, Continuous Integration (CI) and Continuous Deployment (CD), etc. and a peek into the future of Software Quality Assurance.

Challenges with Traditional Testing

There were quite a few challenges with Traditional Testing that led to changes in its form in the modern world. Let's understand the challenges faced and how it led to the modern-day Testing requirements.

• Mostly Manual Testing and Requiring Too Much Effort, Resources and Turnaround Time

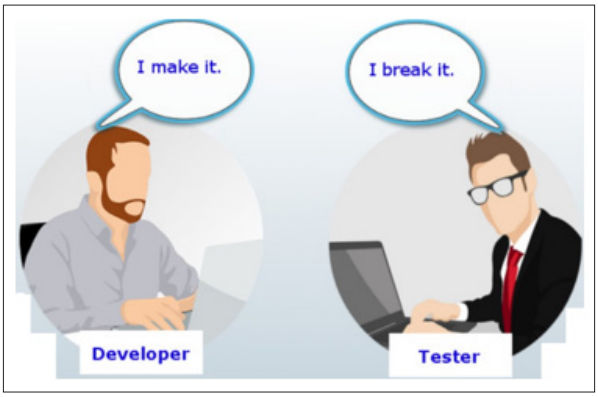
Traditionally, Testing has been mostly Manual Testing. There were multiple steps involved for the Software to be fully Tested. There was a long list of requirements, Scenarios were created first, then the Test Cases were then created. It required Requirement Traceability Matrix to map Test Scenarios with Test Cases, which was done with the help of an excel. After Development was completed, then the Test Cases were run. This required a huge turnaround time for a Software to be pushed into Production. The turnaround time from gathering Requirements from Customers to pushing the Software into Production took at least 3–4 months, even for small changes. Although it worked in yester years, when Software Development was a slow process and Information Technology was not that advanced, delivering Software in 3–4 months was considered good. But in the modern world of AIs, with Technology has advanced so much that You will be out of the competition, if Your Software for small changes is taking 3-4 months' time to develop and pushed to Production.



Citation: Bhupinder Paul Singh Sahni (2022) Quality Assurance / Software Testing – Traditional to Modern. Journal of Mathematical & Computer Applications. SRC/JMCA-195. DOI: doi.org/10.47363/JMCA/2022(1)162

• Separate Team Running Manual Tests and is Independent of Development Team

Testing Team used to be a different Team from Business & Development Teams. Although it had some advantages that Testing Team tested code as an independent party like an end user, but then there were gaps between Business, Development and Testing Team's understanding of the Product, which required too much documentation and time to identify and resolve Bugs or issues found.



• Documentation Dependency

Everything in Traditional Testing was Documentation dependent. Business Owners write the Requirements in detail to be understood by Development and Testing Teams. Traditional Testing was done more in Waterfall Model.

• Developers Didn't Have Testing Awareness, Testers Lacked Coding Knowledge

At the time when Information Technology was just picking up, Development and Testing were too separate streams requiring different skillset and knowledge. Developers were not aware of QA Processes and Quality Assurance Engineers lacked Coding knowledge. This led to separate Teams handling their portion of expertise, requiring big Development and Testing Teams, resulting in Cost to the Company.

Modern Day Software Testing and Its Future

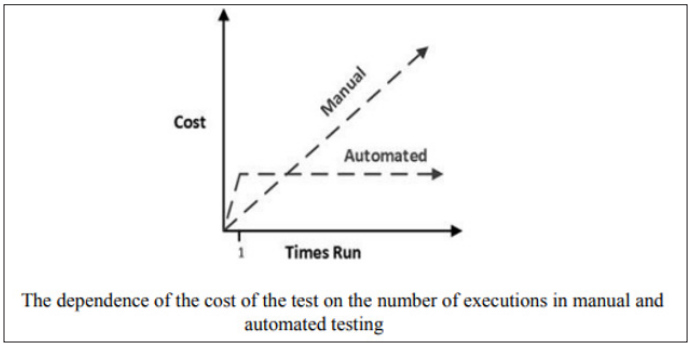
Although Traditional type of Testing worked for many years when Information Technology was in its initial phases, Software Testing evolved with change in requirements, cost measures and timelines to deliver software. Having understood the challenges faced with Traditional type of Testing, let's understand how the modern-day Software Testing has evolved over the years and its future in coming years.

• Business, Development and Testing is one Team

Small Teams are created with Business, Development and Test Engineers working on a common goal. There's a lot more interaction between Business Owners, Developers and Quality Assurance Engineers with less documentation involved.

• Automation

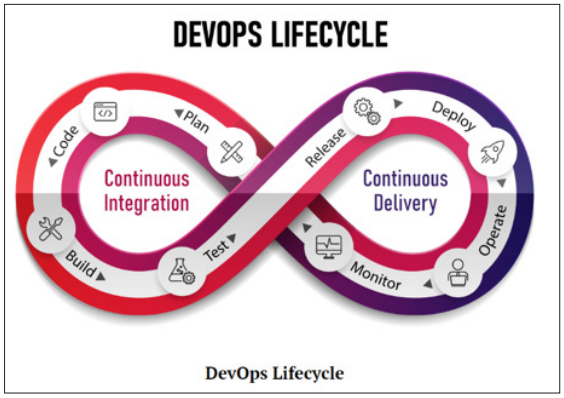
The major change that has come up now is Automation in Software Testing. Automation plays a key role in keeping up with the pace of modern Software development, thereby reducing the turnaround time to deliver Software to Customers. Automation of repetitive Tests ensure faster Testing and reduces chances of finding Critical Bugs in Production, leading to reliable Software [1].



The dependence of the cost of the test on the number of executions in manual and automated testing

• Agile and Integration of DevOps with Testing

Agile methodology has made the overall Software Development and Testing Processes faster. There are good Tools like Jira, Kanban, etc. to keep track of Software Development and Testing progress in check and within the timelines. Currently, in the professional community of Information and Communication Technologies (ICT) there is a growing consensus, in practice, that DevOps can be understood as "DevOps = Agile + Lean + IT service management (ITSM)" [2]. In its method and processes, DevOps adopts characteristics of frameworks related to the technical area of Agile software development together with ICT management processes. DevOps practices further integrate development and operations teams, facilitating continuous integration, delivery, and deployment (Continuous Integration (CI) / Continuous Deployment (CD)), which requires robust testing at every stage. Unlike traditional software development practices, DevOps is an ongoing process of building, testing, deploying, and monitoring. Its main goal is to continuously deliver quality software properly [3].



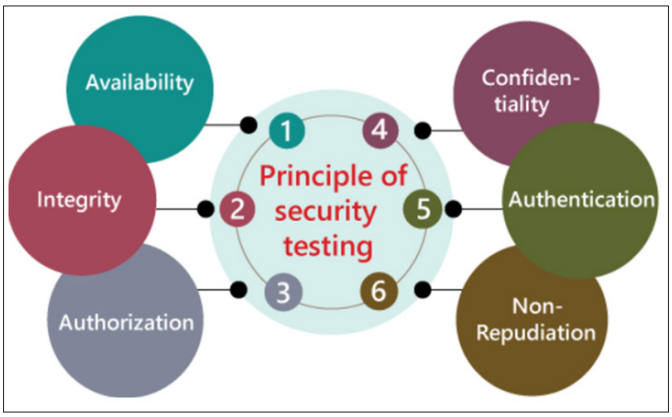
DevOps Lifecycle

• Artificial Intelligence (AI) and Machine Learning (ML) in Testing

For Product Domains where Historical data is available in abundance, AI and ML is effectively used in predictive analysis for identifying high risk areas, test case generation and automating certain aspects of Testing. So, instead of human-driven testing, we're heading towards a situation where robots execute test scripts in place of people [4]. Machine learning and self-improvement will need some human input, but it will be minimal. Hence, the creation of a group focused on the Grand Dream of Testing has become critical, where everything is automated without human interaction and technologies provide superior testing than existing application test teams [5].

• Security and Performance Testing

As Software applications are becoming complex each day, there's an increased risk of cyber threats. Security Testing is performed to make sure cyber threats are in check. Penetration testing, vulnerability scanning, and code analysis are some techniques used to ensure the security of software systems. Also, the requirement to test if Software Applications can handle the expected load of complex Software Applications is also performed using Performance Testing [6].



• Shift-Left Testing - Software Testing early in the lifecycle called the Shift-Left Testing is the approach followed to reduce the Cost of fixing Bugs found later in the Release. Defects in Requirements, Architecture, Design and Code are found early in Software Development lifecycle.

• Cloud based Testing - This paper would not have been complete without mentioning Cloud based Testing. Cloud computing is the next stage of the Internet evolution. Cloud computing has revolutionized testing by providing scalable infrastructure and on-demand resources. Testing cloud resources ensures optimal performance, availability and security of data, and minimizes downtime of the associated infrastructure or platform [7]. With all the modern-day applications moving to Cloud, Cloud based Testing has become paramount.

Conclusion

I think all the above factors have really shaped the Software Testing done today to meet the ever-changing and complex Software requirements that we have today. Delivering a robust and flexible Software meeting Customer's requirement in shortest timeframe would be the real key challenge in future, and to cater to that Artificial Intelligence (AI), Machine Learning (ML), Cloud Computing and Automation would play a major role in the coming time. Of course, these buzz words like AI, ML, Cloud would also evolve, and so will be the Software Testing.

References

1. B Oliinyk, X Bai, V Oleksiuk (2019) Automation in software testing, can we automate anything we want? 2546: 225-229.
2. S Galup, R Dattero, J Quan (2020) What Do Agile, Lean, and ITIL Mean to DevOps? Communications of the ACM 63: 48-53.
3. R Mohanan (2022) DevOps vs. Agile Methodology: Key Differences and Similarities. javatpoint.com https://www.spiceworks.com/tech/devops/articles/devops-vs-agile/.
4. D Kumar, K Mishra (2016) The Impacts of Test Automation on Software's Cost, Quality and Time to Market. Procedia Computer Science 79: 8-15.
5. C Rankin (2002) The Software Testing Automation Framework. IBM Systems Journal 41: 126-139.
6. S. Jaiswal (2022) Security Testing. javatpoint.com https://www.javatpoint.com/security-testing.
7. J Gao, X Bai, W Tsai (2011) Cloud Testing- Issues, Challenges, Needs and Practice. Software Engineering: An International Journal (SEIJ) 1: 10-12.