

A REVIEW OF TOOLS AND TECHNIQUES USED IN SOFTWARE TESTING

Isha Rana¹, Pooja Goswami¹, Himani Maheshwari²

¹Master of Computer Application Scholar, Uttarakhand University, Dehradun, India

²Assistant Professor, Uttarakhand University, Dehradun, India

Abstract: Quality software is results of a methodological and systematic approach the elements of software development: requirement analysis, software requirement specification, scoping and modeling and design and architecture. Testing is an integral part of the development process, and it is recommended as a continuous process throughout the development cycle. Software is tested throughout the development cycle to ensure that no error is detected in the maintenance phase and potential problems affecting quality are detected in the early phase of development. This paper focuses on the review and comparison of software testing types: manual and automated in order to find out the most suitable techniques for Software Testing. In addition to, this paper also highlights various software testing tools.

IndexTerms - software testing, manual testing, automated testing, software testing tools, testing process.

I. INTRODUCTION

The main aim of software development is to ensure that a high quality product is delivered. High quality product has some characteristics like low cost, reliable and user satisfactory. For this developers often perform testing at each level of development to make sure that the software works appropriately, yet software fails when final implementation is done. At this point testing comes into action, tester plays vital role in delivery of high quality product [7, 8]. Software Testing is an essential activity to discover all the errors and bugs in the software before actual deployment of the product [1, 2]. This activity ensures that the expected result matches actual result every time under every scenario, helps in identifying missing requirements in contrary with actual requirements. It is the process to check the functionality and accuracy of the software by executing it. In software testing there are test cases written which needs to be executed, to make sure the software design satisfies all the provided conditions and is bugs free [3,6]. There are over fifty types of testing available with the help of software can be tested under different conditions and according to the need of client and time.

Testing life cycle: Figure 1 depicts the testing life cycle, Figure 2 depicts the software testing process and Figure 3 shows software test execution process.

- **Prerequisite** – Preconditions which must be attained before executing the test.
- **Planning** – Scope of the test, Environment under which test needs to be performed, Testing phase and Methodologies to be used, Manual and Automation Testing, Bugs Management, Configuration Management, liability Management, Evaluation & identification – Testing tools, test scheduling, resource sharing.
- **Designing** - Test Scenarios Identification, Test coverage and Traceability Matrix & Test scripts preparation, Test Case preparation and Test data, Test case reviews and Approval, Base lining under Configuration Management.
- **Test Environment Setup** - Network connectivity, Test Bed installation and configuration, All the tools or Software Installation and configuration, Coordination with Merchants and others.
- **Automation** - Automation Tool Identification and Evaluation, Analysing and Designing Framework and scripting, Script Assimilation, Audit and Approval, Guideline under Configuration Management.
- **Execution and Bug Tracking** - Test case execution, Test Scripts testing, Acquisition, audit and evaluate Test Results, Raise the bugs and finding for its termination.
- **Acceptance and Reports Generation** - Test Metrics, Test reports and process Enhancement made, Build release, Receiving acceptance.

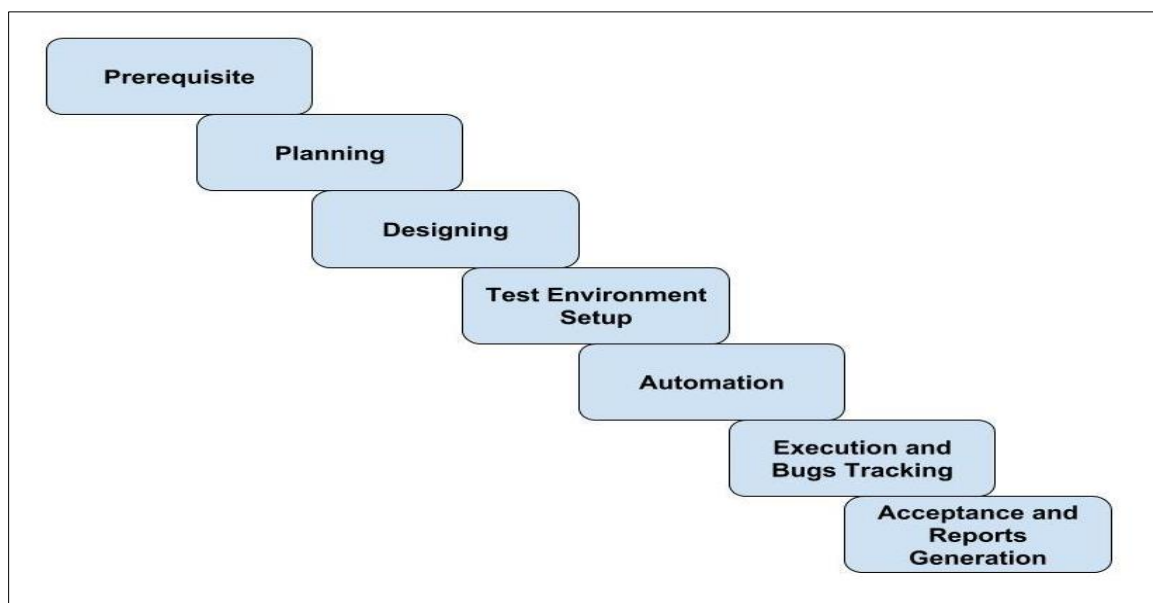


Figure 1: Software Testing Life Cycle

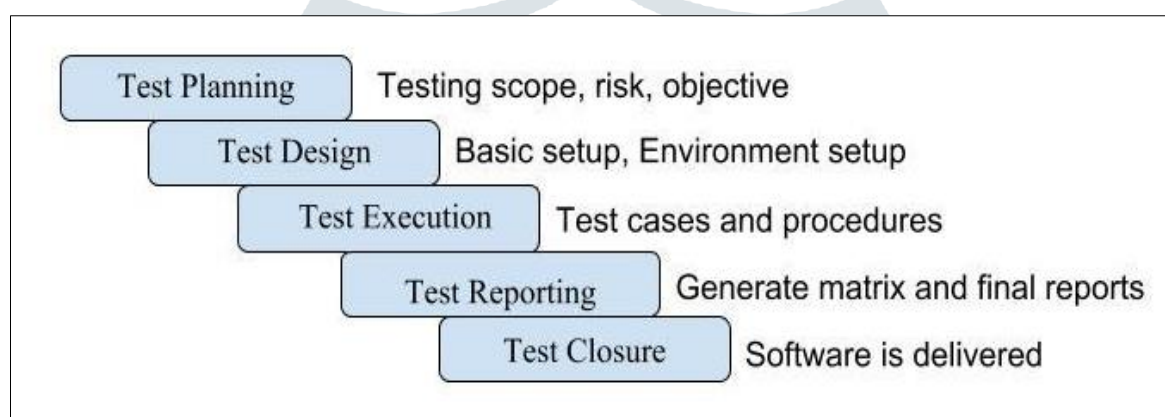


Figure 2: Software Testing Process

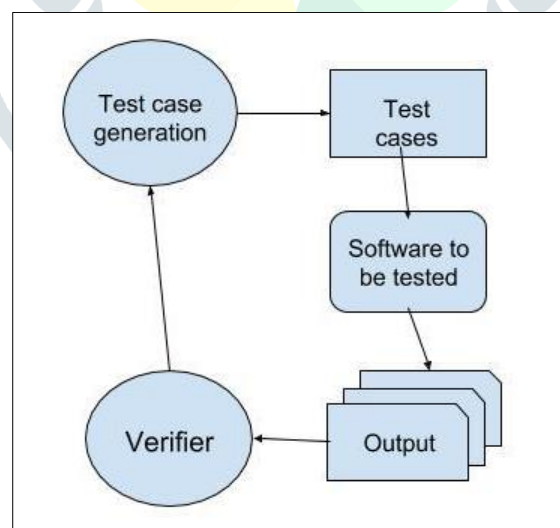


Figure 3: Software Test Execution Process

II. GENERIC SOFTWARE TESTING TERMS

- A. **Software Testing:** Software Testing is an essential activity to discover all the errors and bugs in the software before actual deployment of the product.
- B. **Verification:** Checking of software documents, code, design and program. It doesn't involve test execution. It use uses methods like auditing, inspections, walk-through, etc.
- C. **Validation:** It is the dynamic method of validating and testing the actual product. It does include executions. It uses methods like white box, black box, etc.

- D. **Quality Assurance (QA):** A group of activities to make sure that the maintenance or/and development process is adequate to make sure that the system meets its objectives. A standardized and planned set of activities necessary to provide adequate confidence that requirements are properly established and services or products conform to specified requirements. It doesn't involve executions
- E. **Quality Control (QC):** The process by which the quality of product is correlated with applicable standards and the action is taken when non uniformity is detected. It always involves executions.

III. TYPES OF TESTING

There are three broad types of testing as shown in Figure 4. The type of testing which checks that every function of a software application works in accordance to the requirement specification is known as **functional testing**. It basically includes unit testing, integration testing, user acceptance testing, etc. Functionality of the system is checked providing some input (valid and invalid) and observing the respective output produced. This type of testing can be easily carried out manually. The type of testing which checks the non-functional conditions (scalability, usability, endurance, etc.) of a software application is known as **non-functional testing**. Non-functional testing is equally important as functional testing. It basically includes performance testing, load testing, usability testing, etc. This type of testing is bit difficult to be carried out manually. When the software application has been deployed and some enhancements or changes have been done in the application then the **maintenance testing** is done. It basically includes regression testing and maintenance testing.

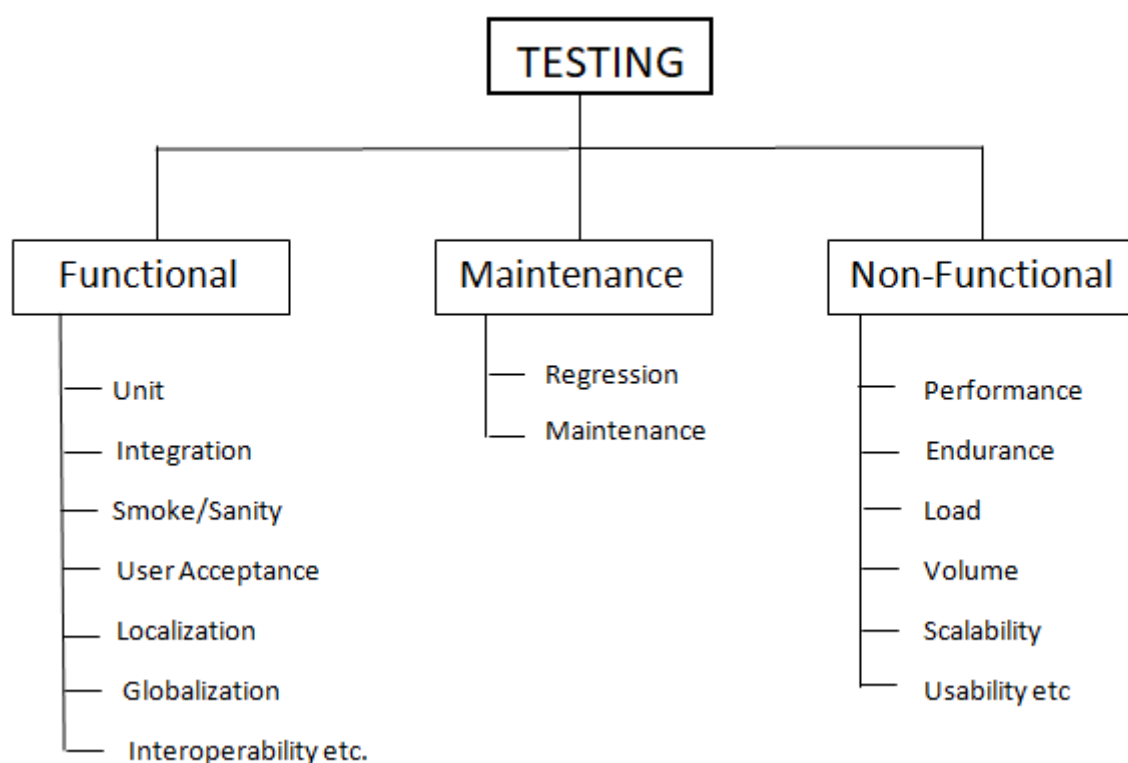


Figure 4: Types of Testing

IV. MANUAL VS. AUTOMATED TESTING

Testing of an application is done manually by human it is known as Manual Testing. A Quality Assurance Specialist (tester) ensures application works properly by operating written test cases. Tester checks the functionality, design, and performance of an application by clicking through different elements of an application. When automation testing is not possible then manual testing is helpful. For example: Exploratory testing, usability testing, ad-hoc testing, etc. The pre-scripted test which runs automatically is known as Automated Testing. The tests run to assert test results comparing actual result with expected result [4, 5]. Automated tests comes into picture when same test has to be performed multiple types under different environment. For example: Integration testing, system testing, unit testing, acceptance testing, etc. Table 1 presents the comparative study between automated and manual testing process.

Table 1: Comparison between Manual and Automated Testing

Automated Testing	Manual Testing
Automated testing is more reliable. It performs same test N number of time, reducing human error.	Manual Testing is less reliable. Due to human errors, manual testing cannot be accurate every time.
The cost of Automated test depends on the cost of testing tools used for testing.	The cost of Manual Testing depends on the human resources used to perform testing.
Due to scripts Automated testing is faster and less time consuming	Due to human interference Manual testing is slow and time consuming.
Automated testing is used when test cases needs to be executed repeatedly.	Manual testing in used when two or three test cases have to be executed.
Automated testing ensures the user friendliness of the application.	Manual testing doesn't ensure the user friendliness of the application.
Automated testing can be executed in parallel.	Manual testing can also be done in parallel but it hypes the testing cost.
Programming knowledge is a must in Automated testing.	Programming knowledge is not necessary in Automated testing.

V. SOFTWARE TESTING TOOLS

There is various software testing tools available in the market which is given in Table 1 [9].

Table 1: Software Testing Tools Description

Tool	Description
Selenium	Selenium is a testing tool, used to automate tests which are performed on web browser. It can be executed in multiple browsers. It is compatible to multiple programming languages.
Cucumber	Cucumber is an open source tool which supports Behavior Driven Development (BDD). It can be defined as testing framework, which uses plain English (Gherkin Language). It is multiple platform compatible, for example- Ruby On Rails, Selenium, PicoContainer, etc.
Ranorex	Ranorex is an all in one tool for mobile, web and desktop testing. It provides an easy click and go interface for beginners and an powerful IDE for automation experts. It is an licensed software.
Testsigma	Testsigma is one of the best automation tools available in market. It is best suited for DevOps and Agile market. It is an AI-driven tool which automates complex tests using simple English. No programming. CI/CD support.
LamdaTest	LamdaTest is one of the best cross browser test automation tool. Run Selenium automation tests on a secure, scalable and reliable cloud based Selenium Grid.

VI. CONCLUSION

Software testing is a continuous process and is affected in an incremental fashion, by testing the lowest level unit processes, applications, modules, subsystems and systems. The testing methodology distinguishes error, fault and failure in the software to attack the overall problem of quality attainment. It defines error as discrepancy, fault as malfunction and failure is inability. In all these cases, the software must be tested; the testing should be of a kind that these discover these inadequacies in the software. Manual testing is cumbersome process and hence cost effective whereas automated testing tools reduced the cost and time of testing.

VII. REFERENCES

- [1] Glenford J. Myers, "The Art of Software Testing, Second Edition" Published by John Wiley & Sons, Inc., Hoboken, New Jersey. 1.1. P. Mathur, "Foundation of Software Testing", Pearson/Addison Wesley, 2008.
- [2]. IEEE Standard 829-1998, "IEEE Standard for Software Test Documentation", pp.1-52, IEEE Computer Society, 1998.
- [3]. "The Theory of Software Testing" Adtha. Lawanna Department of Information Technology, Faculty of Science and Technology Assumption University, Bangkok, Thailand
- [4]. "Software Testing Techniques and Strategies" Abhijit A. Sawant¹, Pranit H. Bari and P. M. Chawan Department of Computer Technology, VJTI, University of Mumbai, INDIA.
- [5]. Mohd. Ehmer Khan, "Different Forms of Software Testing Techniques for Finding Errors", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 1, May 2010.
- [6]. Jovanovic and Irena, "Software Testing Methods and Techniques" May 26, 2008.
- [7]. J. A. Whittaker, "What is Software Testing? And Why Is It So Hard?" IEEE Software, January 2000, pp. 70-79 [hard software testing].
- [8]. Web Url: <http://reqtest.com/testing-blog/functional-vs-non-functionaltesting/> (Last accessed on: April, 2019)
- [9] Ankita Sethi, "A review paper on levels, types & techniques in software testing", International Journal of Advanced Research in Computer Science, vol:8, pp: 269-271.

