Software Testing Techniques: A Literature Review

Mrs Akula Swathi
Assistant Professor
Dept. of Computer Science and Engineering
Avanthi Institute of Engineering and Technology, Cherukupally (Village), Near Thagarapuvalasa-531162

Mrs P.Monika
Assistant Professor
Dept. of Computer Science and Engineering
Avanthi Institute of Engineering and Technology, Cherukupally (Village), Near Thagarapuvalasa-531162

Mr.Kesavarao Seerapu
Assistant Professor
Dept. of Computer Science and Engineering
Avanthi Institute of Engineering and Technology, Cherukupally (Village), Near Thagarapuvalasa-531162

***Abstract*— With the growing complexity of today's software applications injunction with the increasing competitive pressure has pushed the quality assurance of developed software towards new heights. Software testing is an inevitable part of the Software Development Lifecycle, and keeping in line with its criticality in the pre and post development process makes it something that should be catered with enhanced and efficient methodologies and techniques. This paper aims to discuss the existing as well as improved testing techniques for the better quality assurance purposes.**

Keywords*— *Testing Methodologies, Software Testing Life Cycle, Testing Frameworks, Automation Testing, Test Driven Development, Test optimisation, Quality Metrics

I. INTRODUCTION

Testing is defined as a process of evaluation that either the specific system meets its originally specified requirements or not. It is mainly a process encompassing validation and verification process that whether the developed system meets the requirements defined by user. Therefore, this activity results in a difference between actual and expected result. Software Testing refers to finding bugs, errors or missing requirements in the developed system or software. So, this is an investigation that provides the stakeholders with the exact knowledge about the quality of the product.

Software Testing can also be considered as a risk-based activity. The important thing during testing process the software testers must understand that how to minimise a large number of tests into manageable tests set, and make wise decisions about the risks that are important to test or what are not [1].

Figure 1 shows the testing cost and errors found a relationship. The Figure 1 clearly shows that cost goes up dramatically in testing both types i.e. functional and non-functional. The decision making for what to test or reduce tests then it can cause to miss many bugs. The effective testing goal is to do that optimal amount of tests so that extra testing effort can be minimised [1].

According to Figure 1, Software testing is an important component of software quality assurance. The importance of testing can be considered from life-critical software (e.g., flight control) testing which can be highly expensive because of risk regarding schedule delays, cost overruns, or outright cancellation [2], and more about this [3][4].

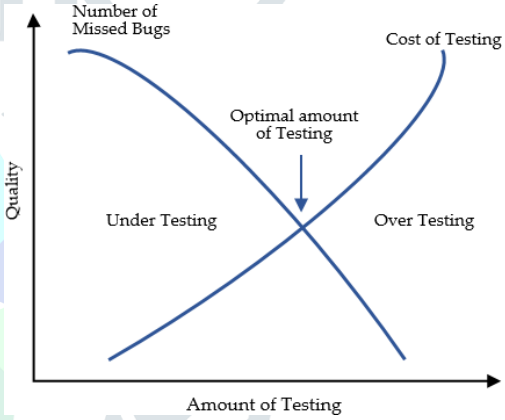

Figure 1: Every Software Project has optimal test effort (Courtesy [1]).

Testing has certain levels and steps according to which the person who does the testing differs from level to level. The three basic steps in the software testing are Unit testing, Integration testing and System testing. Each of these steps is either tested by the software developer or the quality assurance engineer who is also known as a software tester [5]. The testing mentioned above steps is inclusive in the Software Development Lifecycle (SDLC). It is essential to break the software development into a set of modules where each module assigned to a different team or different individual. After the completion of each module or unit, it is tested by the developer just to check whether the developed module is working by the expectation or not, this is termed as Unit Testing. The second step of testing within the SDLC is Integration Testing. Once the modules of a single software system have been developed independently, they are integrated together and often errors arise in the build once the integration has been done. The final testing step in the SDLC is System Testing, which is testing of the whole software from each and every perspective. Also, software testing ensures that the integrated units do not interfere or disturb the programming of any other module. However, testing of a large or intensely complex systems might be an extremely time-consuming and lengthy procedure as the more components within the application, the more difficult it gets to test each combination