



# INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

## A REVIEW ON SOFTWARE TESTING AND QUALITY PROCESS IMPROVEMENT

MS. NILAJA A. DESHMUKH<sup>1</sup>, DR. H. R. DESHMUKH<sup>2</sup>

1. M.E (Scholar) Computer Science and Engg, I. B. S. S. C. O. E, Amravati.
2. Head of Department, Computer Science and Engg, I. B. S. S. C. O. E, Amravati.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

**Abstract:** In our daily life we increasingly depend on software based systems deployed as embedded software control system. Increasing emphasis placed on high quality and user satisfaction of software calls for rethinking on the objectives and management of testing. The successful introduction of such system into business however does depend whether we trust the system or not. Testing is commonly applied as the predominant activity in industry to ensure high software quality providing a variety of methods and techniques to detect different types of errors in software based system. The key to improving the effectiveness of testing is to improve the attitude of software developer. Towards testing and nature and culture of the organisation. The panel goal is to discuss software testing strategy and techniques to improve the quality if the software and at the same time to build the trust.

**Keywords:** QA, QC, SDLC, RTM, GSD

Corresponding Author: MS. NILAJA A. DESHMUKH



PAPER-QR CODE

Access Online On:

[www.ijpret.com](http://www.ijpret.com)

How to Cite This Article:

Nilija A. Deshmukh, IJPRET, 2015; Volume 3 (9): 1068-1075

## INTRODUCTION

Testing is one of the most important parts of quality assurance (**QA**) and the most commonly performed **activity**. Software testing is a process which helps to identify the correctness, completeness, security and quality of the developed software application. And test the software application to detect the defect before they are seen by the end user. The QA & Software Testing Guide describes the standardized model for Quality Assurance in product development and project work from the perspective of the project team members and QA personnel. The basic idea of testing involves the execution of software and the observation of its concepts, issues, and techniques related to testing in this chapter. Although QA & Software Testing Guide is primarily intended for QA personnel and project team members. [1]

## 2. MATERIALS AND METHODS

### 2.1 TESTING OVERVIEW.

#### 2.1.1. Who does testing?

- Software Tester
- Software Developer
- Project Lead/Manager
- End User

Different companies have difference designations for people who test the software on the basis of their experience and knowledge such as Software Tester, Software Quality Assurance Engineer, and QA Analyst etc.

#### 2.1.2. When to Start Testing

An early start to testing reduces the cost, time to rework and error free software that is delivered to the client. However in Software Development Life Cycle (SDLC) testing can be started from the Requirements Gathering phase and lasts till the deployment of the software.

Testing is done in different forms at every phase of SDLC like during Requirement gathering phase, the analysis and verifications of requirements are also considered testing.

### **2.1.3. When to Stop Testing**

Unlike when to start testing it is difficult to determine when to stop testing, as testing is a never ending process and no one can say that any software is 100% tested. Following are the aspects which should be considered to stop the testing:

- Testing Deadlines.
- Completion of test case execution.
- Completion of Functional and code coverage to a certain point.[1]

## **3. TESTING TYPE.**

### **3.1. Verification:**

Verification focuses on “find defects as early as possible”. Verification makes sure that the software application is getting developed in the correct way with respect to the requirements- Are we doing the job right? Verification takes place first and includes the checking for documentation, code etc. Have static activities as it includes the reviews, walkthroughs, and inspections to verify that software is correct or not. It is an objective process and no subjective decision should be needed to verify the Software.

### **3.2. Validation:**

The software application should functionality do what it is supposed to and it satisfy all the requirements set by the customer and end user. Validation is done during or at the end of the development process to determine whether the application satisfies specified requirements. Ensure that functionalities meet the intended behaviour. Validation occurs after verification and mainly involves the checking of the overall product. Validation and verification processes go hand on hand, but visibly validation process starts after verification process end.[2]

## **4. TESTING METHOD**

### **4.1. Black Box Testing:**

The process of giving the input to the system and checking the output of the system, without knowing about how the system generates the output is called black box testing. It is also called as functional testing or behaviour testing.

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code.

#### **4.2. White Box Testing:**

White box testing is done by checking each line of program code. It is the process of giving input and checking how it process the input and generate output. White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.[2]

### **5. FUNCTINAL TESTING**

#### **5.1. Unit Testing:**

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team.

#### **5.2. Integration Testing:**

The testing of combined parts of an application to determine if they function correctly together is Integration testing. There are two methods of doing Integration Testing Bottom-up Integration testing and Top Down Integration testing.

**5.2.1. Bottom-up integration** testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

**5.2.2. Top-Down integration** testing, the highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

## 5.2. System Testing:

System Testing is the first step in the Software Development Life Cycle, where the application is tested as a whole. The application is tested thoroughly to verify that it meets the functional and technical specifications. Is tested in an environment which is very close to the production environment where the application will be deployed. It enables us to test, verify and validate both the business requirements as well as the Applications Architecture.

## 5.3. Acceptance Testing:

This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements. Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors or Interface gaps, but also to point out any bugs in the application that will result in system crashers or major errors in the application. By performing acceptance tests on an application the testing team will deduce how the application will perform in production.

### 5.3.1. Alpha Testing:

This test is the first stage of testing and will be performed amongst the teams (developer and QA teams). Unit testing, integration testing and system testing when combined are known as alpha testing

### 5.3.2. Beta Testing:

This test is performed after Alpha testing has been successfully performed. Beta testing is also known as pre-release testing. Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program a "real-world" test and partly to provide a preview of the next release. [2]

## 6. NONFUNCTIONAL TESTING

### 6.1. Performance testing:

It is mostly used to identify any bottlenecks or performance issues rather than finding the bugs in software. There are different causes which contribute in lowering the performance of software:

1. Network delay.
2. Client side processing.
3. Database transaction processing.
4. Load balancing between servers.

#### **6.1. Load testing:**

A load test is conducted to understand the behaviour of the system under a specific expected load as per requirement document.

#### **6.2. Endurance Testing:**

Endurance testing determines whether a system can sustain a continuous expected load for a longer duration of time.

#### **6.3. Stress testing:**

Stress testing is used to evaluate the ability of an application to maintain a certain level of performance effectiveness under unfavourable conditions.

#### **6.4 .Volume testing:**

It refers to testing an application with a huge amount of data and checking its limitations in term of performance.

### **7. QUALITY AND PROCESS IMPROVEMENT**

#### **7.1. Quality Assurance:**

QA is a set of procedures that should be followed while developing product or survive to assure of good quality to the customers. It include measuring the processes, identifying the deficiencies or weakness in processes, suggest improvement and refine the processes. Conducting quality Audits based on QMS is an example of QA process.

#### **7.2. Quality Control:**

QC is a set of activities performed to ensure that the product meets the requirements. It is done during the development process and also once the product is developed. QC measures

the product, identifies the dependencies. It is an activity which verifies the whether or not the product meets all the standards [4][5]

## 8. TEST DOCUMENTATION

Testing documentation involves the documentation which should be developed before or during the testing of Software.

### 8.1. Test Plan:

A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment in which testing will be performed, the limitations of the testing and the schedule of testing activities. Typically the Quality Assurance Team Lead will be responsible for writing a Test Plan.

### 8.2. Test Scenario:

Test Scenarios are used to ensure that all process flows are tested from end to end. The term test scenario and test cases are used interchangeably however the main difference being that test scenarios has several steps however test cases have a single step. When viewed from this perspective test scenarios are test cases, but they include several test cases and the sequence that they should be executed. Apart from this, each test is dependent on the output from the previous test

### 8.3. Test Case:

Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks. The main intent of this activity is to ensure whether the Software Passes or Fails in terms of its functionality and other aspects.

### 8.4. Traceability Matrix:

Traceability Matrix (also known as Requirement Traceability Matrix - RTM) is a table which is used to trace the requirements during the Software development life Cycle. It can be used for forward tracing i.e. from Requirements to Design or Coding or backward i.e. from Coding to Requirements. [5][6]

**9. CONCLUSION:** In this study, we identified benefits related to the process which improves the quality of software product. We studied approaches related to systematic review, systematic review procedure, acceptance test case specification and process-based learning to obtain the

expected quality result. It highlights human factor and management issues in current software testing practices and offers suggestions for improvement. And the quality of software based system is determine by many properties such as

Completeness, consistency, safety, maintainability, security, and usability among others.. In future we intend to propose a model which covers quality factors in GSD.

## **REFERENCES**

1. Marnie L. Hutcheson. Software Testing Fundamentals: Methods and Metrics. Willy 2003.
2. PRESSMAN R.S .Software engineering: A Practioer's approach. McGraw-Hill, 1997.
3. IEEE STD 1061-1998.IEEE Standard for a Software Quality Metrics Methodology. Institute of Computer Engineers. New York: IEEE Computer society, 2005.
4. Rosaria S. Robinson H." Applying model in your testing process". Information and software Technology, 2000.
5. Hu xiuwen, Tian zhonghe," Research on measurement of Software Quality: Information Technology, 1997.
6. J. H. August Joint Application design: The group session approach to system design, 1991: Yourdon Press.