Software Testing Techniques and Documentation

Navya Garimella¹, Ch.V.Phani Krishna² and V.Samson Deva Kumar³

¹Student, IV/IV B.Tech, Department of Computer Science and Engineering at K L University, Vaddeswaram, Andhra Pradesh
²Associate Professor at K L University, Department of Computer Science and Engineering Vaddeswaram, Andhra Pradesh
³Project Manager at South Central Railways WWO, S/W Training and Development center, E-world, Vijayawada, Andhra Pradesh

ABSTRACT

Software testing is a process that validates and verifies the quality of the program. It also improves the quality by finding the defects and problems. Many people are not aware of the methods in testing. Also the documentation plays main role. This paper mainly describes about the types of testing and the documentation involved in it. Furthermore, it describes about the limitations involved in documentation. Many people are not aware of the steps need to be followed while performing testing. In order to make them aware of the procedure we discussed about the documentation part.

Keywords: Software Testing, Methods, Types of Software Testing and Documentation

1. INTRODUCTION

Software testing is nothing but a process which describes about the verification and validation about the product meets the requirement that guided its design and development, works as expected, can be implemented with the same characteristics and satisfies the needs of the stake holders. It can be implemented at any time in the development process. A primary purpose of software testing is to detect software failures so that they may be discovered and corrected. Software testing methods are usually divided into the box approaches. And are used to describe the point of view that a test engineer takes while testing. Many people do not organize the work they need to complete/perform. Documentation helps in solving this problem. A tester needs to know the methods of testing and how to do it. This paper describes the process. Also the limitations involved in testing.

2. TESTING TYPES

There are many types of software testings. Some of them are acceptance testing, load testing, regression testing, performance testing, smoke testing, stress testing and system testing etc. Each type has its advantages and limitations. The software testing methods are traditionally divided into white and black box testing.

2.1 REGRESSION TESTING

To explain in brief, suppose if we have tested a product and found no errors and then some changes are done to the product a new version of product is released. Then this testing allows consistent and repeatable validation of new product. It also ensures that defects are corrected after changes are done at each stage and no new quality problems are encountered. The practical way to manage regression testing is to automate it.

2.2 SMOKE TESTING

This type of testing is carried out by software testers to check if the new build provided by development team is stable enough i.e., major functionality is working as expected in order to carry out further or detailed testing. Smoke testing is intended to find “show stopper” defects that can prevent testers from testing the applications in detail. Smoke testing is carried out for a build is also known as build verification test.

2.3 STRESS TESTING

In this testing, the software is subjected to peak loads and even to a breakpoint to observe how software behaves at breakpoint. It also tests the behavior of the software with insufficient resources like CPU, memory, network bandwidth, disk space etc. Also it enables to check some of the quality attributes like robustness and reliability. It has its own advantages and disadvantages. Spike testing is a subset of stress testing. Spike testing mainly focuses on validating the performance characteristics of the product under test when subjected to overload work modules that repeatedly increase beyond the anticipated production operations for short periods of time.

2.4 ACCEPTANCE TESTING

This testing is mainly done by the end users to check if the software required by them is met by their own business requirements. Also to the requirements provided earlier. Such types of testings are usually documented but the
document test cases for smaller test cases are not documented. But there are some disadvantages with this testing which many people might not be aware of. Firstly, it requires some planning of resources and it may be re-implementation of system tests. And coming to next point it may not care about the subjective defects in the software because you are looking only for the defects you are expecting to find. [9]

3. METHODS OF TESTING

Coming to methods of testing, I would like to discuss mainly about the black box testing and white box testing. Let’s start with the black-box testing.

3.1 Black box testing:
Black box testing also known as functional testing does not take into consideration the internal mechanism, code structure and implementation details. It just cares about the input and output. This testing is mainly based on the software requirements and specifications. The following diagram represents the procedure.

![Black box testing diagram](image)

**Figure 1**: Black box testing. A black-box test takes into account only the input and output of the software without regard to the internal code of the program.

Now let’s discuss about the steps followed to carry out any type of black box testing. There are basically some steps need to be followed to implement black box testing or else the process becomes difficult. They are as follows.

- Firstly, the requirements and specifications of the system are examined.
- Next the tester chooses some valid inputs to check if the software under test processes them correctly. Also he takes some invalid inputs to verify if the software under test is able to detect them.
- Tester then assesses the expected outputs for all those inputs.
- Next the software tester conducts test cases with the selected inputs.
- The test cases are executed.
- Then the tester compares the actual outputs to the expected outputs.
- If any defects they are fixed and re-tested.

3.1.1 Black box testing strategies:
There are some testing strategies to be followed while performing black box testing. Some of the prominent ones are:

1. Equivalence class testing: It helps in minimizing the number of possible test cases to an optimum level still maintaining the reasonable test coverage.
2. Boundary Value testing: It focuses on the values at boundaries. It helps in determining whether a certain range of values are acceptable by system or not. This technique is so helpful in reducing the number of test cases. It is mostly suitable for the systems where input is within certain range.
3. Decision table testing: This testing puts causes and effects in a matrix. There is unique combination in each row.

3.2 White box testing:
White box testing is quite contrast to black box testing. In this testing the internal mechanism and coding is taken into consideration. It requires access to source code. White box testing requires knowing what makes software secure or insecure, how to think like an attacker and how to use different testing tools and techniques. Similar to black box testing some specific steps needs to be followed to process white box testing. Following are the steps.

1. Thread modeling is performed [Howard 06] i.e. risk analysis to guide whole testing process is performed.
2. A test strategy is developed which defines what testing activities are required to accomplish testing goals.
3. A detailed test plan is developed that organizes the subsequent testing process.
4. A test environment for test execution is prepared.
5. Execute test cases and communicate results.
6. Prepare a report.

Many people are in a dilemma whether to use white box testing or black box testing. When we considered the pros and cons of both the testings it is proved that white box testing is considered to be more preferable than black box testing.
4. DOCUMENTATION

Testing documentation involves the documentation of the artifacts which should be developed before or after the testing. Documentation for software testing helps in estimating the testing effort required, test coverage, requirement tracing etc. Commonly used artifacts are

- Test Plan
- Test Scenario
- Test Case
- Traceability Matrix

As mentioned above in the types of testing a certain sequence of steps are to be performed to test a document. Each and every tester needs to follow these steps. But many people do not follow these steps and so as a result sometimes defects may not be checked properly. By taking these steps into consideration a software tester can become successful and can avoid mistakes in testing. Step wise explanation is done in the following paragraphs. Many of them are not aware of this documentation phase. So step wise explanation is done. Some examples are mentioned below.

4.1 Test Plan

Test plan outlines the strategy that will be used to test an application, the resources that will be used, the testing environment in which testing will be performed, the limitations of the testing and the scheduling of testing activities. A test plan will include the following.

- Introduction to the test plan document
- Assumptions while testing the application
- List of test cases included in testing the application
- List of features to be tested
- What sort of approach to use when testing the software
- List of deliverables that need to be tested
- The resources allocated for testing the application
- Any risks involved during the testing process
- A schedule of tasks and milestones as testing is started

4.2 Test Scenario

Test scenario tells in what area the application will be tested. It is used mainly to ensure that all process flows are tested from end to end. A particular area of an application can have as little as one test scenario to a few hundred scenarios depending on the magnitude and complexity of the application. On the contrary, the terms test cases and test scenarios are used interchangeably but the main difference between them is that test scenarios have several steps whereas test cases have one step. From this perspective test scenarios are test cases, but they include several test cases and the sequence that they should be executed. Also each test is dependent on the output from the previous test.

The following diagram shows the pictorial representation of the test scenarios.

![Diagram representation of test scenarios](image)

**Figure 2:** A diagram representing the presentation of the test scenario.[10]

4.3 Test Case

The main aim of the test cases is to ensure whether the software passes or fails in terms of its functionality and other aspects. Test cases involve the set of conditions, steps etc. There are many types of test cases like functional, negative, error, logical test cases, physical test cases and UI test cases etc. Also test cases help in keeping track of the testing coverage status. Though a specific template is not followed while writing a test case, there are some main components available and included in every test case: They are

- Test case ID
Sometimes, multiple test cases are written for single software which is collectively known as test suites. Let’s take an example test case which involves a sample test case about entering into Gmail account.

Description of test case:

<table>
<thead>
<tr>
<th>SNO</th>
<th>ACTION PERFORMED</th>
<th>EXPECTED OUTPUT</th>
<th>ACTUAL OUTPUT</th>
<th>TEST CASE FAIL OR PASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Double click on Mozilla icon</td>
<td>Mozilla will be opened and mouse pointer will be on the address bar</td>
<td>Mozilla will be opened and mouse pointer will be on the address bar</td>
<td>PASS</td>
</tr>
<tr>
<td>2</td>
<td>Type <a href="http://www.gmail.com">www.gmail.com</a> in the address bar of Mozilla and press ENTER</td>
<td>1. Gmail page will be opened and we can see the login details. 2. We can see Gmail id, password text boxes and sign in and cancel buttons.</td>
<td>1. Gmail page will be opened and we can see the login details. 2. We can see Gmail id, password text boxes and sign in and cancel buttons</td>
<td>PASS</td>
</tr>
<tr>
<td>3</td>
<td>Insert valid Gmail id and password in the text boxes and press ENTER.</td>
<td>Cursor should not move from the id and password test boxes.</td>
<td>Cursor should not move from the id and password test boxes</td>
<td>PASS</td>
</tr>
</tbody>
</table>

Figure 3: A table representing an example of test case which describes about the logging into Gmail account.

4.4 Traceability Matrix

It is also known as Required Traceability Matrix (RTM). It is a table which is used to trace the requirements during the Software development life cycle. It can be used for forward tracing which means from requirements to design or coding or backward i.e. from coding to requirements. There are many user defined template for RTM.

Each requirement in the RTM document is linked with its associated test case, so that testing can be done as per the mentioned requirements. The main goals for the matrix are:

1. Make sure software is developed as per the customer requirements.
2. Helps in finding the root cause of any bug.
3. Helps in tracing the developed documents during different phases of Software Development Life Cycle.

5. CONCLUSION

This paper on Software testing describes about the types of testing, methods involved in testing and the documentation of the application. Software testing is not that formal because we have struggled to define its strategies and importance. To perform testing effectively one should be aware of advantages, goals, principles and disadvantages of it. Here we also discussed about the types of testing techniques and the methods of testing. Also many people are not aware of the documentation of testing. The process of testing should be learnt before testing an application. Finally, there is comparison between testing and debugging. But testing is more prominent. Testing is not only used to locate defects but also used in validation, verification process and measurement.
REFERENCES