

Fundamentals of Software Testing

Introduction

- IEEE definition,
- “Software testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item”.
- IEEE’s Guide to the Software Engineering Body of Knowledge, SWEBOK, states that
 - “Software testing is an activity that should be done throughout the whole development process”.
- The main focus of any testing technique is to make the software bug free

Testing Strategies and Techniques

- A test strategy is a concise statement that describes how the objectives of the software testing are met.
- The test strategy views the test event at a high level, concentrates on the objectives of the test event, the techniques that can be used and the resources that are required.
- Developers should plan their approach to testing at every stage of the project and establish a framework for testing the project.

Testing Technique

- A testing technique is a process that ensures that the application being tested functions in a structured way. A testing strategy and technique is based on the method of testing adopted.
- Methods of testing.
 1. Structural or functional testing
 2. Static or dynamic testing
 3. Manual or automated testing

Structural VS Functional Testing

- If the test cases are developed to check the **actual structure of the program code**, then it is called **structural testing**.
- Structural testing is also known as **white box testing**,
 - where the tester checks the actual code of the software.
- **Functional testing**, the tester checks only the **behavior of the software** and will not check the actual code.
- The tester checks the response of the software for predefined inputs and tests whether the software produces the desired output.
- Therefore, this is called **black box testing**.

Structural VS Functional Testing

- If the test cases are developed to check how the entire system works, then it is called **functional testing**.
- functional testing, the tester uses test cases to check how the software works, *i.e., whether it produces the desired outputs for a set of given inputs.*
- Example
 - Functional testing for a calculator could check whether the software does the addition operation correctly.

Static VS Dynamic Testing

- Static testing refers to the analysis of the program, which is carried out without executing the program.
- A typical white box testing technique, where the developer checks the code to find errors in it. It is preventive in nature and is completed in the verification phase.
- The common methods include feasibility review and code review.
- Example
 - Software developers perform syntax check to test syntactical correctness

Static VS Dynamic Testing

- **Dynamic testing** refers to the analysis of the program in its executable form.
- This is performed by supplying valid entries and is validated against the expected results. It is a curative method and is performed during the validation phase.
- Example
 - Software developers perform unit test to check for correctness in a module.

Static testing is done to check mainly the correctness and logic of the code whereas, dynamic testing is performed to check the response of the system for predefined inputs.

Points to note

- Some of the important differences between static testing that makes it more effective and efficient are:
 1. Since static testing is carried out during the initial stage, it is cost effective compared to dynamic testing, which is carried out once the entire software, a module, or unit is complete.
 2. Static testing detects bugs at the earliest and hence the time required to fix them is less.

Manual versus Automated Testing

- When the software is tested by people to find the bugs, it is called manual testing process.
- The tester acts as an end user and uses all the features of the software, and checks to ensure that they behave correctly.
- Example
 - Performing a manual test for a login screen involves some of the following:
 1. Checking whether username and password can be entered.
 2. Implementing masking of password character.
 3. Verifying whether the screen navigates to next page if valid login details are provided.
 4. Checking the error message displayed when incorrect login details are entered.
 5. Checking the maximum number of characters that can be entered in the login and password fields.

Manual versus Automated Testing

- In automated testing, a software program, commonly referred to as 'a testing tool', runs the test software, provides proper inputs, and checks the output against the expected output.
- A tester writes the test case, and the automated testing tools run the test software according to the test case without any human intervention.
- At the end of the test, a detailed report is generated to indicate the result of any condition which is defined in the test case.

Manual versus Automated Testing

- Manual testing requires human intervention at every stage of the testing process right from writing test cases, providing the inputs, recording the output and analyzing the actual and expected output.
- Automated testing uses testing tools to perform the test. These tools need initial human intervention for supplying test cases. These tools are very powerful and reduce the time required for testing and are very efficient in finding out bugs in software.
- Investment cost is very high

Limitations of Manual Testing:

1. During the process of testing, the possibility of recurrence of a bug can actually have an impact on the time taken for testing the software.
2. Frequent changes in the user scenarios can lead to high maintenance costs in manual testing.
3. Repetitive tests reduce the cycle time for executing a test, and hence automation testing is opted for.

Advantages of Automation Testing

1. It reduces the time consumed to perform repetitive tests.
2. It requires less human effort and less number of resources.
3. It generates a test report which provides information of the test execution unlike a manual test which is written and documented.
4. It helps in regression testing (*Testing which is done to check whether the changes made in a module affects the working of the other existing modules*) and also helps in re-running tests against new releases.
5. It helps in testing large sequences of data and transactions and also in randomly searching for errors in the software.
6. It helps in testing several simultaneous users at a time virtually and can also analyze the load Generated for the program -- which cannot be done in manual testing.
7. It helps in testing web-based systems for performance reliability.

Role of a Software Tester

- The main goal of a software tester is to *find bugs, find them early* and also ensure that the changes incorporated due to the correction does not affect other functionalities.
- Software developers have good problem solving skills and always demonstrate that the software works as intended.
- A tester, on the other hand, demonstrates the weakness of an application. The application developed is checked with test cases or configurations, which gives errors or unexpected results to show where exactly the software breaks.

Role of a Software Tester

- Some of the responsibilities of a software tester during the process of testing are:
 1. Understanding the product/application by analyzing the specifications.
 2. Implementing a test strategy, which includes writing appropriate test plans, prioritizing the testing by assessing the risks, and setting up test data.
 3. Setting up the test environment for manual and automated tests.
 4. Providing reports that list the product defects and metrics.

Tasks of a Software Tester

- Testers co-ordinate with developers and conduct test case reviews of project areas.
- The tasks of a software tester:
 1. To ensure that the test methodology, techniques, and standards are established, developed and documented.
 2. To study and understand the requirements and accordingly prepare verification and validation test plans.
 3. To impact product quality by understanding customer needs.
 4. To develop test plans, test scenarios, and test cases.
 5. To prepare the test data and execute test cases.

Tasks of a Software Tester

6. To automate test cases.
7. To perform both verification and validation testing of the hardware and the software.
8. To prepare test reports and maintain test records.
9. To submit reports that details the schedule progress, the defects and the usability of the product.
10. To track defects and ensure closure of defects through reviews.

Qualities of a Software Tester

- A good software tester should possess strong analytical skills with good domain knowledge.
- The following are the qualities of a good software tester:
 1. Have a strong desire for quality.
 2. Be explorative in approach to venture unknown situations.
 3. Have a creative and relentless approach to discover bugs.
 4. Be tactful and diplomatic with developers while conveying where the software lacks.
 5. Possess good ability to understand customer needs.

Qualities of a Software Tester

6. Be able to compromise between the available resources and be in a position to focus on the most likely areas of bugs when there is insufficient time.
7. Possess good judgment skills to assess high-risk areas of an application.
8. Be sharp enough to observe the small changes.
9. Have a good understanding about the software development process.
10. Be technically aware of testing methods, tools, and criteria.

Software Testing Axioms

- The world of software testing is not just dictated by the models discussed and followed. In reality there are many trade-offs software testing effort faces. In the current day scenario, it is less likely that clients are able to determine every requirement analysis aspect in one-go.
- Requirements can keep changing during the course of time. Hence there are more chances for some of the following realities to occur.
 - a) The specification might not correspond to the customer's needs perfectly.
 - b) Many a time, the time available for testing would not be comprehensive to cover all aspects of testing.
 - c) Tradeoffs and concessions are inevitable.

Realities of Software Testing

- A detailed specification which can perfectly meet the needs of a customer is not given and there is insufficient time to test the software in its entirety. However, if one aims to become a good software tester, he or she needs to:
 - (a) Identify the ideal process involved.
 - (b) Identify the bugs and problems and realize how they affect the project.

Axiom 1

- It is impossible to test a program completely
- A tester may not be completely sure about the number of test cases needed to exhaustively test an application, for example
- Testing an MS Word document with all possible test cases covering all functions would be a difficult task to complete.
 - i. *The only way to ensure that the software works perfectly is to test it with all possible inputs and observe and monitor its outputs.*
 - ii. *questions do arise about the number of possible inputs being very large, the number of possible outputs being very large, the number of paths through the software being very large or the software specification itself being open to interpretation.*

Axiom 2

- Software testing is a risk-based exercise.
- When one does not test the software with all the possible inputs, they may end up taking a fair amount of risk, wherein there are possibilities of skipping some of the inputs which work correctly.
- At this stage, one faces the risk of skipping inputs which can cause a failure and may lead to financial loss or loss of security or even loss of life.

Axiom 2

- Software testing is considered to be a risk-based regime of practice, where one can find that:
 - a) Testing too much can result in high developmental costs.
 - b) Testing too little can result in the failure of the developed software, which can incur heavy cost to an organization.
 - c) The general cost involved in testing the number of missed bugs, over testing and under testing are more.

Axiom 3

Testing cannot show the absence of bugs.

- During the course of testing, we cannot completely eliminate the bugs, unless the software is dismantled to the foundation stage.
- Software testing is a process which can reveal the existence of a bug, but cannot reveal that there are no bugs in the software.
- Although, tests are performed to report and fix the bugs, it is not possible to guarantee bug free software.

Axiom 4

- The more bugs you find, the more bugs there are.
- Bugs in real life and bugs in software are very much alike. They come in groups and when you happen to accidentally notice one of them, there are possibilities of finding another one very soon.
- Most often a tester finds a bug only after long hours of testing, and when he/she encounters one of the bugs, he/she would soon find another one due to programmer's errors, where mistakes often are repeated or different programmers handling a module may have different habits of coding.

Axiom 5

- Not all bugs found will be fixed.
- ***There's Not Enough Time:***
 - Every project has several software features where only few people are involved in coding and testing and hence it becomes difficult to adhere to stringent time schedules to complete assigned tasks.
- ***It's Really Not a Bug:***
 - A bug found need not be a bug but could turn out to be its characteristic feature. There are possibilities of mistaking features as bugs.
- ***It's Too Risky to Fix:***
 - Most often, it has been found that it is indeed very risky to fix bugs. One should first fix the bug which might cause other bugs to appear and it is also necessary to ensure that last moment changes of software do not occur during the release of a product.
- ***It's Just Not Worth it:***
 - Some of the bugs which affect the fringe features are dismissed.

Axiom 6

- It is difficult to say when a bug is indeed a bug.
- We need to analyze the following:
 - a) When a problem in the software is not discovered - is it a bug?
 - b) Is it necessary for a bug to be observable?

Axiom 7

- Specifications are never final.
- The changing specifications make it difficult for complete testing to take place. Specifications can
- Change due to:
 - a) Fierce Competition.
 - b) Rapid release cycles.
 - c) Change requirements.

Axiom 8

- Software testers are not the most popular members of a project.
- Software testers do have goals to:
 - a) Find bugs early and ensure that they are fixed as early as possible.
 - b) Ensure that they adhere to professional behavior without losing their temper.

Axiom 9

- Software testing is a disciplined and technical profession.
- Initially software testers were untrained and did not follow any methodology, as the software was simpler and manageable. However, testing has now become a matured discipline and supports sophisticated techniques with good support of tools and also provides a rewarding career for the testers.

Software Testing Terms and Definitions

- Software Quality:
- Software quality is impacted by bugs. It is essential that the software is bug free or defect free and meets the requirements, specifications, and expectations of the client.
- Verification and Validation:
 - Verification is the process of discovering the possible failures in the software before the commencement of the testing phase. It involves reviews, inspections, meetings, code reviews, and specifications. It answers the question, "Are we building the product right?"
 - Validation occurs after the verification process and the actual testing of the product happens at a later stage. Defects which occur due to discrepancies in functionality and specifications are detected in this phase. It answers the question, "Are we building the right product?"

Software Testing Terms and Definitions

- Quality Assurance (QA) Vs Quality Control:
- Quality Assurance and Quality Control - define the quality management activities of a project.
- Quality assurance –
 - refers to the planned and systematic activities that monitor and ensure that the development and maintenance process meets its objectives,
- quality control –
 - refers to a set of activities that are designed to evaluate a developed product.
- Quality assurance is more a verification process
- Quality control is more a validation process

Software Testing Terms and Definitions

- **Quality assurance**

- The activities of quality assurance can be performed **while** the product is being developed, whereas the activities of quality control are performed **after** the product is being developed.
- Aim is to prevent defects, and thus it focuses on the process of product or application building.
- can be performed by a manager or a third party professional.
- ensures that the process is well defined and is performed through the life cycle of the product.
- Activities include quality management review functions like process checklists, project audits, and standards development for development in coding.
- identifies areas of improvement in the processes through prevention plans and also ensures that the processes followed are effective.

Software Testing Terms and Definitions

- **Quality Control**

- Evaluate whether the deliverables are of acceptable quality
- Detect defects and correction of these defects (correction oriented).
- Activities ensure data integrity, correctness and completeness
- activities include inspections, reviews, and walk throughs of design, code and documentation
- addresses bugs, errors and omissions

Software Testing Terms and Definitions

- Quality assurance
- - check whether the product is “fit for purpose” and also to ensure that it is built “right the first time”
- Quality control
 - works on the principle, “fix the problem”.
 - testing methods like unit testing, integration
 - testing and system testing are the commonly used methods

Software Testing Terms and Definitions

- Test Plan:
- A test plan is a document which gives information about the objectives, scope, approach and the various attributes that the testing project must focus on.
- **Test Case:**
- is a document and is the smallest unit of testing.
- It has a developed set of inputs, execution preconditions and expected outcomes for a specific objective. This is done to ascertain that the feature of a particular application is working as specified.
- test case generally contains test case identifier, test case name, objective, test conditions/setup, input data requirements, steps, and expected results.

Terms and Definitions

| Terms | Definitions |
|---------------------|--|
| Acceptance Testing | Acceptance testing is conducted by the customer or the user to check whether the software product meets the requirements. |
| Agile Testing | Agile testing is for testing the software product from the customer perspective at an early stage. Testing is carried out once the codes become available. |
| Automated Testing | Automated testing is a procedure of using automated tools to execute tests. |
| Bug | A bug is an error or a defect in a program which is unintended. |
| Debugging | Debugging is the process of detecting and eliminating the causes of software errors. |
| Defect | A defect is an error or non-conformance of a specific program. |
| Integration Testing | Integration testing is performed on interfaces between components. |
| System Testing | System testing relates to testing the system after integrating the units, to ensure that specifications are met. |
| Unit Testing | Unit testing is the process of testing the basic unit of software, which is the smallest testable piece of software. |

Questions ?