Types of

Testing Lecture

3

Principles of Software Testing

- There are seven principles of Software Testing. 1. Testing shows presence of defects

  2. Exhaustive testing is impossible

  3. Early testing reduces the cost

  4. Defect clustering

  5. Pesticide paradox

  6. Testing is context dependent

  7. Absence of error – fallacy

Principles of S/W Testing Cont…

- **Testing Shows Presence of Defects:** Testing shows the presence of defects in the software. The goal of testing is to make the software fail. Sufficient testing reduces the presence of defects. In case testers are unable to find defects after repeated regression testing doesn’t mean that the software is bug-free.

- **Exhaustive Testing is Impossible:** Testing all the functionalities using all valid and invalid inputs and preconditions is known as Exhaustive testing. If we keep on testing all possible test conditions then the software execution time and costs will rise. So instead of doing exhaustive testing, risks and priorities will be taken into consideration whilst doing testing and estimating testing efforts.

Principles of S/W Testing Cont…

- **Early Testing:** Defects detected in early phases of SDLC are less expensive to fix. So conducting early testing reduces the cost of fixing defects. It is cheaper to change the incorrect requirement compared to fixing the fully developed functionality which is not working as intended.

- **Defect Clustering:** Defect Clustering in software testing means that a small module or functionality contains most of the bugs or it has the most operational failures. As per the Pareto Principle (80-20 Rule), 80% of issues comes from 20% of modules and remaining 20% of issues from remaining 80% of modules. So we do emphasize testing on the 20% of modules where we face 80% of bugs.

Principles of S/W Testing Cont…

- **Pesticide Paradox:** The process of repeating the same test cases again and again, eventually, the same test cases will no longer find new bugs is called

Pesticide Paradox in software testing. So to overcome this Pesticide Paradox, it is necessary to review the test cases regularly and add or update them to find more defects.

- **Testing is Context Dependent:** Testing approach depends on the context of the software we develop. We do test the software differently in different contexts. For example, online banking application requires a different approach of testing compared to an e-commerce site.

Principles of S/W Testing Cont…

- **Absence of Error – Fallacy:** The absence of error does not necessarily mean that the software is error free. 99% of bug-free software may still be unusable, if wrong requirements were incorporated into the software and the software is not addressing the business needs. The software which we built not only be a 99% bug-free software but also it must fulfill the business needs otherwise it will become an unusable software. The testing team must start with the hypothesis that there are errors in the software. Using test cases and other methods, the testing team desires to prove the hypothesis wrong.

Software Assessment

- Two types-
  - Static Analysis: based on factual system documents available such as system requirement study document,

design document, and source code.

– Dynamic Analysis: based on the behavioral and performance properties.

Types of Software Testing •

Given below is the list of some common types of Software Testing:

Functional Testing include: – Unit Testing

– Integration Testing
– System Testing
– Sanity Testing
– Smoke Testing
– Interface Testing
– Regression Testing
– Acceptance Testing
– Localization Testing

Non-functional Testing include:

– Performance Testing – Load Testing

– Stress Testing

– Volume Testing

– Security Testing – Compatibility Testing – Install Testing

– Recovery Testing – Reliability Testing – Usability Testing – Compliance Testing

Functional vs Non Functional Test.

- Functional testing verifies each

function/feature of the software whereas Non Functional testing verifies non functional aspects like performance, usability, reliability, etc.

- Functional testing can be done manually whereas Non Functional testing is hard to perform manually. • Functional testing is based on customer's requirements whereas Non Functional testing is based on customer's expectations.

- Functional testing has a goal to validate software actions whereas Non Functional testing has a goal to validate the performance of the software.

Functional vs Non Functional Test.

- A Functional Testing example is to check the login functionality whereas a Non Functional testing example is to check the dashboard should load in 2 seconds.
- Functional describes what the product does whereas Non Functional describes how the product works.
- Functional testing is performed before the non-functional testing.

Alpha Testing

- It is the most common type of testing used in the Software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user.

- Alpha Testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing.

- Alpha Testing is conducted at the developer's site. In house virtual user environment can be created for this type of testing.

Acceptance Testing

- An Acceptance Test is performed by the client and verifies whether the end to end flow of the system is as per the business requirements or not and if it is as per the needs of the end-user. Client accepts the software only when all the features and functionalities work as expected.

- It is the last phase of the testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

Ad-hoc Testing

- The name itself suggests that this testing is performed on an Ad-hoc basis i.e. with no reference to the test case and also without any plan or documentation in place for such type of testing.
- The objective of this testing is to find the defects and break the application by executing any flow of the

application or any random functionality.

- Ad-hoc Testing is an informal way of finding defects and can be performed by anyone in the project.
- It is difficult to identify defects without a test case but sometimes it is possible that defects found during ad-hoc testing might not have been identified using existing test cases.

Accessibility Testing

- The aim of Accessibility Testing is to determine whether the software or application is accessible for disabled people or not.

- Here, disability means deaf, color blind, mentally disabled, blind, old age and other disabled groups.

- Various checks are performed such as font size for visually disabled, color and contrast for color blindness, etc.

Beta Testing

- Beta Testing is a formal type of Software Testing which is carried out by the customer. It is performed in the Real Environment before releasing the product to the market for the actual end-users.
- Beta Testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Usually, this testing is typically done by end-users or others. The Beta version of the software or product is released to a limited certain number of users in a specific area.

- So end-user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

Back-end Testing

- Whenever an input or data is entered on front-end application, it stores in the database and the testing of such database is known as Database/Back-end Testing.
- There are different databases like SQL Server, MySQL, and Oracle, etc. Database Testing involves

testing of table structure, schema, stored procedure and so on.

- In Back-end Testing GUI is not involved, testers are directly connected to the database with proper access and testers can easily verify data by running a few queries on the database.
- There can be issues identified like data loss, deadlock, data corruption etc during this back-end testing and these issues are critical to fixing before the system goes live into the production environment

Compatibility Testing

- It is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment.

- Compatibility testing ensures that software can run on a different configuration, different database, different browsers, and their versions.

- Compatibility testing is performed by the testing team.

Browser Compatibility Testing •

It is a subtype of Compatibility Testing and is performed by the testing team.

- Browser Compatibility Testing is performed for web applications and it ensures that the software can run with the combination of different browser and operating system.

- This type of testing also validates whether web application runs on all versions of all browsers or not.

Backward Compatibility Testing

- It is a type of testing which validates whether the newly developed software or updated software works well with the older version of the environment or not.

- Backward Compatibility Testing checks whether the new version of the software works properly with file format created by an older version of the software; it also works well with data tables, data files, data structure created by the older version of that software.
- If any of the software is updated then it should work well on top of the previous version of that software.

Boundary Value Testing

- This type of testing checks the behavior of the application at the boundary level.
- Boundary Value Testing is performed for checking if defects exist at boundary values. Boundary Value Testing is used for testing a different range of numbers.
- There is an upper and lower boundary for each range

and testing is performed on these boundary values. • If testing requires a test range of numbers from 1 to 500 then Boundary Value Testing is performed on values at 0, 1, 2, 499, 500 and 501.

Equivalence Partitioning

- It is a testing technique and a type of Black Box Testing. During this Equivalence Partitioning, a set of the group is selected and a few values or numbers are picked up for testing. It is understood that all values from that group generate the same output.

- The aim of this testing is to remove redundant test cases within a specific group which generates the same output but not any defect.
- Suppose, the application accepts values between -10 to +10 so using equivalence partitioning the values picked up for testing are zero, one positive value, one negative value. So the Equivalence Partitioning for this testing is -10 to -1, 0, and 1 to 10.

Graphical User Interface

(GUI) Testing

- The objective of this GUI Testing is to validate the GUI as per the business requirement. The expected GUI of the application is mentioned in the Detailed Design Document and GUI mockup screens.
- The GUI Testing includes the size of the buttons and input field present on the screen, alignment of all text, tables, and content in the tables.
- It also validates the menu of the application, after selecting different menu and menu items,

it validates that the page does not fluctuate and the alignment remains same after hovering the mouse on the menu or sub-menu.

Gorilla Testing

- Gorilla Testing is a testing type performed by a tester and sometimes by the developer the as well.
- In Gorilla Testing, one module or the functionality in the module is tested thoroughly and heavily.
- The objective of this testing is to check the robustness

of the application.

Happy Path Testing

- The objective of Happy Path Testing is to test an application successfully on a positive flow.

- It does not look for negative or error conditions.
- The focus is only on the valid and positive inputs through which application generates the expected output.

Incremental Integration Testing

- Incremental Integration Testing is a Bottom-up approach for testing i.e continuous testing of an application when new functionality is added.
- Application functionality and modules should be independent enough to test separately. This is done by programmers or by testers.

Install/Uninstall Testing

- Installation and Uninstallation Testing is done on full, partial, or upgrade install/uninstall processes on different operating systems under different hardware or software environment.

Load Testing

- It is a type of Non-Functional Testing and the objective of Load Testing is to check how much load or maximum workload a system can handle without any performance degradation.
- Load Testing helps to find the maximum capacity of the system under specific load and any issues that cause software performance degradation.
- Load testing is performed using tools like JMeter, LoadRunner, WebLoad, Silk performer, etc.

Monkey Testing

- Monkey Testing is carried out by a tester assuming that if the monkey uses the application then how random input, values will be entered by the Monkey without any knowledge or understanding of the application.
- The objective of Monkey Testing is to check if an

application or system gets crashed by providing random input values/data.

- Monkey Testing is performed randomly and no test cases are scripted and it is not necessary to be aware of the full functionality of the system.

Mutation Testing

- Mutation Testing is a type of white box testing in which the source code of one of the program is changed and verifies whether the existing test cases can identify these defects in the system.

- The change in the program source code is very minimal so that it does not impact the entire application, only the specific area having the impact and the related test cases should able to identify those errors in the system.

Negative Testing

- Testers having the mindset of “attitude to break” and using Negative Testing they validate that if system or application breaks.
- A Negative Testing technique is performed using incorrect data, invalid data or input.

- It validates that if the system throws an error of invalid input and behaves as expected.

Recovery Testing

- It is a type of testing which validates how well the application or system recovers from crashes or disasters. • Recovery Testing determines if the system is able to continue the operation after a disaster.

- Assume that application is receiving data through the network cable and suddenly that network cable has been unplugged. Sometime later, plug the network cable; then the system should start receiving data from where it lost the connection due to network cable unplugged.

Sanity Testing

- Sanity Testing is done to determine if a new software version is performing well enough to accept it for a major testing effort or not.
- If an application is crashing for the initial use then the system is not stable enough for further testing. Hence a build or an application is assigned to fix it.

Security Testing

- It is a type of testing performed by a special team of testers. A system can be penetrated by any hacking way.
- Security Testing is done to check how the

software or application or website is secure from internal and external threats.

- This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are.
- It also checks how software behaves for any hackers attack and malicious programs and how software is maintained for data security after such a hacker attack.

Smoke Testing

- Whenever a new build is provided by the development team then the Software Testing team validates the build and ensures that no major issue exists.
- The testing team ensures that the build is stable and a detailed level of testing is carried out further. Smoke Testing checks that no show stopper defect exists in the build which will prevent the testing team to test the application in detail.

- If testers find that the major critical functionality is broken down at the initial stage itself then testing team can reject the build and inform accordingly to the development team.
- Smoke Testing is carried out to a detailed level of any Functional or Regression Testing.

Stress Testing

- This testing is done when a system is stressed beyond its specifications in order to check how and when it fails.
- This is performed under heavy load like putting large number beyond storage capacity, complex

database queries, continuous input to the system or database load.

Usability Testing

- Under Usability Testing, User-friendliness check is done. • The application flow is tested to know if a new user can understand the application easily or not, Proper help documented if a user gets stuck at any point.

- Basically, system navigation is checked in this testing.

Volume Testing

- Volume Testing is a type of Non-Functional Testing performed by the Performance Testing team.

- The software or application undergoes a huge amount of data and Volume Testing checks the system behavior and response time of the application when the system came across such a high volume of data.
- This high volume of data may impact the system’s performance and speed of the processing time.

Regression Testing

- Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing.
- It is difficult to cover all the system in Regression Testing, so typically Automation Testing Tools are used for these types of testing.

Smoke vs Sanity vs Regression

Smoke Testing	Sanity Testing	Regression Testing
Performed on initial builds	Performed on stable builds	Performed on stable builds

Tests the stability of a new build	Tests the stability of a new functionality or code changes in the existing build	Tests the functionality of all affected areas after new functionality /code changes in the existing build
Covers end-to-end basic functionalities	Covers certain modules, in which code changes have been made	Covers detailed testing targeting all the affected areas after new functionalities are added
Executed by testers & sometimes also by developers	Executed by testers	Executed by testers, mostly via automation
A part of basic testing	A part of regression testing	Regression Testing is a superset of Smoke and Sanity Testing

Smoke vs Sanity vs

Regression