Testing

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.

Differences between Testing and Debugging

Testing

Testing is the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by its design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

Debugging

Debugging is the process of fixing a bug in the software. It can define as the identifying, analyzing and removing errors. This activity begins after the software fails to execute properly and concludes by solving the problem and successfully testing the software. It is considered to be an extremely complex and tedious task because errors need to be resolved at all stages of debugging.

Types of Software Testing

- **Manual Testing**
- **Automation Testing**

Manual Testing

The process of checking the functionality of an application as per the customer needs without taking any help of automation tools is known as manual testing. While performing the manual testing on any application, we do not need any specific knowledge of any testing tool, rather than have a proper understanding of the product so we can easily prepare the test document.

Manual testing can be further divided into three types of testing, which are as follows:

- **White box testing**
- **Black box testing**
- **Gray box testing**

Automation Testing

Automation testing is a process of converting any manual test cases into the test scripts with the help of automation tools, or any programming language is known as automation testing. With the help of automation testing, we can enhance the speed of our test execution because here, we do not require any human efforts. We need to write a test script and execute those scripts.

Testing Methods:

- **Black Box Testing:**

  Technique of testing without having any knowledge of the interiorworkings of the application.
  Tester will interact with the system's user interface by providinginputs and examining outputs without knowing how and where the inputs are worked upon.

- **White Box Testing:**

  detailed investigation of internal logic and structure of the code.

Black box	White box
w/o knowledge of internal structure of program	knowledge of internal structure of program
Main goal: test behavior of software	Test the internal operation
Focused on external or end user perspective	Focused on code structure conditions path and branches
Less time consumption	More time consuming

Levels of testing:

- **Functional**
- **Non-functional**

Functional:

It is a type of Blackbox testing. Types:

1. Unit Testing:
   - performed by the developers before the setup is handed over to thetesting team to formally execute the test cases

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

1. Integration Testing:
   - The testing of combined parts of an application to determine if theyfunction correctly together

2. System Testing:
    - tests the system as a whole
    - System Testing is the first step in the Software Development Life Cycle
    - performed by a specialized testing team.

3. Regression Testing:
    - Ensure that a change, such as a bug fix did not result in another faultbeing uncovered in the application
4. Acceptance testing:
    - Conducted by the Quality Assurance Team.
    - The main aim of this testing is to determine the working process of the system by satisfying the required specification and it is acceptablefor delivery

      2 stages:

Alpha Testing	Beta testing
Test each user journey and confine they work as intended	Test how software performs in real world
Done by developers or testers at dev. Site	Performed by customers at their site
Address and fix bugs and correct minor issue before software release	User provide feedback

- Sanity Testing and Smoke testing: Subset of Acceptance testing

Smoke Testing	Sanity testing
goal to verify “stability”	goal to verify “rationality”.
done by both developers or testers	done by testers.
verifies the critical functionalities of the system	verifies the new functionality like bug fixes.
subset of acceptance testing	subset of Regression Testing.
verifies the entire system from end to end	verifies only a particular component.

Non-Functional Testing:

- Involves testing the Software from the requirements which are non-functional.
- Includes performance, security, user interface etc.
- Types:

1. Performance Testing:
   - It is mostly used to identify any bottlenecks or performance issuesrather than finding the bugs in software.
   - Divided into different sub types such as ***Load testing and Stresstesting***.
2. Static testing:
   - performed to check the defects in software without actuallyexecuting the code of the software application.
   - Eg: Walkthrough, code review etc.

Fundamentals of testing:

- **RTM (Requirement Traceability Matrix):**

  - Traceability matrix is a table type document that is used in the development of software application to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as Requirement Traceability Matrix (RTM) or Cross Reference Matrix (CRM).

  - It is prepared before the test execution process to make sure that every requirement is covered in the form of a Test case so that we don't miss out any testing. In the RTM document, we map all the requirements and corresponding test cases to ensure that we have written all the test cases for each condition.

- The test engineer will prepare RTM for their respective assign modules, and then it will be sent to the Test Lead. The Test Lead will go repository to check whether the Test Case is there or not and finally Test Lead consolidate and prepare one necessary RTM document.

- This document is designed to make sure that each requirement has a test case, and the test case is written based on business needs, which are given by the client. It will be performed with the help of the test cases if any requirement is missing, which means that the test case is not written for a particular need, and that specific requirement is not tested because it may have some bugs. The traceability is written to make sure that the entire requirement is covered.

- **Test plan:**

A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.

The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.

- **Test case**

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios.

- **Test Scenario**
    - It is a group of test case
    - Test Scenarios are used to ensure that all process flows are testedfrom end to end.
    - A particular area of an application can has as little as one test scenario to a few hundred scenarios depending on the magnitudeand complexity of the application.

TEST CASE	**TEST SCENARIO**
Test cases have a single step	Test scenarios has several steps
Set of actions executed to verify particular features or functionality	Any functionality that can be tested.
It is a detailed document consisting of application requirements, preconditions, Test data, post conditions and expected results	It is a detailed test procedure consisting of test cases which helps to find problems in the system and evaluating the results
Mostly derived from test scenarios	Derived from test artifacts
Helps in exhaustive testing of an application	helps in an agile way of testing the end-to-end functionality.
Low-level actions	High-level actions

- **Test Strategy:**
    - A test strategy is an outline that describes the testing approach of software development cycle.
    - It gives a set of guidelines that explains test design and determineshow testing needs to be done.

- **Test Script:**

Test Scripts are a line-by-line description containing the information about the system transactions that should be performed to validate the application or system under test. Test script should list out each step that should be taken with the expected results.

- This automation script helps software tester to test each level on a wide range of devices systematically. The test script must contain the actual entries to be executed, and the expected results.

STLC-Software Testing Life Cycle

- The Software Testing Life Cycle (STLC) is a sequence of specific actions performedduring the testing process to ensure that the software quality objectives are met.
- The STLC includes both verification and validation.
- It consists of a series of methodological activities to help certify your software product.
- It includes
  - Requirement is confirmed->Test planning->RTM->Test strategy->Test plan-
    >Test scenario->Test case->Test execution->go/ no go(defect fixing bydevelopers)

SOFTWARE DEVELOPMENT METHODOLOGIES:

i. Waterfall model
ii. Agile Model

WATER Fall MODEL	AGILE MODEL
Linear sequential life cycle model	Continuous iteration of development & Testing
Methodology is structured	Methodology is flexible
Sequential design process	Incremental approach
Testing comes after the build phase	Testing occurs concurrently with development
Changing the requirements once the project development starts is not possible	Changing is possible at any time

Risk Analysis:

- The probability of any unwanted incident is defined as Risk
- Risk analysis in software testing is an approach to software testingwhere software risk is analyzed and measured.
- helps businesses identify, quantify and prioritize potential risks that could negatively affect the organization's operations

➢ How to perform Risk Analysis?

There are three steps:

1. Searching the risk
2. Analyzing the impact of each individual risk
3. Measures for the risk identified

Test Management:

- Test Management is a process of managing the testing activities in orderto ensure high quality and high-end testing of the software application.
- Facilitate organizing test case and reusing them.
- The method consists of organizing, controlling, ensuring traceability andvisibility of the testing process in order to deliver the high-quality software application.

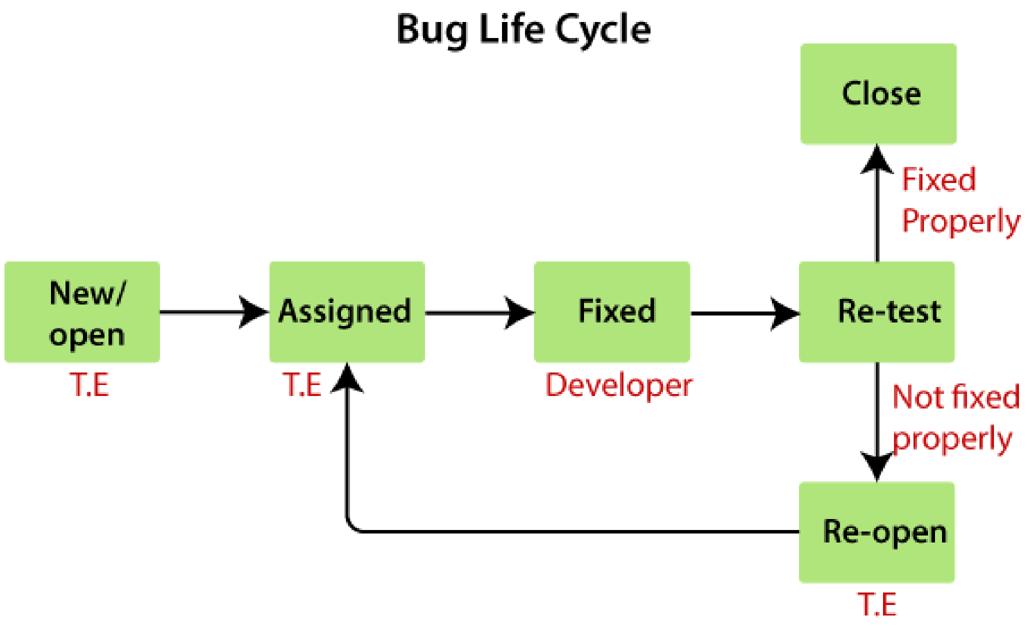
- It ensures that the software testing process runs as expected.
- Helps to objectively measure quality and track progress

Difference between QA & QE:

QA-Quality Analyst	**QE-Quality engineer**
One who ensures/maintains the quality of a product by executing on Code Science's quality procedures.	One who automates quality procedures to minimize manual testing efforts.
Focus on testing for defects	Focus on building in quality
With a background in a non-computer related field that ends up testing software quality issues	More robust background in quality assurance, engineering, and computer science.
Goal-Defect identification	Goal-Defect prevention

Defect Life Cycle:

1. Defect Life Cycle or Bug Life Cycle in software testing is the specific setof states that defect or bug goes through in its entire life.
2. Life cycle starts as soon as a bug is reported by a tester and ends whentest assures that issue is fixed.



Test Pyramid:

3. Helps both QA and developers create high quality software’s
4. Reduces time required for developers to identify the change theyintroduced in code break the code
5. Also be helpful in building a more reliable test suite
6. Faster and efficient
7. Model or structure that should be followed to have a quality productwhich is made faster and more efficient

Verification and Validation (V&V):

- **Verification:** it refers to the set of tasks that ensure that software correctly implements a specific function.
- **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements

Difference between Bug and Defect:

- Bug: An Error found in the development environment before theproduct is shipped to the customer.
- Defect: Difference between expected and actual result in the context oftesting.

  Defect is the deviation of the customer requirement.

Requirement Testing

1. Based on requirement provided by client
2. All the test cases and test scenarios are inclined from requirement.

Types of Requirement Testing:

- Explicit Requirements-It is a first type of requirements, found mostly indocuments. It includes the things you wrote down

- Implicit requirements- It is the second type of requirements. It includes thethings that users are going to expect that were not captured explicitly.

Eg: performance, security, usability, availability etc.

- Latent requirement- It represents the behavior that users do not expect based on their previous experiences but which will make them like the softwaremore.

Eg: when I transfer money from one account to another they show the transaction is successful, which I didn’t expect but I am delight.

User stories:

A user story is an informal, general explanation of a software feature writtenfrom the perspective of the end user. It includes

FRD: Functional Requirement Document

BRD: Business Requirement Document

HLD: High Level Design Document

LLD: Low Level Design Document