

TESTING CONCEPTS

It is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not.

Difference between Testing and Debugging:

Testing:

It involves the identification of bug/error/defect in the software without correcting it.

Done in testing phase

Debugging:

It involves identifying, isolating and fixing the problems/bug.

Debugging is the part of White box or Unit Testing.

Done in developing phase

Testing Types:

- i. Manual Testing
- ii. Automated testing

Manual Testing:

Include testing of the Software manually

Tester identify any un-expected behaviour or bug without using any automated tool.

There are different stages for manual testing like unit testing, Integration testing, System testing and User Acceptance testing.

Testers use test plan, test cases or test scenarios to test the Software to ensure the completeness of testing.

Automation Testing:

Tester writes scripts and uses another software to test the software.

Used to re-run the test scenarios that were performed manually, quickly and repeatedly.

Testing Methods:

i. Black Box Testing:

Technique of testing without having any knowledge of the interior workings of the application.

Tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

ii. White Box Testing:

detailed investigation of internal logic and structure of the code.

<u>Black box</u>	<u>White box</u>
w/o knowledge of internal structure of program	knowledge of internal structure of program
Main goal: test behaviour 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:

- i. Functional
- ii. 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 the testing 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.

2. Integration Testing:

- The testing of combined parts of an application to determine if they function correctly together

3. 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.

4. Regression Testing:

- Ensure that a change, such as a bug fix did not result in another fault being uncovered in the application

5. 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 acceptable for delivery
- 2 stages:

<u>Alpha Testing</u>	<u>Beta testing</u>
Test each user journey and confine they work as intended	Test how software performs in real world
Done by developers or testers at devp.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

<u>Smoke Testing</u>	<u>Sanity testing</u>
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 issues rather than finding the bugs in software.
 - Divided into different sub types such as **Load testing and Stress testing**.
 2. Static testing:
 - performed to check the defects in software without actually executing the code of the software application.
 - Eg: Walkthrough, code review etc.

Fundamentals of testing:

- **RTM (Requirement Traceability Matrix):**
 - It 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).
 - There are many users defined templates 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.
- Furthermore, Bug ID is also including and linked with its associated requirements and test case.
- The main goals for this matrix are: Make sure Software is developed as per the mentioned requirements, helps in finding the root cause of any bug, helps in tracing the developed documents during different phases of SDLC.

- **Test plan:**

- Detailed document which describes the test strategy.
- Defines scope, objective and approach to test the software application
- Defines roles and responsibilities, risk, entry and exit criteria.
- Designed based on requirement.
- A test plan will include the following.
 - Introduction to the Test Plan document
 - Assumptions when 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

- **Test case:**

- Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.
- There are many types of test cases like: functional, negative, error, logical test cases, physical test cases, UI test cases etc.
- The main intent of this activity is to ensure whether the Software Passes or Fails in terms of its functionality and other aspects.

- **Test Scenario**

- It is a group of test case
- Test Scenarios are used 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.

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 determines how testing needs to be done.

- **Test Script**

- Same as a test case but created programmatically.

STLC-Software Testing Life Cycle

- The Software Testing Life Cycle (STLC) is a sequence of specific actions performed during 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 by developers)

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 testing where 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. Analysing 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 order to 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 and visibility 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:

- **Defect Life Cycle** or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life.
- Life cycle starts as soon as a bug is reported by a tester and ends when test assures that issue is fixed.
- Steps involved:

New→ Assign→ open→test→retest→verified→closed

Test Pyramid:

- Helps both QA and developers create high quality softwares
- Reduces time required for developers to identify the change they introduced in code break the code
- Also be helpful in building a more reliable test suite
- Faster and efficient
- Model or structure that should be followed to have a quality product which is made faster and more efficient

Verification and Validation (V&V):

- Verification:
 - Static analysis technique
 - Testing is done without executing the code
 - Eg: inspection, reviews
- Validation:
 - Dynamic analysis technique
 - Done by executing code
 - Eg: functional and non-functional testing

Difference between Bug and Defect:

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

Defect is the deviation of the customer requirement.

Requirement Testing

- Based on requirement provided by client
- All the test cases and test scenarios are inclined from requirement

Types of Requirement Testing:

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

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

Eg: performance, security, usability, availability etc.

3) Latent requirement- It represents the behaviour that users do not expect based on their previous experiences but which will make them like the software more.

Eg: when I transfer money from one account to another they shows 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 written from 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

