TESTING CONCEPTS

It is the process of evaluating a system or its components 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 identifies 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.
- Testerwill 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.

Black box testing	**White box testing**
Testing is done without the knowledge of the internal structure of program or application	Testing is done with the knowledge of internal structure of program
Main goal- test the behaviour of software	Main goal- test the internal operation of software
Focused on external or end user perspective	Focused on code, structure, condition paths & branches
Less time consumption	More time consumption

Levels of Testing:

Levels of testing include the different methodologies that can be used while conducting Software Testing. It includes,

- Functional Testing
- Non- functional Testing

Functional Testing

This is a type of black box testing that is based on the specifications of the software that is to be tested.

➢ **Unit Testing**

- This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases.

- Unit testing is performed by the respective developers on the *individual units* of source code assigned areas.
- 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.
- Limitations of Unit Testing- Testing cannot catch each and every bug in an application, it is impossible to evaluate every execution path in every software application, There is a limit to the number of scenarios and test data that the developer can use to verify the source code.

- **Integration Testing**
  - The testing of *combined parts* of an application to determine if they function correctly together is Integration testing.
  - There are two methods of doing Integration Testing Bottom-up Integration testing and Top Down Integration testing.

- **System Testing**
  - Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.
  - This type of testing is performed by a specialized testing team.

- **Regression Testing**
  - Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change.
  - To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing.
  - The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.

- **Acceptance Testing**
  - Check whether the application meets the intended specifications and satisfies the client's requirements.
  - The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.
  - 2 stages-Alpha & Beeta testing.

ALPHA TESTING	BEETA TESTING
Test each user journey and confirm they work as intended- It always done by developers or tester at the software development site.	Test how the software perform in the real world scenario-It always performed by the customers at their own site
Done before the software release where it will address & fix bugs and correct any minor issue	User provide feedback & result to the product team for in co-operate in the next version

SMOKE TESTING	SANITY TESTING
Goal-To verify stability	Goal-To verify rationality
Done by both developers &Testers	Done by Testers
Verifies the critical functionalities of the system	Verifies the new functionalities 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

Non-functional testing of Software involves testing the Software from the requirements which are non-functional in nature related but important a well such as performance, security, user interface etc.

- **Performance Testing**
  - It is mostly used to identify any performance issues rather than finding the bugs in software.
  - Performance testing is considered as one of the important and mandatory testing type in terms of some aspects such as Speed (i.e. Response Time, data rendering and accessing), Capacity, Stability and Scalability.
  - 2 types- Load testing and Stress testing.
  - **Load testing-** A process of testing the behaviour of the Software by applying maximum load in terms of Software accessing and manipulating large input data. This type of testing identifies the maximum capacity of Software and its behaviour at peak time.
  - **Stress testing-** This testing type includes the testing of Software behaviour under abnormal conditions. Taking away the resources, applying load beyond the actual load limit. The main intent of this testing is to identify the breaking point of software.

- **Static Testing**
  - Static Testing is a type of a Software Testing method which is performed to check the defects in software without actually executing the code of the software application.
  - Static testing includes activities like code Review, Walkthrough, etc.

Testing Documentation:

- **Test Plan**
  - Test plan can be defined as a document for a software project which defines the approach, scope, and intensity on the effort of software testing.

- A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment in which testing will be performed, the limitations of the testing and the schedule of testing activities.
- 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.
- The main intent of this activity is to ensure whether the Software Passes or Fails in terms of its functionality and other aspects.
- There are many types of test cases like: functional, negative, error, logical test cases, physical test cases, UI test cases etc.

➢ **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
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

- **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.

- **Requirement 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 (i.e. from Requirements to Design or Coding) or backward (i.e. from Coding to Requirements).
    - There are many user 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 include 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.

- **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)

PROJECT DEVELOPMENT METHODOLOGIES:

- There are 2 models Water fall Model & Agile Model

- **Water fall model-** It is a linear life cycle model, Water fall model followed in the sequential order and so the project development team only moves to next phase of development or testing if the previous step is completed successfully.
- **Agile methodology-** It is a continuous iteration of development & testing. Agile means in the form of sprints, each sprint spans up to around 2 weeks.

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 occur concurrently with development
Changing the requirements once the project development starts is not possible	Changing is possible at any time

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

Bug and a defect:

- Bug is a mistake in the program -> unit / dev-Integration phases.
- Defect: this is a deviation in the requirement. The program is not behaving as per the requirement.

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.
- 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.

Defect Life cycle:

- A **defect lifecycle**, or **bug lifecycle**, is a specific set of states that a software bug goes through from discovery to fixation.

- The lifecycle may vary from organization to organization depending on factors like company policy, software developmental model (e.g., Agile, Waterfall, etc.), and project timeline.

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
Goal-Defect identification	Goal-Defect prevention

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 Pyramid:

- Helps both QA and developers create high quality Software’s
- 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