Module–2(Manual Testing)

- **What is Exploratory Testing?**

  Exploratory testing is a software testing approach where testers actively explore the application while simultaneously designing and executing tests.

- **What is traceability matrix?**

  Traceability matrix is a tool used in software testing to track the relationship between requirements and test cases, making sure that all the specified functionalities are tested.

- **What is Boundary value testing?**

  Boundary Value Testing is a black-box testing technique where test cases are created based on the boundaries of input ranges. Errors often occur at the edges rather than in the middle of input data, so this testing focuses on values:

- **What is Equivalence partitioning testing?**

  Equivalence Partitioning divides input data into equivalence classes. These are sets of inputs that should be treated the same by the system. You then select one value from each partition to test.

- **What is Integration testing?**

  Integration Testing checks the interaction between components or systems**.** It ensures that combined units/modules work correctly when integrated together.

- **What determines the level of risk?**

  1) Likelihood of failure

  How likely is it that the risk will occur?

- High: Very likely to happen
- Medium: Might happen
- Low: Unlikely to happen

2. Impact (Severity)

If the risk occurs, how bad will the consequences be?

- High: Could cause system failure, major business loss
- Medium: Moderate disruption or loss
- Low: Minor issues, easy to recover

- **What is Alpha testing?**

  Alpha Testing is **internal acceptance testing** done by testers (often within the company) **before the product is released** to customers. It's typically done in a **controlled environment**.

- **What is beta testing?**
  Beta Testing is external user testing performed by a group of real users in a real environment. It helps gather feedback and find issues that weren't caught during internal testing.

- **What is component testing?**

  Component Testing is the testing of individual software components or modules in isolation. It ensures that each part works as intended before combining them.

- **What is functional system testing?**

  Functional System Testing is a black-box testing technique where testers validate the entire system's functionality based on the requirements/specs.

- **What is Non-Functional Testing?**

  Non-Functional Testing checks aspects of the system that aren't related to specific behaviors or functions, but instead focus on how the system performs**.**

- **What is GUI Testing?**

  GUI (Graphical User Interface) Testing checks the look and feel of the application.

- **What is Adhoc testing?**

  Ad-hoc Testing is informal testing without any planning, documentation, or structured test cases.

- **What is load testing?**

Load Testing is a type of performance testing where you test the system under expected (or slightly above expected) user loads.

- **What is stress Testing?**

  Stress Testing is a type of **performance testing** that evaluates how a system behaves **under extreme conditions** — beyond normal or expected load.

- **What is white box testing and list the types of white box testing?**

  White Box Testing is a method where the internal structure, logic, and code of the application is tested.

  Types of white box testing

1. Unit Testing.
2. Integration Testing
3. Code Coverage Testing
   - Statement Coverage
   - Branch Coverage
   - Condition Coverage
   - Path Coverage
4. Path Testing
5. Loop Testing
6. Condition Testing
7. Data Flow Testing
8. Control Flow Testing

- **What is Black Box Testing?**

  Black Box Testing is a software testing method where the tester doesn't know the internal code or structure of the application. The focus is on input -output.

- Black Box Testing Techniques
- Equivalence Partitioning
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing
- Error Guessing

- **Mention what are the categories of defects?**

- **Functional Defects** – App doesn’t work as expected.
- **Performance Defects** – Slowness, lag under load.
- **UI Defects** – Misalignment, bad font, wrong color, etc.
- **Security Defects** – Data leaks, authentication failures.
- **Compatibility Defects** – Doesn’t work on certain browsers/devices.
- **Logical Defects** – Incorrect calculation or logic.

- **What is Big Bang Testing?**

Big Bang Testing is an integration testing approach where all components are integrated and tested at once, rather than gradually.

- **What is the Purpose of Exit Criteria?**

Exit Criteria defines the conditions under which testing can be considered complete.

Examples:

- All planned test cases are executed.
- No critical bugs remain.
- Test coverage reaches 90%.
- All requirements are validated.

- **When Should Regression Testing Be Performed?**

Regression Testing is done after any change in code, such as:

- Bug fixes
- New feature additions
- Performance improvements

Goal: Ensure existing functionality hasn’t broken due to new changes.

- **What Are the 7 Key Principles of Testing?**

1. **Testing shows presence of defects** – It can prove bugs exist, not that they don’t.
2. **Exhaustive testing is impossible** – You can’t test every possible input.
3. **Early testing saves time and money** – Start testing as early as possible.
4. **Defects cluster together** – Most bugs are found in a few modules.
5. **The pesticide paradox** – Repeating the same tests won’t find new bugs.
6. **Testing is context-dependent** – How you test depends on the project.
7. **Absence-of-errors fallacy** – Just because it’s bug-free doesn’t mean it meets user needs.

- **Difference Between QA, QC, and Tester**

Role	**Description**
QA (Quality Assurance)	Process-focused. Prevents defects by improving development processes.
QC (Quality Control)	Product-focused. Identifies defects in the actual product.
Tester	Performs actual testing to find bugs and report them.

- **Difference Between Smoke and Sanity Testing**

Feature	**Smoke Testing**	**Sanity Testing**
Purpose	To check whether the build is stable enough for further testing.	To verify specific functionalities or bug fixes after changes.
Depth	Shallow and wide (basic features tested).	Narrow and deep (focused on particular areas).
Time	Done early and quickly.	Done after receiving a stable build with fixes.

Performed by	Usually QA team or developers.	QA team.
Example	App launches and login screen appears.	A new bug fix in login module works correctly.

- **Difference Between Verification and Validation**

Type	**Verification**	**Validation**
Focus	Are we building the product right?	Are we building the right product?
Methods	Reviews, walkthroughs, inspections	Testing, UAT
Phase	During development	After development

- **Types of Performance Testing**

- **Load Testing** – Tests normal user load.
- **Stress Testing** – Tests extreme load to find limits.
- **Spike Testing** – Tests sudden, sharp increases in load.
- **Endurance Testing** – Tests system performance over a long period (aka soak testing).
- **Scalability Testing** – Checks how the app handles increasing workloads.

- **What is Error, Defect, Bug and failure?**

Term	**Meaning**
Error	Human mistake made by a developer (e.g., logic mistake)
Defect	Flaw in the code detected during development or testing
Bug	Another name for a defect

Term	**Meaning**
Failure	When the system behaves unexpectedly in production due to a defect

- **What is Bug Life Cycle?**

The Bug Life Cycle is the process a bug goes through during its life, from identification to closure.

Common Bug Life Cycle Stages

1. **New** – Bug is found and reported.
2. **Assigned** – Assigned to a developer.
3. **Open** – Developer starts analysing it.
4. **In Progress / Fixed** – Developer fixes the bug.
5. **Pending Retest** – Sent to testers for verification.
6. **Retest** – Tester checks if bug is really fixed.
7. **Verified** – Bug is fixed and confirmed.
8. **Closed** – Tester closes the bug.
9. **Reopened** (if issue still exists after fix).

- **Explain the difference between Functional Testing and Non-Functional Testing**

Functional Testing	**Non-Functional Testing**
Tests what the system does.	Tests how the system performs.
Based on requirements/functions.	Based on performance, usability, reliability, etc.
Examples: Login, registration, form validation.	Examples: Load testing, speed, UI/UX, security testing.
Validates user interactions and features.	Validates system behavior under various conditions.

Functional Testing	Non-Functional Testing

- **What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

Aspect	SDLC (Software Development Life Cycle)	STLC (Software Testing Life Cycle)
Definition	SDLC is a process followed for developing software.	STLC is a process followed for testing software.
Purpose	To design, develop, and deliver high-quality software.	To ensure the software meets quality standards and is bug-free.
Focus Area	Involves planning, designing, coding, testing, deployment, and maintenance.	Involves only the testing phase, including planning, designing test cases, execution, and reporting.
Involvement	Developers, architects, project managers, testers, etc.	Primarily involves testers and QA team.
Starts When	As soon as the project is initiated.	Starts after the requirement gathering phase of SDLC is complete.

- **What is the difference between test scenarios, test cases, and test script?**

Term	Definition	Example
Test Scenario	A high-level idea of what to test.	Check login functionality for valid and invalid inputs."
Test Case	A detailed document that includes input, steps, expected results, and actual results.	**Title**: Login with valid credentials **Steps**: Enter username, password, click login **Expected Result**: User logs in successfully
Test Script	Automated or manual script that performs the actual test steps.	A Selenium script that inputs credentials and clicks login button.

- **Explain what Test Plan is? What is the information that should be covered.**

A Test Plan is a document that outlines the strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product.

Information Covered in a Test Plan:

1. **Test Plan ID**
2. **Introduction**

   Purpose and scope of testing

3. **Objectives**

   What is to be tested and why

4. **Test Items**

   Features or modules to be tested

5. **Features Not to Be Tested**

   Out-of-scope items

6. **Test Strategy**

   Approach (manual/automation), levels of testing, types (functional, regression, etc.)

7. **Test Environment**

   Hardware, software, network setups

8. **Test Deliverables**

   Test cases, test scripts, defect reports, etc.

9. **Schedule**

   Timeline for each phase of testing

10. **Resources**

    Who will do what (testers, tools)

11. **Risks & Mitigation**

    Potential risks and their backup plans

12. **Entry and Exit Criteria**

    When to start and end testing

➢ **Write a scenario of only Whatsapp chat messages.**

- Verify users can successfully send and receive text messages.
- Test the ability to initiate and accept voice and video calls.
- Check if multimedia files (images, videos, documents) can be sent and viewed without errors.

➢ **Write a scenario of only Whatsapp chat messages.**

- Verify users can successfully send and receive text messages.
- Test the ability to initiate and accept voice and video calls.
- Check if multimedia files (images, videos, documents) can be sent and viewed without errors.

➢ **Write a Scenario of Door.**

- Verify if the door is single door or bi-folded door. Check if the door opens inwards or outwards.
- Verify that the dimension of the door as per the specifications.
- Verify that the material used in the door body and its parts is as per the specifications.

➢ **Write a Scenario of ATM.**

- Verify the 'ATM Card Insertion Slot' is as per the specification.
- Verify the ATM machine accepts card and PIN details.
- Verify the error message by inserting a card incorrectly.
- Verify the error message by inserting an invalid card (Expired Card)
- Verify the error message by entering an incorrect PIN.

➢ **When to used Usability Testing?**

- Once you've got an idea, conduct usability testing before putting any design resources to work. Identify specific areas where testing and validation can enhance your concept. After you get the results from your initial test, share them with your team. Then, continue testing users as you build a prototype.

➢ **What is the procedure for GUI Testing?**

- GUI (Graphical User Interface) testing ensures the application's interface functions correctly and provides a satisfactory user experience. It involves verifying the layout, functionality, usability, and compatibility of the interface elements. The procedure typically includes planning, execution, and reporting.

➢ **Write a scenario of Microwave Owen.**

- Verify that the dimensions of the oven are as per the specification provided.
- Verify that the oven's material is optimal for its use as an oven and as per the specification.
- Verify that the oven heats the food at the desired temperature properly.
- Verify that the oven heats food at the desired temperature within a specified time duration.
- Verify the ovens functioning with the maximum attainable temperature.
- Verify the ovens functioning with minimum attainable temperature.
- Verify that the oven's plate rotation speed is optimal and not too high to spill the food kept over it.
- Verify that the oven's door gets closed properly.
- Verify that the oven's door opens smoothly.
- Verify the battery requirement of the microwave oven and check that it function's smoothly at that power.

- **Write a scenario of Coffee vending Machine.**

  - UI scenario – Verify that the dimension of the coffee machine is as per the specification.

  - Verify that outer body, as well as inner part's material, is as per the specification.

  - Verify that the machine's body colour as well brand is correctly visible and as per specification.

  - Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc.

  - Verify that the quantity of hot water, milk, coffee powder per serving is correct.

  - Verify the power/voltage requirements of the machine.

  - Verify the effect of suddenly switching off the machine or cutting the power. The machine should stop in that situation and in power resumption, the remaining coffee should not get come out of the nozzle.

  - Verify that coffee should not leak when not in operation.

  - Verify the amount of coffee served in single-serving is as per specification.

  - Verify that the digital display displays correct information.

- **Write a scenario of chair.**

  - Verify that the chair is stable enough to take an average human load.
  - Check the material used in making the chair-wood, plastic etc.
  - Check if the chair's leg are level to the floor.
  - Check the usability of the chair as an office chair, normal household chair.

- **Write a Scenario of Wrist Watch.**

- Verify the type of watch – analog or digital.
- In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch.
- In the case of a digital watch, check the digital display for hours, minutes, and seconds is correctly displayed.
- Verify the material of the watch and its strap.
- Check if the shape of the dial is as per specification.
- Verify the dimension of the watch is as per the specification.
- Verify the weight of the watch.
- Check if the watch is waterproof or not.
- Verify that the numbers in the dial are clearly visible or not.
- Check if the watch is having a date and day display or not.

- **Write a Scenario of Lift.**

- Verify the dimensions of the lift.

- Verify the type of door of the lift is as per the specification.

- Verify the type of metal used in the lift interior and exterior.

- Verify the capacity of the lift in terms of the total weight.

- Verify the buttons in the lift to close and open the door and numbers as per the number of floors.

- Verify that the lift moves to the particular floor as the button of the floor is clicked.

- Verify that the lift stops when the up/down buttons on a particular floor are pressed.

- Verify if there is an emergency button to contact officials in case of any mishap.

- Verify the performance of the floor – the time taken to go to a floor.

- Verify that in case of power failure, the lift doesn’t free-fall and gets halted on the particular floor.

- **Write a Scenario of What Sapp payment.**

- Check if the payment gateway can connect with the bank systems.
- Ensure that the payment gateway allows users to enter their payment details securely.
- Verify that users can initiate payments using valid credentials.
- Validate if the payment gateway processes the payments accurately.