

Module-2

- What is Exploratory Testing?
 - Exploratory Testing means testing a software by exploring it freely, without following written test steps, to find problems while you learn how it works.
- What is traceability matrix?
 - A traceability matrix is a table that links requirements to test cases to make sure every requirement is tested and nothing is missed.
- What is Boundary value testing?
 - It is a testing method where you check the edge values (just at, below, and above the limits) because bugs often happen at the boundaries of input ranges.
- What is Equivalence partitioning testing?
 - It is a testing method where you divide inputs into groups (partitions) that should behave the same, and then test just one value from each group instead of testing everything.
- What is Integration testing?
 - Integration Testing is a type of software testing where two or more modules are combined and tested together to check if they work correctly as a group.
- What determines the level of risk?
 - The level of risk is determined by two main factors:
 1. Probability (Likelihood) – How likely it is that a problem or failure will happen.
 2. Impact (Severity) – How bad the damage or effect will be if the problem happens.
$$\text{Risk Level} = \text{Likelihood} \times \text{Impact}$$
- What is Alpha testing?
 - Alpha Testing is a type of software testing done by the internal team (developers or testers) before releasing the product to real users.
- What is beta testing?
 - It is testing where real users try the software before it's officially released to find problems in real-life use.
- What is component testing?
 - Component testing are also unit testing where individual modules are tested separately to make sure each module works correctly
- What is functional system testing?
 - Functional System Testing is a type of testing where the entire software system is tested to ensure it performs all its functions correctly according to the requirements.

- It is testing the whole software system to check if it works according to the specified requirements and performs all its intended functions correctly.
- What is Non-Functional Testing?
 - Non-Functional Testing is a type of software testing that checks how well the software works, rather than what it does.
 - It focuses on aspects like performance, usability, reliability, security, and speed, instead of specific features or functions.
- What is GUI Testing?
 - GUI Testing (Graphical User Interface Testing) is a type of software testing that checks whether the user interface of an application works correctly. It verifies elements like buttons, menus, icons, text fields, and layout to ensure they look right and respond properly.
- What is Adhoc testing?
 - Adhoc testing is an informal testing type with an aim to break the system.
- What is load testing?
 - Load Testing is a type of performance testing that checks how a software behaves under heavy use or high traffic.
- What is stress Testing?
 - Stress testing means checking how well something works when it is put under very heavy pressure or difficult conditions. It helps find out when it might fail, how strong it is, and whether it can recover after a problem.
- What is white box testing and list the types of white box testing?
 - White box testing is a software testing technique in which the internal structure, design, and code of the software are known to the tester. It is used to verify the logic, flow, and structure of the program.

Types of white box testing:

- Unit Testing
- Statement Coverage
- Branch Coverage
- Path Coverage
- Condition Coverage
- Loop Testing
- What is black box testing?
 - Black box testing means testing software without looking at the code—only checking whether it gives the correct output for given inputs.

- What are the different black box testing techniques?
 - Black box testing focuses on testing the functionality of an application without knowing its internal code or structure. Test cases are designed based on inputs and expected outputs.

Common Black Box Testing Techniques

1. Equivalence Partitioning
Divide inputs into groups and test one value from each group.
 2. Boundary Value Analysis
Test values at the edges (start and end) of a range.
 3. Decision Table Testing
Test different rules and conditions using a table.
 4. State Transition Testing
Test how the system behaves when it moves from one state to another.
 5. Use Case Testing
Test the system the same way a user uses it.
 6. Error Guessing
Try common mistakes users might make.
 7. Cause-Effect Graphing
Check how inputs (causes) affect outputs (results).
 8. Pairwise Testing
Test combinations of two inputs at a time to save effort.
- Mention what are the categories of defects?
 1. Functional Defects
 - The system does not work as expected.
 - Example: Login button not working.
 2. Performance Defects
 - The system is slow or not responsive.
 - Example: Page takes too long to load.
 3. Usability Defects
 - The application is hard to use or confusing.
 - Example: Poor layout, unclear error messages.
 4. Compatibility Defects
 - The application does not work on all devices, browsers, or OS.
 - Example: Works in Chrome but not in Firefox.
 5. Security Defects
 - The system allows unauthorized access or data leakage.
 - Example: Weak password validation.
 6. UI (User Interface) Defects
 - Visual issues in the application.
 - Example: Misaligned buttons, wrong font size.
 7. Data Defects
 - Incorrect data storage or retrieval.
 - Example: Wrong calculation results.
 8. Documentation Defects
 - Errors in manuals or help documents.
 - Example: Incorrect steps mentioned.

- Mention what bigbang testing is?
 - Big Bang Testing is a software testing technique in which all system modules are integrated simultaneously and the complete system is tested as a whole, without prior individual module testing.
- What is the purpose of exit criteria?

The purpose of exit criteria is to define the conditions that must be met to stop testing and move to the next phase of the software development lifecycle.

In simple terms, exit criteria ensure that testing is completed to an acceptable level of quality before release.

Key purposes:

- Determine when testing is sufficient
 - Ensure test objectives are met
 - Confirm critical defects are fixed
 - Reduce the risk of releasing poor-quality software
 - Provide a clear, measurable decision point for release or next phase
- When should "Regression Testing" be performed?
 - Regression Testing should be performed whenever changes are made to the software to ensure that existing functionality still works correctly.
 - What is 7 key principles? Explain in detail?
 1. Testing shows presence of defects:-

Testing can show that defects are present, but cannot prove that there are no defects.
 2. Exhaustive testing is impossible:-

Testing everything is not possible, so instead if doing exhaustive testing we can use risks priorities to focus testing efforts.
 3. Early testing:-

Testing activities should start as early as possible in development life cycle.
 4. Defect clustering:-

Defects are not evenly speard in a system they are “clustered”.
 5. The pesticide paradox:-

If same tests are repeated over and over again, then no longer find new defects. To overcome this test cases need to review and revised.
 6. Testing is context dependent:-

Basically testing is context dependent because testing is done differently in different contexts.
 7. Absence of errors fallacy:-

Even after defects have been resolved it may still be unusable or does not fulfil the users need and expectations.

- Difference between QA v/s QC v/s Tester

Testing	QC	QA
Subset of QA	Quality control	Quantity assurance
Focus on test execution	Focus on product	Focus on process
Actual testing	Finds defect	Prevents defect
Done during development or after development	Done after development	Done before development

- Difference between Smoke and Sanity?

Smoke	sanity
To check whether build is stable or not	To check specific function is working or not
When we get a new build	When any new feature is added or bug fix
Basic and critical functionality	Bug fix or related feature
Explore:- app install, login, dashboard, logout, app crash or not	Scenario:- app crashes with invalid data login feature (login, dashboard, logout)

- Difference between verification and Validation

Verification	Validation
Checks whether the product is built according to requirements	Checks whether the product meets user needs
“Are we building the product right?”	“Are we building the right product?”
Static testing (review of document)	Dynamic testing (live testing of actual software)
Performed during development	Performed after development
Includes reviews, inspections, walkthroughs	Includes testing activities

- Explain types of Performance testing.
 - The main types of performance testing are:
 1. Load Testing
 - Tests system behavior under expected user load
 - Ensures the application performs well under normal conditions
 2. Stress Testing
 - Tests system behavior under extreme load
 - Identifies breaking point and system recovery
 3. Spike Testing
 - Tests system response to sudden increase or decrease in load
 - Checks stability during traffic spikes
 4. Endurance (Soak) Testing
 - Tests system performance over a long period
 - Detects memory leaks and performance degradation
 5. Volume Testing
 - Tests system with large amounts of data
 - Ensures database and storage handling efficiency
 6. Scalability Testing
 - Tests system's ability to scale up or down
 - Verifies performance with increasing users or resources
- What is Error, Defect, Bug and failure?
 - Error: - A mistake in coding is called error
 - Defect: - Error found by tester is called defect
 - Bug: - defect accepted by development team then it is called bug
 - failure: - build does not meet the requirements then it is failure

- Difference between Priority and Severity

Priority	severity
priority indicates the impact of a defect on the system	Severity indicates the urgency to fix the defect
Focus on business importance	Focus on technical impact
Example Minor UI issue needed urgently	Example Application crash

- What is Bug Life Cycle?
 - Bug Life Cycle is the sequence of states a defect goes through from the time it is identified until it is fixed and closed.
- Typical Bug Life Cycle Stages:
1. New – Bug is identified and logged.
 2. Assigned – Bug is assigned to a developer.
 3. Open – Developer starts working on the bug.
 4. Fixed – Bug is fixed by the developer.
 5. Retest – Tester retests the fix.
 6. Verified – Tester confirms the bug is fixed.
 7. Closed – Bug is closed.

Other Possible States:

- Reopen – Bug persists after fix.
- Deferred – Fix postponed to a later release.
- Rejected – Bug is invalid or not reproducible.
- Duplicate – Bug already reported.

- Explain the difference between Functional testing and Nonfunctional testing

Functional testing	Non-functional testing
<ul style="list-style-type: none">• Functional testing is executed first	<ul style="list-style-type: none">• Non-functional testing should be performed after functional testing
<ul style="list-style-type: none">• Describes what the product does	<ul style="list-style-type: none">• Describes how good the product works
<ul style="list-style-type: none">• Easy to do manual testing	<ul style="list-style-type: none">• Tough to do manual testing
<ul style="list-style-type: none">• Types:- Unit testing Smoke testing Sanity testing Black box testing White box testing	<ul style="list-style-type: none">• Types:- Performance testing Load testing Volume testing Stress testing Security testing

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

STLC (Software Testing Life Cycle)	SDLC (Software Development Life Cycle)
STLC is to ensure software quality by identifying defects	SDLC is to develop and deliver software as per requirements
STLC is limited to testing activities only	SDLC is covers end-to-end software development
Quality-assured and defect-free software	Fully developed and functional software

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

Test Scenario: What to test

→ A big idea of testing

Example: “Check login feature”

Test Case: How to test

→ Step-by-step instructions

Example: “Enter username, enter password, click login, check result”

Test Script: Code to test automatically

→ Program that runs the test

Example: A script that logs in by itself

- Explain what Test Plan is?
 - A test plan is a document that explains how testing will be done for a project.
- What is the information that should be covered.
 - Test objectives – What needs to be tested
 - Scope of testing – What is in scope and out of scope
 - Test strategy – How testing will be done
 - Test environment – Hardware and software setup
 - Test schedule – Start and end dates of testing
 - Resources – Testers and their roles
 - Test tools – Tools used for testing
 - Entry and Exit criteria – When to start and stop testing
 - Risks and assumptions – Possible issues and conditions
- What is priority?
 - Priority is the level of importance or urgency assigned to a defect, indicating how soon it should be fixed.
- What is severity?
 - Severity is the measure of how much impact a defect has on the system or application functionality.
- Bug categories are...
 - Functional – something doesn't work
 - UI – looks wrong
 - Usability – hard to use
 - Performance – slow
 - Security – not safe
 - Compatibility – works on one device/browser but not another
 - Data – wrong data shown or saved
- Advantage of Bugzilla .
 - Easy to report bugs
 - Easy to track bug status (new, fixed, closed)
 - Many people can work together
 - Keeps all bug details in one place
 - Good for large projects

- Difference between priority and severity

priority	severity
priority indicates the impact of a defect on the system	Severity indicates the urgency to fix the defect
Focus on business importance	Focus on technical impact
Example Minor UI issue needed urgently	Example Application crash

- What are the different Methodologies in Agile Development Model?

1. Scrum

- Most widely used Agile framework
- Work is divided into Sprints (2–4 weeks)
- Key roles: Product Owner, Scrum Master, Development Team

2. Kanban

- Visual workflow using a Kanban board
- Focuses on continuous delivery
- Limits work in progress (WIP)

- Explain the difference between Authorization and Authentication in Web testing.

Authentication	Authorization
Authentication is the process of verifying who the user is. It checks user credentials such as username, password, OTP, or biometrics.	Authorization is the process of verifying what the authenticated user is allowed to do. It controls access to features, pages, or data based on user roles (admin, user, guest).
Authentication verifies who the user is (login).	Authorization verifies what the user can access (permissions).

- What are the common problems faced in Web testing?

- Common problems faced in web testing include Browser issues, Slow loading, UI issues, Broken links, Login issues, Data issues, Security issues, Network issues, Device issues

- When to use usability testing?

- Before releasing a website or app
- While designing it, to see if it makes sense
- After making changes, to avoid new problems
- When users are confused or make mistakes
- Example:

If people can't find a button or don't know what to do next, usability testing helps you see that.

- What is the procedure for gui testing?

Procedure for GUI testing

1. Understand the design
Look at the design or mockups and know how the screen should look.
2. Check screen elements
Make sure buttons, icons, menus, text boxes, and images are visible and placed correctly.
3. Test actions
Click buttons, links, and menus to see if they work properly.
4. Check text and labels
See if text is readable, spelled correctly, and makes sense.
5. Test different screens
Open the app on different screen sizes or browsers (if needed).
6. Check error messages
Enter wrong input and see if helpful messages appear.
7. Report problems
Note anything that looks wrong or doesn't work.