Module – 2

1)What is Exploratory?

- Exploratory testing is a software testing method where testers actively explore the application under test to identify defects and assess user experience without relying on predefined test cases. It's a spontaneous and unstructured approach, allowing testers to navigate the application like real users would, discovering issues and learning about the software as they go.

2)What is traceability matrix?

- A traceability matrix in testing, also known as a test matrix, is a document that links and maps test cases to their respective requirements.

3) What is Boundary value?

- Boundary Value Analysis (BVA) is a black-box testing technique that focuses on testing values at the extreme boundaries of input domains, also known as edge cases.

4) What is Equivalence partitioning testing?

- Equivalence partitioning is a black-box testing technique that divides the input data of a software system into groups, called equivalence classes, that share similar characteristics and are expected to produce the same output.

5) What is Integration testing?

- Integration testing in software development is a method where individual components or modules of a software application are combined and tested together to ensure they interact and communicate effectively.

6) What determines level of risk?

- The “level of risk” is a measure that combines the potential severity of a security breach with the probability of its occurrence. It is a concept that allows organisations to prioritise their security efforts, focusing on the most significant threats that could impact their operations.

7) What is Alpha testing?

- Alpha testing is the initial phase of validating whether a new product will perform as expected. Alpha tests are carried out early in the development

process by internal staff and are followed by beta tests, in which a sampling of the intended audience actually tries the product.

8) What is beta testing?

- Beta testing is a phase in software development where a select group of real users, who are not part of the development team, test a pre-release version of a product before it's officially released to the public.

9) What is component testing?

- Component testing, also known as module testing, is a software testing technique where individual components or modules of a software system are tested in isolation before they are integrated with other modules or the overall system.

10) What is functional system testing?

- Functional system testing verifies that a software system or application's features and functions work as expected and meet the specified requirements.

11) What is Non-Functional Testing?

- Non-functional testing evaluates aspects of a software application beyond its core functions, focusing on how well it performs and operates under various conditions, such as speed, scalability, and user experience.

12) What is GUI Testing?

- GUI (Graphical User Interface) testing, also known as UI (User Interface) testing, is a type of software testing that focuses on the visual elements of a software application to ensure they function correctly and meet user expectations.

13) What is Ad-hoc testing?

- Ad-hoc testing is a software testing method where testing is performed without any pre-defined or documented test cases, test plans, or procedures. It's an informal, unstructured approach that relies on the tester's intuition, experience, and creativity to identify defects in the software.

14) What is load testing?

- Load testing is a type of performance testing that evaluates how a system behaves under expected or peak user loads. It simulates multiple users

accessing a website, app, or server simultaneously to identify performance bottlenecks, response delays, or stability issues.

15) What is stress Testing?

- Stress testing in software testing evaluates a system's or application's stability, reliability, and performance under extreme conditions, pushing it beyond its normal capacity to identify breaking points and assess recovery from failures.

16) What is white box testing and list the types of white box testing?

- White box testing, also known as glass box, clear box, or structural testing, is a software testing approach that examines the internal structures and logic of a program to ensure its correctness.

1. Unit Testing
2. Integration Testing
3. Branch Coverage
4. Statement Coverage
5. Condition Coverage
6. Basis Path Testing

17) What is black box testing? What are the different black box testing techniques?

- Black box testing is a software testing method that focuses on verifying the functionality of a system by examining its inputs and outputs without knowing its internal code or implementation.

1. Equivalence Partitioning
2. Boundary Value Analysis
3. Decision Table Testing
4. State Transition Testing
5. Error Guessing
6. Syntax Testing
7. Functional Testing
8. Non-Functional Testing

18) Mention what are the categories of defects?

- Defects are commonly categorized into three severity levels: critical, major, and minor. These categories help prioritize and manage defects during quality control and software development.

19) Mention what big-bang testing is?

- Big bang testing, also known as Big Bang Integration Testing, is a software testing approach where all components of a system are combined and tested as a whole at once.

20) What is the purpose of exit criteria?

- Exit criteria define the conditions that must be met before a task, process, or testing phase can be considered complete. They serve as a benchmark to ensure that all necessary requirements have been met, the software meets quality standards, and the testing process is thorough before moving to the next stage.

21) When should "Regression Testing" be performed?

- Regression testing should be performed whenever changes are made to a software application, including after bug fixes, feature additions, or code modifications. This ensures that the new changes haven't inadvertently affected existing functionality.

22) What is 7 key principles? Explain in detail.

1. Testing shows the Presence of Defects

- The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn't talk about the absence of defects. Software testing can ensure that defects are present but it cannot prove that software is defect-free. Even multiple tests can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.

2. Exhaustive testing is impossible

- Consider a simple real-world example in which you have a screen that takes two numbers as input and prints their sum. It would take infinite time to validate this screen for all possible numbers.
  If a simple screen like this is virtually impossible to test exhaustively, what about a full-fledged application?
  Trying to test exhaustively will burn time and money without affecting the overall quality.

3. Early testing saves time and money

- The proverb: a stitch in time saves nine, applies to most things in life, and software testing is no exception. As shown in a study conducted by IBM, what costs a dollar to fix in the design phase can cost fifteen in the testing phase and a whopping hundred if detected in a production system.

4. Defects cluster together

- This principle is an example of the 80:20 rule (also called the Pareto principle) – 80 percent of defects are due to 20 percent of code. While most believe this is some divine mandate, it is based on the observation that 80 percent of users use 20 percent of the software. It is this 20 percent of the software that will contribute most towards the defects.

5. Beware of the pesticide paradox

- The use of a pesticide makes pests immune to its use. Similarly, subjecting a module to the same test cases can make the module immune to the test cases. This principle is compelling while testing complex algorithms.

6**. Testing is context-dependent**

- There is no one-strategy-fits-all in software testing. Yes, testing a web application and ATM will be different but what this principle says is more profound.

7. Beware of the absence-of-errors fallacy

- It is a common belief that a low defect rate implies the product is a success. This idea is the absence-of-errors delusion.
- Zero defects do not mean the software solves end-user problems successfully. Linux always had very few bugs, while Microsoft Windows was (is?) notorious for its bugs.

23) Difference between QA v/s QC v/s Tester.

- **Focus:**
    - **QA:** Process-oriented, focusing on preventing defects from occurring.
    - **QC:** Product-oriented, focusing on identifying and correcting defects in the final product.
    - **Testing:** A subset of QC, focusing on identifying defects through execution and inspection.
- **Timing:**

- **QA:** Implemented throughout the software development life cycle (SDLC).
- **QC:** Primarily conducted after the product is developed, during the testing phase.
- **Testing:** A component of QC, typically conducted after the development phase.

➢ **Responsibility:**

- **QA:** Involves all team members.
- **QC:** Primarily the responsibility of the test team.
- **Testing:** Conducted by the test team and other members of the development team.

24) Difference between Smoke and Sanity?

Smoke Testing	**Sanity Testing**
Smoke testing is done to assure that the acute functionalities of program is working fine.	Sanity testing is done to check the bugs have been fixed after the build.
Smoke testing is also called subset of acceptance testing.	Sanity testing is also called subset of regression testing.
Smoke testing is documented.	Sanity testing isn't documented.
Smoke testing is performed by either developers or testers.	Sanity testing is normally performed by testers.
Smoke testing may be stable or unstable.	Sanity testing is stable.
Smoke testing is scripted.	Sanity testing is usually not scripted.
Smoke testing is done to measure the stability of the system/product by performing testing.	Sanity testing is done to measure the rationality of the system/product by performing testing.
Smoke testing is used to test all over function of the system/product.	Sanity testing is used in the case of only modified or defect functions of system/products.
Smoke testing can be performed either manually or by using automation tools.	Sanity testing is commonly executed manually, not by using any automation approach.
Smoke testing is performed when new product is built.	Sanity testing is conducted after the completion of regression testing.
It includes all the system's essential basic functionality.	It includes only those modules where change in code is made.
Smoke Testing firstly performs on the initial build. Smoke testing is done first.	Sanity Testing is done on stable builds or for the introduced new features in the software.

25) Explain types of Performance testing.

- Common types include:
- Load testing: To measure performance under normal traffic.
- Stress testing: To identify the breaking point.
- Volume testing: To assess large amounts of data.
- Endurance testing: To check for memory leaks or degradation over time.

26) What is Error, Defect, Bug and failure?

Error:

- A human mistake, often in coding, that introduces a problem into the software.

Defect:

- A flaw or deviation from the expected behaviour of the software.

Bug:

- An informal term for a defect, often used to describe an issue causing unexpected software behaviour.

Failure:

- The manifestation of a defect that leads to the software not functioning as intended.

27) What is Bug Life Cycle?

- The bug life cycle, also known as the defect life cycle, is the process a bug follows from its initial discovery to its resolution in software testing. It's a systematic approach to managing bugs, ensuring they are identified, tracked, prioritized, fixed, and validated effectively.

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

- SDLC (Software Development Life Cycle) encompasses the entire process of building and deploying software, while STLC (Software Testing Life Cycle) is a subset of SDLC focused solely on the testing phase. In essence, STLC is a part of SDLC, ensuring the software meets requirements and is defect-free.

- **SDLC (Software Development Life Cycle):**
  - The overall process of building software, from initial planning to maintenance.
  - Includes phases like requirements gathering, design, implementation (coding), testing, deployment, and maintenance.
  - Focuses on creating high-quality software that meets user needs.
  - Can be structured using various models like Waterfall, Agile, etc.

- **STLC (Software Testing Life Cycle):**
    - A specific phase within the SDLC dedicated to testing the software.
    - Focuses on verifying that the software functions as intended, is free of bugs, and meets quality standards.
    - Includes phases like test planning, design, execution, and closure.
    - Helps ensure the software is reliable and meets the specified requirements.

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

- Test scenarios, test cases, and test scripts are all related to software testing, but they serve different purposes. A test scenario is a high-level description of a functionality or user interaction to be tested. Test cases are specific, detailed instructions on how to test a particular scenario, while test scripts automate those instructions using code.

1. Test Scenario:

- **Definition:**

A test scenario is a broad, high-level description of a feature or functionality to be tested.

- **Purpose:**

It helps determine the scope of testing and ensures that all critical functionalities are covered.

- **Example:**

A test scenario for an e-commerce website could be "A user should be able to add items to their shopping cart and proceed to checkout.

- **Key Characteristics:**
    - Focuses on what to test, not how to test.
    - Describes the conditions under which the test will be performed.
    - Identifies critical functionalities that need to be tested.
    - Serves as a foundation for creating more detailed test cases.

2. Test Case:

- **Definition:**

A test case is a detailed set of instructions that specifies how to test a particular test scenario.

- **Purpose:**

To verify that a specific feature or functionality is working as expected.

- **Example:**

A test case for the scenario "A user should be able to add items to their shopping cart" would specify the steps:

  - Go to the product page.
  - Click "Add to Cart".
  - Navigate to the shopping cart page.
  - Verify that the item is added to the cart.
- **Key Characteristics:**
  - Provides a step-by-step guide for testers.
  - Includes expected results.
  - Can be used for both manual and automated testing.
  - Helps identify errors or defects in the system.

3. Test Script:

- **Definition:**

A test script is a program or set of instructions written in a programming language to automate the execution of test cases.

- **Purpose:**

To automate the testing process and save time and resources.

- **Example:**

A test script for the test case above would use a programming language to automate the steps of navigating to a product page, clicking "Add to Cart", navigating to the shopping cart, and verifying the item is added.

- **Key Characteristics:**
  - Automates the execution of test cases.
  - Can be used to run multiple test cases simultaneously.
  - Requires programming knowledge.

30) Explain what Test Plan is? What is the information that should be covered.

- A Test Plan is a document that outlines the objectives, scope, approach, resources, and schedule for testing a software product or system. It serves as a roadmap for the testing process, guiding testers and stakeholders in ensuring the product meets requirements and expectations.

1. Scope and Objectives:

- **What to test:**
  Clearly defines the specific components or features to be tested, including what's "in-scope" and "out-of-scope".

- **Test Objectives:**
  Outlines the specific goals and desired outcomes of the testing process, such as identifying defects, verifying functionality, or evaluating performance.

2. Test Approach and Methodology:

- **Test Strategy:**
  Describes the overall approach to testing, including the types of testing (e.g., unit, integration, system) and the tools and techniques to be used.
- **Test Methods:**
  Specifies the specific testing techniques, such as manual testing, automated testing, or performance testing, to be employed.

31) What is priority?

- In software testing, priority determines the order in which defects should be fixed, focusing on the urgency and impact of a bug on the business. It helps prioritize defect resolution based on business needs and customer expectations, influencing the timeline for bug fixes.

32) What is severity?

- A bug's severity reflects its impact on the end-user, while its priority indicates its effect on the business. For example, a spelling mistake on Amazon's home page would be low severity (since it doesn't affect the end-user) but high priority (since it reflects poorly on Amazon as a company).

33) Bug categories are…

- Typical bug types are syntax errors, which happen when the code deviates from the rules of the programming language; runtime problems, which happen when the program is running; and logic errors, which happen when the program doesn't execute as expected owing to incorrect reasoning.

34) Advantage of Bugzilla.

- Improved Product Quality
- Advanced Features
- Enhanced Communication
- Scalability and Security
- Cost-Effective
- Customization and Extension

35) What are the different Methodologies in Agile Development Model?

- 1.Scrum
  2.Kanba
  3.Extreme Programming
  4.Feature-Driven Development
  5. Dynamic Systems Development Method

36) Write a Scenario of Pen.

- Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen.
- Verify that the user is able to write clearly over different types of papers.
- Check the weight of the pen.
- Verify if the pen is with a cap or without a cap.
- Verify the colour of the ink on the pen.

37) 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.

38) 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.

39) 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.

40) When to used Usablity 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.

41) 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.

42) 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.

43) 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.

44) 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.

45) 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.

46) 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.

47) Write a Scenario of Whatsapp 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.

48) Difference between verification and Validation.

Feature	**Verification**	**Validation**
Timing	Before validation, during development	After verification, before release
Method	Static analysis, reviews, inspections	Dynamic testing (e.g., black box, white box)
Focus	Meeting specified requirements	Meeting user needs and expectations
Purpose	Preventing errors	Detecting errors
Internal/External	Internal, within the development team	External, involves users and testing teams

49) Difference between Priority and Severity.

severity	**Priority**
Defect Severity is specified as the degree of impact that a defect has on the operation of the product.	Defect Priority has specified the order in which the developer should fix a defect.
Severity means the seriousness of the defect in the product functionality.	Priority means how soon the bug should be fixed.

The test engineer determines the severity level of the defect.	Priority of defects is decided in discussion with the manager/client.
It is driven by functionality.	It is driven by business value.
Severity status is established on the technical aspect of the product.	Priority status is established on customer requirements.

50) Explain the difference between Functional testing and Non-Functional testing.

Aspect	**Functional Testing**	**Non-Functional Testing**
Objective	Validates what the system does	Evaluates how the system performs
Focus	Business logic and feature correctness	Performance, usability, security, and reliability
Test types	Unit, integration, system, acceptance tests	Load, stress, security, and performance tests
Execution method	Often manual or automated	Mostly automated with specialized tools