**ASSIGNMENT-2**

**MODULE-2(MANUAL TESTING)**

**Q-1 What is exploratory testing?**

**ANS..** It is a simultaneous learning, test design, and test execution process. In this testing test planning, analysis, design and test execution, are all done together and instantly. The tester will create or write down a test idea to give direction, and explore the system while testing to further create critical, practical and useful tests for the successful testing of an application.

Example…. A home care service provider website with following components: login , services , cart , payment , order history , technician allotment. A general idea to start exploratory testing will be to login or book a service.

**Q-2 What is traceability value testing?**

**ANS..** Traceability value testing is a software testing technique that focuses on verifying the extent to which the requirements of a software system are traceable throughout the development and testing process. It involves ensuring that each requirement can be traced back to its source, such as user stories, business requirements, or system specifications, and that these requirements are effectively translated into test cases and ultimately into the delivered product. The goal is to ensure transparency, accountability, and alignment between the software's intended functionality and its actual implementation.

**Q-3 What is boundary value testing?**

**ANS..** The process of testing between extreme ends or boundaries between partitions of the input values.

Boundary value analysis is a method which refines equivalence partitioning. So these extreme ends like start-end, lower-upper, maximum-minimum, just inside-just outside values are called “boundary testing”.

The basic idea in boundary value testing is to select input variable values at their: minimum , just above the minimum , a normal value , just below the maximum , maximum

Example: in our earlier example instead of checking, one value for each partition you will check the values at the partitions like 0, 1, 10, 11 and so on. As you may observe, you test values at both valid and invalid boundaries. Boundary value analysis is also called range checking.

**Q-4** **What is equivalence partitioning testing?**

**ANS..**  Aim is to treat groups of inputs as equivalent and to select one representative input to test them all. Ep can be used for all levels of testing. Equivalence partitioning is the process of defining the optimum number of tests by: reviewing documents such as the functional design specification and detailed design specification, and identifying each input condition within a function.

Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.

If we want to test the following IF statement:” if value is between I and 100(inclusive) (e.g value>=1 and value <=100) Then….”

We could put a range of number as shown in the below figure.

Devides the input data of a software unit into partitions of equivalent data from which test cases can be derived.

In principle, test cases are designed to cover each partition at least once. An advantage of this approach is reduction in the time required for testing software due to lesser number of test cases.

**Q-5** **What is integration testing?**

**ANS..** Testing performed to expos defects in the interface and in the interaction between integrated components or system. It is level of the software testing process where individual units are combined and tested as a group.

The purpose is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in integration testing.

It tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between system.

Done by a specific integration tester or test team.

There are 2 levels of integration testing .

* Component integration testing.
* System integration testing.

**Q-6 What determines the level of risk?**

**ANS..** Assessing the likelihood of the event occurring and the impact of that event.

The level of risk is determined by a variety of factors depending on the context. In finance, for example, the level of risk associated with an investment is influenced by factors such as market volatility, economic conditions, and the specific characteristics of the investment itself. In the context of health and safety, risk can be determined by factors such as the likelihood of exposure to hazards, the potential severity of harm, and the effectiveness of risk mitigation measures. In general, the level of risk is influenced by the interplay of various factors that can impact the likelihood and potential impact of adverse events.

Types of risk: project risk

Product risk

Project risk: is senior team member leaving the project abruptly.

Product risk: would be flight reservation system not installing in test environment.

**Q-7 What is Alpha testing?**

**ANS..** It is always performed by the developers at the software development site.

Sometimes it is also performed by Independent Testing Team.

Alpha Testing is not open to the market and public

It is conducted for the software application and project.

It is always performed in Virtual Environment.

It is always performed within the organization.

It is the form of Acceptance Testing.

Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.

It comes under the category of both White Box Testing and Black Box Testing.

**Q-8 What is beta testing?**

**ANS..** It is always performed by the developers at the software development site.

Sometimes it is also performed by Independent Testing Team.

Alpha Testing is not open to the market and public

It is conducted for the software application and project.

It is always performed in Virtual Environment.

It is always performed within the organization.

It is the form of Acceptance Testing.

Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.

It comes under the category of both White Box Testing and Black Box Testing.

Beta Testing is always performed at the time when software product and project are marketed.

It is always performed at the user's premises in the absence of the development team.

It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area.

Beta testing can be considered "pre-release" testing.

Pilot Testing is testing to product on real world as well as collect data on the use of product in the classroom

**Q-9 What is component testing?**

**ANS..** a level of software testing that focuses on verifying the individual components or units of a system. Component testing, also known as program or module testing, is done after unit testing. In this type of testing those test objects can be tested independently as a component without integrating with other components e.g. modules, classes, objects, and programs. This testing is done by the development team.

Component testing is like unit testing with the difference that the developer uses real data instead of dummy data for testing of the written code. Suppose there is a software application which consists of five components modules. The testing of each module is done independently by the developer as part of the development cycle before it is ready for integration testing. By doing component testing, bugs can be found at a very early stage in the cycle and helps save time.

Debugging tools or unit test structure tools are used for this type of testing since this is done by programmers on the code written by them and with the support of integrated development environment. Defects are fixed as soon as possible when they are found without formally recording incidents.

Component testing plays an important role in finding the issue. Before we proceed with the integration testing it’s always advised to do the component testing in order to ensure that each module of an application is working correctly and as per requirement.

**Q-10 What is functional system testing?**

**ANS..** Functional testing is a type of software testing where the basic functionalities of an application are tested against a predetermined set of specifications. A requirement that specifies a function that a system or system component must perform

There is two type of test approach: 1. requirement based functional testing. 2. Process based testing.

Functional testing aims to figure out whether given functionality works as specified. System testing aims to figure out whether the whole system fulfills the requirements given to it.

Testing based on an analysis of the specification of the functionality of a component or system.

Requirement specification , use case this testing mainly involves black box testing and it is not concerned about the source code of the application.

Functional testing is executed first manual testing or automation tools can be used for functional testing.

**Q-11 What is non-functional testing?**

**ANS..** Non-Functional Testing is a type of testing used to evaluate a software application's performance, usability, dependability, and other non-functional characteristics. It is intended to test a system’s readiness according to non-functional criteria that functional testing never considers.

Non Functional testing is essential for confirming the software’s reliability and functionality. The Software Requirements Specification (SRS) serves as the basis for this software testing method, which enables quality assurance teams to check if the system complies with user requirements. Increasing the product’s usability, effectiveness, maintainability, and portability is the goal of non-functional testing. It aids in lowering the manufacturing risk associated with the product’s non-functional components.

Non-functional testing is also very important as functional testing because it plays a crucial role in customer satisfaction.

For example, non-functional testing would be to test how many people can work simultaneously on any software.

**Q-12 What is GUI testing?**

**ANS..** Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screen with the controls like menus, buttons, icons, and all types of bars - tool bar, menu bar, dialog boxes and windows etc.

WHAT DO YOU CHECK IN GUI TESTING?

Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.

Check you can execute the intended functionality of the application using the GUI

Check Error Messages are displayed correctly

Check for Clear demarcation of different sections on screen

Check Font used in application is readable Check the alignment of the text is proper

Check the Color of the font and warning messages is aesthetically pleasing

Check that the images have good clarity

Check that the images are properly aligned

Check the positioning of GUI elements for different screen resolution.

* Approach of GUI Testing

MANUAL BASED TESTING:

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

RECORD AND REPLAY:

GUI testing can be done using automation tools. This is done in 2 parts. During Record, test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test. Example of such tools - QTP.

MODEL BASED TESTING:

A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements

* GUI Testing Examples

Web Based Testing & Desktop Based Testing:

The scrollbar should be enabled only when necessary.

Font size, style, and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS.

The description text box should be multi-lined.

Enough space should be provided between field labels, columns, rows, error messages, etc.

* Mobile Based Testing:

If mobile is in every orientation mode so display image, video properly.

Every app will display in responsive type.

Alignment should be apply properly of every field.

* Game Based Testing:

Game infra design will showing properly

Game points or score will display proper with its background color

Game sound manage with its background effect

Can be also conducted in advance of designing page layouts or navigation menus.

**Q-13 What is Adhoc testing?**

**ANS..** Adhoc Testing(Error Guessing)

* NO DOCUMENTATION
* NO TEST DESIGN
* NO TEST CASE

Adhoc testing is an informal testing type with an aim to break the system. It does not follow any test design techniques to create test cases. This testing is primarily performed if the knowledge of testers in the system under test is very high. Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing technique called Error Guessing. Error guessing can be done by the people having enough experience on the Activate Windows system to "guess" the most likely source of errors.

* Types of Adhoc testing:

1. Buddy Testing:

Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing. Team. Buddy

testing helps the testers develop better test cases and development team can also make design changes early. This testing usually happens after unit testing completion.

1. Pair testing:

Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the tests and another person can take notes on the findings. Roles of the persons can be a tester and scriber during testing

1. Monkey Testing:

Randomly test the product or application without test cases with a goal to break the system.

**Q-14 What is load testing?**

**ANS..** Load testing - It’s a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails. Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

* This testing usually identifies -

The maximum operating capacity of an application

Determine whether current infrastructure is sufficient to run the application

Sustainability of application with respect to peak user load

Number of concurrent users that an application can support, and scalability to allow more users to access it.

It is a type of non-functional testing. Load testing is commonly used for the Client/Server, Web based applications - both Intranet and Internet.

* Goals of Load Testing:

- Loading testing identifies the following problems before moving the application to market or Production:

- Response time for each transaction

- Performance of System components under various loads

- Performance of Database components under different loads

- Software design issues

- Network delay between the client and the server

- Server configuration issues like Web server, application server, database server etc.

- Hardware limitation issues like CPU maximization, memory limitations, network bottleneck, etc.

- Load testing will determine whether system needs to be fine-tuned or modification of hardware and software is required to improve performance.

**Q-15 What is stress testing?**

**ANS..** Stress testing - System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database

Queries , continuous input to system or database load. Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions. Stress Testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing. Under Stress Testing, Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks. It also checks whether system demonstrates effective error management. under extreme conditions. The application under testing will be stressed when 5GB data is copied from the website and pasted in notepad. Notepad is under stress and gives 'Not Responded' error message.

Examples of stress conditions include:

Excessive volume in terms of either users or data; examples might include a denial of service (DoS) attack or a situation where a widely viewed news item prompts a large number of users to visit a Web site during a three-minute period.

Resource reduction such as a disk drive failure. Application components fail to respond.

* Types of stress testing:

Application stress testing:

Transactional stress testing:

Systemic stress testing:

Exploratory stress testing:

**Q-16 What is white box testing and list the types of white box testing?**

**ANS..** White-box testing, also known as clear box testing, glass box testing, or structural testing, is a method of software testing that examines the internal structure and workings of an application. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. White-box testing can be applied at the unit, integration, and system levels of the software testing process.

* types of white-box testing:

Unit Testing: This is the most basic form of white-box testing and involves testing individual functions or methods of the software under test.

Integration Testing: This type of testing verifies that the units or components of the software work together as expected.

System Testing: This type of testing ensures that the entire system functions correctly and meets the specified requirements.

Path Testing: This is a white-box testing technique that focuses on testing all possible paths through a piece of code, which can be represented using a control flow graph.

Branch Testing: This technique involves testing all possible branches in the code, where a branch is a decision point in the code that could result in different outcomes.

Statement Testing: This technique involves testing each statement in the code at least once, ensuring that every statement is executed and behaves as expected.

Loop Testing: This technique involves testing loops in the code to ensure that they execute correctly, including the boundary conditions and exit conditions.

Data Flow Testing: This technique focuses on testing how data flows through the software, including how it is stored, retrieved, and manipulated.

Control Flow Testing: This technique focuses on testing the control flow of the software, including how control is transferred between different parts of the code.

Mutation Testing: This technique involves making small changes to the code (mutations) and then running the test suite to see if the tests can detect the changes.

**Q-17 What is black box testing? What are the different black box testing techniques?**

**ANS..** Black box testing is a software testing method where the internal structure or workings of the system being tested are not known to the tester. In this method, the tester only interacts with the system's inputs and outputs, without knowing how the system processes them. The objective is to verify the system's functionality, rather than its internal logic.

* black box testing techniques:

Equivalence Partitioning: This technique divides the input data into different equivalence classes, where the behavior of the system is expected to be similar or identical for any value within each class.

Boundary Value Analysis: It focuses on testing the system at the boundaries of input values. The idea is to select test cases at the edges of the equivalence classes.

State Transition Testing: Used for systems where the behavior depends on the state of the system. The tester defines different states of the system and the transitions between them, then tests how the system responds to those transitions.

Decision Table Testing: A technique where test cases are derived from a table that describes the conditions that should be met for different outputs.

Use Case Testing: Tests are derived from use cases, which describe the interactions between the system and its users.

Boundary Value Testing: Similar to Boundary Value Analysis, this technique focuses on testing values at the edges or boundaries of input parameters.

Error Guessing: Testers make educated guesses about potential errors in the system and design test cases to verify if those errors occur.

Ad-hoc Testing: Also known as random or monkey testing, this method involves testing the system without any formal test case or plan.

**Q-18 Mention what are the categories of defects?**

**ANS..** Functional defects: These defects occur when the software doesn't perform according to its specifications. For example, a login function that doesn't allow users to log in, or a report that doesn't display the correct data.

Performance defects: These defects relate to how efficiently the software performs. Examples include slow loading times, high memory usage, or excessive CPU usage.

Usability defects: These defects impact the user experience. They could include elements like confusing or misleading user interface elements, poor navigation, or unclear instructions.

Compatibility defects: These defects occur when the software doesn't work as intended on certain platforms, browsers, or devices. For example, a website that looks broken in a particular browser or a mobile app that crashes on certain devices.

Security defects: These defects involve vulnerabilities that could allow unauthorized access to the system or data, or that could lead to data

loss or corruption. This could include issues like SQL injection, cross-site scripting (XSS), or authentication bypass.

Data defects: These defects involve problems with the data itself, such as incorrect or missing data, or data that is not properly formatted or validated.

Integration defects: These defects occur when different components or systems don't work together as expected. For example, an online shopping cart that doesn't communicate properly with the inventory system, causing incorrect stock levels to be displayed.

Documentation defects: These defects involve problems with the documentation or user manuals, such as missing or incorrect information, or outdated screenshots.

Regression defects: These defects occur when a change or update to the software causes previously working functionality to break. This could be due to changes in the code, configuration, or environment.

Syntax defects: These defects involve issues with the programming language syntax, such as missing or incorrect punctuation, or incorrect variable names.

Standards defects: These defects occur when the software doesn't adhere to industry or internal standards, such as coding standards, security standards, or accessibility standards.

Design defects: These defects involve problems with the overall design of the software, such as poor architecture, inadequate scalability, or lack of modularity.

Requirements defects: These defects occur when the software doesn't meet the requirements specified by the stakeholders, such as missing features, incorrect functionality, or incorrect performance expectations.

**Q-19 Mention what bigband testing is?**

**ANS..** In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

Big Bang testing has the advantage that everything is finished before integration testing starts

The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration

Here all component are integrated together at once, and then tested.

Advantages:

Convenient for small systems

Disadvantages:

Fault Localization is difficult.

Since the integration testing can commence only after "all" the modules are designed, testing team will have less time for execution in the testing phase. Since all modules are tested at once, high risk critical modules are not isolated and tested on priority.

Peripheral modules which deal with user interfaces are also not isolated and tested on priority.

**Q-20 What is the purpose of exit criteria?**

**ANS..** Exit criteria are established to provide a set of conditions that must be met before a phase or a project can be considered complete or ready for transition to the next phase.

Quality Assurance: Ensuring that the product or service meets the quality standards set for it.

Risk Management: Identifying and addressing potential risks and issues before moving forward.

Progress Monitoring: Tracking the progress of the project and ensuring that it stays on schedule and within budget.

Client Satisfaction: Ensuring that the client's needs and expectations are met or exceeded.

Regulatory Compliance: Ensuring that all applicable regulations and standards are adhered to.

By defining clear exit criteria, project managers can ensure that all stakeholders are aligned on what constitutes a successful outcome and what needs to be accomplished before moving to the next phase or closing the project.

**Q-21 When should “regression Testing ” be performed?**

**ANS..** Regression testing" should typically be performed whenever a change is made to the software.

Bug Fixes: After fixing a bug, regression testing should be performed to ensure that the bug fix did not introduce new issues elsewhere in the codebase.

New Features or Enhancements: When new features or enhancements are added, regression testing should be performed to verify that these changes didn't break existing functionality.

System Upgrades: After upgrading system components or dependencies, regression testing should be performed to ensure that the upgrade didn't introduce new problems.

Changes in Environment: If the software is moved to a new environment (e.g., a new server or platform), regression testing should be performed to ensure that it works correctly in the new environment.

Scheduled Basis: Regression testing can also be performed on a regular schedule to ensure that the software is working as expected over time, even if no specific changes have been made.

The frequency of regression testing depends on the project, but it's generally recommended to perform it frequently, especially when the software is actively being developed or changed.

**Q-22 What is 7 key principles? Explain in detail?**

**ANS..**  The 7 key principles of software testing emphasize the importance of early and continuous testing, as well as the need to focus testing efforts on areas that are most likely to have defects. They also highlight the fact that testing is context-dependent and that the absence of defects does not necessarily mean that a software system is of high quality.

**Testing shows the presence of defects**: Software testing is a process that aims to identify defects or bugs in a software system. It does not aim to prove the absence of defects, but rather to show that defects are present.

**Exhaustive testing is impossible:** It is impossible to test all possible inputs and scenarios of a software system. Therefore, testing must be focused on areas that are most likely to have defects or impact the system the most.

**Early testing:** Testing should begin as early as possible in the software development life cycle. This allows defects to be identified and fixed early, which reduces the cost and time of fixing defects later in the development process.

**Defect clustering:** Defects tend to cluster in certain parts of a software system. By focusing testing efforts on these areas, it is possible to identify and fix a large number of defects.

**Pesticide paradox:** Repeated testing of a software system with the same test cases will eventually lead to the same defects being found over and over again. To prevent this, test cases should be regularly reviewed and updated to include new scenarios.

**Testing is context-dependent:** The effectiveness of testing depends on the context in which it is performed. This includes factors such as the requirements of the software system, the technology used to develop it, and the skills and experience of the testing team.

**Absence-of-errors fallacy:** The absence of defects in a software system does not guarantee that it is of high quality. The quality of a software system is determined by a combination of factors, including its functionality, performance, usability, and reliability.

**Q-23 Difference between OA v/s QC v/s Tester**

**ANS..** **Quality assurance:**

1: Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

2: Focuses on processes and procedures rather than conducting actual testing on the system.

3: Process oriented activities.

4: Preventive activities.

5: It is a subset of Software Test Life Cycle (STLC).

**Quality Control:**

1: Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.

2: Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.

3: Product oriented activities.

4: It is a corrective process.

5: QC can be considered as the subset of Quality Assurance

**Testing:**

1: Activities which ensure the identification of bugs/error/defects in the Software.

2: Focuses on actual testing.

3: Product oriented activities.

4: It is a preventive process.

5: Testing is the subset of Quality Control.

**Q-24 Difference between smoke and sanity?**

**ANS..** **Smoke Testing:**

Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.

The objective of this testing is to verify the "stability" of the system in order to proceed with more rigorous testing.

This testing is performed by the developers or testers.

Smoke testing is usually documented or scripted.

Smoke testing is a subset of Regression testing.

Smoke testing exercises the entire system from end to end.

Smoke testing is like General Health Check Up.

**Sanity Testing:**

Sanity Testing is done to check the new functionality/bugs have been fixed.

The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing.

Sanity testing is usually performed by testers.

Sanity testing is usually not documented and is unscripted.

Sanity testing is a subset of Acceptance testing.

Sanity testing exercises only the particular component of the entire system.

Sanity Testing is like specialized health check up.

**Q-25 Difference between verification and validation**

**ANS..** **Verification:**

Definition: The process of evaluating work- products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.

Objective : To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.

Question: Are we building the product right?

Evaluation items: Plan, Requirement Specs, Design Specs, Code, Test Cases

Activities: Reviews , Walkthroughs , Inspections.

**Validation:**

Definition: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements

Objective: To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

Question: Are we building the right product?

Evaluation items: The actual product/software.

Activities: Testing

**Q-26 Explain types of performance testing.**

**ANS..** Performance testing is a critical aspect of software development and ensures that applications meet the desired performance standards. There are several types of performance testing, each focusing on different aspects of performance.

**Load Testing:** This type of testing is used to evaluate how a system behaves under normal and peak load conditions. It helps to identify the maximum number of concurrent users the system can handle before performance starts to degrade.

**Stress Testing:**  This type of testing is used to evaluate how a system behaves when pushed beyond its normal or peak load conditions. It helps to identify the breaking point of the system and how it recovers from failure.

**Endurance Testing:** Also known as Soak Testing, it evaluates how a system behaves under a sustained load over an extended period. It helps to identify memory leaks, database connection leaks, and other issues that may arise due to continuous usage.

**Spike Testing:**  This type of testing evaluates how a system behaves when the load is suddenly increased or decreased. It helps to identify how the system responds to sudden surges or drops in traffic.

**Volume Testing:** This type of testing evaluates how a system behaves when processing a large volume of data. It helps to identify performance issues related to data handling and storage.

**Scalability Testing:**  This type of testing evaluates how a system can handle an increasing amount of work by adding more resources like CPU, memory, or storage. It helps to identify how the system scales as the workload increases.

**Q-27 What is Error, Defect, Bug and failure?**

**ANS..** **Error:** An error is a human action that produces an incorrect result. It can be a mistake made by the programmer during coding or logic errors in the program.

**Defect:** A defect is an imperfection or shortcoming in the software that causes it to produce incorrect or unexpected results. It is the result of an error. A defect can exist in any phase of the software development lifecycle and can be found by a developer or a tester.

**Bug:** A bug is a specific type of defect that causes the software to behave in an unintended manner. The term "bug" is often used interchangeably with "defect," but technically a bug is a type of defect. Bugs are usually found during testing or after the software has been deployed.

**Failure:** A failure occurs when the software does not meet its specified requirements and does not perform its intended function. A failure can result from a defect or a bug, but not all defects or bugs lead to failures.

**Q-28 Difference between priority and severity**

**ANS..**  **Priority:** This measures the urgency and order in which the issue needs to be addressed. It is determined by factors such as the potential damage the bug could cause, the number of users affected, or the importance of the feature affected. Priority levels often include

classifications like High, Medium, Low, or Immediate, High, Normal, Low. For example, a bug that causes a security vulnerability might be

considered Immediate priority, while a bug that affects a less critical feature might be considered Low priority.

**Severity:** This measures the impact of a bug on the system's functionality. It is mainly concerned with how much the bug disrupts the proper functioning of the software. Severity levels often include classifications like Critical, High, Medium, Low, or Minor. For example, a bug that causes the application to crash would be considered Critical, while a bug that only causes a minor inconvenience might be rated as Low.

**Q-29 What is bug life cycle?**

**ANS..** The bug life cycle is a sequence of steps that a bug goes through, starting from its discovery through to its resolution. It typically consists of the following stages:

**New:** When a bug is discovered, it is initially reported or logged, and its status is marked as "New."

**Assigned:** The bug is then assigned to a developer or a team who will be responsible for investigating and fixing the bug. At this stage, the status changes to "Assigned" or "Open."

**In Progress:** The developer starts working on the bug, trying to reproduce it and identify its root cause. The status of the bug is changed to "In Progress."

**Fixed:** Once the developer identifies and fixes the bug, they mark it as "Fixed."

**Verified:** The testing team or the quality assurance team then verifies the fix by retesting the functionality. If the fix works, the bug status is updated to "Verified."

**Closed:** Finally, if the bug is verified and the fix is successful, the bug is marked as "Closed," indicating that the issue is resolved.

**Q-30 Explain the difference between Functional testing and Non-Functional testing**

**ANS..** **Functional Testing:**

Functional testing in performed using the Non-Functional functional specification provided by the client and Verifies the system against the functional requirements.

Functional testing executed first

Manual testing or automation tools can be used for functional testing

Business requirements are the inputs to functional testing.

Functional testing describes what the product does

Easy to do manual testing

Types of Functional testing are:

* Unit Testing
* Smoke Testing
* Sanity Testing
* Integration Testing
* White box testing
* Black Box testing
* User Acceptance testing
* Regression Testing

**Non-Functional Testing:**

Testing checks the Performance,reliability, scalability and other non-functional aspectsof the software system.

Non functional testing should be performed after functional testing.

Using tools will be effective for this testing

Performance parameters like speed, scalability are inputs to non-functional testing.

Nonfunctional testing describes how good the product works.

Tough to do manual testing

Types of Nonfunctional testing are

* Performance Testing
* Load Testing
* Volume Testing
* Stress Testing
* Security Testing
* Installation Testing
* Penetration Testing
* Compatibility Testing,
* Migration Testing

**Q-31 To create HLR & Test case of 1) (Instagram , Face book) only first page. 2) Facebook login page:** [**https://www.facebook.com/**](https://www.facebook.com/)

**Q-32 What Is the difference between the STLC and SDLC?**

**ANS..** **SDLC (Software Development Life Cycle):**

SDLC encompasses the entire process of software development, from planning and analysis to maintenance and deployment.

It includes various phases such as requirements gathering, design, implementation, testing, deployment, and maintenance.

It is primarily concerned with the overall development process, ensuring that software is built efficiently and meets the desired requirements.

**STLC (Software Testing Life Cycle):**

STLC focuses specifically on the testing aspect within the software development process.

It involves planning, designing, executing, and evaluating the effectiveness of tests to ensure the quality and reliability of the software.

STLC runs parallel to SDLC, with testing activities integrated into each phase of the development life cycle.

In essence, while SDLC covers the entire software development process, STLC zooms in on the testing component within that process.

**Q-33 What is the difference between test scenarios, test case, and test script?**

**ANS..** **Test Scenario:**

A test scenario is a high-level description of what will be tested. It outlines the objectives and conditions under which testing will be conducted.

Test scenarios are often written in natural language and focus on specific functionalities or features of the software.

They are used to ensure that testing efforts are focused on important aspects of the software and to guide the creation of more detailed test cases.

**Test Case:**

A test case is a detailed specification of a particular test scenario. It describes the steps to be taken, the inputs to be used, and the expected results for a specific test. Test cases are more detailed than test scenarios and are often written in a structured format, including preconditions, steps, expected outcomes, and postconditions. They serve as executable documentation for testing and are used by testers

to verify that the software behaves as expected under different conditions.

**Test Script:**

A test script is a set of instructions or code that automates the execution of one or more test cases. Test scripts are typically written in scripting languages such as Python, JavaScript, or Ruby.

Test scripts automate the manual execution of test cases, saving time and effort during the testing process.

They can be used for both functional and non-functional testing, including regression testing, performance testing, and load testing.

**Q-34 Explain what test plan is? What is the information that should be covered.**

**ANS..** A test plan is a document that outlines the objectives, scope, approach, resources, and schedule for a testing project. It defines the strategy for how testing will be conducted, including what will be tested, who will perform the tests, the test environment, and the criteria for success. The purpose of a test plan is to ensure that testing is systematic, thorough, and aligned with the project's goals and requirements.

A comprehensive test plan typically includes the following information:

**Introduction:** Provides an overview of the document and its purpose.

**Objectives:** Clearly states the goals and objectives of the testing effort.

**Scope:** Defines the boundaries of the testing, including what will and will not be tested.

**Approach:** Describes the overall strategy for testing, including methodologies, techniques, and tools to be used.

**Test Items:** Lists the specific components or features to be tested.

**Features to be tested:** Describes the features or functionalities to be tested, often with reference to requirements or specifications.

**Features not to be tested:** Identifies any features or functionalities that will not be tested and the reasons for exclusion.

**Test Environment:** Specifies the hardware, software, and other resources needed for testing.

**Test Deliverables:** Lists the documents, reports, and other deliverables expected from the testing process.

**Test Schedule:** Outlines the timeline for testing activities, including milestones and deadlines.

**Test Team:** Identifies the roles and responsibilities of individuals involved in testing.

**Risks and Assumptions:** Identifies potential risks to the testing process and any assumptions made during planning.

**Dependencies:** Lists any dependencies or constraints that may impact testing activities.

**Approvals:** Specifies the individuals or stakeholders who must review and approve the test plan.

**Appendices:** Includes any additional information or reference materials relevant to the testing effort.

**Q-35 What is priority?**

**ANS..** **Defect Priority:**

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements. For example: If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

**Priority can be of following types:**

**Low:** The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.

**Medium:** The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

**High:** The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.

**Critical:** Extremely urgent, resolve immediately

**Q-36 What is severity?**

**ANS..** **Defect Severity:**

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

**For example:** If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

**Types:**

**Critical:** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.

**Defect Severity (Cont)**

**Types:**

**Major (High):** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.

**Moderate (Medium):** The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or

inconsistent results then the severity will be stated as moderate.

**Minor (Low):** The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

**Cosmetic:** The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

**Q-37 Bug categories are…**

**ANS..** **Functional Bugs:** These are issues that affect the intended functionality of the software. For example, a button that doesn't work as expected or a feature that produces incorrect results.

**Performance Bugs:** These bugs affect the performance of the software, such as slow loading times, high memory usage, or inefficient algorithms.

**Compatibility Bugs:** These bugs occur when the software doesn't work correctly on different platforms, browsers, or devices. For instance, a website that displays improperly on certain browsers or a mobile app that crashes on specific devices.

**Usability Bugs:** Usability bugs impact the user experience of the software. This can include confusing user interfaces, unclear instructions, or difficulty navigating through the application.

**Security Bugs:** These are vulnerabilities in the software that could be exploited by attackers. Examples include SQL injection, cross-site scripting (XSS), or insecure authentication mechanisms.

**Localization Bugs:** Localization bugs arise when the software doesn't support or display content correctly in different languages or regions.

**Documentation Bugs:** These bugs involve errors or inconsistencies in the software documentation, such as outdated instructions, missing information, or inaccuracies in the user manual or help files.

**Installation/Configuration Bugs:** Issues related to installing or configuring the software, including errors during installation, conflicts with other software, or incorrect setup instructions.

**Q-38 Advantage of Bugzila..**

**ANS..** **Centralized Issue Tracking:** Bugzilla provides a centralized platform for tracking bugs, issues, and feature requests, making it easier for teams to manage and prioritize tasks.

**Customizable Workflow:** Users can customize Bugzilla to match their development workflow, including defining statuses, resolutions, and fields to suit their project needs.

**Collaboration:** Bugzilla facilitates collaboration among team members by allowing them to comment, assign, and monitor the status of bugs, promoting transparency and accountability.

**Search and Reporting:** Its robust search and reporting capabilities enable users to quickly find relevant information and generate customized reports, aiding in project analysis and decision-making.

**Integration:** Bugzilla can be integrated with other development tools and systems, such as version control systems and continuous integration tools, enhancing overall workflow efficiency.

**Open Source:** Being open source, Bugzilla is freely available and can be customized further by developers to meet specific requirements without licensing costs.

**Community Support:** With a large user base and active community, Bugzilla benefits from ongoing development, support, and a wealth of resources, including documentation and user forums.

**Q-39 Difference between priority and severity**

**ANS..** Priority and severity are terms commonly used in software development, particularly in the context of bug tracking and issue management. Here's the difference between the two:

**Priority:** Priority refers to the level of urgency or importance assigned to an issue or bug. It helps in determining the order in which issues should be addressed. Priority levels are typically defined based on factors such as impact on users, business impact, deadlines, and overall project goals. Examples of priority levels include high, medium, and low.

**Severity:** Severity, on the other hand, refers to the extent of impact that a bug or issue has on the system or software's functionality. It indicates how serious the problem is in terms of its consequences. Severity levels are usually based on the extent of damage or loss caused by the issue. Examples of severity levels include critical, major, minor, and trivial.

**Q-40 What are the difference methodologies in agile development model?**

**ANS.. Conducting meetings:** The team conducts frequent meetings for 10-15 minutes daily, and they think that conducting frequent meetings will be Agile. However, only the following meetings will not be Agile.

**Requirements changing anytime:** Requirements can be changed at any time, then it is not Agile. For example, a client wants to add some new features and want the changes to be updated at the same time, then this will not be Agile.

**Unstructured development:** Suppose you are not following any plan and you are working on Adhoc basis then it is not Agile wherein Adhoc testing, testers randomly test the application without following any documents and test design techniques.

**No documentation:** If the company does not make the documentation, then it is not Agile.

**What is agil?** The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.Agile is a philosophy, i.e., a set of values and principles to make a decision for developing software.

Agile is based on the iterative-incremental model. In an incremental model, we create the system in increments, where each increment is developed and tested individually.

**What are values?**

**Individuals and interactions, Over processes and tools:** Suppose the team finds any issue in software then they search for another process or tool to resolve the issue. But, in Agile, it is preferable to interact with client, manager or team regarding issue and make sure that the issue gets resolved.

**Working software, Over comprehensive documentation:** Documentation is needed, but working software is much needed. Agile is not saying that documentation is not needed, but working software is much needed. For example, you have 20-page documents, but you do not have a single prototype of the software. In such a case, the client will not be happy because, in the end, the client needs a document.

**Customer collaboration, Over contract negotiation:** Contract negotiation is important as they make the budget of software, but For example, if you stuck with the requirements or process, then do not go for a contract which we have negotiated. You need to interact with the customer, gather their requirements.

**Responding to change, over following a plan**

In the waterfall model, everything is planned, i.e., at what time, each phase will be completed. Sometimes you need to implement the new requirements in the middle of the software, so you need to be versatile to make changes in the software.

**Agile Principles: Customer satisfaction through early and continuous software delivery :** Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.

**Accommodate changing requirements throughout the development process –** The ability to avoid delays when a requirement or feature request changes.

**Frequent delivery of working software –** Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.

**Collaboration between the business stakeholders and developers throughout the project –** Better decisions are made when the business and technical team are aligned.

**Support, trust, and motivate the people involved –** Motivated teams are more likely to deliver their best work than unhappy teams.

**Enable face-to-face interactions –** Communication is more successful When development teams are co-located.

**Working software is the primary measure of progress –** Delivering functional software to the customer is the ultimate factor that measures progress.

**Agile processes to support a consistent development pace –** Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.

**Attention to technical detail and design enhances agility –** The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.

**Simplicity –** Develop just enough to get the job done for right now.

**Self-organizing teams encourage great architectures, requirements, and designs –** Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.

**Regular reflections on how to become more effective –** Self improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

**Methodologies:**

**Scrum:** A framework for managing and controlling iterative and incremental projects, emphasizing communication, collaboration, and delivering working software in short iterations called sprints.

**Kanban:** A visual management method that aims to optimize the flow of work by visualizing the workflow, limiting work in progress, and continuously improving the process.

**Extreme Programming (XP):** Emphasizes principles such as rapid feedback, continuous testing, continuous integration, and simple design to deliver high-quality software incrementally.

**Lean Software Development:** Adapted from Lean manufacturing principles, it focuses on eliminating waste, amplifying learning, empowering teams, and delivering as fast as possible.

**Crystal:** A family of methodologies that prioritize communication and teamwork, with different variants suited for various project sizes and contexts.

**Dynamic Systems Development Method (DSDM):** Emphasizes active user involvement, frequent delivery of products, and adapting to changing requirement.

**Q-41 Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?**

**ANS..** Authorization and authentication are both essential aspects of web testing, but they serve different purposes:

**Authentication:** Authentication is the process of verifying the identity of a user or entity attempting to access a system or resource. In web testing, this typically involves validating that a user is who they claim to be by verifying their credentials, such as username and password. Common authentication methods include username/password, biometric authentication, single sign-on (SSO), and multi-factor authentication (MFA).

**Authorization:** Authorization, on the other hand, determines what actions or resources a user is allowed to access after they have been

authenticated. It is the process of granting or denying access rights and privileges to authenticated users based on their identity and the permissions associated with their account or role. In web testing, authorization testing involves verifying that users can only access the resources and perform the actions they are authorized to, based on their role or privileges.

Common problems faced in web testing include:

**Security vulnerabilities:** Web applications are often targets for malicious attacks, so security testing is crucial to identify and address vulnerabilities such as injection flaws, cross-site scripting (XSS), cross-site request forgery (CSRF), and insecure direct object references.

**Compatibility issues:** Web applications need to work across various browsers, devices, and operating systems. Compatibility testing ensures that the application functions correctly and displays properly across different environments.

**Performance issues:** Poor performance, slow loading times, and high server response times can negatively impact user experience and lead to decreased engagement and customer dissatisfaction. Performance testing helps identify and address bottlenecks, scalability issues, and other performance-related problems.

**Usability concerns:** Usability testing evaluates the user interface, navigation, and overall user experience of the web application to ensure it is intuitive, user-friendly, and meets the needs of its target audience.

**Functionality bugs:** Functional testing verifies that the web application functions correctly according to its specifications and requirements.

Testers need to identify and report any bugs, glitches, or inconsistencies in the application's functionality.

**Q-42 To create HLR & TestCase of WebBased (WhatsApp web , Instagram) 1. WhatsApp**

**Web :** [**https://web.whatsapp.com/**](https://web.whatsapp.com/) **2. Instagram Web :** [**http://www.instagram.com/accounts/login/?hl=en**](http://www.instagram.com/accounts/login/?hl=en)

**ANS..**

**Q-43 To create HLR and Test Case on this link.** [**http://artoftesting.com/**](http://artoftesting.com/)

**ANS..**

**Q-44 Write a scenario of only whatsapp chat messages**

**ANS..** **Individual Chats:**

• Verify that the Chat window contains the entire chat list.

• Verify that the Chat window displays the contact numbers whose numbers not saved in mobile.

• Verify that Chat window displayed with all contacts with DP or without DP.

• Verify that the Chat window is displayed on the group chat list.

• Verify that the Chat window displays the last updated chatting time.

• Verify that the Chat window displays the name of all contacts on the chat window.

• Verify that clicking on one Chat contact then a new window should open with history.

• Verify that the user can see all delivered and received messages.

• Verify that the user can see the read or send time of messages.

• Verify that the user can send and receive text messages in the individual chat box.

• Verify that user can send and receive documents in the individual chat box.

• Verify that user can send and receive photos in individual chat box.

• Verify that user can send and receive videos in individual chat box.

• Verify that user can send and receive audio in individual chat box.

• Verify that user can send and receive emotions icons in individual chat box.

• Verify that user can send and receive Contacts in individual chat box.

• Verify that user can send and receive Location in individual chat box.

• Verify that user can send and receive GIF in individual chat box.

• Verify that user can send and receive Stickers in individual chat box.

• Verify that user can delete text, video, audio, locations, documents in individual chat box.

• Verify that user can send recorded voice mail in individual chat box.

• Verify that user can delete the entire chat history in individual chat box.

• Verify that user is able to see contact details in individual chat box.

• Verify that user is able to share images, links, and documents from media in individual chat box.

• Verify that user is able to search specific chat history using search option in individual chat box.

• Verify that user is able to video call in individual chat box.

• Verify that user is able to voice call in individual chat box.

• Verify that user is able to mute the individuals in individual chat box.

• Verify that user is able to share images, links, and documents from media in individual chat box.

• Verify that user is able to search specific chat history using search option in individual chat box.

• Verify that user is able to change the wallpaper.

• Verify that users have options like Report, Block, Clear Chat, Export Chat and Add Shortcut.

**Q-45 Write a scenario of pen.**

**ANS..** • Verify that user is able to hold pen comfortably.

• Verify that the pen is writing smoothly or not.

• Verify that pen is not making any type of sound while writing.

• Verify that the ink flow of the pen is smooth or not. There should not be any break in ink flow.

•Verify that the quality of the material of pen is good or bad.

• Verify that the company name of pen is visible clearly or not.

• Verify that the design on pen or the branding of the pen is not removed easily.

• Verify that the line drawn by the pen is according to the mentioned specifications.(0.5,0.7)

• Verify that the ink colour of the pen should be consistent from start to end.

• Verify that the pen is able to write on different types of paper. (Smooth, Glossy, rough, thick, thin.)

• Verify that the ink of pen is waterproof or not.

• Verify that the ink of pen is dried quickly on paper or not.

• Verify that other refills fit the pen or not.

• Verify that the pen shouldn't have any sharp edges and corners.

• Verify that the body or pen and the ink is made up of non toxic material or not.

• Verify that pen is working normally or not after putting the pen in water for sometime.

• Verify that pen is working normally or not after dropping the pen from some height.

• Verify that ink spelled or not after dropping the pen from some height.

•Verify that pen is able to write when left open for some time with out pen cap.

•Verify that nib of pen is able to withstand the pressure while writing.

• Verify that the pen is working under high pressure or not.

• Verify that pen is working under high temperature or not.

• Verify that pen is working under low pressure or not.

• Verify that p t pen is working where gravity is zero.

**Q-46 Write a scenario of pen stand.**

**ANS..**

• Verify that the pen stand is sturdy and stable.

• Verify any visible defects or damages in the material.

• Verify that maximum number of pens or other stationery items that can be accommodated in the pen stand.

• Verify if the material is resistant to scratches, stains, or other forms of damage that may occur during regular use

• Verify Evaluate the weight distribution of the pen stand by placing pens of varying weights and sizes in different compartments.

• Verify Ensure that the pen stand remains stable and balanced, even when loaded with items unevenly.

• Verify the compatibility of the pen stand with different desk surfaces, including wood, glass, or plastic.

• Verify Ensure that the pen stand does not scratch or damage the desk surface during normal use.

• Verify Assess the ease of access to pens or other items stored in the pen stand.

Determine how easy it is to clean the pen stand, considering factors such as material type and design intricacies.

• Verify the pen stand is resistant to water or if it requires special cleaning instructions to maintain its appearance.

• Verify if the base of the pen stand provides adequate support to prevent tipping or wobbing.

• Verify the pen stand arrives without damages or defects that may have occurred during transit.

• Consider user reviews and ratings to understand overall satisfaction with the pen stand.

**Q-47 Write a scenario of door.**

**ANS..**

• Verify the door is single door or bi-folded door.

• Verify the door opens inwards or outwards.

• Verify that the dimension of the doors are as per the specifications.

• Verify that the material used in the door body and its parts is as per the specifications.

• Verify that color of the door is as specified.

• Verify if the door is sliding door or rotating door.

• Verify the position, quality and strength of hinges.

• Verify the type of locks in the door.

• Verify the number of locks in the door interior side or exterior side.

• Verify if the door is having peek-hole or not.

• Verify if the door is having stopper or not.

• Verify if the door closes automatically or not – spring mechanism.

• Verify if the door makes noise when opened or closed.

• Verify the door condition when used extensively with water.

• Verify the door condition in different climatic conditions- temperature, humidity etc.

• Verify the amount of force- pull or push required to open or close the door.

• Verify the handle to open/close the door, can be used by various size of human hands

• Verify that we are able to lock and unlock the door

• Verify if someone knocking the door we are able to listen or not

• Verify that someone people knocking the door we are able to see

• Verify the door is water proof or not.

**Q-48 Write a scenario of ATM.**

**ANS..**

• Verify that power backup should be present at ATM.

• Verify that card reader should be present.

• Verify that receipt printer should be present and working.

• Verify that cash dispenser is working as expected.

• Verify that the key pad should be working and covered.

• Verify that buttons are displayed on screen of ATM machine.

• Verify the font of text on the screen, it should be clearly visible.

• Verify that when card in inserted in ATM, pin should be asked from user.

• Verify that when user enters incorrect pin for a particular number of times, the card is blocked.

•Verify that when user enters correct pin, the user details should be displayed on ATM screen.

• Verify that ATM machine asks to user for the amount to be withdrawn.

• Verify that if use enters amount greater than daily withdraw limit, error message is displayed.

• Verify that if doesn't enter amount in round off digits, error message is displayed.

• Verify that if user enters valid amount, the exact amount of cash should be dispensed from ATM machine.

• Verify that how much time is taken in a transaction.

• Verify how much time is taken by system to logout user.

• Verify that user is able to use card of other bank on the ATM.

• Verify that message is displayed when the cash in ATM is finished.

• Verify that correct message is displayed after the transaction.

• Verify that user is presented with an option to select language of operation.

• Verify that pin is displayed in masked format.

• Verify that error message is displayed when entered amount is greater than account balance.

• Verify that session is timeout is no activity is performed for a particular time.

• Verify that the user is not allowed to exceed one transaction limit amount.

• Verify that the user is not allowed to exceed the one-day transaction limit amount.

• Verify the functionality by entering a wrong pin number for a particular number of times.

• Verify the card ATM machine functionality by inserting an expired card.

•Verify the cash withdrawal functionality by entering invalid amount such as 10, 20, 50.

•Verify the ATM machine functionality by entering amount greater than available balance. y entering amour

• Verify the ATM machine functionality by entering amount greater than per day and per transaction limit.

**Q-49 When to used Usability Testing?**

**ANS..** Web Based Testing, Desktop Based. Mobile based & Game based Testing:

All fields on a page (For Example, text box, radio options, drop-down lists) should be aligned properly

The user should not be able to type in drop-down select lists.

Tab and Shift +Tab order should work properly.

All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard

All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard,

All pages should have a title

Confirmation messages should be displayed before performing any update or delete operation

Hourglass should be displayed when the application is busy

Page text should be left-justified.

The user should be able to select only one radio option and any combination for checkboxes

* Goal of Usability Testing

Effectiveness of the system

Efficiency

Accuracy

User Friendliness

HOW MANY USERS DO YOU NEED?

5(five) users are enough to uncover 80% of usability problems. Some researchers suggest other numbers. The truth is, the actual number of user required depends on the complexity of the given application and your usability goals. Increase in usability participant's results into increased cost, planning, participant management and data analysis.

But as a general guideline, if you on a small budget and interested in DIY usability testing 5 is a good number to start with. If budget is not a constraint its best consult experienced professionals to determine number of users.

**Q-50 What is the procedure for GUI Testing?**

**ANS..** Graphical User Interface (GUI) testing is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of barstool bar, menu bar, dialog boxes and windows etc.

WHAT DO YOU CHECK IN GUI TESTING?

Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.

Check you can execute the intended functionality of the application using the GUI

Check Error Messages are displayed correctly

Check for Clear demarcation of different sections on screen

Check Font used in application is readable

Check the alignment of the text is proper

Check the Color of the font and warning messages is aesthetically pleasing

Check that the images have good clarity

Check that the images are properly aligned

Check the positioning of GUI elements for different screen resolution.

* Approach of GUI Testing

**MANUAL BASED TESTING**

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

**RECORD AND REPLAY**

GUI testing can be done using automation tools. This is done in 2 parts. During Record, test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test. Example of such tools - QTP.

**MODEL BASED TESTING**

A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements.

* GUI Testing Examples:

Web Based Testing & Desktop Based Testing:

The scrollbar should be enabled only when necessary.

Font size, style, and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS.

The description text box should be multi-lined.

Enough space should be provided between field labels, columns, rows, error messages, etc.

* Mobile Based Testing:

If mobile is in every orientation mode so display image, video properly.

Every app will display in responsive type.

Alignment should be apply properly of every field.

* Game Based Testing:

Game infra design will showing properly

Game points or score will display proper with its background color.

Game sound manage with its background effect

Can be also conducted in advance of designing page layouts or navigation menus.

**Q-51 Write a scenario of microwave Owen.**

**ANS..**

• Verify that type of microwave oven like solo, grill or convection

• Verify that the company name is properly displayed or not

• Verify that the Brand logo is properly displayed on the microwave oven or not

• Verify that size of the microwave oven

• Verify that colour of the microwave oven

• Verify that material of the microwave oven

• Verify that capacity of the microwave oven

• Verify that the compact design of the microwave oven

• Verify that glass is turn able or not

• Verify that weight of the microwave oven

• Verify that dimensions of the microwave oven

• Verify that voltage of the microwave oven

• Verify that batteries are required or not

• Verify that all buttons are properly worked or not

• Verify that food is properly reheating or not

• Verify that food is grilled properly or not

• Verify that the digital displayed screen should be properly visible to users

• Verify that oven's door is properly opened and get closed

• Verify that different kind of food at different temperature

• Verify that different kind of containers

• Verify that temperature functionality is properly working or not

• Verify that the alarm sound system is properly working or not

• Verify that glass rotation speed is as expected

• Verify oven's condition when it runs for specific hours

• Verify that disconnecting power while cooking is in progress

**Q-52 Write a scenario of coffee vending machine.**

**ANS..**

• 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 color as well brand is correctly visible and as per specification.

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

• 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.

• Verify the machine can be switched on and off using the power buttons.

• Verify for the indicator lights when the machine is switched on-off.

• Verify that the functioning of all the buttons work properly when pressed.

• Verify that each button has an image/text with it, indicating the task it performs.

• Verify that complete quantity of coffee should get poured in a single operation, no residual coffee should be present in the nozzle.

• Verify that the coffee served has the same and correct temperature each time it is served by the machine.

• Verify that system should display an error when it runs out of ingredients.

• Verify that pressing the coffee button multiple times leads to multiple serving of coffee.

• Verify that there is the passage for residual/extra coffee in the machine.

• Verify that machine should work correctly in different climatic, moistures and temperature conditions.

• Verify that machine should not make too much sound when in operation.

• Verify the amount of time the machine takes to serve a single serving of coffee.

• Verify the performance of the machine when used continuously until the ingredients run out of the requirements.

• Verify the functioning of the coffee machine when two/multiple buttons are pressed simultaneously.

• Verify the functioning of coffee machine with a lesser or higher voltage than required.

• Verify the functioning of the coffee machine if the ingredient container’s capacity is exceeded.

• Verify for the cup holder dimension as per specification/ or market standard

• Verify for guarantee & warranty of the machine, in case provided.

**Q-53 Write a scenario of chair.**

**ANS..**

•Verify that the chair is stable enough to take an average human load.

•Verify that material used in making the chair-wood, plastic etc.

•Verify that chair’s leg are level to the floor.

•Verify that usability of the chair as an office chair, normal household chair.

•Verify there is back support in the chair.

•Verify there is support for hands in the chair.

•Verify the paint’s type and color.

•Verify if the chair’s material is brittle or not.

•Verify cushion is provided with chair or not.

•Verify condition when washed with water or effect of water on chair.

•Verify that the dimension of chair is as per the specifications.

•Verify that the weight of the chair is as per the specifications.

•Verify the height of the chair’s seat from floor.

•Verify whether distance between all legs it should be same.

•Verify by design because there is a chairs with 3 legs or modern design when two and two are different distance

•Verify that wheels are needed specified in chair.

•Verify the wheels are turning 360 degrees or not.

•Verify the wheels are oiled well or not.

•Verify that chair can be upgraded (different back) or add the wheels

•Verify chair can be stored with other chairs on top/bottom

•Verify that strength and material of the chair

•Verify that the Main Function check stability

•Verify edged of chair is needed to be in round shape or sharp

**Q-54 To Create scenario (positive & negative) 1. Gmail(Receiving mail) 2. Online shopping to buy product(flipkart).**

**ANS..** **1. Gmail (receiving mail):**

• Verify that a newly received email is displayed as highlighted in the Inbox section.

• Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to a single line).

• Verify that the user receives the email in their inbox.

• Verify that on clicking the newly received email, the user is navigated to email content.

• Verify that the email contents are correctly displayed with the desired source formatting.

• Verify that any attachments are attached to the email and are downloadable.

• Verify that the attachments are scanned for viruses before download.

• Verify that all the emails marked as read are not highlighted.

• Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.

• Verify that count of unread emails is displayed alongside ‘Inbox’ text in the left sidebar of Gmail.

• Verify that unread email count increases by one on receiving a new email.

• Verify that unread email count decreases by one on reading an email (marking an email as read).

• Verify that email recipients in cc are visible to all users.

• Verify that email recipients in bcc are not visible to the user.

• Verify that all received emails get piled up in the ‘Inbox’ section and get deleted in cyclic fashion based on the size availability.

• Verify that email can be received from non-Gmail email Ids like – yahoo, Hotmail etc.

2. **Online shopping to buy product(flipkart).**

• Verify the initiation of the buy flow.

• Verify the accuracy of product details.

• Verify responsiveness to changes in quantity.

• Verify the behavior with out-of-stock items.

• Verify the adding products from different categories.

• Verify the accuracy and function of applied discounts.

• Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc.

• Verify that users can add products to the wishlist.

• Verify that the user can see the previously added products on the cart page, after signing in to the application.

• Verify that the user can successfully buy more than one products that were added to his/her cart.

• Verify that the user cannot add more than the available inventory of the product.

• Verify that the limit to the number of products a user can buy is working correctly. Also, an error message gets displayed, preventing the user from buying more than the limit.

• Verify that the delivery can be declined during checkout for the places where shipping is not available.

• Verify that the Cash on Delivery option of payment is working fine.

• Verify that the different prepaid methods of payments are working fine.

• Verify that product return functionality works correctly

• Verify the visibility and accuracy of delivery options.

• Verify responsiveness of the payment selection step.

• Verify the behavior with saved addresses.

• Verify the accuracy of order summary details.

• Verify responsiveness to changes in user location.

• Verify the redirection to the Order Confirmation page.

• Verify the cancelling the payment process midway.

**Q-55 Write a scenario of wrist watch.**

**ANS..**

•Verify the type of watch – analog or digital.

•Verify the correctness time displayed by the second, minute, and hour hand of the watch.

•Verify the digital display for hours, minutes, and seconds is correctly displayed.

•Verify the material of the watch and its strap.

•Verify 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.

•Verify the watch is waterproof or not.

•Verify that the numbers in the dial are clearly visible or not.

•Verify the watch is having a date and day display or not.

•Verify the color of the text displayed in the watch – time, day, date, and other information.

•Verify that clock’s time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.

•Verify the second hand of the watch makes ticking sound or not.

•Verify the brand of the watch and check if its visible in the dial.

•Verify the clock is having stopwatch, timers, and alarm functionality or not.

•Verify the format of the watch 12 hours or 24 hours.

•Verify the watch comes with any guarantee or warranty.

•Verify the dial has glass covering or plastic, check if the material is breakable or not.

•Verify the dial’s glass/plastic is resistant to minor scratches or not.

•Verify the battery requirement of the watch.

•Varify belt or chain used is comfortable or not and it’s length

•Verify chain material and belt for damage.

**Q-56 Write a scenario of lift(Elevator)**

**ANS..**

• Verify the maximum passenger capacity of lift.

• Verify the maximum weight capacity of lift.

• Verify the Height, Width and length of lift.

• Verify the type of material used in lift interior and exterior.

• Verify the type of door of lift.

• Verify that buttons are present according to the number of floors in building.

• Verify that buttons to open and close lift door should be present.

• Verify that the buttons should be self-explanatory.

• Verify that blind person should be able to use the buttons.

• Verify that the controls are guided by a voice message.

• Verify that emergency button and contact details should be present in lift.

• Verify the performance of lift, the time taken to go a particular floor.

• Verify that fan is present in lift or not.

• Verify that light is present in lift or not.

• Verify that lift moves to particular floor when the floor button is pressed.

• Verify that lift comes to floor is up/down button are pressed at a particular floor.

• Verify that the door should not be opened if open is door button is pressed in between two floors.

• Verify the behaviour of lift in case of power failure, lift should not free fall or should not stop in between two floors.

• Verify that lift door closes or not if an object is placed in between the door, it should not close.

• Verify the time for which the door remains open at floor.

• Verify that door should not open while the lift is in motion.

• Verify that backup mechanism should be present in case of power loss.

• Verify that in case multiple floor number button is clicked, lift should stop at each floor.

• Verify that in case of capacity limit is reached users are prompted with warning alert.

• Verify that inside lift user are prompted with current floor and direction information the lift is moving towards.

• Verify the behaviour of lift when the capacity of lift exceeds.

Verify the behaviour of lift when there is smoke or fire inside the lift.

• Verify the behaviour by pressing open button while lift is moving.

• Verify the behaviour by pressing stop button before reaching the specific floor.

**Q-57 Write a scenario of whatsapp Group (generate group)**

**ANS..** **Group Chat:**

• Verify that user is able to create a new or not.

• Verify that user is able to add multiple contacts from contact list.

• Verify that user is able to insert group name and select image for DP.

• Verify that user is able to add and remove contacts from group.

• Verify that user is able to delete a group.

• Verify that user can send and receive text messages in group.

• Verify that user can send and receive documents in group chat box.

• Verify that user can send and receive photos in group chat box.

• Verify that user can send and receive videos in in group chat box.

• Verify that user can send and receive audio in in group chat box.

• Verify that user can send and receive emotions icons in in group chat box.

• Verify that user can send and receive Contacts in group chat box.

• Verify that user can send and receive Location in group chat box.

• Verify that user can send and receive GIF in in group chat box.

• Verify that user can send and receive Stickers in group chat box.

• Verify that user can delete text, video, audio, locations, documents in group chat box.

• Verify that user can send recorded voice mail in group chat box.

• Verify that user is able make multiple video call in group chat box.

• Verify that user is able to see the group contact information from Group Info in group chat box.

• Verify that user is able to shared images, links, and documents from Group Media in group chat box.

• Verify that user is able to search specific chat history using search option in group chat box.

• Verify that user is able to mute the group in group chat box.

• Verify that users have options like Report, Block, Clear Chat, Export Chat and Add Shortcut.

.

**Q-58 Write a scenario of whatsapp payment.**

**ANS..** **Whatsapp payment:**

• Verify that users can register for WhatsApp Payment.

• Verifying the user's phone number for whatsapp payment.

• Verify that users can link their bank account to WhatsApp for transactions.

• Verify that sending money from one user to another within the WhatsApp application.

• Verify that users can select a contact from their address book to send money to.

• Verify that users can enter the amount they wish to send and add a note if necessary.

• Verify that users can review the transaction details before confirming.

• Verify that receiving money from another user.

• Verify that users receive a notification when they have received money.

• Verify that the transaction details are visible within the chat interface.

• Verify that users can view their transaction history.

• Verify that transaction history includes details such as date, time, amount, sender/receiver, and transaction status (completed, pending, failed, etc.).

• Verify that security features such as PIN or biometric authentication before making a payment.

• Verify that users receive confirmation messages for each transaction, including details about the recipient and amount sent.

• Verify that adding a bank account for transactions.

• Verify that users can add multiple bank accounts if supported.

• Verify that users can edit or remove bank accounts as needed.

• Verify that there are limits on the amount of money that can be sent or received per transaction and per day

• Verify that performance of the payment feature under various network conditions (3G, 4G, 5G, Wi-Fi).

• Verify that transactions are processed promptly and reliably without delays or errors.

.