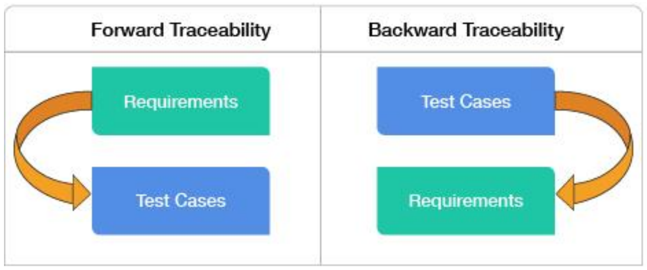
Software Testing Assignment
Module–2(Manual Testing)

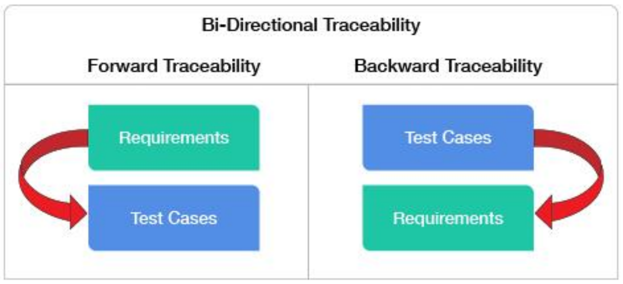
1. <u>What is Exploratory Testing?</u>

❖ Exploratory testing is a simultaneous process of test design and test execution all done at the same time.
❖ In this tester will explore application without any documentation to look how system will work and write scenario.
❖ In exploratory testing, tester will be exploring the application in all possible ways, understanding the flow of the application, preparing a test document and then testing the application, this approach is known as exploratory testing.

2. <u>What is traceability matrix?</u>

❖ A Traceability Matrix is a document that maps or traces the relationship between two baseline documents.
❖ The Traceability Matrix is an essential document used during the software development lifecycle of a product, and it ensures completeness and transparency of the product. It is also called Requirement Traceability Matrix (RTM).
❖ In short, **Requirement Traceability Matrix (RTM)** is a document that maps and traces user requirement with test cases.
❖ Types of Traceability matrix
  - **Forward Traceability:** Mapping requirements to test cases
  - **Backward Traceability:** Mapping test cases to requirements
  - **Bi-directional Traceability:** Combination of forwarding and backward traceability matrix, it ensures that all requirements are covered by test cases.

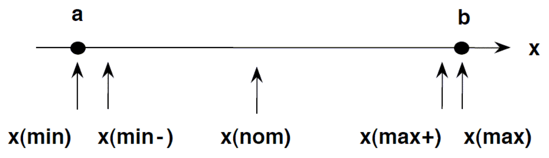




Advantage of Requirement Traceability Matrix
- It confirms 100% test coverage
- It highlights any requirements missing or document inconsistencies
- It shows the overall defects or execution status with a focus on business requirements
- It helps in analyzing or estimating the impact on the QA team’s work with respect to revisiting or re-working the test cases

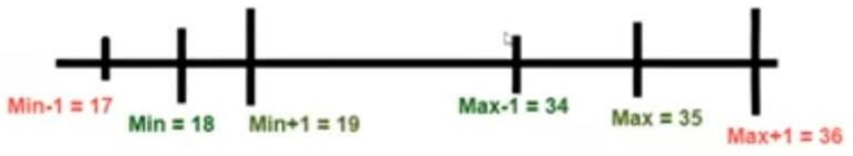
3. What is Boundary value testing?

- Boundary value analysis is one of the widely used case design technique for black box testing.
- Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions.
- A boundary value for a valid partition is a valid boundary value.
- A boundary value for an invalid partition is an invalid boundary value.
- For each variable we check-
    1. Minimum value.
    2. Just above the minimum.
    3. Nominal Value.
    4. Just below Max value.
    5. Max value



- **Example:**

Enter Age: *Allow digits from 18—35



Min = 18 (PASS)
Min-1 = 17(FAIL)
Min+1 = 19(PASS)

Max = 35 (PASS)
Max-1 = 34 (PASS)
Max+1 = 36 (FAIL)

4. What is Equivalence partitioning testing?

- This technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.
- It divides the input data of software into different equivalence data classes.
- This technique is particularly useful when dealing with a large range of input values.

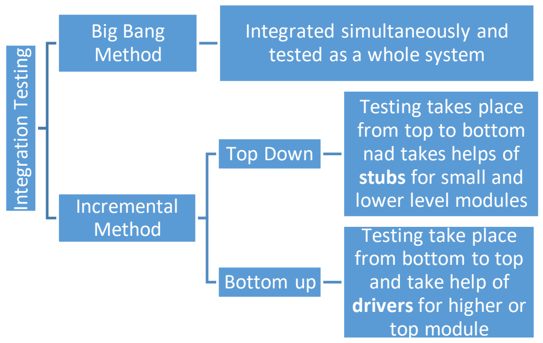
- **Example:**

Percentage: *Accept value between 50 to 90

Equivalence Partitioning		
Value range	Value range	Value range
<50	50-90	>90
Invalid	Valid	Invalid

5. What is Integration testing?

- ❖ Integration testing is the second level of the software testing process comes after unit testing.
- ❖ In integration testing components of the software are gradually integrated and then tested as a unified group.
- ❖ Usually, these components are already working well individually, but they may break when integrated with other components.
- ❖ With integration testing, testers want to find defects that surface due to code conflicts between software modules when they are integrated with each other.
- ❖ There are two types of integration testing:
  - Component integration testing
  - System integration testing
- ❖ Integration testing done by 2 different methods:
  - Big Bang
  - Incremental


Integration Testing
Big Bang Method
Integrated simultaneously and tested as a whole system
Incremental Method
Top Down
Testing takes place from top to bottom nad takes helps of **stubs** for small and lower level modules
Bottom up
Testing take place from bottom to top and take help of **drivers** for higher or top module

6. What determines the level of risk?

- ❖ Risk is the probability of occurrence of an undesirable event.
- ❖ It could be any future event with a negative consequence. You need to identify the risks associated with your project
- ❖ There are 2 level of risk may come on your project.
  - **Project Risks** (Example of Project risk is Senior Team Member leaving the project abruptly.)
  - **Product Risk** (Example of product risks would be Flight Reservation system not installing in test environment)

7. What is Alpha testing?

- ❖ Alpha testing is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly.
- ❖ It always performs by developers at the software development site.
- ❖ It is always performed in virtual environment.
- ❖ It comes under the category of White box testing and Black box testing.

8. What is beta testing?

- ❖ Beta testing is the process of testing a software product or service in a real-world environment before its official release.
- ❖ Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.

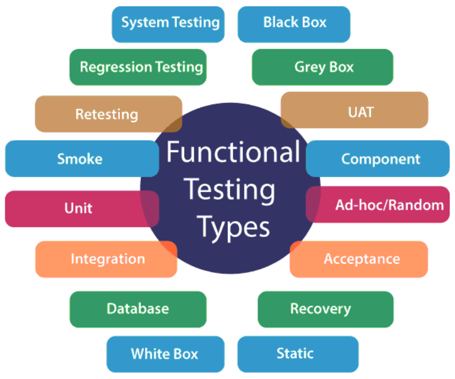
- Beta testing is the last phase of the testing, which is carried out at the client's or customer's site.
- It is performed in Real Time Environment. It is always performed outside the organization
- It is only a kind of Black Box Testing.

9. <u>What is component testing?</u>

- Component means a smallest part/entity/unit of software.
- Component testing means the testing is performed on each individual component/unit separately without integrating with other components.
- Component testing also known as Unit Testing, Module Testing or Program Testing.
- Unit testing is the first level of testing and is performed prior to Integration Testing.
- Unit testing is mostly done by developers by using white box techniques.

10. <u>What is functional system testing?</u>

- Functional testing is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification.
- This testing is not concerned with the source code of the application
- Each functionality of the software application is tested by providing appropriate test input, expecting the output, and comparing the actual output with the expected output.
- It defines "What the product does?"



Type of Functional Testing Techniques:

1. **Unit Testing:**
   Unit testing is the type of functional testing technique where the individual units or modules of the application are tested. It ensures that each module is working correctly.

2. **Integration Testing:**
   In Integration testing, combined individual units are tested as a group and expose the faults in the interaction between the integrated units.

3. **System Testing:**
   System testing is a type of software testing that is performed on the complete integrated system to evaluate the compliance of the system with the corresponding requirements.

4. **User Acceptance Testing:**
   User acceptance testing is done by the client to certify that the system meets the requirements and works as intended. It is the final phase of testing before the product release.

5. **White box Testing:**

White box testing is a type of software testing that allows the tester to verify the internal workings of the software system. This includes analyzing the code, infrastructure, and integrations with the external system.

6. **Black box Testing:**
   Black box testing is a type of software testing where the functionality of the software system is tested without looking at the internal working or structures of the software system.

7. **Smoke Testing:**
   Smoke testing is a type of functional testing technique where the critical functionality or feature of the application is tested as it ensures that the most important function works properly.

8. **Sanity Testing**:
   Sanity testing is a subset of regression testing and is done to make sure that the code changes introduced are working as expected.

9. **Regression Testing:**
   Regression testing is done to make sure that the code or environment changes should not affect the existing functionality and the features of the application. It concentrates on whether all parts are working or not.

10. **Re-Testing:**
    Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions

11. **Adhoc Testing:**
    Adhoc testing also known as monkey testing or random testing is a type of software testing that does not follow any documentation or test plan to perform testing and its main aim to break the system.

12. **Experience Testing:**
    testing technique is solely based on the experience of the tester.

13. **End to End testing:**
    End-to-end testing is a software testing technique that verifies the functionality and performance of an entire software application from start to finish by simulating real-world user scenarios and replicating live data.

14. **Static Testing:**
    Static testing is a type of software testing which is performed to check the defects in software without actually executing the code of the software application.

15. **Grey box Testing:**
    Grey box testing is a type of software testing that includes black box and white box testing but with limited knowledge of the internal workings of an application.

16. **Component Testing:**

Component testing also known as program testing or module testing is a type of software testing that is done after the unit testing. In this, the test objects can be tested independently as a component without integrating with other components.

11. What is Non-Functional Testing?

- ❖ Non-functional testing is a type of software testing that verifies non-functional aspects of the product, such as performance, stability, and usability.
- ❖ Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability
- ❖ May be performed at all Test levels
- ❖ It is the testing of "how" the system works.

Type of Non-functional Testing Techniques:

1. **Performance testing**
   Performance testing verifies whether the system meets the non-functional requirements identified in the SRS document or not. It verifies how the system behaves and how well the system works based on response time, throughput, etc.

2. **Load testing**
   Load testing checks how the system responds when the load is increased. When the load is at its peak, it verifies if the application works as expected. One way to constantly perform this testing is to increase the number of users using the same application until the time load reaches its maximum.

3. **Stress testing**
   Stress testing requires increasing the system's capacity to check the point at which it fails. An example is by feeding a system with illegal or invalid inputs to stress the capabilities of a product.

4. **Volume testing**
   In volume testing, the amount of data is increased and fed to the software. With this, the efficiency and response time of the software is checked for any data loss.

5. **Security testing**
   This type of testing is known as one of the most important types. Security testing ensures that an application is secured. Security deals with authentication, authorization, sessions, etc.

6. **Usability testing**
   Usability testing is focused on how usable a system is. The main focus is the experience of using the application. The application should be user-friendly.

7. **Compatibility testing**
   Compatibility testing tests if the application or software is compatible with other hardware or software platforms where it will be used**.**

8. **User Interface testing**
   User Interface testing is done regarding the application designs. It helps ascertain that all the elements that make up the interface work as they should.

12. <u>What is GUI Testing?</u>

- GUI is the abbreviation of 'Graphical User Interface'.
- It contains several visual elements, such as buttons, text boxes, menus, checkboxes, images, etc.
- GUI testing refers to the validating UI functions or features of an application that are visible to the users, and they should comply with business requirements.

13. <u>What is Ad hoc testing?</u>

- Ad-hoc testing is a type of software testing that is performed informally and randomly after the formal testing is completed to find any loophole in the system.
- Ad hoc Testing does not follow any structured way of testing and it is randomly done on any part of application.
- Main aim of this testing is to find defects by random checking. Ad-hoc testing can be achieved with the Software testing technique called **Error Guessing.**
- Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.

Types of Adhoc Testing

1. **Buddy Testing**
   Buddy testing is a type of Adhoc testing where two bodies will be involved one is from the Developer team and one from the tester team. So that after completing one module and after completing Unit testing the tester can test by giving random inputs and the developer can fix the issues too early based on the currently designed test cases.

2. **Pair Testing**
   Pair testing is a type of Adhoc testing where two bodies from the testing team can be involved to test the same module. When one tester can perform the random test another tester can maintain the record of findings. So when two testers get paired they exchange their ideas, opinions, and knowledge so good testing is performed on the module.

3. **Monkey Testing**
   Monkey testing is a type of Adhoc testing in which the system is tested based on random inputs without any test cases the behavior of the system is tracked and all the functionalities of the system are working or not is monitored. As the randomness approach is followed there is no constraint on inputs so it is called Monkey testing.

14. <u>What is load testing?</u>

- Load testing is an integral part of performance testing under non-functional testing and used to check the performance of the software by applying some load.

- Load testing is 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.
- Here, load means that when N-number of users using the application simultaneously or sending the request to the server at a time.
- Load testing will help to detect the maximum operating capacity of an application and any blockages or bottlenecks.
- It governs how the software application performs while being accessed by several users at the same time.
- The load testing is mainly used to test the Client/Server's performance and applications that are web-based.

  Advantages
  - Load testing helps us to detect the bottlenecks and performance-related issues before production.
  - The load testing enhances the scalability regarding network, software and database for the system or software application.
  - The major advantage of performing the load testing is reducing the cost failures, which also helps us minimize the system interruption risks.
  - Customer's satisfaction is enhanced while using the load testing.

  Disadvantages
  - Load testing can only be executed if we have enough knowledge of any programming language as well as testing tools.
  - Usage of load testing tools can be an expensive process because pricing depends on the number of virtual users supported.

15. What is stress Testing?

- **Stress Testing** is a type of software testing that verifies stability & reliability of software application.
- The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn’t crash under crunch situations.
- It even tests beyond normal operating points and evaluates how software works under extreme conditions.
- In other words, we can say that Stress testing is used to verify the constancy and dependability of the system and also make sure that the system would not crash under disaster circumstances.

Advantages
- Stress testing signifies the system's behavior after failure and makes sure that the system recovers quickly from the crashes.
- The most important advantage of executing the stress testing will make the system work in regular and irregular conditions in a suitable way.
- It determines the scalability and enhance the performance of the software.

Disadvantages

- Even in open-source tools like JMeter, a load testing environment is required, which should be as close to the production environment setup as possible.
- If we are writing the Stress test script, the person should have enough scripting knowledge of the language supported by the particular tool.
- If we are using stress testing, it will require additional resources, which makes this testing bit costlier.
- If we perform the **Stress Testing** manually, it became a tedious and complicated task to complete, and it may also not produce the expected results.

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

- ❖ Testing based on an analysis of the internal structure of the component or system is known as white box testing.
- ❖ white box testing is also known as glass box is testing, structural testing, clear box testing, open box testing and transparent box testing.
- ❖ It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs.
- ❖ It is based on inner workings of an application and revolves around internal structure testing.
- ❖ In this type of testing programming skills are required to design test cases.
- ❖ The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

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

- ❖ Testing, either functional or non-functional, without reference to the internal structure of the component or system is known as Black box testing.
- ❖ Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding.
- ❖ The primary source of black box testing is a specification of requirements that is stated by the customer.
- ❖ In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed.
- ❖ It is also known as Specification based testing techniques.

Black box testing techniques:

1. Decision Table Technique

   Decision Table Technique is a systematic approach where various input combinations and their respective system behavior are captured in a tabular form. It is appropriate for the functions that have a logical relationship between two and more than two inputs.

1. Boundary Value Technique

   Boundary Value Technique is used to test boundary values, boundary values are those that contain the upper and lower limit of a variable. It tests, while entering boundary value whether the software is producing correct output or not.

2. State Transition Technique

   State Transition Technique is used to capture the behavior of the software application when different input values are given to the same function. This applies to those types of applications that provide the specific number of attempts to access the application.

3. Equivalence Partitioning Technique

   Equivalence partitioning is a technique of software testing in which input data divided into partitions of valid and invalid values, and it is mandatory that all partitions must exhibit the same behavior.

4. Error Guessing Technique

   Error guessing is a technique in which there is no specific method for identifying the error. It is based on the experience of the test analyst, where the tester uses the experience to guess the problematic areas of the software.

5. Use Case Technique

   Use case Technique used to identify the test cases from the beginning to the end of the system as per the usage of the system. By using this technique, the test team creates a test scenario that can exercise the entire software based on the functionality of each function from start to end.

18. **Mention what are the categories of defects?**

- ❖ Defects can be categorized into different types basing on the core issues they address. Some defects address security or database issues while others may refer to functionality or UI issues.

Types of defects are as follow:

1. **Database Defects**: Deals with improper handling of data in the database.

   Examples: Values not inserted into the database properly Improper/wrong/null values inserted in place of the actual values

2. **Critical Functionality Defects**: The occurrence of these bugs hampers the crucial functionality of the application.

   Examples: - Exceptions

3. **Functionality Defects**: These defects affect the functionality of the application.

   Examples: All JavaScript errors Buttons like Save, Delete, Cancel not performing their intended functions A missing functionality (or) a feature not functioning the way it is intended to Continuous execution of loops.

4. **Security Defects**: Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.

   Examples: Authentication: Accepting an invalid username/password Authorization: Accessibility to pages though permission not given

5. **User Interface Defects**: As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

   Examples: Improper error/warning/UI messages Spelling mistakes Alignment problems

19. <u>Mention what bigbang testing is?</u>

- ❖ Big-bang integration testing is a type of integration testing that combines all the modules or components of a system into a single unit and tests them 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.

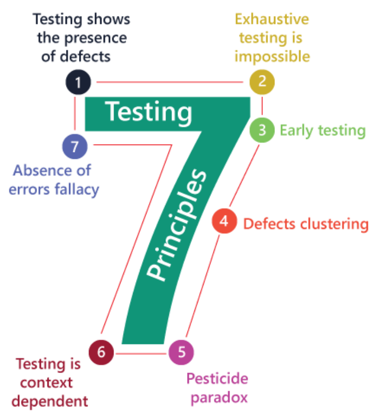
20. <u>What is the purpose of exit criteria?</u>

- ❖ Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

21. <u>When should "Regression Testing" be performed?</u>

- ❖ Regression testing is a type of software testing conducted after a code update to ensure that the update introduced no new bugs.
- ❖ This is because new code may bring in new logic that conflicts with the existing code, leading to defects.
- ❖

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


Testing shows the presence of defects
1
Exhaustive testing is impossible
2
Testing
3 Early testing
7
Absence of errors fallacy
Principles
4 Defects clustering
6
5
Testing is context dependent
Pesticide paradox

❖ It is important that you achieve optimum test results while conducting software testing without deviating from the goal. But how you determine that you are following the right strategy for testing? For that, you need to stick to some basic testing principles. Here are the common seven testing principles that are widely practiced in the software industry.

Testing shows the presence of defects

The test engineer will test the application to make sure that the application is bug or defects free. While doing testing, we can only identify that the application or software has any errors. By doing testing on any application, we can decrease the number of bugs, which does not mean that the application is defect-free because sometimes the software seems to be bug-free while performing multiple types of testing on it. But at the time of deployment in the production server, if the end-user encounters those bugs which are not found in the testing process.

Exhaustive Testing is not possible

Sometimes it seems to be very hard to test all the modules and their features with effective and non- effective combinations of the inputs data throughout the actual testing process.

Hence, instead of performing the exhaustive testing as it takes boundless determinations and most of the hard work is unsuccessful. So, we can complete this type of variations according to the importance of the modules because the product timelines will not permit us to perform such type of testing scenarios.

Early Testing

Here early testing means that all the testing activities should start in the early stages of the software development life cycle's **requirement analysis stage** to identify the defects because if we find the bugs at an early stage, it will be fixed in the initial stage itself, which may cost us very less as compared to those which are identified in the future phase of the testing process.

Defect clustering

The defect clustering defined that throughout the testing process, we can detect the numbers of bugs which are correlated to a small number of modules. We have various reasons for this, such as the modules could be complicated; the coding part may be complex, and so on.

Pesticide paradox

This principle defined that if we are executing the same set of test cases again and again over a particular time, then these kinds of the test will not be able to find the new bugs in the software or the application. To get over these pesticide paradoxes, it is very significant to review all the test cases frequently. And the new and different tests are necessary to be written for the implementation of multiple parts of the application or the software, which helps us to find more bugs.

Testing is context-dependent

Testing is a context-dependent principle states that we have multiple fields such as e-commerce websites, commercial websites, and so on are available in the market. There is a definite way to test the commercial site as well as the e-commerce websites because every application has its own needs, features, and functionality. To check this type of application, we will take the help of various kinds of testing, different technique, approaches, and multiple methods. Therefore, the testing depends on the context of the application.

Absence of errors fallacy

Once the application is completely tested and there are no bugs identified before the release, so we can say that the application is 99 percent bug-free. But there is the chance when the application is tested beside the incorrect requirements, identified the flaws, and fixed them on a given period would not help as testing is done on the wrong specification, which does not apply to the client's requirements. The absence of error fallacy means identifying and fixing the bugs would not help if the application is impractical and not able to accomplish the client's requirements and needs.

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

Quality Assurance	Quality Control	Testing
QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	It includes activities that ensure the verification of a developed software with respect to documented (or not in some cases) requirements.	It includes activities that ensure the identification of bugs/error/defects in a software.
Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing the software with an aim to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
Process-oriented activities.	Product-oriented activities.	Product-oriented activities.
Preventive activities.	It is a corrective process.	It is a preventive process.
It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

24. <u>Difference between Smoke and Sanity?</u>

Smoke Testing	Sanity Testing
To check critical functionality is working properly	To check new functionality/bugs have been fixed.
Objective: Stability	Objective: Rationality
Perform by developers or testers	Perform by testers
Subset of acceptance testing	Subset of regression testing
It checks entire system from end to end	It checks only particular components.
It is usually documented or scripted.	It is not documented or unscripted.
Eg: General Health check up	Eg. Specialized Health check up

25. <u>Difference between verification and Validation</u>

Verification	Validation
It includes checking documents, design, codes and programs.	It includes testing and validating the actual product.
Verification is the static testing.	Validation is the dynamic testing.
It does not include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing.
It can find the bugs in the early stage of the development.	It can only find the bugs that could not be found by the verification process.
The goal of verification is application and software architecture and specification.	The goal of validation is an actual product.
Quality assurance team does verification.	Validation is executed on software code with the help of testing team.
It comes before validation.	It comes after verification.
Verification refers to the set of activities that ensure software correctly implements the specific function.	Validation refers to the set of activities that ensure that the software that has been built is traceable to customer requirements.
Verification is for prevention of errors.	Validation is for detection of errors.
Verification is also termed as white box testing or static testing as work product goes through reviews.	Validation can be termed as black box testing or dynamic testing as work product is executed.
Verification is about process, standard and guideline.	Validation is about the product.

26. <u>Explain types of Performance testing.</u>

- ❖ Checking the behavior of an application by applying some load is known as performance testing.
- ❖ While doing performance testing on the application, we will concentrate on the various factors like **Speed, Scalability, Stability** of the application.
    - o Speed – Determines whether the application responds quickly
    - o Scalability – Determines maximum user load the software application can handle.
    - o Stability – Determines if the application is stable under varying loads

Types of performance testing:

1. **Load testing:**
   checks the application's ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live**.**

2. **Stress testing:**
   Involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.

3. **Endurance testing:**
   Is done to make sure the software can handle the expected load over a long period of time.

4. **Spike testing:**
   Tests the software's reaction to sudden large spikes in the load generated by users.

5. **Volume testing:**
   Under Volume Testing large no. of Data is populated in a database, and the overall software system's behavior is monitored. The objective is to check software application's performance under varying database volumes.
6. **Scalability testing:**
   The objective of scalability testing is to determine the software application's effectiveness in "scaling up" to support an increase in user load. It helps plan capacity addition to your software system.

27. <u>What is Error, Defect, Bug and failure?</u>

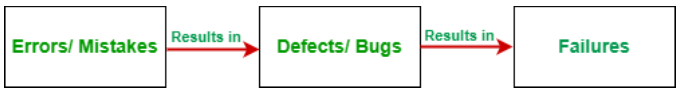
A **bug** is a problem or an issue in the software that causes unexpected behavior. Bugs can be introduced during the development process, or they may arise from changes in the environment where the software runs. Bugs can manifest in various ways, such as crashes, incorrect outputs, or user interface issues.

A **defect** is a non-conformance to a specification or requirement. A defect can be thought of as a deviation from the expected behavior of the software. Defects can be introduced during the development process, and they often result from a failure to understand the requirements or design of the software.

An **error** is a human action that produces an incorrect or unexpected result. Errors are often the result of a misunderstanding of the requirements or design of the software. Errors can occur at any stage of the software development process, from requirements gathering to coding to testing.

A **failure** is the inability of the software to perform its intended function. Failures occur when the software is unable to meet the user's needs or expectations. Failures can result from bugs, defects, errors, or faults.

A simple diagram depicting Bug vs Defect vs Fault vs Failure:

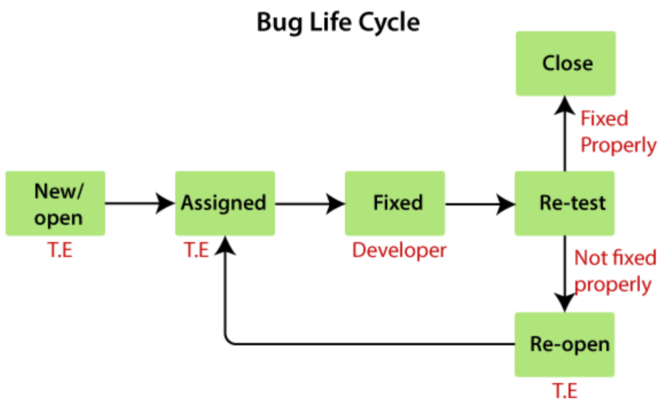


28. Difference between Priority and Severity

Parameters	Severity in Testing	Priority in Testing
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameter	Severity is basically a parameter that denotes the total impact of a given defect on any software.	Priority is basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Change of Value	The value of Severity changes continually from time to time.	The value of Priority changes from time to time.
Who Decides the Defect	The testing engineer basically decides a defect's severity level.	The product manager basically decides a defect's priority level.
Types	There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.	There are 3 types of Priorities: High, Medium, and Low.

29. What is Bug Life Cycle?

- A defect/bug life cycle is the sequence of steps a bug or defect goes through from its identification to its resolution in software development.
- This life cycle standardizes the bug management process, ensuring teams can manage and resolve them more effectively.



- As soon as the test engineer finds the bug, status is given as **New**, which indicates that a bug is just found.
- This new bug needs to be reported to the concerned Developer by changing the status as **Assigned** so that the responsible person should take care of the bug.
- Then the Developer first go through the bug, which means that the Developers read all the navigation steps to decide whether it is a valid bug or not.
- Based on this, if the bug is valid, the Developer starts reproducing the bug on the application, once the bug is successfully reproduced, the Developer will analyze the code and does the necessary changes, and change the status as **Fixed**.
- Once the code changes are done, and the bug is fixed, the test engineer **re-test** the bug, which means that the test engineer performs the same action once again, which is mentioned in the bug report, and changes the status accordingly:
- **Close**, if the bug fixes properly, and functionally working according to the requirement.

30. Explain the difference between Functional testing and Non-Functional testing?

Functional Testing	Non-functional Testing
It verifies the operations and actions of an application.	It verifies the behavior of an application.
It is based on requirements of customer.	It is based on expectations of customer.
It helps to enhance the behavior of the application.	It helps to improve the performance of the application.

Functional testing is easy to execute manually.	It is hard to execute non-functional testing manually.
It tests what the product does.	It describes how the product does.
Functional testing is based on the business requirement.	Non-functional testing is based on the performance requirement.
Examples: 1. Unit Testing 2. Smoke Testing 3. Integration Testing 4. Sanity Testing 5. Black box Testing 6. Whitebox Testing	Examples: 1. Performance Testing 2. Load Testing 3. Stress Testing 4. Scalability Testing 5. Compatibility Testing

31. <u>What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?</u>

S.NO	Comparison basis	SDLC	STLC
1.	Explanations	It is primarily connected to software development, which means that it is the procedure of developing a software application.	It is mainly linked to software testing, which means that it is a software testing process that contains various phases of the testing process.
2.	Representation	SDLC stands for Software Development Life Cycle.	STLC stands for Software Testing Life cycle.
3.	Resources	While performing the SDLC process, we needed a greater number of developers to complete the development process.	The STLC process needed a smaller number of testers to complete the testing process.
4.	Focuses on	Besides the development phase, other phases like testing are also included.	The STLC concentrate only on testing the software.
5.	Objective	The objective of the Software development life cycle is to complete the development of software successfully.	The objective of the Software testing life cycle is to complete the testing of software successfully.
6.	Help in	The SDLC will help us to develop a good quality software product.	The STLC will helps to create the software bug-free.

7.	Different phases	The various phase includes in Software Development Life Cycle are as follows: ○ Requirements Collection ○ Feasibility Study ○ Design ○ Programming or Coding ○ Testing ○ Installation ○ Maintenance	The various phase includes in Software Testing Life Cycle are as follows: ○ Requirement collection or System study ○ Test Plan ○ Write test case ○ Traceability Matrix ○ Defect Tracking ○ Test Execution Report ○ Retrospect meeting
8.	Requirement collection phase	In the SDLC Requirement collection phase, the BA [Business Analyst] and PA [Product Analyst] will collect the requirements and interpret business language into software language.	In the Requirement Analysis phase of the STLC, the QA [Quality Assurance] team will study requirement documents and prepare the System Test Plan.
9.	Designing phase	Based on the requirement understanding, the development team will develop the HLD [High-Level Design] and LLD [Low-Level Design] of the software.	Generally, in STLC, the Test Architect or a Test Lead plan the test strategy. And also finds the testing points.
10.	Coding phase	In the SDLC coding phase, the developer will start writing the code as per the designed document and beginning of building the software.	In STLC, the QA team writes the test scenarios to authenticate the quality of the product.
11.	Environment Set up	After writing the code, the development team sets up a test environment with the developed product to validate the code.	Based on the prerequisites, the Test team confirms the environment set up. And do one round of smoke testing to ensure that the environment is stable for the product and ready for testing.
12.	Testing Phase	Once the environment has been set, the test engineer will perform various types of testing, such as Unit, Integration, System, Retesting, Regression testing, and so on. And the development team is also involving to fixing the bugs and report back to the tester.	Based on the test cases, the tester will do one round of integration and system testing. While performing the testing, if they encounter with any bugs, it will be reported and fixed after the retesting.
13.	Deployment/ Product Release phase	In the SDLC deployment phase, when we received sign-off from various testing teams, the application is deployed or installed in a production environment for real end-users.	In STLC, the Smoke and sanity testing are performed in the production environment as soon as the product is deployed. And the testing team will prepare the test reports and matrix to analyze the product.

14.	Maintenance Phase	Once the product has been deployed, the development team includes support and release updates.	To check maintenance code deployed, the QA team performs the regression suites.
15.	Performed	The SDLC phases are done before the STLC phases.	The STLC phases are completed after SDLC phases.

32. <u>What is the difference between test scenarios, test cases, and test script?</u>

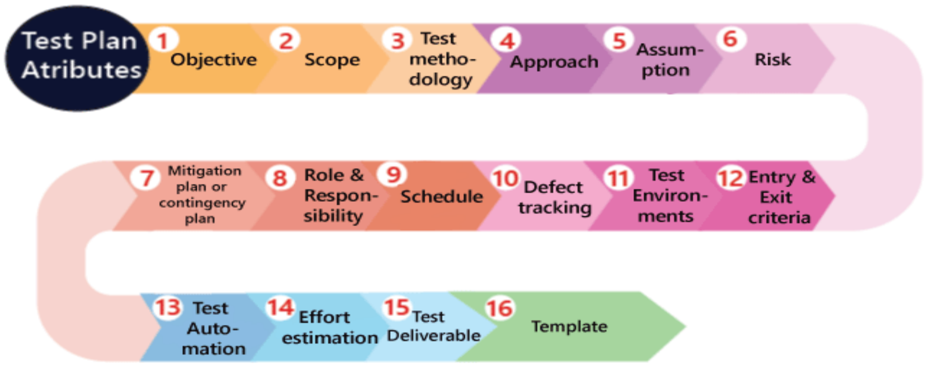
Test Scenario	Test Case	Test Script
It is any functionality that can be tested.	It is a set of actions executed to verify particular feature or functionality.	It is a set of instructions to test an application automatically.
It is derived from BRS (Business requirement specifications) and SRS (Software requirement specifications)	It is derived from test scenario.	It is derived from test cases.
It is more focused on what to test.	It if focused on what to test and how to test.	It is focused on expected result.
Takes less time and fewer resources to create	Require more resources and time.	Require less time for testing but more resources for script creating and updating
Includes end-to-end functionality to be tested	Includes test steps, data, expected result for testing	Includes different commands to developed a script.
The main task to check the full functionality of a software application.	The mail task is to verify compliance with the applicable standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.

33. <u>Explain what Test Plan is? What is the information that should be covered.</u>

- ❖ A test plan is a detailed and comprehensive software testing document that outlines the strategy, objectives, schedule, resources, and overall approach that are needed to complete a project.
- ❖ In many ways, a test plan is like a blueprint for a building.
- ❖ The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.
- ❖ The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager.

- The test plan is prepared by the Test Lead (60%), Test Manager (20%), and by the test engineer (20%).

Test plan components or attributes



1. **Objective**:
   It describes the aim of the test plan, whatever the good process and procedure they are going to follow to give quality software to customers. The overall objective of the test is to find as many defects as possible and to make software bug-free. The test objective must be broken into components and sub-components. In every component following activities should be performed.
   - List all the functionality and performance to be tested.
   - Make goals and targets based on the application feature.

2. **Scope:**
   It consists of information that needs to be tested concerning an application. The scope can be divided into two parts:
   - In-Scope: The modules that are to be tested rigorously.

3. **Testing Methodology:**
   The methods that are going to be used for testing depend on application to application. The testing methodology is decided based on the feature and application requirements. Since the testing terms are not standard, one should define what kind of testing will be used in the testing methodology. So that everyone can understand it.

4. **Approach:**
   The approach of testing different software is different. It deals with the flow of applications for future reference. It has two aspects:
   - High-Level Scenarios: For testing critical features high-level scenarios are written. For Example, login to a website, and book from a website.
   - The Flow Graph: It is used when one wants to make benefits such as converging and merging easy.

5. **Assumption:**
   In this phase, certain assumptions will be made.
   Example:
   - The testing team will get proper support from the development team.
   - The tester will get proper knowledge transfer from the development team.

- ❖ Proper resource allocation will be given by the company to the testing department.

6. **Risk:**
All the risks that can happen if the assumption is broken. For Example, in the case of wrong budget estimation, the cost may overrun. Some reason that may lead to risk is:
   - ❖ Test Manager has poor management skills.
   - ❖ Hard to complete the project on time.
   - ❖ Lack of cooperation.

7. **Mitigation Plan:**
If any risk is involved then the company must have a backup plan, the purpose is to avoid errors. Some points to resolve/avoid risk:
   - ❖ Test priority is to be set for each test activity.
   - ❖ Managers should have leadership skills.
   - ❖ Training course for the testers.

8. **Roles and Responsibilities:**
All the responsibilities and role of every member of a particular testing team has to be recorded.
Example:
   - ❖ Test Manager: Manages the project, takes appropriate resources, and gives project direction.
   - ❖ Tester: Identify the testing technique, verify the test approach, and save project costs.

9. **Schedule:**
Under this, it will record the start and end date of every testing-related activity.

10. **Defect Tracking:**
It is an important process in software engineering as lots of issue arises when you develop a critical system for business. If there is any defect found while testing that defect must be given to the developer team. There are the following methods for the process of defect tracking:
    - ❖ Information Capture: In this, we take basic information to begin the process.
    - ❖ Prioritize: The task is prioritized based on severity and importance.
    - ❖ Communication: Communication between the identifier of the bug and the fixer of the bug.
    - ❖ Environment: Test the application based on hardware and software.

11. **Test Environments**:
It is the environment that the testing team will use i.e. the list of hardware and software, while testing the application, the things that are said to be tested will be written under this section. The installation of software is also checked under this.
Example:
    - ❖ Software configuration on different operating systems, such as Windows, Linux, Mac, etc.
    - ❖ Hardware Configuration depends on RAM, ROM, etc.

12. **Entry and Exit Criteria:** The set of conditions that should be met to start any new type of testing or to end any kind of testing.

    Entry Condition:
    - ❖ Necessary resources must be ready.
    - ❖ The application must be prepared.
    - ❖ Test data should be ready.

    Exit Condition:
    - ❖ There should not be any major bugs.
    - ❖ Most test cases should be passed.
    - ❖ When all test cases are executed.

13. **Test Automation:**
    It consists of the features that are to be automated and which features are not to be automated.
    - ❖ If the feature has lots of bugs, then it is categorized as Manual Testing.
    - ❖ If the feature is frequently tested then it can be automated.

14. **Effort Estimation:**
    This involves planning the effort that needs to be applied by every team member.

15. **Test Deliverables:**
    It is the outcome from the testing team that is to be given to the customers at the end of the project.
    Before the testing phase:
    - ❖ Test plan document.
    - ❖ Test case document.
    - ❖ Test design specification.

During the testing phase:
- ❖ Test scripts.
- ❖ Test data.
- ❖ Error logs.

After the testing phase:
- ❖ Test Reports.
- ❖ Defect Report.
- ❖ Installation Report.

34. <u>What is priority?</u>

- ❖ Priority is a parameter to decide the order in which defects should be fixed.
- ❖ Priority means how fast the defect has to be fixed.
- ❖ It can be urgent, high, medium, and low.
    - o High: it is a major impact on the customer application, and it has to be fixed first.
    - o Medium: In this, the problem should be fixed before the release of the current version in development.
    - o Low: The flow should be fixed if there is time, but it can be deferred with the next release.

35. <u>What is severity?</u>

- ❖ Severity is a parameter to denote the impact of a particular defect on the software.
- ❖ Severity means how severe the defect is affecting the functionality.
- ❖ Types of Severity of bug/defect can be categorized into the following
    - o Critical: This defect indicates complete shut-down of the process, nothing can proceed further
    - o Major: It is a highly severe defect and collapses the system. However, certain parts of the system remain functional
    - o Medium: It causes some undesirable behavior, but the system is still functional
    - o Low: It won’t cause any major break-down of the system

36. <u>Bug categories are…</u>

1. **Database Defects**: Deals with improper handling of data in the database.

   Examples: Values not inserted into the database properly Improper/wrong/null values inserted in place of the actual values

2. **Critical Functionality Defects**: The occurrence of these bugs hampers the crucial functionality of the application.

   Examples: - Exceptions

3. **Functionality Defects**: These defects affect the functionality of the application.

   Examples: All JavaScript errors Buttons like Save, Delete, Cancel not performing their intended functions A missing functionality (or) a feature not functioning the way it is intended to Continuous execution of loops.

4. **Security Defects**: Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.

   Examples: Authentication: Accepting an invalid username/password Authorization: Accessibility to pages though permission not given

5. **User Interface Defects**: As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

   Examples: Improper error/warning/UI messages Spelling mistakes Alignment problems

37. <u>Advantage of Bugzila.</u>

- ❖ Bugzilla is a bug tracking tool developed by the Mozilla Foundation in 1998. It is a quite popular reporting tool which is having a simple user interface. Due to its simple interface, beginners take no time to understand its workflow.
- ❖ Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.

Advantages of Bugzilla

- It improves the quality of the product.
- It enhances the communication between the developing team and the testing team.
- It has the capability to adapt to multiple situations.

38. <u>Difference between priority and severity</u>

Repeat question (pls. ref. Question No: 28)

39. <u>What are the different Methodologies in Agile Development Model?</u>

- 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 Testing Methods:

1) Scrum

SCRUM is an agile development process focused primarily on ways to manage tasks in team-based development conditions.

There are three roles in it, and their responsibilities are:

- **Scrum Master:** The scrum can set up the master team, arrange the meeting and remove obstacles for the process
- **Product owner:** The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.
- **Scrum Team:** The team manages its work and organizes the work to complete the sprint or cycle.

2) eXtreme Programming (XP)

This type of methodology is used when customers are constantly changing demands or requirements, or when they are not sure about the system's performance.

3) Crystal:

There are three concepts of this method-

1. Chartering: Multi activities are involved in this phase such as making a development team, performing feasibility analysis, developing plans, etc.

2. Cyclic delivery: under this, two more cycles consist, these are:
    A. Team updates the release plan.
    B. Integrated product delivers to the users.
3. Wrap up: According to the user environment, this phase performs deployment, post-deployment.

4) Dynamic Software Development Method (DSDM):

DSDM is a rapid application development strategy for software development and gives an agile project distribution structure. The essential features of DSDM are that users must be actively connected, and teams have been given the right to make decisions. The techniques used in DSDM are:

1. Time Boxing
2. MoSCoW Rules
3. Prototyping

The DSDM project contains seven stages:

1. Pre-project
2. Feasibility Study
3. Business Study
4. Functional Model Iteration
5. Design and build Iteration
6. Implementation
7. Post-project

5) Feature Driven Development (FDD):

This method focuses on "Designing and Building" features. In contrast to other smart methods, FDD describes the small steps of the work that should be obtained separately per function.

6) Lean Software Development:

Lean software development methodology follows the principle "just in time production." The lean method indicates the increasing speed of software development and reducing costs. Lean development can be summarized in seven phases.

40. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Authentication	Authorization
Authentication is the process of identifying a user to provide access to a system.	Authorization is the process of giving permission to access the resources.
In this, the user or client and server are verified.	In this, it is verified that if the user is allowed through the defined policies and rules.
It is usually performed before the authorization.	It is usually done once the user is successfully authenticated.
It requires the login details of the user, such as user name & password, etc.	It requires the user's privilege or security level.
Data is provided through the Token Ids.	Data is provided through the access tokens.
Example: Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software.	Example: After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles.
Authentication credentials can be partially changed by the user as per the requirement.	Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner/manager of the system, and he can only change it.

The challenges faced in web application testing are as follow:

- Ensuring Cross-Browser Compatibility. Read Also: Web Application Testing Types: Ensuring Quality and Performance.
- Handling Dynamic Content.
- Performance and Load Testing.
- Security Vulnerabilities.
- Insufficient Bandwidth.
- UI Testing Challenges.
- Altering Environments.
- Checking Compliance & Standards.

41. Write a scenario of only Whatsapp chat messages

Pls. Excel sheet for scenario

42. Write a Scenario of Pen

Pls. Excel sheet for scenario

43. Write a Scenario of Pen Stand

Pls. Excel sheet for scenario

44. <u>Write a Scenario of Door</u>

Pls. Excel sheet for scenario

45. <u>Write a Scenario of ATM</u>

Pls. Excel sheet for scenario

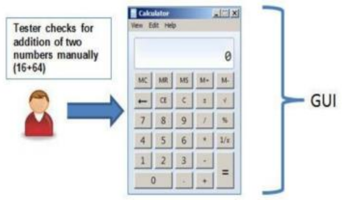
46. <u>When to used Usability Testing?</u>

- Usability Testing is a significant type of software testing technique, which is comes under the non-functional testing.
- The implementation of usability testing requires an understanding of the application, as it is extensive testing.
- Generally, usability testing is performed from an end-user viewpoint to verify if the system is efficiently working or not.
- Checking the user-friendliness, efficiency, and accuracy of the application is known as Usability Testing.
- Usability Testing identifies usability errors in the system early in development cycle and can save a product from failure.
- We need usability testing because usability testing is to build a system with great user experience. Usability is not only used for software development or website development, but it is also used for product designing.
- Goals of usability testing are:
    - Efficiency
    - Memorability
    - Accuracy
    - Learnability
    - Satisfaction
    - Errors

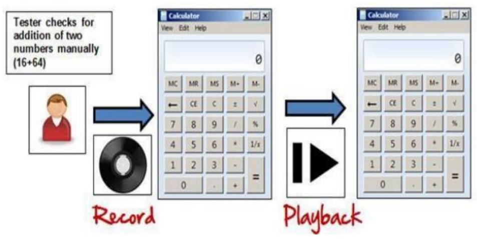
47. <u>What is the procedure for GUI Testing?</u>

Approaches of GUI testing

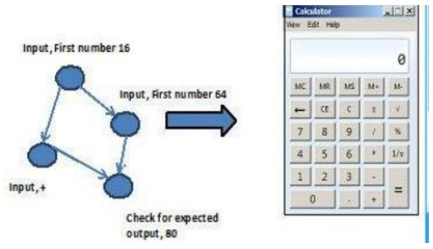
1. **MANUAL BASED TESTING**
   Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.



2. **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.



3. **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.



48. <u>Write a scenario of Microwave Owen</u>

Pls. Excel sheet for scenario

49. <u>Write a scenario of Coffee vending Machine</u>

Pls. Excel sheet for scenario

50. <u>Write a scenario of chair</u>

Pls. Excel sheet for scenario

51. <u>Write a Scenario of Wrist Watch</u>

Pls. Excel sheet for scenario

52. <u>Write a Scenario of Lift (Elevator)</u>

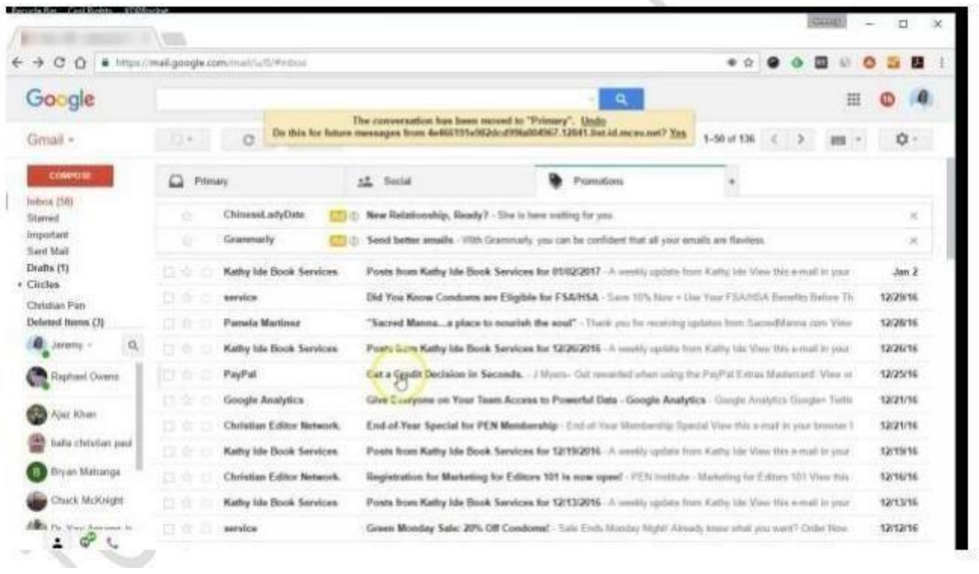
Pls. Excel sheet for scenario

Pls. Note: (Answers for all below questions are done in Excel sheet)
(All test scenarios, HLR and test cases done in Excel sheet)

To Create Scenario (Positive & Negative)

1.Gmail receiving mail

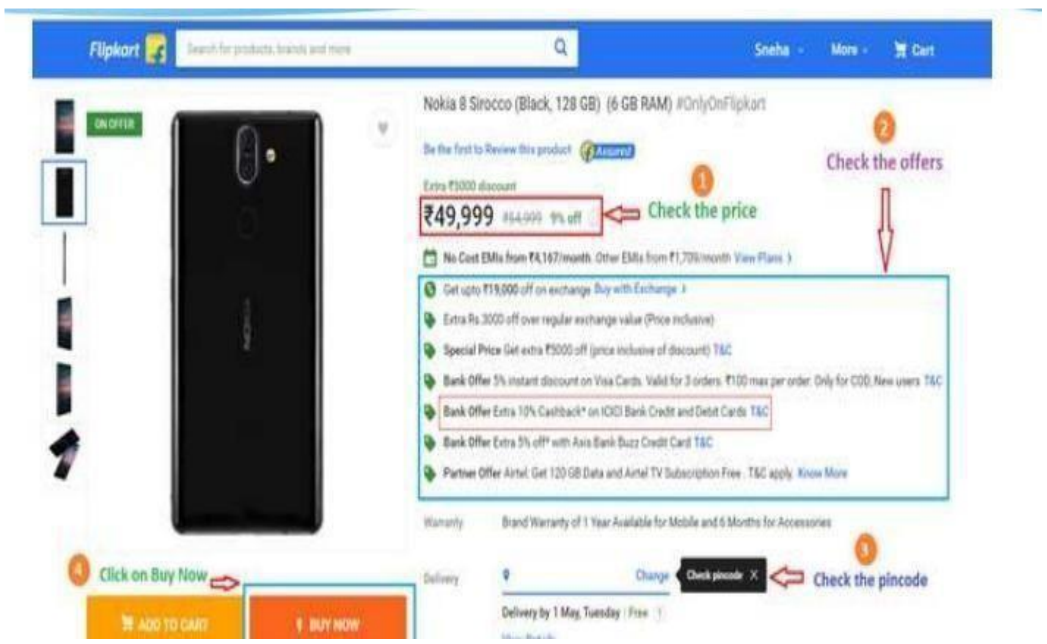
2. Gmail (Receiving mail)



The conversation has been moved to "Primary". Undo
Do this for future messages from 4e465151e982dcd99a004067.12041.list-id.mcsv.net? Yes

ChineseLadyDate	New Relationship, Ready? - She is here waiting for you	
Grammarly	Send better emails - With Grammarly, you can be confident that all your emails are flawless.	
Kathy Ide Book Services	Posts from Kathy Ide Book Services for 01/02/2017 - A weekly update from Kathy Ide View this e-mail in your	Jan 2
service	Did You Know Condoms are Eligible for FSA/HSA - Save 10% Now + Use Your FSA/HSA Benefits Before Th	12/29/16
Pamela Martinez	"Sacred Manna...a place to nourish the soul" - Thank you for receiving updates from SacredManna.com View	12/28/16
Kathy Ide Book Services	Posts from Kathy Ide Book Services for 12/26/2016 - A weekly update from Kathy Ide View this e-mail in your	12/26/16
PayPal	Get a Credit Decision in Seconds. - J Myers- Get rewarded when using the PayPal Extras Mastercard. View in	12/25/16
Google Analytics	Give Everyone on Your Team Access to Powerful Data - Google Analytics - Google Analytics Google+ Twitte	12/21/16
Christian Editor Network.	End-of-Year Special for PEN Membership - End-of-Year Membership Special View this e-mail in your browser I	12/21/16
Kathy Ide Book Services	Posts from Kathy Ide Book Services for 12/19/2016 - A weekly update from Kathy Ide View this e-mail in your	12/19/16
Christian Editor Network.	Registration for Marketing for Editors 101 is now open! - PEN Institute - Marketing for Editors 101 View this	12/16/16
Kathy Ide Book Services	Posts from Kathy Ide Book Services for 12/13/2016 - A weekly update from Kathy Ide View this e-mail in your	12/13/16
service	Green Monday Sale: 20% Off Condoms! - Sale Ends Monday Night! Already know what you want? Order Now	12/12/16

2. Online shopping to buy product (flipkart)



Nokia 8 Sirocco (Black, 128 GB) (6 GB RAM) #OnlyOnFlipkart

Be the first to Review this product

Extra ₹5000 discount

₹49,999 ~~₹54,999~~ 9% off

No Cost EMIs from ₹4,167/month. Other EMIs from ₹1,709/month View Plans

Get upto ₹19,000 off on exchange Buy with Exchange

Extra Rs 3000 off over regular exchange value (Price inclusive)

Special Price Get extra ₹5000 off (price inclusive of discount) T&C

Bank Offer 5% instant discount on Visa Cards. Valid for 3 orders. ₹100 max per order. Only for COD. New users T&C

Bank Offer Extra 10% Cashback* on ICICI Bank Credit and Debit Cards T&C

Bank Offer Extra 5% off* with Axis Bank Buzz Credit Card T&C

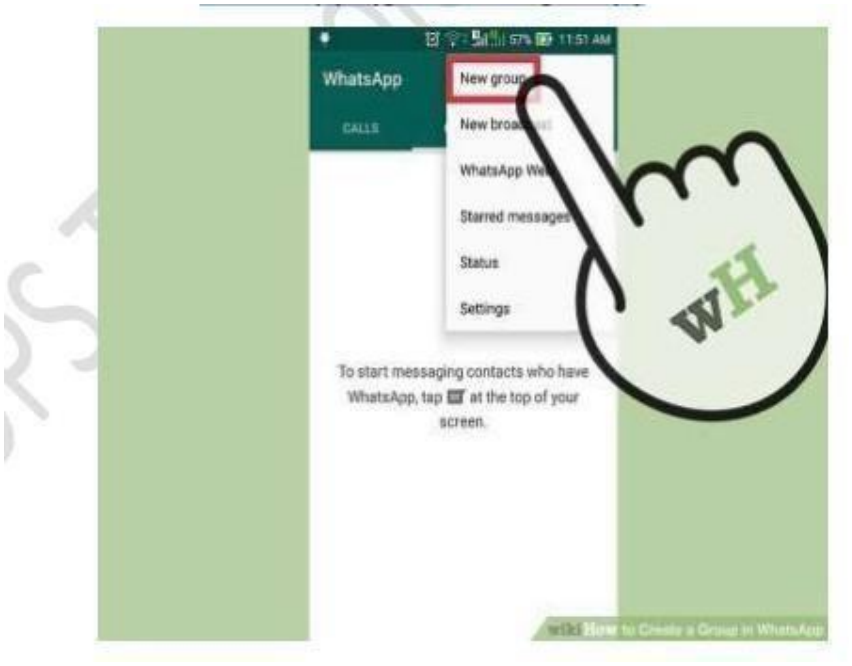
Partner Offer Airtel: Get 120 GB Data and Airtel TV Subscription Free . T&C apply. Know More

Warranty Brand Warranty of 1 Year Available for Mobile and 6 Months for Accessories

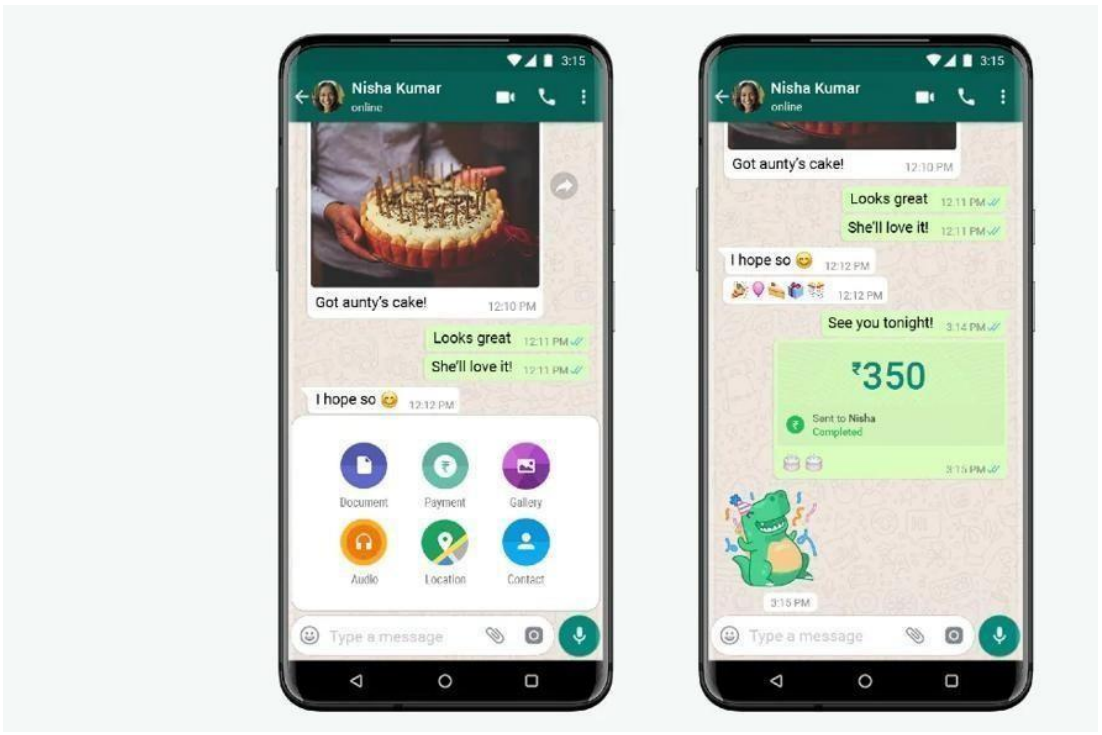
Delivery

Delivery by 1 May, Tuesday | Free

3. **whatsapp Group (generate group)**

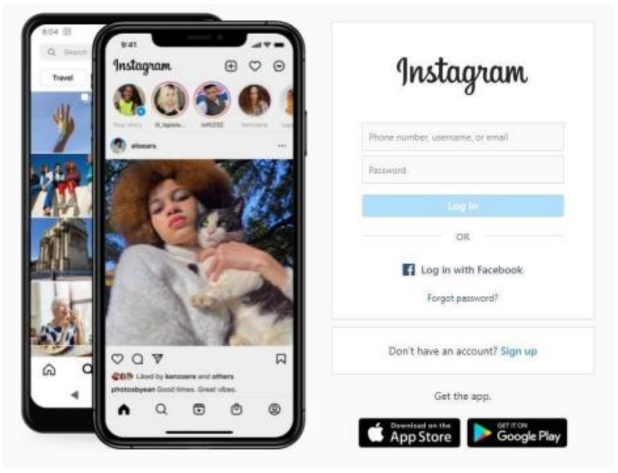

WhatsApp
CALLS
New group
New broa
WhatsApp We
Starred messages
Status
Settings
To start messaging contacts who have WhatsApp, tap at the top of your screen.
wH
How to Create a Group in WhatsApp

4. **Write a Scenario of Whatsapp payment**



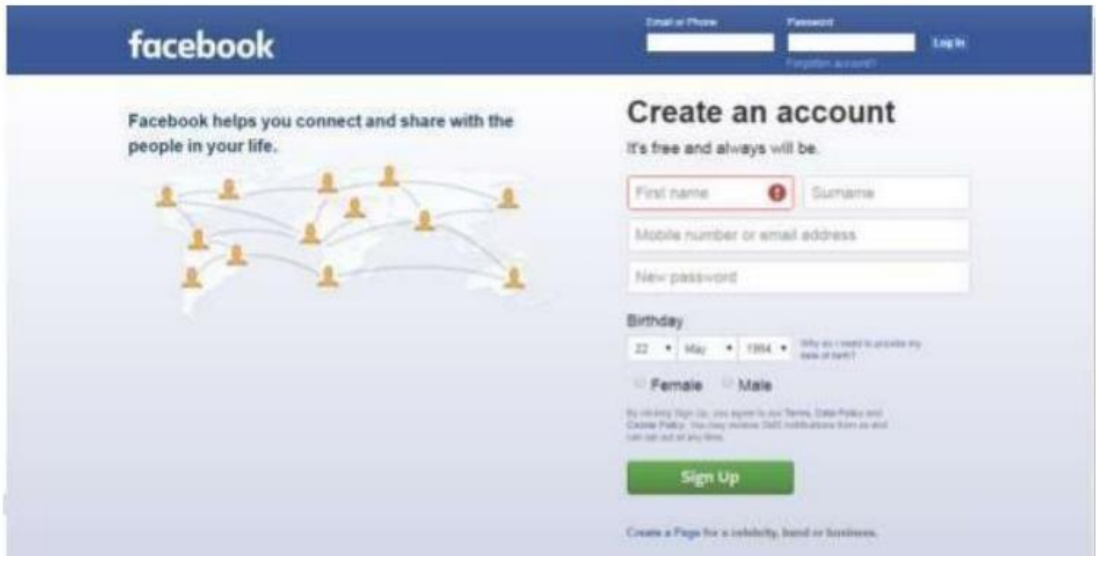
To create HLR & TestCase

1) Instagram only first page



Don't have an account? Sign up

2) Facebook Login Page: https://www.facebook.com/



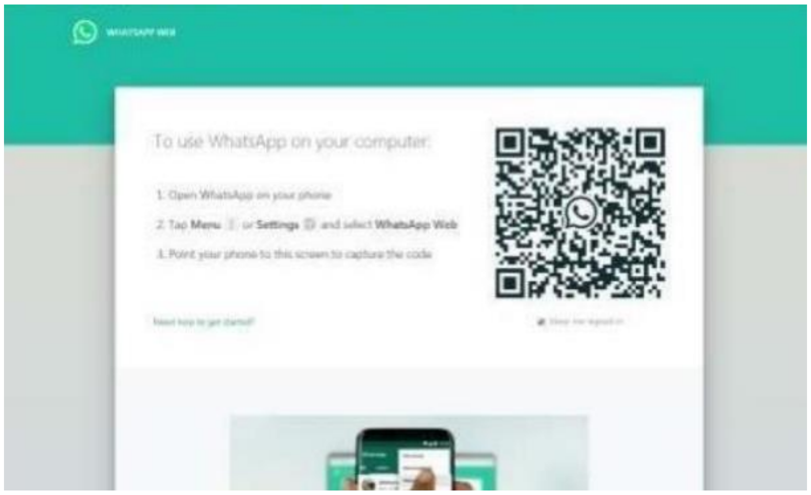
Facebook helps you connect and share with the people in your life.

Create an account

It's free and always will be.

To create HLR & TestCase of WebBased (WhatsApp web , Instagram)

1. WhatsApp Web: https://web.whatsapp.com/



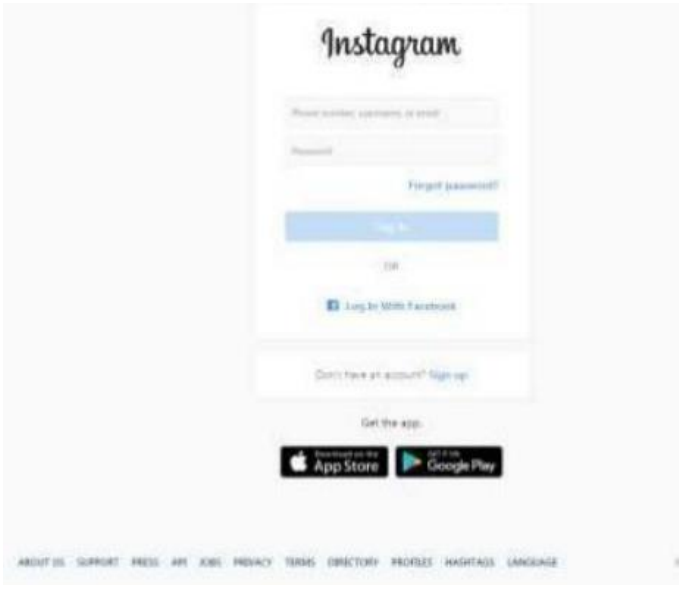
To use WhatsApp on your computer:

1. Open WhatsApp on your phone
2. Tap **Menu** or **Settings** and select **WhatsApp Web**
3. Point your phone to this screen to capture the code

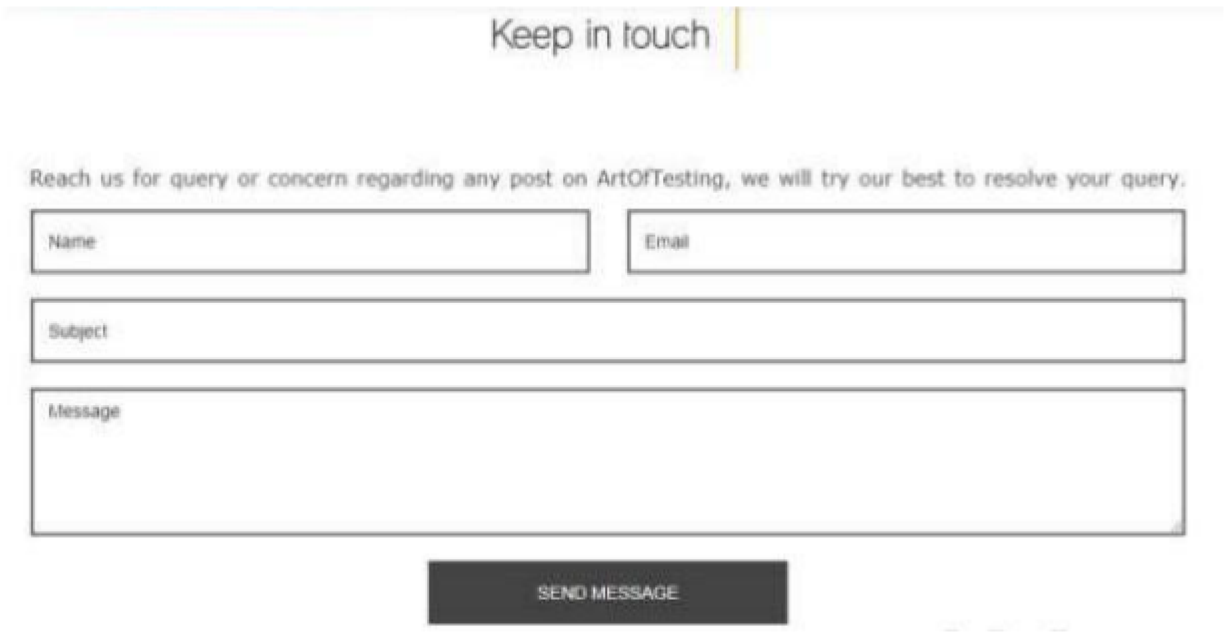
Need help to get started?

Keep me signed in

2. Instagram web: https://www.instagram.com/accounts/login/?hl=en


Instagram
Phone number, username, or email
Password
Forgot password?
Log In
OR
Log In With Facebook
Don't have an account? Sign up
Get the app.
App Store
Google Play
ABOUT US SUPPORT PRESS API JOBS PRIVACY TERMS DIRECTORY PROFILES HASHTAGS LANGUAGE

To create HLR and Test Case on this Link. https://artoftesting.com/


Keep in touch

Reach us for query or concern regarding any post on ArtOfTesting, we will try our best to resolve your query.