Module -2 (Manual Testing)

Q -1 What is traceability matrix:

Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability.
To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

✓ Types of Traceability Matrix

- Forward Traceability
- Backward Traceability
- Bi-Directional Traceability

✓ Pros of Traceability Matrix

- Make obvious to the client that the software is being developed as per the requirements.
- To make sure that all requirements included in the test cases
- To make sure that developers are not creating features that no one has requested
- Easy to identify the missing functionalities.
- If there is a change request for a requirement, then we can easily find out which test cases need to update.
- The completed system may have “Extra” functionality that may have not been specified in the design specification, resulting in wastage of manpower, time and effort.

✓ Cons of Traceability Matrix

- No traceability or Incomplete Traceability Results into:
- Poor or unknown test coverage, more defects found in production
- It will lead to miss some bugs in earlier test cycles which may arise in later test cycles. Then a lot of discussions arguments with other teams and managers before release.
- Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays, etc.

Q-2 What is Boundary value testing:

- Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges
- Boundary value analysis is a method which refines equivalence partitioning.

- Boundary value analysis generates test cases that highlight errors better than equivalence partitioning.
- The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.
- At those points when input values change from valid to invalid errors are most likely to occur.
- Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design

Q- 3 What is Equivalence partitioning testing:

- 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:
    1. Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function.
    2. 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 1 and 100 (inclusive) (e.g value >=1 and value <=100) Then...”
- EP says that by testing just one value we have tested the partition (typically a-point value is used). It assumes that:
- If one value finds a bug, the others probably will too
- If one doesn't find a bug, the others probably won't either
- In EP we must identify Valid Equivalence partitions and Invalid
- Equivalence partitions where applicable (typically in range tests)
- The Valid partition is bounded by the values 1 and 100
- Plus there are 2 Invalid partitions

Q-4 What is Integration testing?

- Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems
- Integration Testing is a level of the software testing process where individual units are combined and tested as a group.
- The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

- Integration testing is done by a specific integration tester or test team.
- There are 2 levels of Integration Testing
  1. Component Integration Testing
  2. System Integration Testing

Need of Integration Testing;

- A Module in general is designed by an individual software developer who understanding and programming logic may differ from other programmers. Integration testing becomes necessary to verify the software modules work in unity.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.

❖ Component Integration Testing;

- Component Integration Testing: Testing performed to expose defects in the interfaces and interaction between integrated components
- **Functional Testing using** Black Box Testing techniques against the interfacing requirements for the component under test
- **Non-functional Testing** (where appropriate, for performance or reliability testing of the component interfaces, for example)

❖ System Integration Testing:

- It tests the interactions between different systems and may be done after system testing.
- It verifies the proper execution of software components and proper interfacing between components within the solution.
- As testing for dependencies between different components is a primary function of SIT Testing, this area is often most subject to Regression Testing.

❖ Integration Testing Methods;

- Any of Black Box Testing, White Box Testing, and Gray Box Testing methods can be used. Normally, the method depends on your definition of 'unit'.
- There is two types methods of Integration Testing:
  Bing Bang Integration Testing
  Incremental Integration Testing
  Top Down Approach
  Bottom Up Approach

When is Integration Testing Performed?

Integration Testing is performed after Unit Testing and before System Testing.

Who performs Integration Testing?

Either Developers themselves or independent Testers perform Integration Testing.

- ❖ Big Bang Integration Testing
  - 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.

- ❖ Top Down Approach
  - Testing takes place from top to bottom, following the control flow or architectural structure (e.g. starting from the GUI or main menu). Components or systems are substituted by stubs.
  - In Top to down approach, testing takes place from top to down following the control flow of the software system.
  - Takes help of stubs for testing.

- ❖ Bottom Up Approach
  - Testing takes place from the bottom of the control flow upwards. Components or systems are substituted by drivers.
  - In the bottom up strategy, each module at lower levels is tested with higher modules until all modules are tested. It takes help of Drivers for testing

- ❖ Continuous Integration Approach
  - Continuous Integration is a software development method where team members integrate their work at least once a day.
  - In this method, every integration is checked by an automated build to detect errors.
  - In Continuous Integration after a code commit, the software is built and tested immediately.
  - In a large project with many developers, commits are made many times during a day. With each commit code is built and tested.
  - This commit, build, test, and deploy is a continuous process, and hence the name continuous integration/deployment.

- ❖ Entry and Exit Criteria

Entry Criteria:

- Unit Tested Components/Modules
- All High prioritized bugs fixed and closed
- All Modules to be code completed and integrated successfully.
- Integration test Plan, test case, scenarios to be signed off and documented.
- Required Test Environment to be set up for Integration testing

Exit Criteria:

- Successful Testing of Integrated Application.
- Executed Test Cases are documented
- All High prioritized bugs fixed and closed
- Technical documents to be submitted followed by release Notes.

Limitations

- Any condition not specified in integration tests, apart from the confirmation of the execution of the design items is usually not tested.

Q-5 What is component testing:

- Component (Unit) – A minimal software item that can be tested in isolation. It means "A unit is the smallest testable part of software."
- Component Testing – The testing of individual software components.
- Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.
- Unit testing is the first level of testing and is performed prior to Integration Testing.
- Component can be tested in isolation – stubs/drivers may be employed
- Functional and Non-Functional testing.

Unit testing is performed by using the White Box Testing method**.**

Test Approach:
Test-First/Test-Driven approach – create the tests to drive the design and code construction.

Below we look at some of what extreme programming brings to the world of unit testing:

1. Tests are written before the code
2. Rely heavily on testing frameworks
3. All classes in the applications are tested
4. Quick and easy integration is made possible

Q-6 What is functional system testing?

- Functional System Testing : A requirement that specifies a function that a system or system component must perform
- A Requirement may exist as a text document and/or a model

There are two types of Test Approach

- Requirement Based Functional Testing
- Process Based Testing

Functional System Testing Functionality As below:

- Accuracy - Provision of right or agreed results or effects
- Interoperability - Ability to interact with specified systems
- Compliance - Adhere to applicable standards, conventions, regulations or laws
- Auditability - Ability to provide adequate and accurate audit data
- Suitability - Presence and appropriateness of functions for specified tasks

Requirement Based Testing:

- Testing against requirements and specifications
- Starts by using the most appropriate black-box testing technique
- May support this with white-box techniques (e.g. menu structures, web page navigation)
- Risk based approach

Business Process Based Testing:

- Test procedures and cases derived from:
- Testing should reflect the business environment and processes in which the system will operate.
- Therefore, test cases should be based on real business processes.

Q -7 What is Non-Functional Testing?

- It is the testing of "**how**" the system works. Non-functional testing may be performed at all test levels.
- The term non-functional testing describes the tests required to measure characteristics of systems and software that can be quantified on a varying scale, such as response times for performance testing.
- Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

Example:

Web Based Testing :

- Identify the software processes that directly influence the overall performance of the system.
- In website number of user/customer will increase , how the website will handled to every customer/user.
- Desktop Based Testing :
- Numerous other such GUI test cases, the desktop application tester must view
- Guarantee that error messages are instructive and helpful for the client
- Memory, and different other issues

Mobile Based Testing :

- In mobile , automatically will switch off without any reason.
- To stop the application which is not in our hand.

Game Based Testing :

- Confirms workability and stability of the software.
- Validate whether the user interface of the app is as per the screen size of the device and ensure high quality

Q -9 What is Adhoc testing?

- 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.
- In fact is does not create test cases altogether
- Testers randomly test the application without any test cases or any business requirement document.
- 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 system to "guess" the most likely source of errors.

The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.
Using experience to postulate errors.
Use Error Guessing to Complement Test Design Techniques

Types of Adhoc Testing;

1. Buddy Testing
2. Pair testing
3. Monkey Testing

Q -10 What is Exploratory Testing?

- Test design, execution and logging happen simultaneously
- Testing is often not recorded
- Makes use of experience, heuristics and test patterns
- Testing is based on a test charter that may include
- Scope of the testing (in and out)
- The focus of exploratory testing is more on testing as a "thinking" activity.
- A brief description of how tests will be performed
- Expected problems
- Is carried out in time boxed intervals
- More structured than Error guessing

Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits.

- Is not random testing but it is Adhoc testing with purpose of find bugs
- Is structured and rigorous
- Is cognitively (thinking) structured as compared to procedural structure of scripted testing. This structure comes from Charter, time boxing etc.
- Is highly teachable and manageable
- Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently.

Q – 11 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.
- Structure-based testing technique is also known as 'white-box' or 'glass-box' testing technique because here the testers require knowledge of how the software is implemented, how it works.
- White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.
- Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.

Types:

1. Statement coverage
2. Decision coverage
3. Condition coverage

Q – 12 What are 7 key principles? Explain in detail?

1. Testing shows presence of Defects
2. Exhaustive Testing is Impossible!
3. Early Testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent
7. Absence of Errors Fallacy

1. Testing shows presence of Defects:
    - Testing can show that defects are present, but cannot prove that there are no defects.
    - Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.\
    - We test to find Faults
    - However Testing cannot prove that there are no defects present

2. Exhaustive Testing is Impossible!

- Testing everything including all combinations of inputs and preconditions is not possible.
- So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
- This is very unlikely that the project timescales would allow for this number of tests.
- We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions).
- That is we must Priorities our testing effort using a Risk Based Approach.

3. Early Testing

- Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
- Remember from our Definition of Testing, that Testing doesn't start
- Once the code has been written!

4. Defect Clustering

- A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
- They are 'clustered'
- In other words, most defects found during testing are usually confined to a small number of modules
- An important consideration in test prioritization!

5. Pesticide Paradox

- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
- To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
- Testing identifies bugs, and programmers respond to fix them
- As bugs are eliminated by the programmers, the software improves

6. Testing is Context Dependent

- Testing is basically context dependent.
- Testing is done differently in different contexts
- Different kinds of sites are tested differently.
- Safety – critical software is tested differently from an e-commerce site.

7. Absence of Errors Fallacy

- If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.
- If we build a system and, in doing so, find and fix defects

- It doesn't make it a good system

Q – 13 What determines the level of risk?

- 'A factor that could result in future negative consequences; usually expressed as impact and likelihood'
- When testing does find defects, the Quality of the software system increases when those defects are fixed
- Testing should be integrated as one of the Quality assurance activities
- A Risk could be any future event with a negative consequence You

Risks are of two types

- Project Risks
- Product Risk

Example of **Project risk** is Senior Team Member leaving the project abruptly
Example of **product risks** would be Flight Reservation system not installing in test environment

Q- 14 What is GUI Testing?

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 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
  - Record and replay
  - Model based testing

Q-15 Mention what big bang testing is?

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

Error:

- What is Error, Defect, Bug and failure?
- A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.

Defect:

- Error found by tester is called defect
- Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

Bug:

- Defect accepted by development team then it is called bug
- A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.

Failure:

- Build does not meet the requirements then it is
- The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

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

- Black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.
- The testers have no knowledge of how the system or component is structured inside the box.
- Specification-based testing technique is also known as 'black-box' or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
- The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.

Techniques of Black Box Testing:

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition testing
- Use-case Testing
- Other Black Box Testing
- Syntax or Pattern Testing

1. Equivalence Partitioning (E.P.)

- 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
- If we want to test the following IF statement: "If value is between 1 and 100 (inclusive) (e.g value >=1 and value <=100) Then..."
- In EP we must identify Valid Equivalence partitions and Invalid Equivalence partitions where applicable (typically in range tests) The Valid partition is bounded by the values 1 and 100
- Plus there are 2 Invalid partitions

2. Boundary Value Analysis(B.V.A.)

- Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges
- Boundary value analysis is a method which refines equivalence partitioning.
- Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design

3. Decision Table

- The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs.

- However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface.
- The other two specification-based software testing techniques, decision tables and state transition testing are more focused on business logic or business rules.
- A decision table is a good way to deal with combinations of things (e.g. inputs).
- Inputs are usually defined in terms of actions which are Boolean.(true or false)

4. State Transaction Testing

- State transition testing is used where some aspect of the system can be described in what is called a 'finite state machine'. This simply means that the system can be in a (finite) number of different states, and the transitions from one state to another are determined by the rules of the 'machine'. This is the model on which the system and the tests are based.
- One of the advantages of the state transition technique is that the model can be as detailed or as abstract as you need it to be.
- Where a part of the system is more important (that is, requires more testing) a greater depth of detail can be modeled.
- Where the system is less important (requires less testing), the model can use a single state to signify what would otherwise be a series of different states.

Q-18 What is the purpose of exit criteria?

- How do we know when to stop testing?
  - Run out of time?
  - Run out of budget?
  - The business tells you it went live last night!
  - Boss says stop?
  - All defects have been fixed?
- Purpose of exit criteria is to define when we STOP testing either at the:
  - End of all testing – i.e. product Go Live
  - End of phase of testing (e.g. hand over from System Test to UAT)
- Exit Criteria typically measures:
  - Thoroughness measures, such as coverage of requirements or of code or risk coverage
  - Estimates of defect density or reliability measures. (e.g. how many defects open by category)
  - Cost.
  - Residual Risks, such as defects not fixed or lack of test coverage in certain areas.
  - Schedules - such as those based on time to market.

Q-19 Difference between QA v/s QC v/s Tester

QA	QC	Testing
Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	Activities which ensure the identification of bugs/error/defects in the Software.
Process oriented activities.	Process 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

Q – 20 Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.	Sanity Testing is done to check the new functionality / bugs have been fixed.
The objective of this testing is to verify "stability" of the system in order to proceed with more rigorous testing.	The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing.
This testing is performed by the developers or testers.	Sanity testing is usually performed by testers.

Smoke testing is a subset of Acceptance testing.	Sanity testing is a subset of Regression testing.
Smoke testing is usually documented or scripted.	Sanity testing is usually not documented and is unscripted.

Q-21 Difference between verification and Validation

Verification	Validation
The verifying process includes checking documents, design, code, and program.	It is a dynamic mechanism of testing and validating the actual product.
It finds bugs early in the development cycle.	It can find bugs that the verification process cannot catch.
Are we building the product right?	Are we building the right product?
· Reviews · Walkthroughs · Inspections	Testing
It comes before validation.	It comes after verification.

Q-22 Explain the difference between Functional testing and Non Functional testing

Functional Testing	Non-functional Testing
It is based on customer's requirements.	It focuses on customer's expectation.
Functional testing is executed first.	Nonfunctional testing should be performed after functional testing.
Manual testing or automation tools can be used for functional testing.	Using tools will be effective for this testing.
Functional testing describes what the product does.	Nonfunctional testing describes how good the product works.
Business requirements are the inputs to functional testing.	Performance parameters like speed , scalability are inputs to non-functional 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	Types of Nonfunctional testing are · Performance Testing · Load Testing · Volume Testing · Stress Testing · Security Testing · Installation Testing · Penetration Testing · Compatibility Testing · Migration Testing

Q-23 What is alpha testing?

- 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-24 What is Beta testing?

- It is always performed by the customers at their own site.
- It is not performed by Independent Testing Team.
- Beta Testing is always open to the market and public.
- It is usually conducted for software product.
- It is performed in **Real Time Environment**.
- It is always performed outside the organization.
- It is also the form of Acceptance Testing.
- 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.
- It is only a kind of Black Box Testing.

Q-25 Difference between Priority and Severity.

Severity	Priority
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.
The value of severity is objective	The value of priority is subjective
The testing engineer basically decides a defect's severity level.	The product manager basically decides a defect's priority level.
5 Types 1. Cosmetic 2. Minor 3. Moderate 4. Major 5. Critical	3 Types 1. Priority 2. High Medium 3. Low

Q-26 What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

STLC (Software Testing Life Cycle)	SDLC (Software Development Life Cycle)
STLC is generally related to software testing.	SDLC is generally related to software development.
STLC focuses only on testing the software.	Including the development, it focuses on other phases like testing are also included.
STLC focuses only on testing the software.	SDLC consist of a total of six steps or phases.
In this, less number of members (testers) are needed for completing the whole process.	In this more members (developers) are required for completing the whole process.
STLC helps in making the software defects/error-free	SDLC helps in developing good quality software.

Q-26 What is the difference between test scenarios, test cases, and test script?

Test Scenario	Test Cases	Test script
Is any functionality that can be tested	Is a set of functional executed to verify particular features or functionality	Is a set of instruction to test an app automatically
Is derived from test artifacts like Business Requirement of Specification(BRS) and Software Requirements Specification	Is mostly derived from test scenario	Is mostly derived from test cases
Is more focused on what to test	Is more focused on what to test and how to test	Is more focused on expected results

Q-27 When "Regression Testing" should be performed?

- when the system is stable and the system or the environment changes
- when testing bug-fix releases as part of the maintenance phase
- It should be applied at all Test Levels
- It should be considered complete when agreed completion criteria for regression testing have been met
- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation

Q-28 Explain what Test Plan is? What is the information that should be covered?

➢ A Test Plan is a detailed document that catalogs the test strategies, objectives, schedule, estimations, deadlines, and resources required to complete that project

- Test Strategy and Objectives. Identify the main purpose of testing (in light of the product requirements) and what a successful completion of a testing cycle looks like. ...
- Schedule, Estimation, and Deliverables. ...
- Resources Required Performing Testing.

Q-29 Advantage of Bugzilla

- Open source, free bug tracking tool.
- Automatic Duplicate Bug Detection.
- Search option with advanced features.
- File/Modify Bugs By Email.
- Move Bugs Between Installs.

- Multiple Authentication Methods (LDAP, Apache server).
- Time Tracking.
- Automated bug reporting; has an API to interact with system.

Q-30 Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Authorization	Authentication
Authorization is the process of giving permission to access the resources.	Authentication is the process of identifying a user to provide access to a system.
In this, it is verified that if the user is allowed through the defined policies and rules.	In this, the user or client and server are verified.
It is usually done once the user is successfully authenticated.	It is usually performed before the authorization.
It requires the user's privilege or security level.	It requires the login details of the user, such as user name & password, etc.

- Integration. Integration testing exposes problems with interfaces among different program components before deployment. ...
- Interoperability. ...
- Security. ...
- Performance. ...
- Usability. ...
- Quality Testing, Exceptional Services.

➢ Write a scenario of only Whatsapp chat messages

1. Verify that on downloading the Whatsapp application, users can register using a new mobile number.
2. Verify that for a new mobile number user will get a verification code on his mobile and filling in the same verifies the new user account.
3. Check the maximum number of incorrect attempts allowed while filling out the verification code.
4. Verify that registering an existing mobile number for new user account registration is not allowed.
5. Verify that on successful registration all the contacts in the user’s contact directory get imported to the Whatsapp contact list.
6. Verify that the user can send messages to any individual selected from his contact list.
7. Verify that ‘Chats’ window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated.
8. Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person.
9. Verify that the user can check the message delivered and read the time for a message in the ‘Message Info’ section.
10. Verify that the user can share or receive contact with the other person.
11. Verify that the user can create a group by adding multiple people from his contact list.
12. Verify that the user can send and receive the message in group chats.
13. Verify that users can send and receive images, audio, video, and emoticons in the chat with individuals.
14. Verify that users can send and receive images, audio, video, and emoticons in group chats.
15. Verify that the user can send and receive chats in the secondary languages available.
16. Verify that users can delete text, images, audio, and video messages within a chat.
17. Verify that users can clear their complete chat history in an individual or group chat.
18. Verify that users can archive chats in an individual or group chat.
19. Verify that users can block a user to prevent any message from getting received from the blocked contact.
20. Verify that the user makes WhatsApp calls to the person in his contact list.
21. Verify that the user can receive WhatsApp calls from the person in his contact list.
22. Verify that users can mark chats as favorites and access all chats marked as favorites from the ‘Favorites’ section.

➢ Write a Scenario of Pen

1. Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen.
2. Verify that the user is able to write clearly over different types of papers.
3. Check the weight of the pen. It should be as per the specifications. In case not mentioned in the specifications, the weight should not be too heavy to impact its smooth operation.

4. Verify if the pen is with a cap or without a cap.
5. Verify the color of the ink on the pen.
6. Check the odor of the pen’s ink on writing over a surface.
7. Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc.
8. Verify that the text written by the pen should have consistent ink flow without leaving any blob.
9. Check that the pen’s ink should not leak in case it is tilted upside down.
10. Verify if the pen’s ink should not leak at higher altitudes.
11. Verify if the text written by the pen is erasable or not.
12. Check the functioning of the pen by applying normal pressure during writing.
13. Verify the strength of the pen’s outer body. It should not be easily breakable.
14. Verify that text written by pen should not get faded before a certain time as mentioned in the specification.
15. Check if the text written by the pen is waterproof or not.
16. Verify that the user is able to write normally by tilting the pen at a certain angle instead of keeping it straight while writing.
17. Check the grip of the pen, and whether it provides adequate friction for the user to comfortably grip the pen.
18. Verify if the pen can support multiple refills or not.
19. In the case of an ink pen, verify that the user is able to refill the pen with all the supported ink types.
20. For ink pens, verify that the mechanism to refill the pen is easy to operate.
21. In the case of a ballpoint pen, verify the size of the tip.
22. In the case of a ball and gel pen, verify that the user can change the refill of the pen easily.

➢ Write a Scenario of Door

1. Verify if the door is single door or bi-folded door
2. Check if the door opens inwards or outwards
3. Verify that the dimension of the doors are as per the specifications
4. Verify that the material used in the door body and its parts is as per the specifications
5. Verify that color of the door is as specified
6. Verify if the door is sliding door or rotating door
7. Check the position, quality and strength of hinges
8. Check the type of locks in the door
9. Check the number of locks in the door interior side or exterior side
10. Verify if the door is having peek-hole or not
11. Verify if the door is having stopper or not
12. Verify if the door closes automatically or not – spring mechanism
13. Verify if the door makes noise when opened or closed
14. Check the door condition when used extensively with water
15. Check the door condition in different climatic conditions- temperature, humidity etc
16. Check the amount of force- pull or push required to open or close the door

➢ Write a scenario of ATM

1. Verify the type of ATM machine, if it has a touch screen, both keypad buttons only, or both.
2. Verify that on properly inserting a valid card different banking options appear on the screen.
3. Check that no option to continue and enter credentials is displayed to the user when the card is inserted incorrectly.
4. Verify that the touch of the ATM screen is smooth and operational.
5. Verify that the user is presented with the option to choose a language for further operations.
6. Check that the user is asked to enter a pin number before displaying any card/bank account detail.
7. Verify that there is a limited number of attempts up to which the user is allowed to enter the pin code.
8. Verify that if the total number of incorrect pin attempts gets surpassed then the user is not allowed to continue further. And operations like temporary blocking of the card, etc get initiated.
9. Check that the pin is displayed in masked form when entered.
10. Verify that the user is presented with different account type options like- saving, current, etc.
11. Verify that the user is allowed to get account details like available balance.
12. Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal.
13. Verify that the user is only allowed to enter the amount in multiple denominations as per the specifications.
14. Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured.
15. Check that the user cannot withdraw more amount than the total available balance and a proper message should be displayed.
16. Verify that the user is provided the option to get the transaction details in printed form.
17. Verify that the user's session timeout is maintained.
18. Check that the user is not allowed to exceed one transaction limit amount.
19. Verify that the user is not allowed to exceed the one-day transaction limit amount.
20. Verify that the user is allowed to do only one transaction per pin request.
21. Check that in case the ATM machine runs out of money, a proper message is displayed to the user.
22. Verify that the applicable fee gets deducted along with the withdrawn amount in case the user exceeds the limit of the number of free transactions in a month.
23. Verify that the applicable fee gets deducted along with the withdrawn amount in case the user uses a card of a bank other than that of an ATM.
24. Check that the user is not allowed to proceed with the expired ATM card and that a proper error message gets displayed.
25. Verify that in case of sudden electricity loss before withdrawing cash, the transaction is marked as null and the amount is not withdrawn from the user's account.

➢ Write a scenario of Microwave Owen

1. Verify that the dimensions of the oven are as per the specification provided.
2. Verify that the oven’s material is optimal for its use as an oven and as per the specification.
3. Verify that the oven heats the food at the desired temperature properly.
4. Verify that oven heats food at the desired temperature within a specified time duration.
5. Verify the ovens functioning with maximum attainable temperature.
6. Verify the ovens functioning with minimum attainable temperature.
7. Verify that the oven’s plate rotation is speed is optimal and not too high to spill the food kept over it.
8. Verify that the oven’s door gets closed properly.
9. Verify that the oven’s door opens smoothly.
10. Verify the battery requirement of the microwave oven and check that it function’s smoothly at that power.
11. Verify that the text written over the oven’s body is clearly readable.
12. Verify that the digital display is clearly visible and functions correctly.
13. Verify that the temperature regulator is smooth to operate.
14. Verify that the temperature regulator works correctly.
15. Check the maximum capacity of the oven and test its functioning with that volume of food.
16. Check oven’s functionality with different kinds of food – solid, liquid.
17. Check the oven’s functionality with different food at different temperatures.
18. Verify the oven’s functionality with different kinds of container material.
19. Verify that the power cord of the oven is long enough.
20. Verify that the usage instruction or user manuals have clear instructions.

➢ Write a scenario of Coffee vending Machine

1. Verify that outer body, as well as inner part’s material, is as per the specification
2. Verify that the machine’s body color as well brand is correctly visible and as per specification
3. Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc
4. Verify that the quantity of hot water, milk, coffee powder per serving is correct
5. Verify the power/voltage requirements of the machine
6. 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.
7. Verify that coffee should not leak when not in operation
8. Verify the amount of coffee served in single-serving is as per specification
9. Verify that the digital display displays correct information
10. Check if the machine can be switched on and off using the power buttons
11. Check for the indicator lights when the machine is switched on-off
12. Verify that the functioning of all the buttons work properly when pressed
13. Verify that each button has an image/text with it, indicating the task it performs

14. Verify that complete quantity of coffee should get poured in a single operation, no residual coffee should be present in the nozzle
15. Verify the mechanism to clean the system work correctly- foamer
16. Verify that the coffee served has the same and correct temperature each time it is served by the machine
17. Verify that system should display an error when it runs out of ingredients
18. Verify that pressing the coffee button multiple times leads to multiple serving of coffee
19. Verify that there is the passage for residual/extra coffee in the machine
20. Verify that machine should work correctly in different climatic, moistures and temperature conditions
21. Verify that machine should not make too much sound when in operation
22. Performance test – Check the amount of time the machine takes to serve a single serving of coffee
23. Performance test – Check the performance of the machine when used continuously until the ingredients run out of the requirements
24. Negative Test – Check the functioning of the coffee machine when two/multiple buttons are pressed simultaneously
25. Negative Test – Check the functioning of coffee machine with a lesser or higher voltage than required
26. Negative Test – Check the functioning of the coffee machine if the ingredient container's capacity is exceeded.

When to used Usablity Testing?

If possible, usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you've begun the strategy work around a brand new site or app. This will quickly identify areas for opportunity, and reduce the amount of assumptions your design team will make with regard to what the user wants. Additionally, after the usability tests analysis, the team should have the ability to pinpoint the steps needed to achieve the project goals with as little disruption as possible.

Once you've gotten results from an initial usability test, it's then important to use those results throughout your design phase and keep re-testing users. We typically design at the wireframe level first, followed by the high-fidelity final designs. At both of those stages, we create clickable prototypes using In Vision, which allow us to perform user tests and continue to optimize the design and usability of the site.

What is the procedure for GUI Testing?

- Testing the size, position, height, width of the visual elements

- Verifying and testing the error messages are displayed or not
- Testing different sections of the display screen
- Verifying the usability of carousel arrows
- Checking the navigation elements at the top of the page
- Checking the message displayed, frequency and content
- Verifying the functionality of proper filters and ability to retrieve results.
- Checking alignment of radio buttons, drop downs
- Verifying the title of each section and their correctness
- Cross-checking the colors and its synchronization with the theme

➢ Write a scenario of chair

1. Verify that the chair is stable enough to take an average human load
2. Check the material used in making the chair-wood, plastic etc
3. Check if the chair’s leg are level to the floor
4. Check the usability of the chair as an office chair, normal household chair
5. Check if there is back support in the chair
6. Check if there is support for hands in the chair
7. Verify the paint’s type and color
8. Verify if the chair’s material is brittle or not
9. Check if cushion is provided with chair or not
10. Check the condition when washed with water or effect of water on chair
11. Verify that the dimension of chair is as per the specifications
12. Verify that the weight of the chair is as per the specifications
13. Check the height of the chair’s seat from floor

➢ Write a Scenario of Wrist Watch

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

10. Check if the watch is having a date and day display or not.
11. Verify the color of the text displayed in the watch – time, day, date, and other information.
12. 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.
13. Check if the second hand of the watch makes ticking sound or not.
14. Verify if the brand of the watch and check if its visible in the dial.
15. Check if the clock is having stopwatch, timers, and alarm functionality or not.
16. In the case of a digital watch, verify the format of the watch 12 hours or 24 hours.
17. Verify if the watch comes with any guarantee or warranty.
18. Verify if the dial has glass covering or plastic, check if the material is breakable or not.
19. Verify if the dial’s glass/plastic is resistant to minor scratches or not.
20. Check the battery requirement of the watch.

➢ **Write a Scenario of Lift(Elevator)**

1. Verify the dimensions of the lift
2. Verify the type of door of the lift is as per the specification
3. Verify the type of metal used in the lift interior and exterior
4. Verify the capacity of the lift in terms of the total weight
5. Verify the buttons in the lift to close and open the door and numbers as per the number of floors
6. Verify that lift moves to the particular floor as the button of the floor is clicked
7. Verify that lift stops when up/down buttons at particular floor are pressed
8. Verify if there is an emergency button to contact officials in case of any mishap
9. Verify the performance of the floor – the time is taken to go to a floor
10. Verify that in case of power failure, lift doesn’t free-fall and get halted in the particular floor
11. Verify lifts working in case button to open the door is pressed before reaching the destination floor
12. Verify that in case door is about to close and an object is placed between the doors if the doors sense the object and again open or not
13. Verify the time duration for which door remain open by default
14. Verify if lift interior is having proper air ventilation
15. Verify lighting in the lift
16. Verify that at no point lifts door should open while in motion
17. Verify that in case of power loss, there should be a backup mechanism to safely get into a floor or a backup power supply
18. Verify that in case multiple floor number button is clicked, lift should stop at each floor
19. Verify that in case of capacity limit is reached users are prompted with warning alert-audio/visual
20. Verify that inside lift user are prompted with current floor and direction information the lift is moving towards- audio/visual prompt

- **To Create Scenario (Positive & Negative) Face book chat on mobile**

1. Verify that the massager app install in mobile
2. Verify that the successfully login on messenger
3. Check the received messages count should be displayed on the Facebook Messages icon
4. Check the user gets all received messages in his inbox
5. Check that only message contact will display on the left hand side of the message box
6. Check the Active users display with a green got in the message box
7. Check the unread messages are highlighted so that the user can identify it
8. Check the user can send or recived message from massager app
9. Check the user can search contact in the message box
10. Check the user can delete the message or not
11. Check the user can show the all contact profile pic in massager app
12. Check the user can send or received text, picture ,documents, videos, audio so on
13. Check the user can call audio or video
14. Check the user should get message after uploading an images or file of an unsupported type
15. Check that the user is able to send messages to other offline users
16. Check the user can open more account in massager app
17. Check the user should successfully logout or not

- **To create scenario Gmail (receiving mail)**

1. Check that the user can receive email are correctly displayed or not
2. Check that the recently received unread emails is highlighted and bold in the Inbox section
3. Check the user can get the notification of receiving mail
4. Check the user should open the receiving mail properly or not
5. Check the user can get the information ID who receving mail
6. Check the attached documents of the email are download or not
7. Check the already read emails should not be the highlight
8. The number of unread email counts should be displayed beside the inbox test box
9. Check if the count is increased asper the number of new emails as unread
10. Check the name are visible to all the user whose names are present in CC and To section
11. Check those name and emails are present in the BCC section and should not display to other
12. Check that the you can receive emails from other domains like yahoo, skype, twiter

- **To create scenario Online shopping to buy product (flipkart)**

1. Verify that user can search easily as per your choice

2. Verify that user get proper product displayed which they search on the flipcart
3. Verify that the user can all information about the product
4. Verify that the user can easily add or not in add to cart
5. Verify that the user can show the all product which they add in to the card with details
6. Verify that the user can easily buy the one product from cart
7. Verify that the user should logon successfully
8. Verify that the user get more option for payment
9. Verify that the user get product order details
10. Verify that the user get the product into the delivery time
11. Check the return policy is available or not for the product
12. Verify that user get product on right address
13. Verify that the user can see the review of the product

- **Write a Scenario of whatsapp Group (generate group)**

1. Check whether the user can create a new one or note
2. Check the user can add multiple contacts from the contact list
3. Verify the user can insert the group name and select an image for DP
4. Check the user can add and remove contacts from the group
5. Check the user is able to delete group
6. Check the user can send and receive text message in the group
7. Check the user can send and documents in the group chat box
8. Check the user can send and receive photos in the group chat box
9. Check the user can send and receive videos in the group chat box
10. Check the user can send and receive emotion icons in the group chat box
11. Check the user can send and receive contact
12. Check the user can send and receive the live location
13. Check the user can send and receive GIFs
14. Check the user can delete text, video, audio, location and documents
15. Check the user can send the recorded voice
16. Check the user is able to invite or add multiple video call
17. Verify the user can see the group contact information from group info in the chat box
18. Check the user is able to search specific chat history using the search option in the group chat box
19. Check the user is able to mute the group in the group chat box
20. Check the users have options like Report, block, clear chat, export chat and add shortcut
21. Check the how many user get authority of add other person
22. Check the minimum user get admin authority.

➢ Write a Scenario of instagram (video call with chat)

1. Verify that the Instagram application is install or not
2. Verify that the camera should be available on the mobile phone
3. Verify that the Internet connection should be on both person
4. Check the voice clarity during videocall
5. Verify that the user can send the massage when video call is on
6. Verify that the picture clarity during the video call
7. Verify that the front camera and top camera available or not
8. Verify that the video call with microphone is working or not
9. Verify that the group video call is available or not
10. Verify that the user can recorded video call or not
11. Verify that the user can send video during the videocall
12. If you tab the message when the video call is on, the video call stops
13. Verify that use can invite people during the video call

➢ Write a Scenario of Whatsapp payment

1. Verify the whatsapp open properly or not
2. Verify the payment option should be open or when we go to right side on whatsapp and tick on the three dots
3. Verify the show the payment page
4. Verify the security protect functionality should be work properly or not
5. Verify the add payment method functionality should open
6. Check the all bank name show into the page
7. Verify that the we can select the multiple bank in a whatsapp payment
8. Check the verification should properly or we get the otp in verify the mobile number
9. Verify that the we can see the transaction history
10. Verify the QR code generate properly or not
11. Verify that how many time get for transaction payment
12. Verify that the internet speed is affect on payment time