MODULE - 2 ASSIGNMENT

Que: What is Exploratory Testing?

Exploratory testing is a concurrent process where test design, execution and logging happen simultaneously. The focus of exploratory testing is more on testing as a "thinking" activity.

Que: What is traceability matrix?

Traceability matrix is a table type document that is used in the development of software application to trace requirements. In this document, the test cases are mapped to the corresponding requirement to ensure the coverage of test cases, to find any gap between requirement and test cases. This document is designed to make sure that each requirement has a test case. Generally it is prepared before the test execution process.

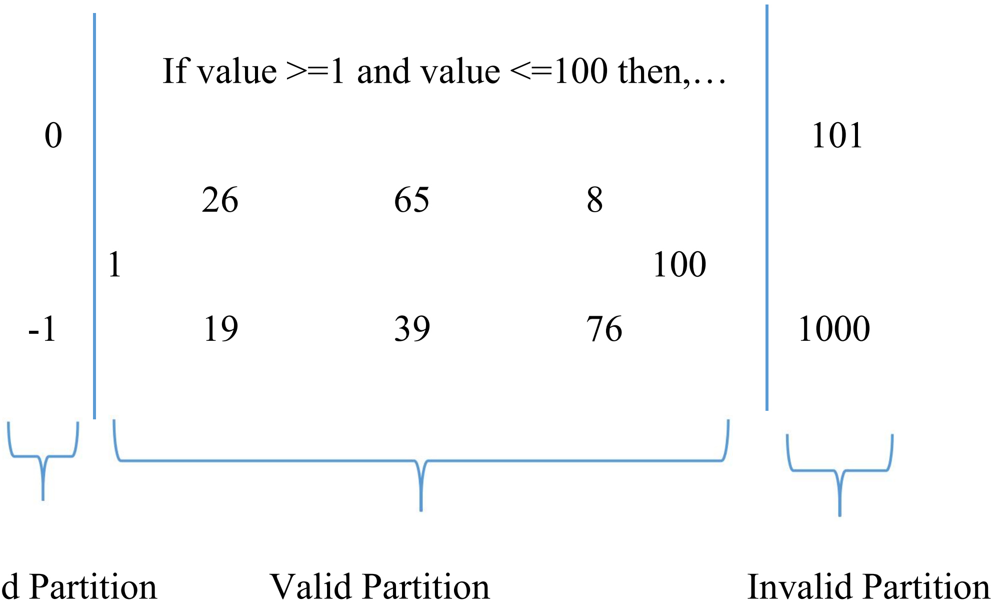
Que: What is Boundary value testing?

Boundary value testing is a type of technique for black box testing. This is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.

Que: What is Equivalence partitioning testing?

Equivalence partitioning is a method of dividing up tests into sections with similar values and then performing a test on each section. This black box testing method eliminates the need to test every potential value, since each test in that specific partition should have the same outcome. The aim is to treat groups of inputs as equivalent and to select one representative input to test all of them.

For example, 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...” We could put a range of numbers as shown in the below figure.



Que: What is Integration testing?

Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

Que: What determines the level of risk?

A properly designed test that passes determines the overall level of Risk in a system.

Que: What is Alpha testing?

Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. It is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly.

Que: 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.

Que: What is component testing?

Unit Testing is a level of the software testing process where individual units of a system are tested. The purpose is to validate that each unit of the software performs as designed.

Que: What is functional system testing?

A requirement which may exist as a text document and/or a model that specifies a function that a system or system component must perform.

Que: What is Non-Functional Testing?

Non-functional testing means testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability. 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 time for performance testing.

Que: 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. For example, Mobile Based Testing : If mobile is in every orientation mode so display image-video properly, Every app will display in responsive type, Alignment should be apply properly of every field.

Que: What is Adhoc testing?

Adhoc testing is a software testing technique performed without any specific test plan or predefined set of steps. The main aim of this testing is to find defects by random checking.

Que: What is load testing?

Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. 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. This testing helps determine how the application behaves when multiple users access it simultaneously.

Que: What is stress Testing?

Stress testing is used to test the stability and reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. System is stressed beyond its specifications and performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load to check how and when it fails.

Que: What is white box testing and list the types of white box testing?

White box testing is based on an analysis of the internal structure of the component or system. In this method, the testers require knowledge of how the software is implemented, how it works.

- Types of testing: 1) Branch Condition testing
  2) Branch Condition Combination testing
  3) Modified Condition Decision testing

4) Dataflow testing

5) Linear Code Sequence And Jump (LCSAJ) testing

Que: 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. The tester does not have access to the source code. When performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. There are four black-box technique:

1) Equivalence partitioning

2) Boundary value analysis

3) Decision tables

4) State transition testing

5) Use-case Testing

Que: Mention what are the categories of defects?

There are five types of defects:

1) Database Defects

2) Critical Functionality Defects:

3) Functionality Defects

4) Security Defects

5) User Interface Defects

Que: Mention what bigbang 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.

Que: What is the purpose of exit criteria?

The Purpose of exit criteria is to define when we STOP testing either at the: 1) End of all testing – i.e. product Go Live

2) End of phase of testing (e.g. hand over from System Test to UAT)

Que: When should "Regression Testing" be performed?

Regression testing should be performed to confirm that a recent program or code change has not adversely affected existing features. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

Que: What is 7 key principles? Explain in detail.

1) Testing shows presence of Defects:-

Testing can show that defects are present but cannot prove that there are no defects in a system. By 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.

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.

3) Early Testing:-

The testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives. These activities should be focused on defined objectives which are outlined in the test strategy.

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. Most defects found during testing are usually confined to a small number of modules. Most operational failures of a system are usually confined to a small number of modules.

5. The 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.

6. Testing is Context Dependent:-

Testing is done differently in different contexts different kinds of sites are tested differently. Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing.

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

Que: Difference between QA v/s QC v/s Tester

Quality Assurance	Quality Control	Tester
Activities which ensure the implementation of processes and standards in context to verification of developed software & 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.
Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing Software with intend to identify bug/defect through an 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.

Que: 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.

Smoke testing is usually documented or scripted.	Sanity testing is usually not documented and is unscripted.
Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.
The objective of this testing is to verify “stability" of the system in order to do with more rigorous testing.	The objective of this 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 Regression testing.	Sanity testing is a subset of Acceptance testing.
Smoke testing is like general health check up.	Sanity Testing is like specialized health check up.

Que: Difference between Verification and Validation

Verification	Validation
The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
To ensure that the product is being built according to the requirements and design specifications.	To ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place.

Evaluation items in verification are plans, requirement specs, design specs, code, test cases.	Evaluation items in validation is The actual product/software.
Activities in verification is considered as reviews, walkthroughs, inspections.	Activities in verification is considered as testing.

Que: Explain types of Performance testing.

There are six types of performance testing which are as follows:

1) Load Testing:

Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps to determine how the application behaves when multiple users access it simultaneously. It gives confidence in the system & its reliability and performance. Load Testing helps to identify the bottlenecks in the system under heavy user stress scenarios before they happen in a production environment. Load testing is needed as some extremely popular sites have suffered serious downtime when they get massive traffic volumes. For example, Popular toy store Toysrus.com, could not handle the increased traffic generated by their advertising campaign resulting in loss of both marketing dollars, and potential toy sales.

2) Stress Testing:

Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. Stress Testing is done to make sure that the system would not crash under crunch situations. It is also known as endurance testing. Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.

3) Endurance testing:

This type of performance test is meant to test the system performance for longer periods of time (i.e. 8 hours or more), in order to reveal other types of problems. These may come up following the accumulation of problems, as in the case of a memory leak and to analyze how the system behaves while following a specific period in operation.

4) Spike Testing:

It is a type of performance testing that involves flooding a site or application with sudden and extreme increases and decreases (spikes) in load. Spike testing is similar to load testing. But unlike load testing, spike testing doesn't apply a uniform or expected load to the system (e.g.,100 requests per minute for 20 minutes). This tests are useful to prepare for situations in which traffic is large, unpredictable, and sudden like, A PR appearance or major marketing campaign.

5) Volume Testing:

Volume testing (also called flood testing) is a type of testing that involves populating a system with large volumes of data. Where load tests are about the number of users, volume tests are about the amount of data. So volume tests are mainly tests of the database.

6) Scalability Testing:

Checking the performance of an application by increasing or decreasing the load in particular scales (no of a user) is known as scalability testing. Upward scalability and downward scalability testing are called scalability testing. It is divided into two parts: **i) Upward scalability testing** where the number of users is being increased on a particular scale until get a crash point. It is used to find the maximum capacity of an application. **ii) Downward scalability testing** is used when the load testing is not passed, then start decreasing the number of users in a particular interval until the goal is achieved. So that it is easy to identify the bug.

Que: What is Error, Defect, Bug and Failure?

- **Error:-**

  An error is a discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. A mistake in coding is called error.

- **Defect:-**

  Defect is commonly refers to several troubles with the software products, with its external behavior or with its internal features. An error found by tester is called defect.

- **Bug:-**

  Bug means a fault in a program which causes the program to perform in an unintended or unanticipated manner. A defect accepted by the development team then it is called bug.

- **Failure:-**

  Failure means a discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. If a build does not meet the requirements then it is called failure.

Que: Difference between Priority and Severity

Priority	Severity
Priority is Relative and Business-Focused.	Severity is absolute and customer-focused.
It defines the order in which we should resolve a defect.	It defines the impact that a given defect has on the system.
The category decided by testers.	The category decided by developers

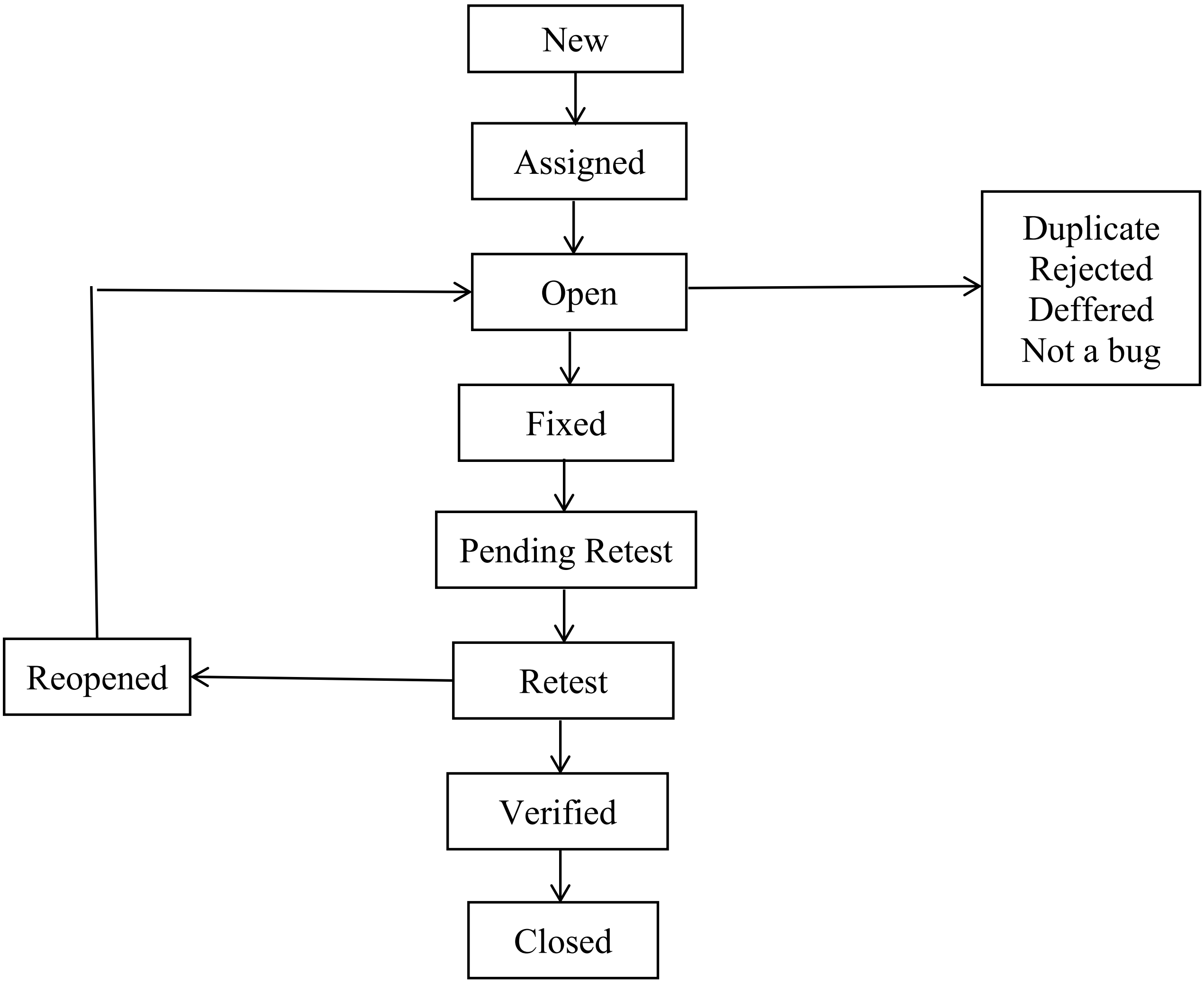
	or product owners.
It deals with the technical aspects of the application.	It deals with the time frame or order to fix the defects.
The value does not change with time, it’s fixed.	The priority value is subjective and may change after comparing with other defects.
For instance, if the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.	For instance, if an application or web page crashes when a remote link is clicked, in this case, clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

Que: What is Bug Life Cycle?

The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as ‘Bug(Defect) Life Cycle’. Bug life cycle includes follows:

- **New:** When a new defect is logged and posted for the first time, it is assigned a status as New.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team.
- **Open:** The developer starts analyzing and works on the defect fix
- **Fixed:** When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
- **Pending retest:** Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”

- **Retest:** Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test”.



- **Verified:** The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
- **Reopen:** If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.
- **Closed:** If the bug is no longer exists then tester assigns the status “Closed.”

- **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."
- **Rejected:** If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."
- **Deferred:** If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs
- **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".

Que: Explain the difference between Functional testing and Non-Functional testing.

Functional Testing	Non-functional Testing
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first.	Non-functional 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.
Business requirements are the inputs to functional testing.	Performance parameters like speed , scalability are inputs to non-functional testing.
Functional testing describes what the product does.	Non-functional testing describes how good the product works.
Types of functional testing are unit testing, smoke testing, sanity	Types of non-functional testing are performance testing, load testing,

testing, integration testing, white box testing, black box testing, user acceptance testing and regression testing.	volume testing, stress testing, security testing, installation testing, penetration testing, compatibility testing and migration testing.

Que: Create HLR & Test case of …

Instagram and Facebook only first page

Answer written in excel sheet

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

STLC	SDLC
STLC is a systematic approach to testing a software application to ensure that it meets the requirements and is free of defects.	SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.
SDLC consists of 6 phases which are requirement analysis, test planning, test case development, test environment setup, test execution , test cycle closure.	SDLC consists of 6 phases which are requirements gathering, analysis, design, implementation, testing, maintenance.
Goal of STLC is to complete successful testing of software.	Goal of SDLC is to complete successful development of software.
In STLC, less number of members (testers) are needed.	In SDLC, more number of members (developers) are required for the whole process.

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

A **test scenario** is any functionality that can be tested. It is also called test Condition, or test Possibility. **Test cases** involve the set of steps, conditions and inputs which can be used while performing the testing tasks. **Test script** is a set of sequential instruction that detail how to execute a core business function.

Que: Explain what Test Plan is? What is the information that should be covered.

Test plan is a document describing the scope, approach, resources and schedule of intended test activities. In test plan, following information should be covered:

- **Approach:** Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.
- **Integrating and coordinating the testing activities:** supply, development, operation and maintenance.
- **Making decisions about:**

  - what to test

  - who do testing? i.e. what roles will perform the test activities

  - when and how the test activities should be done and when they should be stopped

  - how the test results will be evaluated
- Assigning resources for the different tasks defined.
- **Test ware:** Defining the amount, level of detail, structure and templates for the test documentation.
- Selecting metrics for monitoring and controlling test preparation and execution, defect resolution and risk issues.

- **Process**: Setting the level of detail for test procedures in order to provide enough information to support reproducible test preparation and execution.

Que: What is priority?

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

Que: What is severity?

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words, it defines the impact that a given defect has on the system. For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So, the severity is high.

Que: Bug categories are…

The bug categories are Security, Database, Functionality (Critical/ General), UI.

Que: Advantage of Bugzilla.

The advantages of bugzilla are as follows:

- 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.
- It is an open-source widely used bug tracker;
- It is easy in usage and its user interface is understandable for people without without technical knowledge;
- It easily integrates with test management instruments;
- It integrates with an e-mailing system;
- It automates documentation.

Que: What are the different Methodologies in Agile Development Model?

The methodologies in agile development model are as follows:

- Scrum
- Extreme Programming (XP)
- lean product development
- Kanban
- Feature-Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)

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

Authorization	Authentication
Authorization determines what resources a user can access.	Authentication verifies who the user is.

Authorization works through settings that are implemented and maintained by the organization.	Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.
Authorization always takes place after authentication.	Authentication is the first step of a good identity and access management process.
Authorization isn’t visible to or changeable by the user.	Authentication is visible to and partially changeable by the user.
Example: By verifying their identity, employees can gain access to a human resources application that includes their personal pay information, vacation time, and 401K data.	Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization.

There are some challenges which are faced in web testing and they are as follows:

1) Integration:-

Integration testing exposes problems with interfaces among different program components before deployment. Additionally, integration testing can show the different issues an application may have when interacting with other applications, allowing the developer to tweak things. Environment and infrastructure inconsistency, different interaction models, and overall performance are just a few of the issues associated with integration testing.

2) Interoperability:-

Proving end-to-end functionality between communicating systems is always a challenging obstacle. Different users utilize different browsers and operating systems. To pull data, testing each one to confirm a clear information pathway is very important. Even if the browsers are similar, the web application may be rendered differently based on screen resolution and overall software configuration. This can present some serious issues for developers.

3) Security:-

In one of the most important tests, the developer must make sure that the continually evolving cyber threat can be countered and neutralized. Additionally, tests associated with data integrity before and after an attack are equally important when considering data breaches or lost information. Some of the challenges associated with security testing include dealing with unsecured communications, removing malicious files (if security firewalls have been breached), and the utilization (and integration) of different authentication procedures.

4) Performance:-

Slow applications are not successful. Developers understand that the speed of the app is defined by the need of the user, and with more users expecting more speed, the requirement of performance is non-negotiable. Testing large applications on minimal hardware, underestimating software requirements, and overextending application features are just a few of the issues associated with performance testing. Integration and interoperability issues can also have a direct effect on performance, and because of that, should be tested at the beginning.

5) Usability:-

Since web-based applications are dependent on different browsers, consistent usability is crucial. Additionally, since the app is the brand (or a component thereof), any inconsistency within the user experience may translate into a negative experience, affecting the brand and its potential growth. When testing usability, developers face issues with scalability and interactivity. Since every user is different, it is important for developers to utilize a representative group to test the application across different browsers, using different hardware.

Que: Create HLR & Test case of Web based Whatsapp web and Instagram.

Answer written in excel sheet

Que: Create HLR and Testcase on this link https://artoftesting.com/

Answer written in excel sheet

Que: Write a scenario of only Whatsapp chat messages
Que: Write a Scenario of Pen
Que: Write a Scenario of Pen Stand
Que: Write a Scenario of Door
Que: Write a Scenario of ATM
Que: Write a scenario of Microwave Owen
Que: Write a scenario of Coffee vending Machine
Que: Write a scenario of chair
Que: To Create Scenario Gmail(Receiving mail)
Que: Online shopping to buy product (flipkart)
Que: Write a Scenario of Wrist Watch
Que: Write a Scenario of Lift(Elevator)
Que: Write a Scenario of Whatsapp Group (generate group)
Que: Write a Scenario of Whatsapp payment

Answer of all scenarios written into excel sheet

Que: When to use Usability Testing?

Here are the three major times to do usability testing:

- Early and Often Throughout Development:-

- Conduct usability testing **early** in the design process even with low-fidelity prototypes.

- Continue testing **often** as you iterate on your design solutions.

- Identifying problems early saves time and money on rework.

- Before Launch:-

- Ensure a smooth user experience right from the start.

- Validate that your product meets user needs and expectations.

- After Major Updates:-

- Verify that new features are user-friendly and don't disrupt existing functionality.

- Usability testing helps maintain a positive user experience even after significant changes.

Que: What is the procedure for GUI Testing?

The procedure for GUI testing is as follows:

1) Understand the requirement:-

- Begin by fully understanding the software application's intended functionalities.

- Define what the GUl elements (buttons, menus, forms, etc.) should do under different conditions.

2) Define test cases:-

- Create detailed test cases that cover various scenarios related to GUI.

3) Set up the test environment:-

- Prepare the test environment with the necessary hardware, software and configurations.

3) Set up the test environment:-

- Prepare the test environment with the necessary hardware, software and configurations.

Ensure you have access to the application's GUI for testing.

4) Execute the tests:-

- Interact with GUI elements as user would.
- Verify that the buttons, menus, forms and other visual components work as expected.
- Check for responsiveness, alignment, colour consistency, and usability.

5) Report and analyse:-

- Document any defects or issues encountered during testing
- Provide clear descriptions of the problems found along with steps to reproduce them.

6) Make necessary changes:-

- Developers address the reported issues by fixing defects or improving the GUI
- Retest the modified GUI components to ensure they now function correctly.

7) Review and improvement:-

- Conduct a review of the GUI testing Process.
- Identify areas for improvement, such as enhancing test coverage or redefining test cases.