SOFTWARE TESTING ASSIGNMENT

MODULE-2 (MANUAL TESTING)

What is Exploratory Testing?

Exploratory Testing is a type of software testing where Test cases are not created in advance but testers check system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is more on testing as a “thinking” activity.

Exploratory Testing is widely used in Agile models and is all about discovery, investigation, and learning. It emphasizes personal freedom and responsibility of the individual tester.

What is traceability matrix?

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship.

It is used to track the requirements and to check the current project requirements are met.

What is Boundary value testing?

Boundary Value Testing is one of the popular software testing mechanism, where testing of data is done based on boundary values or between two opposite ends where the ends may be like from start to end, or lower to upper or from maximum to minimum.

This testing process was introduced to select boundary values that came from the boundary based on the inputs at different ends of testing values. This black box testing strategy was introduced after equivalence class partitioning where the partition of classes takes place first followed by a partition at the boundaries.

What is Equivalence partitioning testing?

Equivalence Partitioning or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In 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.
- You can apply this technique, where there is a range in the input field.

What is Integration testing?

Integration testing -- also known as integration and testing (I&T) -- is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity. However, these modules may be coded by different programmers.

The aim of integration testing is to test the interfaces between the modules and expose any defects that may arise when these components are integrated and need to interact with each other.

What determines the level of risk?

- Risk should be evaluated at the Business Level, Technological Level, Project Level and Testing Level.
- Risk also used to decide where to start and where more testing is needed

What is Alpha testing?

Alpha Testing is a type of acceptance testing; performed to identify all possible issues and bugs before releasing the final product to the end users. Alpha testing is carried out by the testers who are internal employees of the organization. The main goal is to identify the tasks that a typical user might perform and test them.

To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. The main focus of alpha testing is to simulate real users by using a black box and white box techniques.

What is beta testing?

Beta Testing is performed by “real users” of the software application in “real environment” and it can be considered as a form of external User Acceptance Testing. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in customer’s environment.

Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

What is component testing?

Component testing is defined as a software testing type, in which the testing is performed on each individual component separately without integrating with other components. It’s also referred to as Module Testing when it is viewed from an architecture perspective. Component Testing is also referred to as Unit Testing, Program Testing or Module Testing.

Generally, any software as a whole is made of several components. Component Level Testing deals with testing these components individually.

What is functional system testing?

functional testing of a system includes tests that evaluate the functions that the system must perform. Functional requirements may be described in work products (requirements, specification, business need, user story, use case) and in the functional specification.

What is Non-Functional Testing?

Non-Functional Testing is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

An excellent example of non-functional test would be to check how many people can simultaneously login into a software.

What is GUI Testing?

GUI Testing is a software testing type that checks the Graphical User Interface of the Software. The purpose of Graphical User Interface (GUI) Testing is to ensure the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc.

GUI is what the user sees. Say if you visit guru99.com what you will see say homepage it is the GUI (graphical user interface) of the site. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not.

What is Adhoc testing?

Ad hoc Testing is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

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. Adhoc 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.

This testing requires no documentation/ planning /process to be followed. Since this testing aims at finding defects through random approach, without any documentation, defects will not be mapped to test cases. This means that, sometimes, it is very difficult to reproduce the defects as there are no test steps or requirements mapped to it.

What is load testing?

Load testing is a type of **performance testing** that simulates a real-world load on any software, application, or website. Without it, your application could fail miserably in real-world conditions. That’s why we build tools like **Retrace** to help you **monitor application performance** and fix bugs before your code ever gets to production. Load testing examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end-users. This tool is typically applied when a software development project nears completion.

What is stress Testing?

- Stress Testing – System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity , complex database queries , continuous input to system or database load.
- Stress testing is done to make sure that the system would not crash under crunch situation.
- Stress testing is also known as endurance testing.

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

- 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 on an application, the tester needs to possess knowledge of the internal working of the code..

- Types of coverage
  - Statement coverage
  - Decision coverage
  - Condition coverage

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

- The technique of testing without having any knowledge of the interior workings of the application is Black box testing.

- The tester is oblivious to the system architecture and does not have access to the source code.

- Typically when performing a black box test , a tester interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked.

- 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

Mention what are the categories of defects?

- There are some categories which are as below
  - Data quality/Database Defects
  - Critical Functionality Defects
  - Functionality Defects
  - Security Defects
  - User Interface Defects

What is the purpose of exit criteria?

- 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 (handover from system test to UAT)

When should "Regression Testing" be performed?

- Change in requirements and code Is modified according to the requirements
- New feature is added to the software
- Defect fixing
- Performance issue fix

What is 7 key principles? Explain in detail?

- Testing shows presence of Defects
- Exhaustive Testing is Impossible
- Early testing
- Defect Clustering
- The pesticide Paradox
- Testing is context dependent
- 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 reduce the probability of undiscovered defects remaining in the software but. Even if no defects are found, it is not a proof of correctness.
- As we find more defects. The probability of undiscovered defects remaining in a system reduces

2. Exhaustive Testing is Impossible:

- Testing everything including all combinations of inputs and preconditions is not possible
- For example in an application in one screen there are 15 input fields. Each having 5 possible values then to test all the valid combination you would need
- That is we must Priorities our testing efforts 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
- Testing activities should start as early as possible in the development life cycle.

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
- Defects are not evenly spread in a system
- In other words, most defects found during testing are usually confined to small number of modules

5. The pesticide Paradox:

- If the same tests are repeated over and over again, eventually the same set of test case will no longer find any new defects
- To overcome this “pesticides 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 basically context dependent.
- Testing is done differently context
- Different kinds of sites are tested differently

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 defeclts have been resolved it may still be unusable and/or does not fulfill the users need and expectations

Difference between QA v/s QC v/s Tester

- QA (Quality Assurance)
  - It is a subset of Software Test Life Cycle (STLC)
  - Preventive activities
  - Process oriented activities
  - Focus on processes and procedures rather than conducting actual testing on the system

- QC (Quality Control)
  - QC can be considered as the subset of Quality Assurance
  - It is a corrective process
  - Product oriented activities
  - Focuses on actual testing by executing software with intend to identify bug/defect through implementation of procedures and process

- Tester
  - Testing is the subset of Quality Control
  - It is a preventive process
  - Product oriented activities
  - Focuses on actual testing

Difference between Smoke and Sanity?

- **Smoke Testing**
  - Smoke testing is performed to ascertain that the critical functionalities of the program is working fine
  - This testing is performed by the developers or testers
  - Smoke testing is usually documented or scripted
  - Smoke testing is a subset of Regression testing

- **Sanity Testing**
  - Sanity testing is done to check the new functionality / bugs have been fixed
  - Sanity testing is usually performed by testers
  - Sanity testing is usually not documented and is unscripted
  - Sanity testing is a subset of Acceptance Testing

Difference between verification and Validation

- **Verification :**
  - 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
  - To ensure that the product is being built according to the requirements and design specification. In other words, to ensure that work products meet their specified requirement
  - Are we building the product right?

- **Validation :**
  - 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 actually meets the user's needs and that the specification were correct in the first place in other word , to demonstrate that the product fulfills its intended use when placed in its intended environment
  - Are we building the right product?

Explain types of Performance testing.

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

1. Load testing:

It’s a performance testing to check system behavior under load testing an application under heavy loads such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails

2. Stress testing:

Stress testing system is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity complex capacity queries, continuous input to system or database load.

What is Error, Defect, Bug and failure?

- **“**A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure”

Difference between Priority and Severity

- **Priority :**
  - Priority is relative and business focused. Priority defines the order in which we should resolve a defect, should we fix it now or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mention then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

- **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 given defect has on the system

What is Bug Life Cycle?

- **“A** computer bug is an error, flow, mistake, failure or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bug arise from mistake and errors made by people, in either a program’s source code or its design.”

- When bug is discovered, it goes through several states and eventually reaches one of the terminal states where it becomes inactive and closed

Explain the difference between Functional testing and NonFunctional testing

- **Functional Testing :**
  - Functional testing based on an analysis of the specification of the functionality of a component or system
  - Functional testing is based on the Functions and features – may be applied at all test levels
  - Functional testing verifies that each functions of the software application operates in conformance with the requirement specification

- **Non – Functional testing :**
  - Non – functional testing is the attributes of a component or system that do not relate to functionality e.g. reliability , efficiency , usability , interoperability , maintainability and portability
  - Measuring the characteristics of the system/software that can be quantified on varying scale
  - Non – functional testing includes, but is not limited to, performance testing, load testing , stress testing , usability testing , maintainability testing , reliability testing and portability testing.

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

❖ **Software Testing Life Cycle :**

- STLC is mainly related to software testing
- It focuses only on testing the software
- STLC involves only five phases or steps
- In STLC, less number of members are needed
- Goal of STLC is to complete successful testing of software

❖ **Software Development Life Cycle :**

- SDLC is mainly related to software development
- Beside development other phases like testing is also included
- SDLC involves total six phases or steps
- In SDLC, more number of developers are required for the whole process

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

❖ **Test Scenario:**

- Is any functionality that can be tested
- Is derived from test artifacts like Business Requirements Specification (BRS) and Software Requirements Specification (SRS)
- Helps test the end to end functionality in Agile way
- Is more focused on what to test
- Test less time and fewer resources to create

❖ **Test Cases:**

- Is a set of actions executed to verify particular features or functionality
- Is mostly derived from test scenario
- Helps in exhaustive testing of an app
- Is focused on what to test and how to test
- Requires more resources and time

Test Script :

- Is a set of instructions to test an app automatically
- Is mostly derived from test cases
- Helps to test specific things repeatedly
- Is focused on the expected result
- Requires less time for testing but more resources for scripts creating and updating

What is priority?

- Priority is Relative and Business focused. Priority defines the order in which we should resolve a defect. Should we fix it now or Can it wait? This priority status is set by the tester to the developer mentioning the same frame to fix the defect. If the high priority is mentioned then the developer has to fix it at earliest. The priority status to set based on the customer requirements.

- For example : If the company name is misspelled in the home page of the website then the priority is high and severity is high and severity is low to fix.

What is severity?

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

- For example : if an application of the 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 severe. So the severity is high but priority is low.

Bug categories are…

- There are some categories
  - Security
  - Database
  - Functionality (Critical/General)
  - UI

Advantage of Bugzila .

- Open source, free bug tracking
- Automatic Duplicate bug detection
- Search option with advanced features
- File/Modify bugs by mail
- Move bugs between installs
- Multiple authentication methods
- Time tracking
- Automated bug reporting, has an API to interact with system

Difference between priority and severity

❖ Severity :

- Severity is a term that denotes how severely a defect can affect the functionality of the software
- Severity is basically a parameter that denotes the total impact of a given defect on any software
- The value of severity is objective
- The value of severity changes continually from time to time
- The testing engineer basically decides a defect's severity level

❖ Priority

- Priority is a term that defines how fast we need to fix a defect
- Priority is basically a parameter that decides the order in which we should fix the defect.
- The value of priority is subjective
- The value of priority change from time to time
- The product manager basically decides a defect's priority level

What are the different Methodologies in Agile Development Model?

Scrum methodology:

Scrum is a lightweight framework of Agile Project Management, it can be adopted to conduct iterative and all types of incremental projects.

Due to specific characteristics like simplicity, sustained productivity and strength for blending several underlying approaches adapted by other agile methods, Scrum has obtained popularity over the years.

Kanban:

Kanban is an eminently visual workflow management approach, that can be employed for visualizing and thoroughly maintaining the making of products, it focuses on continual delivery of the product , but is not making stress to the entire software development life cycle.

Similar to scrum, kanban is the process developed for supporting collaborative teamwork more effectively.

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

Authentication	Authorization
Authentication verifies who the user is.	Authorization determines what resources a user can access.
Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.	Authorization works through settings that are implemented and maintained by the organization.
Authentication is the first step of a good identity and access management process.	Authorization always takes place after authentication.
Authentication is visible to and partially changeable by the user.	Authorization isn’t visible to or changeable by the user.
Example: By verifying their identity, employees can gain access to an HR 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.

When to used Usablity Testing?

- Usability testing is a method of testing the functionality of a website, app or other digital product by observing real users as they attempt to complete tasks on it. The users are usually observed by researchers working for a business.

- Usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you have begun the strategy work around a brand new site or app.

What is the procedure for 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.

Write agile manifesto principles?

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

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

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

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

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

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

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

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

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

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

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

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