**Module: 2 (Manual Testing) Assignment**

* **What is software testing?**

- Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not.

- software testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

- Software testing is a process used to identify the corrections, completeness, and quality of developed computer software.

* **What is Exploratory Testing?**

- Exploratory Testing is a concurrent process where

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

* **What is traceability matrix?**

- Traceability Matrix (also known as Requirement Traceability Matrix- RTM) is a table which is used to trace the requirements during the software development life cycle. It can be used for forward tracing (i.e. from Requirements to Design or coding) or backward (i.e. from coding to Requirements).

- Each requirement in the RTM documents is linked with its associated test case, so that testing can be done as per the mentioned requirements.

- It helps in tracing the developed documents during different phases of SDLC.

* **What is Boundary value testing?**

- Boundary value analysis is a methodology for designing test cases that concentrates software testing efforts on cases near the limits of valid ranges.

- It is a type of Black Box testing.

- It is the method which refines equivalence partitioning.

- It generates test cases that highlight errors better than equivalence partitioning.

* **What is Equivalence partitioning testing?**

- Equivalence partitioning is the process of defining the optimum number of tests by:

- Reviewing documents such as the functional design specification and detailed design specification and identifying each input condition within a function.

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

- Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.

- It is type of Black Box type of testing

* **What is Integration testing?**

- Integration Testing is a 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.

- There are two levels of Integration testing:

1) Component Integration testing

2) System integration testing

* **What determines the level of risk?**

- Risk - 'A factor that could result in future negative consequences; usually expressed as impact and likelihood.

- A risk could be any future event with a negative consequence we need to identify the risks associated with our project.

- Risks are of two types:

Project Risks

Product Risks

* **What is Alpha testing?**

- It is the form of Acceptance (Thread) 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.

* **What is beta testing?**

- It is the form of Acceptance 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 team.

- It is usually conducted for software product.

- It is performed in Real time Environment

- It is always performed outside the organization.

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

* **What is component testing?**

- Component (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.

- 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.

- Component testing is also known as Unit testing, Module testing or program testing.

* **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 type of techniques:

Requirement Based Functional testing

Process Based Testing

* **What is Non-Functional Testing?**

- Non-functional testing is a Dynamic Testing Technique

- Noon-functional testing: Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and probability

- Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling

- It is the testing of 'how" the system work. Non-functional testing may be performed at all levels.

* **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 is Adhoc testing?**

- It is the type of Experience Based Testing.

- Adhoc testing can be achieved with the testing technique called Err-or Gussing.

- The Adhoc testing (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.

- 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.

- Main aim of this testing is to find defects by random checking.

* **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.

- White box testing is also known as structure based testing or glass - box testing technique because here the testers require knowledge of how the software is implemented how it works.

White box testing techniques:

Statement coverage

Branch coverage

Decision coverage

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

- Black box testing either functional or non-functional, without reference to the internal structure of the component or system.

black box testing is also known as specification based testing techniques or input/output driven testing techniques because they view the software as a black box with inputs and outputs.

- Black box techniques:

Equivalence Partitioning

Boundary Value Analysis

Decision Tables

State Transition Testing

Use case Testing

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

- There are five types of categories of defects:

- **Data quality/ Database Defects:** Deals with improper handling of data in the database.

- **Critical functionality of defects:** The occurrence of these bugs hampers the crucial functionality of the application.

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

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

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

* **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.

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

* **What is the purpose of exit criteria?**

- The purpose of Exit criteria is 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.

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

* **When should "Regression Testing" 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.

* **What is 7 key principles? Explain in detail?**

- 7 key principal of testing:

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 error fallacy

**This principles in detail:**

**1) Testing shows presence of Defect:** Testing can show that defects are present, but cannot prove that there are no defect.

**-** 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 combination of inputs and precondition 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.

**-** So accessing and managing risk is one of the most important activities and reason for testing in any project.

**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 objectivise.

**-** Testing activities should start as early as possible in the development life cycle.

**-** These activities should be focused on defined objectivise outline in the test strategy.

**4) Defect Clustering:**  A small number of modules contain most of the defects discovered during per-release testing or are responsible for the most operational failures.

**-** Defects are not evenly spread in a system, they are clustered

**-** In other words most defects found during testing are usually confined to small number of modules

**-** Most operational failures of a system are usually confined to a small number of modules.

**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.

**6) Testing is context Dependent:** Testing is basically context dependent, Testing is done differently in different contexts.

**-** Different kinds of sites are tested differently.

**-** For example: safety critical software is tested differently form an e-commerce site.

**7) Absence of Error Fallacy:** If the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing defect does not help.

**-** If we built a system and in doing so find and fix defects, it does not make it good system

**-** Even after defect have been resolved it may still be unusable and/or does not fulfill the users.

* **Difference between QA v/s QC v/s Tester?**

- **QA (Quality Assurance)**: Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

- **QC (Quality Control):** Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.

- **Testing:** Activities which ensure the identification of bugs/ error/ defect in the software.

* **Difference between Smoke and Sanity?**

**- Smoke Testing:** Smoke testing is performed after software build to ascertain that the critical functionalities of the program is working fine.

**-** It is executed "Before" any detailed functional or regression tests are executed on the software build.

**-** Purpose is to reject badly broken application.

**- Sanity Testing:** After receiving a software build with minor changes in code, or functionality, sanity testing is performed to ascertain that the bugs have been fixed and no further issues are introduced due to these changes.

**-** The goal is to determine that the proposed functionality works roughly as expected.

**-** If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

* **Difference between verification and Validation**

**- Verification Phase:** 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.

**- Validation phase:** The process of evaluating software during or at the end of the development process to determine whether is satisfies specified business requirements.

* **Explain types of Performance testing?**

**-** Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.

**-** Feature and Functionality supported by a software system is not the only concern. A software application's performance like its response time, do matter. The goal of performance testing is not to find bugs but to eliminate performance bottlenecks.

- Types of performance testing:

Load testing

Stress testing

Endurance testing

Spike testing

Volume testing

Scalability testing

* **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"

- **Error:** 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 value, etc.

- **Failure:** The inability of a system or component to perform its required functions within specified performance requirements.

- **Bug:** A fault in a program which causes the program to perform in an unintended or unanticipated manner.

- **Defect:** commonly refers to several troubles with the software products, with its external behaviour or with its internal features.

* **Difference between Priority and Severity**

**- 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 impacts that a given defects has on the system.

**-** Severity can be of following type:

- Critical

- Major

- Moderate

- Minor (low)

- Cosmetic

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

- Priority can be of following type:

- Low

-Medium

- High

* **What is Bug Life Cycle?**

**-** The duration or time span between the first time defects is found and the time that is closed successfully, rejected, postponed or deferred is called as "Defect Life Cycle".

* **Explain the difference between Functional testing and Non Functional testing**

**-** **Functional testing** is performed using the functional specification provided by the client and verifies the system against the functional requirements.

**-** Is executed first

**-** Manual testing or automation tools can be used for functional testing.

**-** Functional testing describes what the product does.

**-** Easy to do manual testing

**-** Types of Functional testing; - Unit testing

- Smoke testing

- Sanity testing

- Integration testing

- White box testing

- Black box testing

- User Acceptance testing

- Regression testing

- **Non Functional testing:** Non-functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system.

- Non-functional testing should be performed after functional testing.

- using tools will be effective for this testing.

- Non-Functional testing describes how good the product works.

- Tough to do manual testing.

- Type of Non-functional testing: - Performance testing

- Load testing

- Volume testing

- Stress testing

- Security testing

- Installation testing

- Penetration testing

- Compatibility testing

- Migration testing

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

- **Software Development Life Cycle (SDLC):** SDLC is a structured imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support there are a number of different development models.

- A software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

SDLC phases: - Requirement collection/Gathering

- Analysis

- Design

- Implementation

- testing

- Maintenance

**- Software Testing Life Cycle (STLC):** STLCis the structured imposed on the testing of the software product that also defines the different types of process.

**-** The process always starts with the planning and ends with test closure activities

**-** Each phase may have to be executed a number of times in order to fulfil exit or completion criteria

- Stages of STLC: - Test planning and controlling

- Test Analysis and design

- Test implementation and execution

- Evaluating Exit criteria and reporting

- Test closure and Activities

* What is the difference between test scenario, test case and test script?

- **Test Scenario:** A scenario is any functionality that can be tested. It is also called test condition or test possibility.

- Test Scenario is 'What to be tested'

- Test scenario is nothing but test procedure.

- **Test Case:** Test case involve the set of steps, conditions and inputs which can be ysed while performing the testing tasks.

- Test case is 'how to be tested'

- Test case consist of set of input values, execution precondition expected results and executed post condition developed to cover certain test condition

- **Test Script:**  Test script is also known as a test procedures specification

- The test procedures specification specifies the sequence of actions for a test, i.e one or more test cases.

- A set of sequential instruction that details how to execute a core business function.

* Explain what is test plan is? What is the information that should be covered?

- It is a high level document in which how to perform testing is described. The test plan document is usually prepared by the test lead or test manager and the focus of the document is to describe what to test, how to test when to test and who will do what test.

- The test plan will include the following:

- Introduction to the test plan document

- Assumptions when testing the application

- List of test cases included in testing the application

- List of features to be tested

- What sort of Approach to use when testing the software

- List Deliverables that need to be tested

- The resource allocated for testing the application

- Any risks involved during the testing process

- A schedule of tasks and milestones as testing is started

* **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 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.

* **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 impacts that a given defects has on the system.

* **Bug Categories are...**

**-** There are different types of bug/defects:

**- Data Quality/Database Defects:** Deals with improper handling of data in the database

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

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

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

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

* **Advantages of bugzilla**

- Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in perl and uses MYSQL database.

- Bugzilla is a defect tracking tool however it can be used as a test management tool as such it can be easily linked with other test case management tools like quality center. Test link etc.

- This open bug-tracker enables users to stay connected with their clients or employees to communicate about problems effectively throughout the data-management chain

* **What are the different methodology in Agile Development Model?**

**-** There are various methodologies present in agile testing and those are listed below:

**- scrum**

**- Extreme programming**

**-** Other are used less frequently:

**- Dynamic system Development method (DSDM**):

This is an Iterative and incremental approach that emphasizes on the continuous user involvement.

**- Test Driven Development (TDD):**

This is a technique which has short iterations where new test cases covering the desired improvement or new functionality are written first.

**- Feature Driven Development:**

This is an iterative and incremental software development process and this can aim depends on the features.

- **XBreed:**

Agile enterprise previously known as Xbreed. It is agile way of managing architecting and monitoring the enterprise.

- **Crystal:**

Crystal is an adaptive technique mainly used for software development methodologies.

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

- **Authorization:** Accepting an invalid username/password

- **Authorization:** Accessibility to pages though permission not given

- The common problems faced in web testing are:

**- Integration:** Integration testing exposes problems with interfaces among different program components before deployment. Environment and infrastructure inconsistency, different interaction models, and overall performance are just a few of the issues associated with integration testing

**- Interoperability:**  proving end-to-end functionality between communicating systems is always a challenging obstacle. Different users utilize different browsers and operating systems. The web application may be rendered differently based on screen resolution and overall software configuration. This can present some serious issues for developers.

**- Security:**  In one of the most important tests, the developer must make sure that the continually evolving cyber threat can be countered and neutralized. 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.

**- 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. Integration and interoperability issues can also have a direct effect on performance, and because of that, should be tested at the beginning.

**- Usability:** Since web-based applications are dependent on different browsers, consistent usability is crucial. 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.

* **When to used Usability Testing?**

- This testing is recommended during the initial design phase of SDLC which gives more visibility on the expectations of the users.

- There are two methods available to do usability testing-

- Laboratory usability testing

- Remote usability testing

* **What is the procedure for 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 minus, buttons, icons, and all types of bars - tool bar, menu bar, dialog boxes and windows etc.