Module 3

1. What is load testing?

**Load Testing** is a non-functional software testing process in which the performance of software application is tested under a specific expected load. It determines how the software application behaves while being accessed by multiple users simultaneously. The goal of Load Testing is to improve performance bottlenecks and to ensure stability and smooth functioning of software application before deployment.

2. What is stress Testing?

**Stress Testing** is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn’t crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

3. 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.

4. What is the procedure for 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.

5. 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.

6. Mention what are the categories of defects?

List of Defects in Software Testing:

Let’s find the list of defects in software testing.

Design Defects: The algorithms, login and data elements, module interface, the external software and hardware UI descriptions should be correctly designed. The incompatible or incorrectly designed modules lead to defects in the system.

Command Defects:An error in the sequences and logic is known as control flow error or command error. The reasons for such defects are missing command, wrong algorithm, incorrect data and code errors.

Boundary Value Defects:In case the login page is logging in by giving the passport length to 16 characters in the place of 15 characters, then the defect is the boundary value defect.

Error Handling Defects:The error that is raised while the users interacting with the software need to be handled in the correct flow. The flow should indicate the instruction in the popup message for the mandatory fields to alert the users for incorrect information.

Multithreading Defects:

Executing or running multiple tasks at the time. Complex debugging is possible in the multiple threading process. It may also lead to a system crash/failure due to the condition in deadlock.

Security Defect:The defects will be different by their nature of the risks. These defects are weaknesses allowing for a potential security attack.

Interface Defects:

The defects in the interactions of the software and the users. Some of the interfaces in the different kinds of forms are complicated interfaces, unclear interfaces and platform based interfaces.

Priority of Defects:

The impact of the bug of an application should be described.

It is the order of priority which the developer will resolve the defects.

The Priority can be changed based on the comparison with other defects.

At the time of UAT, defects are fixed according to the priority.

Priority can be classified as follow as

Immediate/Critical :This generally occurs when the entire functionality of the task is fault and no more [testing can do further for the result](https://qrsolutions.com.au/testing-best-practices/). Any defects that need immediate attention that affects the flow of the testing comes under this category. Critical severity also fall into the same category.

High :  The defect that does not meet the exit criteria. Due to such defects the testing of the entire application has been stopped until the defects are solved. These defects are resolved once the critical issues are fixed. This kind of defect should be resolved before the release.

Medium :Defects occur when a particular feature cannot be used the way it should be because of defects in a program, environmental issue. These defects should be fixed once all the critical and serious bugs get fixed. These defects can also be fixed in the next release.

Low :If few users of the feature encountered a defect such as minor UI issues, spelling mistakes, alignment issues and colour code mismatch are considered as low priority bugs. Sometimes these defects are opened to suggest enhancements in the existing design. This defect does not need any immediate actions as it can be resolved in future.

Severity of Defects:

It is related to the defect fixing urgency.

The testers will set the level of severity.

Once the Severity is fixed it won’t change with time.

It is based on the functionality that the defects affect.

Severity can be classified as follow as:

Blocker :When the whole functions feature / functionality missing from the project applications and completely crashes the system will be considered as Blocker and these severity defects are having the highest priority.

Major :  The implemented feature that does not meet the requirements, test cases and behaves differently than the normal flow. It does not cause any system kind of failure but not equal to the blocker but avoiding the unnecessary delay of fixing.

Minor :  The defect will not cause a failure in execution of the applications which is not a major impact. Loss of data falls in the minor severity it also falls in major severity based on the classifications.

Low/Trival :   The valid defect that should be fixed even though there is no impact on the functionality.

7. When should "Regression Testing" be performed?

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged.

We do regression testing whenever the production code is modified.

We can perform regression testing in the following scenario, these are:

**1. When new functionality added to the application.**

**Example:** A website has a login functionality which allows users to log in only with Email. Now providing a new feature to do login using Facebook.

**2. When there is a Change Requirement.**

**Example:** Remember password removed from the login page which is applicable previously.

3. **3. When the defect fixed**

**Example:** Assume login button is not working in a login page and a tester reports a bug stating that the login button is broken. Once the bug fixed by developers, tester tests it to make sure Login Button is working as per the expected result. Simultaneously, tester tests other functionality which is related to the login button.

4. **4. When there is a performance issue fix**

**Example:** Loading of a home page takes 5 seconds, reducing the load time to 2 seconds.

5. **5. When there is an environment change**

**Example:** When we update the database.

## How to perform Regression Testing?

The need for regression testing comes when software maintenance includes enhancements, error corrections, optimization, and deletion of existing features. These modifications may affect system functionality. Regression Testing becomes necessary in this case.

Regression testing can be performed using the following techniques:

**1. Re-test All:**

Re-Test is one of the approaches to do regression testing. In this approach, all the test case suits should be re-executed. Here we can define re-test as when a test fails, and we determine the cause of the failure is a software fault. The fault is reported, we can expect a new version of the software in which defect fixed. In this case, we will need to execute the test again to confirm that the fault fixed. This is known as re-testing. Some will refer to this as confirmation testing.

The re-test is very expensive, as it requires enormous time and resources.

2. Regression test Selection:

In this technique, a selected test-case suit will execute rather than an entire test-case suit.

The selected test case suits divided in two cases

1 Reusable Test cases.

2 Obsolete Test cases.

1 Reusable test cases can use in succeeding regression cycle.

2 Obsolete test cases can't use in succeeding regression cycle.

**3. Prioritization of test cases:**

Prioritize the test case depending on business impact, critical and frequently functionality used. Selection of test cases will reduce the regression test suite.

8. What is Bug Life Cycle?

Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

NEW

ASSIGN

OPEN DUPLICATE, REJECTED, DEFAULT, NOT A BUG

FIXED

PENDING RETEST TEST

VERIFIED

CLOSED

## Defect Life Cycle States:

New - Potential defect that is raised and yet to be validated.

Assigned - Assigned against a development team to address it but not yet resolved.

Active - The Defect is being addressed by the developer and investigation is under progress. At this stage there are two possible outcomes; viz - Deferred or Rejected.

Test - The Defect is fixed and ready for testing.

Verified - The Defect that is retested and the test has been verified by QA.

Closed - The final state of the defect that can be closed after the QA retesting or can be closed if the defect is duplicate or considered as NOT a defect.

Reopened - When the defect is NOT fixed, QA reopens/reactivates the defect.

Deferred - When a defect cannot be addressed in that particular cycle it is deferred to future release.

Rejected - A defect can be rejected for any of the 3 reasons; viz - duplicate defect, NOT a Defect, Non Reproducible.